

The 24rd International Conference
on Computer Graphics and Vision

GraphiCon'2014

September 30 – October 3, 2014
Rostov-on-Don, Russia

Conference Proceedings

24-я МЕЖДУНАРОДНАЯ КОНФЕРЕНЦИЯ
ПО КОМПЬЮТЕРНОЙ ГРАФИКЕ И ЗРЕНИЮ

ГрафиКон'2014

30 сентября – 3 октября, 2014
Ростов-на-Дону, Россия

Труды конференции

Академия архитектуры и искусств
Институт механики, математики и компьютерных наук
Южный федеральный университет

Ростов-на-Дону – 2014

УДК 004.9
ББК 32.973.26-018.2
Г 78
Г 78

ГрафиКон'2014: 24-я Международная конференция по компьютерной графике и зрению: Ростов-на-Дону, Академия архитектуры и искусств ЮФУ
Труды конференции

УДК 004.9
ББК 32.973.26-018.2

Выпускающий редактор. Л. Ковалева

Подписано к печати 19.09.2014 г.
Печать офсетная. Бумага офсетная.
Формат 21х29. Усл. п.л. 11,4. Уч.-изд. л. 21,5.
Тираж 100 экз. Заказ 1321

Академия архитектуры и искусств ЮФУ
пр. Будёновский, 39

Типография ООО «Лаки Пак»
ул. Мечникова, 112

СОДЕРЖАНИЕ

Секция: Компьютерная графика

С.

1. Некоторый опыт применения Nvidia OptiX. В. Дебелов (Лаборатория численного анализа и машинной графики Институт вычислительной математики и математической геофизики СО РАН, Новосибирск, Россия) 6
2. Метод реконструкции и параметризации поверхностных моделей компьютерных манекенов на основе их геометрических свойств. С. Грудинин (Факультет автоматизации и вычислительной техники Новосибирский государственный технический университет, Новосибирск, Россия) 9
3. Формирование и использование изображений-англифов трехмерных моделей местности для рационального природопользования речных бассейнов. А. Коротин, Е. Попов (Нижегородский государственный архитектурно-строительный университет, Нижний Новгород) 13
4. Автоматизация процесса чтения чертежа с использованием системы Компас-3D. Н. Жилина, М. Лагунова, Т. Мошкова, С. Ротков, В. Тюрина (Нижегородский государственный архитектурно-строительный университет, Нижний Новгород) 17
5. Автоматическое обнаружение геометрических ошибок на машиностроительных 2d-чертежах. Д. Васин, С.Ротков (НИИ Прикладной математики и кибернетики Нижегородского Государственного Университета им. Н.И. Лобачевского; Нижегородский государственный архитектурно-строительный университет, Нижний Новгород) 20
6. Физически корректный рендеринг прозрачных оптически активных изотропных кристаллов. Д. Козлов, В. Дебелов (Лаборатория численного анализа и машинной графики Институт вычислительной математики и математической геофизики СО РАН, Новосибирск, Россия) 23
7. Моделирование и визуализация 3D виртуальных пультов управления в тренажерных системах. М. Михайлюк, М. Торгашев (Научно-исследовательский институт системных исследований РАН, Москва, Россия) 27
8. Эффективная реализация HDR визуализации на GPU. П. Тимохин, М. Торгашев (Научно-исследовательский институт системных исследований РАН, Москва, Россия) 30
9. Using dual quaternions for Control Cluster Method. A. Bukatov (SFU Computer Center), E. Gridchina, D. A. Zastavnoy (Informatics and Computational Experiment Department, SFU) 34
10. Фотореалистичная модель объемного рассеивания в задаче двунаправленной стохастической трассировки лучей. Д. Жданов (Санкт-Петербургский Национальный Исследовательский Университет ИТМО, Санкт-Петербург; Институт Прикладной Математики им. М.В. Келдыша РАН, Москва; Государственный Оптический Институт им. С.И. Вавилова, Санкт-Петербург), А. Гарбуль (Государственный Оптический Институт им. С.И. Вавилова, Санкт-Петербург), Волобой А.Г., Галактионов В.А., Ершов С.В., Потемин И.С., Соколов В.Г. (Институт прикладной математики им. М.В. Келдыша РАН, Москва) 38
11. Сравнительный анализ современных рендер-систем для 3ds Max. В. Фролов, В. Галактионов, М. Трофимов (Институт прикладной математики им. М.В. Келдыша) 43
12. Практичный подход ускорения синтеза изображений методом трассировки лучей Монте-Карло. А. Груздев, В.Фролов, А.Игнатенко (Московский государственный университет им. М.В. Ломоносова) 47
13. Интерактивный визуальный анализ многомерных данных. П. Масленников, И.Е. Мильман, А.Э. Сафиуллин, А.Е. Бондарев, Ш.У. Низаметдинов, В.В. Пиллогин (НИЯУ МИФИ, Институт прикладной математики им. М.В. Келдыша РАН) 51
14. Постобработка и анализ синтезированных реалистичных изображений. Б. Барладян, Е. Бирюков, И. Валиев, А. Волобой, Л. Шапиро (Институт прикладной математики им. М.В. Келдыша РАН, Москва) 55
15. Графический инструментарий стоматолога-гнатолога на основе DVR. А. Белокаменская, М. Новожилов, В. Турлапов (Нижегородский Государственный Университет им. Н.И. Лобачевского, Нижний Новгород) 59
16. Распознавание символов на базе низкоуровневых моделей описания графических изображений. Д. Васин, М. Ершов (НИИ прикладной математики и кибернетики Нижегородского государственного университета им. Н.И. Лобачевского) 62
17. Визуализация результатов историко-теоретических исследований архитектурного наследия Северного Кавказа. О.Иевлева, А.Кошевой, Т. Карпюк (Академия архитектуры и искусств ЮФУ, Ростов-на-Дону) 65

Секция: Обработка изображений и видео

1. Комбинированный метод коррекции спутниковых изображений съемочной системы сканового типа с неравномерной чувствительностью датчиков. В. Гусев (Московский авиационный институт) 69
2. Нелинейная ортогональная фильтрация цифровых изображений. В. Сазонов, М. Щербаков (Пензенский государственный университет, Пенза) 72
3. Об оптимальном выделении субполосных компонент изображений. А. Черноморец (Белгородский государственный национальный исследовательский университет, Белгород, Россия) 75
4. Использование картографической основы в задаче анализа линейной спектральной смеси на гиперспектральных изображениях. А. Денисова, В. Мясников (Самарский государственный аэрокосмический университет имени академика С.П. Королёва (национальный исследовательский университет), Самара, Россия) 79
5. Representation of Linear Segment Voronoi Diagram by Bezier Curves. L. Mesteteskiy (Department of Mathematical Methods of Forecasting Lomonosov Moscow State University Moscow, Russia) 83
6. Универсальный метод извлечения скрытых изображений из полиграфических документов. В. Сергеев, В. Федосеев (Институт систем обработки изображений РАН, Самарский государственный аэрокосмический университет имени академика С.П. Королёва (национальный исследовательский университет), Самара, Россия) 88
7. Быстрый алгоритм совмещения контуров изображений, связанных изотропным аффинным преобразованием. Д. Сунгатуллина, А. Крылов (Московский государственный университет им. М.В. Ломоносова) 92
8. Оценка состояния венул по МРТ SWI изображениям с применением масок белого и серого вещества. И.Архипов, А. Ятченко, А. Гаврилов, И. Куликов, Инна Кротенкова, В. Брюхов, Р. Коновалов, А. Суслин⁴ (РНИЦХ РАМН, Факультет Вычислительной математики и кибернетики, ЗНИИЯФ МГУ им. М.В.Ломоносова, Отделение лучевой диагностики ФГБУ «НЦН» РАМН, Москва) 96
9. SAR Image Processing for Sea Surface Monitoring. В. Alpatov, O. Balashov, M. Ershov, V. Muraviev, A. Feldman (Department of Automation and Information Technologies in Control, Ryazan State Radio Engineering University) 100
10. Автоматическое построение комбинаций алгоритмов обработки изображений в задаче обработки УЗ изображений сонных артерий. Т. Махно (Институт прикладной математики и механики НАН Украины) 104
11. Визуализация и анализ тонкой тепловой структуры волны горения СВС в программе ImageJ. М. Бороненко, А. Серегин (Политехнический институт, кафедра физико-химии процессов и материалов, Югорский Государственный Университет, Ханты-Мансийск, Россия) 108
12. Face Quality Assessment for Face Verification in Video. M. Nikitin, , A. Konushi (Lomonosov Moscow State University,) V. Konushin (Video Analysis Technologies, LLC) 111
13. Выделение зрачка на изображении глаза преобразованием Хафа границ компонент связности. И. Матвеев (Вычислительный центр им. А.А. Дородницына Российской академии наук, Москва), Н.Н. Чинаев (Московский Физико-Технический Институт, Москва) 115
14. Алгоритмы построения и хранения навигационной когнитивной карты для взаимодействия с человеком. М. Пестун (Институт прикладной математики им. М.В. Келдыша Российской академии наук) 119
15. Кластеризация пикселей для сегментации цветового изображения. М. Харинов (Санкт-Петербургский институт информатики и автоматизации Российской академии наук) 123
16. Детектирование стадии фиброза печени при помощи текстурного анализа ультразвуковых изображений. А. Хвостиков, А. Крылов, Ю. Камалов (факультет вычислительной математики и кибернетики, Московский государственный университет им. М. В. Ломоносова) 127
17. Аппроксимация фазы проекционным методом Эрмита при восстановлении изображения по фазе. Е. Павельева, А. Крылов (Факультет Вычислительной Математики и Кибернетики Московский Государственный Университет имени М.В. Ломоносова, Москва, Россия) 131
18. Встраивание данных в плоскости Грея цифрового полутонового изображения. В. Горбачев, Л. Денисов, Е. Яковлева (Северо-Западный институт печати Санкт-Петербургского государственного университета технологии и дизайна), Е. Кайнарова (Национальный минерально-сырьевой университет «Горный», Санкт-Петербург) 135

Секция: Компьютерное зрение

1. Modification of The Multi-target Tracking Algorithm Based on Energy Minimization. A. Gringauz, E. Shalnov, A. Konushin (Department Of Computational Mathematics And Cybernetics, Lomonosov Moscow State University) 139
2. Модификация нейронной сети неокогнитрон для увеличения качества распознавания текста. А. Кугаевских (Институт математики и компьютерных наук, Тюменский государственный университет) 143
3. Анализ и визуализация многомерных данных в задачах вычислительной газовой динамики. А. Бондарев, В. Галактионов, Л. Шапиро (Институт прикладной математики им. М.В. Келдыша РАН, Москва) 146
4. Сравнение формы изображений на основе спектра ширины объектов. Е. Зак, Н. Ломов, Л. Местецкий (Московский государственный университет имени М.В. Ломоносова, Москва) 150
- А. Осокин, А. Осокин, Д.Ветров (Московский государственный университет им. М.В. Ломоносова; Национальный исследовательский университет Высшая школа экономики) 155
- Отпечатано в редакции и форматировании авторов*
6. Алгоритм поиска областей резкости на стереоизображениях. А. Афанасьева, А. Игнатенко (Московский государственный университет им. М.В. Ломоносова) 160
7. Система поиска минимального пути в среде с полигональными препятствиями. К. Заева, А.Семенов (Факультет прикладной математики и кибернетики, Тверской государственный университет) 164
8. Дополнительные селективные признаки минеральных объектов для оптического метода сепарации. Д. Петухова, Д. Шитов, А. Чертов (Университет ИТМО, Санкт-Петербург) 168
9. Алгоритм нахождения отрезков в задаче анализа аэрофотоснимков. Л. Новоторцев, А. Волобой (Институт прикладной математики им. М.В. Келдыша, Москва) 171
10. Корреляционный алгоритм распознавания символьной информации в телевизионных изображениях. Д. Григорович (ЗАО «Московский научно-исследовательский телевизионный институт», Москва) 174
11. Estimation of Human Age and Facial Expression Using Biologically Inspired Features. A. Spizhevoy, A. Bovyryn (Lobachevsky State University of Nizhni Novgorod, Russia Itseez Inc.) 178
12. Challenges for Graphics and Heterogeneous Architectures: Applications and Technology. T. Paltashev (Graphics IP Engineering, Advanced Micro Devices, Sunnyvale, California, U.S.A.), Ilya Perminov, National Research University ITMO, St. Petersburg, Russian Federation 181

Некоторый опыт применения Nvidia OptiX

В. Дебелов

Лаборатория численного анализа и машинной графики
Институт вычислительной математики и математической геофизики СО РАН,
Новосибирск, Россия
debelov@oapmg.sccc.ru

Аннотация

В данном сообщении рассматривается причина выбора OptiX в противовес CUDA в качестве инструмента для научно-исследовательской разработки при использовании Nvidia GPU.

На примере разработки простейшего рендера для сцен, состоящих из полупрозрачных оптически изотропных объектов, рассматриваются побудительные мотивы к использованию OptiX и основные привлекательные характеристики OptiX. Сообщается также о неприятности, которая может случиться во время работы программы.

Ключевые слова: трассировка лучей, спектральный рендеринг, Nvidia GPU, CUDA, OptiX.

1. ВВЕДЕНИЕ

Достаточно много научно-исследовательских программ разрабатывается для проведения численных экспериментов во время отладки и получения различных характеристик разрабатываемого алгоритма. Во многих случаях алгоритм последовательный, но для качественной проверки требуется получить некоторую целостную картину, просчитывая его для ряда связанных наборов параметров. В нашей области знаний алгоритм работает для серии пикселей.

Простой прямолинейный подход – просчитать алгоритм для всех пикселей последовательно – приводит к длительному затягиванию времени проведения экспериментов, особенно при наличии рекурсии. Часто применяется MPI, но для реального получения выгоды во времени счета требуется наличие подручного кластера. В последнее время стало модно программировать параллельные приложения, используя технологию CUDA, а значит:

- Надо изучить технологию. Знать архитектуру ГПУ. Очень времяземкий процесс. Или приглашать программиста, которому надо будет объяснить суть алгоритма в некотором объеме.
- Имеющийся код на C/C++ необходимо значительно переделать.
- По окончании экспериментов код на CUDA становится ненужным. Также, скорее всего, этот код не подойдет для GPU других производителей.

В принципе, подход напоминает переход с языка высокого уровня на более низкий уровень, что, как правило, ведет к уменьшению производительности труда исследователя, хотя может дать значительный выигрыш в производительности программы.

В последнее время (с 2008 г.) Nvidia популяризирует технологию OptiX, построенную над CUDA для создания параллельных программ для приложений, основанных на трассировке лучей [2]:

- Хотя OptiX построена над CUDA, программист не обязан знать CUDA. Однако для знатоков CUDA, OpenGL и

DirectX возможно применение их знаний для построения более эффективных программ.

- Программист должен немного перестроить свои модули на C или C++ при переходе на OptiX. В основном это касается косметических изменений при передаче параметров и организации основного цикла.
- Очень полезная черта: исходные коды переносимы между Windows и Linux платформами.
- Большой набор примеров приложений в исходных кодах, например, алгоритмы Виттеда, Кука.
- Для Windows обеспечивается автоматизация компиляции и сборки программ в рамках Visual Studio 2010. Для Visual Studio 2012 создан wizard приложений в среде OptiX.

Отрицательной стороной OptiX на современном этапе является ориентация на float, а не на double. Использование последних потребует определенных усилий.

В данной работе не приводится описание OptiX в какой-либо сокращенной форме, а просто показано, как применение механизма ускорения сказывается на времени работы программы.

2. СТРУКТУРА СЦЕНЫ OPTIX

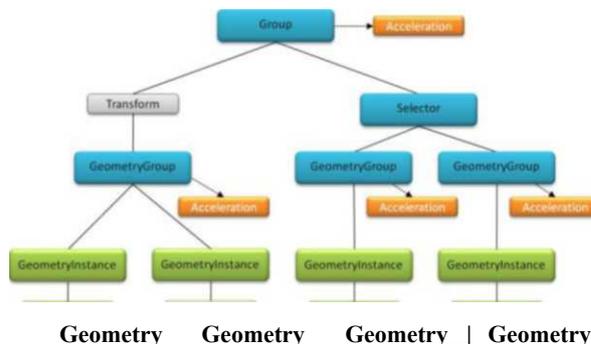


Рис. 1. Граф сцены OptiX

На рис. 1 приведен пример графа сцены, здесь:

- Group – вся сцена или ее подграф.
- Transform, Selector – способ включения подграфа в данный узел – через преобразование координат или как выбор из альтернативы.
- GeometryGroup – геометрия некоторого, возможно, осмысленного объекта сцены.
- Geometry – геометрия, набор однотипных геометрических примитивов. Может быть только один примитив.
- GeometryInstance – геометрия, специфицированная одним или несколькими материалами.
- Acceleration – способ организации геометрии подграфа, предназначенный для ускорения операции пересечения

луча с геометрией, заданной данным подграфом на основе габаритных боксов сыновних узлов данного узла.

Структура описания сцены вполне естественная, поэтому здесь переход на OptiX не вызывает трудности. Явно ни один из примитивов не является в OptiX стандартным. Для каждого используемого примитива программист должен предоставить две процедуры: 1) вычисление габаритов примитива; 2) вычисление точки (точек) пересечения произвольного луча с примитивом. Здесь надо отметить, что богатый набор примерных приложений, поставляемых с OptiX, содержит большое число требуемых процедур, запрограммированных с высоким качеством.

Еще один очень полезный механизм предлагается в OptiX – это *Callable* программы, которые, по сути, C/C++ функции, вызываемые в модулях, работающих на GPU. Получается, что нет ощутимых ограничений на стиль программирования.

3. НЕМНОГО О ТЕСТОВОМ ПРИЛОЖЕНИИ

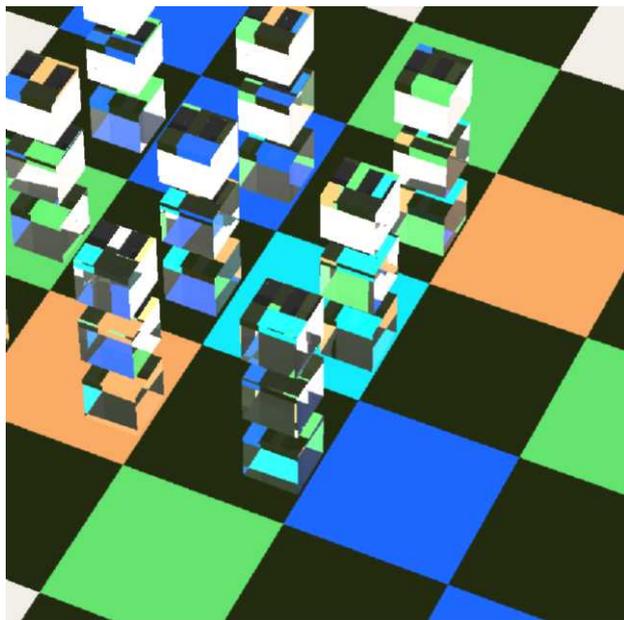


Рис. 2. Общий вид сцены

Описываемое экспериментальное приложение программировалось для отладки расчета взаимодействия луча поляризованного света с границей двух оптически изотропных сред, например, с границей воздуха или воды с кристаллом алмаза или стеклом. Алгоритм расчета взят из работы [1] и вычислительно является очень трудоемким. Тестовая сцена (см. рис. 1) включает в себя 27 прозрачных кубов и излучающего квадратного многоцветного источника света. Воображаемая охватывающая сцену сфера излучает освещенность 40% стандартного источника CIE D65. Как и в работе [1], применяется обратная рекурсивная трассировка лучей и спектральный рендеринг: для 81 длины волны в видимом диапазоне от 380 до 780 нм. Все кубы идентичны, для их простоты коэффициент преломления положен равным 1.2 и не зависит от длины волны.

4. ЭКСПЕРИМЕНТЫ

Итак, для отладки последовательного алгоритма, предназначенного для вычисления одного пересечения луча с границей двух сред, пишем последовательный рендер-П для расчета изображения сцены. Второй рендер-О программируем с применением OptiX. За день добиваемся того, чтобы все обрабатывающие модули имели идентичный код, за исключением способа передачи параметров. Никакого знания об GPU, кроме того, что он

разработан Nvidia и имеет сколько-то CUDA-ядер и его *compute capability* SM_NM (например, SM_21, SM_50). Последнее надо указать в параметрах сборки приложения.

Основная платформа для экспериментов: Intel® Core i7-3630QM @ 2.40GHz, Win 9.1, x84, VS2010, CUDA 5.5, OptiX 3.5.1. GPU1: GeForce GT 650M, 384 CUDA Cores. GPU2: GeForce GTX 560 Ti, 448 CUDA Cores. GPU2 стоит в десктопе, который сам в сравнениях не используется.

Расчет рендером-П сцены рис. 2 (768x768 пикселей) занял:

Глубина трассировки	Время
6	204
8	298

При этом рендер-О показывает:

Глубина тр-ки	Время: GPU1	GPU2
6	23	10
8	47	22

В CUDA есть понятие варпов (warp) из 32-х ядер CUDA. Мы для себя просто считаем, что GPU1 – это $Np1 = 384/32 = 12$, а GPU2 – это $Np2 = 448/32 = 14$. Таким образом, мое видение: GPU1 – это кластер с общей памятью и 12-ю «виртуальными» процессорами, работающими почти независимо. Часть из них занимает под менеджмент сам OptiX, а на остальных параллельно выполняется трассировка лучей.

Приведенные выше времена получены программой без всяких установок, ускоряющих работу программы на GPU, т.е. режим *NoAccel* (без структур ускорения).

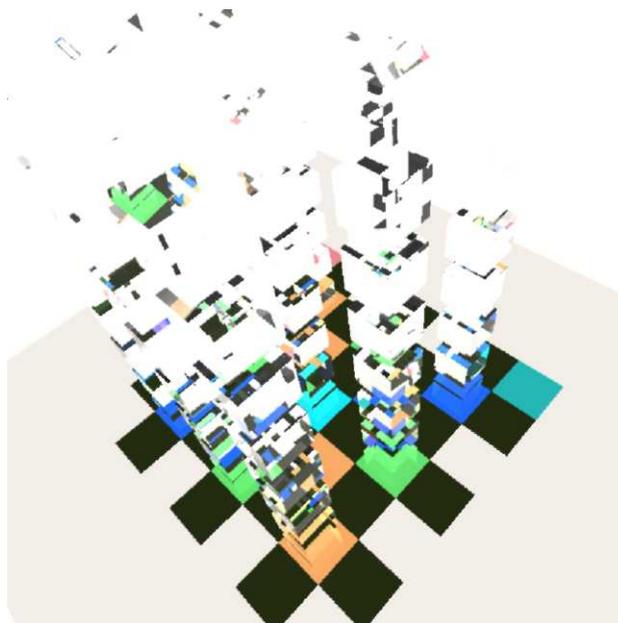


Рис. 3. Усложненная сцена

OptiX предлагает с десятков различных алгоритмов ускорения операции пересечения луча. Здесь не будут рассмотрены все они. Рассмотрим некоторые и то, как они влияют на время решения задачи. Каждый программист должен посвятить определенное время и подобрать подходящий алгоритм ускорения для его задачи, его геометрии.

Усложним сцену. Вместо трех этажей кубов сделаем девять, т.е. 81 куб, как на рис. 3. Все кубы организованы как единая Geometry, состоящая из 81 примитива типа куб. При создании GeometryInstance назначаем ту или иную стратегию ускорения: *NoAccell* – без ускорения; *Lbvh*, *Tbvh* – алгоритмы OptiX.

Получены следующие времена счета (г.т. – глубина трассировки):

Г.т.	Алгоритм	GPU1	GPU2
6	NoAccel	114	29
8	NoAccel	279	80
10	NoAccel	635	Q

Из приведенных цифр ясно, что на GPU2 под собственно трассировку выделено больше виртуальных процессоров, чем на GPU1. Очевидно, что менеджмент OptiX и там, и там занимает одно число процессоров. Символом Q отмечена ситуация, рассмотренная ниже.

Рассмотрим времена, полученные на GPU2 для разных глубин трассировки и разных алгоритмов ускорения.

Алг-м/Г.т.	6	8	10	12
GPU1 NoAccel	116	275	635	-
GPU1 Lbvh	113	277	645	-
GPU2 NoAccel	29	80	222	643
GPU2 Lbvh	35	89	212	552

По этим цифрам можно сделать вывод, что для такой простой сцены выигрыш от применения алгоритмов ускорения начинает проявляться только при больших глубинах трассировки, т.е. когда число операций пересечения луча со сценой превысит некоторую величину. На GPU1 этого порога мы не стали дожидаться. Сравнение GPU1 и GPU2 показывает, что скорость расчета приложения OptiX зависит от числа ядер CUDA на GPU.

Теперь рассмотрим ту же сцену, но каждый куб опишем при помощи 12-ти треугольников, т.е. создадим треугольную сетку из 972 треугольников. Число примитивов увеличилось в 12 раз.

Алг-м/Г.т.	6	8	10
GPU1 NoAccel	666	1623	-
GPU1 Lbvh	99	246	582
GPU1 Tbv	99	247	585
GPU2 NoAccel	433	Q	-
GPU2 Lbvh	37	121	271
GPU2 Tbv	41	121	273

Судя по цифрам, оба алгоритма ускорения работают примерно одинаково. Скорее всего, это связано с тем, что рассматривается очень простая ситуация – статическая сцена с одинаковыми по размерам примитивами.

5. Q – ПРОБЛЕМЫ ВИДЕОДРАЙВЕРА

В приведенных выше таблицах символ Q информирует о том, что задача снялась аварийно. Windows сообщает об этом как "Видеодрайвер перестал отвечать и был успешно восстановлен" ("Display driver has stopped working and has recovered"). Поиск по сети показал, что аналогичные сообщения возникают и в игровых программах при работе в Windows. Имя механизму – Timeout Detection and Recovery (TDR). В сети предлагается ряд способов обхода этого механизма. В моих экспериментах не все случаи удалось просчитать. Возможно возникновение ситуации Q связано с большим объемом вычислений в алгоритме из работы [1] в каждом узле дерева трассировки.

6. ЗАКЛЮЧЕНИЕ

В данном докладе сделана попытка ознакомить с некоторыми экспериментами по применению OptiX для разработки приложения, базирующегося на трассировке лучей (путей).

- OptiX оказался удобным инструментом для приложений, базирующихся на трассировке лучей.
- Появляется возможность превратить свой десктоп или ноутбук в небольшой кластер.
- Поскольку вполне достаточное подмножество API OptiX (без прямого использования DirectX и CUDA) ориентировано только на создание определенного типа приложений, то можно ожидать, что кто-нибудь реализует OptiX на других типах GPU.
- При возникновении непонятных ситуаций и для получения достаточно оперативных консультаций полезен форум для общения с пользователями и разработчиками OptiX [3].

7. БЛАГОДАРНОСТИ

Работа была выполнена при частичной поддержке гранта РФФИ № 12-07-00386 а.

8. ССЫЛКИ

- [1] Debelov V.A., Kozlov D.S. A Local Model of Light Interaction with Transparent Crystalline Media, IEEE Transactions on Visualization and Computer Graphics, 2013. - Vol. 19, No. 8. – P. 1274 - 1287.
- [2] Parker S.G. OptiX: A General Purpose Ray Tracing Engine. Siggraph, 2010.
- [3] Форум Nvidia OptiX: <https://devtalk.nvidia.com/default/board/90/optix/1/>

ОБ АВТОРАХ

Дебелов Виктор – доктор технических наук, ведущий научный сотрудник лаборатории численного анализа и машинной графики Института вычислительной математики и математической геофизики СО РАН.

E-mail: debelov@oapmg.sgcc.ru.

Метод реконструкции и параметризации поверхностных моделей компьютерных манекенов на основе их геометрических свойств

С. Грудинин

Факультет автоматизации и вычислительной техники

Новосибирский государственный технический университет, Новосибирск, Россия

sn_grudinin@mail.ru

Аннотация

В статье описан метод построения трехмерной модели манекена из полигональной входной модели для целей параметрического моделирования виртуальных манекенов. Предложенные алгоритмы позволяют на основе геометрических свойств горизонтальных сечений и контуров исходной модели получить параметризованное представление в виде трехмерного каркаса, аппроксимированного набором патчей.

Ключевые слова: компьютерный манекен, базовая модель, параметрическое моделирование, геометрические характеристики формы

1. ВВЕДЕНИЕ

В настоящее время существует большое количество графических систем, решающих задачу проектирования геометрических объектов. Параметризованные твердотельные объекты хорошо представлены в различных CAD-системах (например, AutoCAD, MicroStation, SolidWorks и пр.). Хорошо развиты методы параметрического проектирования, где геометрические модели представляются в виде теоретико-множественных композиций примитивов, активно развивается характеристическое моделирование примитивов [1]. Однако объекты, имеющие сложную геометрическую форму и представляемые в общем виде сеточными моделями, требуют для параметризации специализированных технологий, учитывающих информацию о форме и принятые в предметной области метрические характеристики. К ним относятся объекты естественного происхождения, в том числе и тело человека.

Среди методов поверхностного моделирования человеческого тела можно выделить три основных группы [2]: создание, реконструкция и интерполяция. К первой группе относятся 3D-сканирование и 3D-скульптинг. Во второй группе объединены методы, позволяющие получить модель по набору неструктурированных или частично структурированных данных [3]. Третья группа включает в себя методы, создающие модель путем деформации заданной базовой модели, в соответствии с установленными ограничениями. К этому подходу относится параметрическое моделирование [4]. В отличие от других методов параметрическое моделирование является менее дорогостоящим и вычислительно емким средством, что позволяет интерактивно создавать новые модели, изменяя заданные параметры. Формально такой подход можно описать следующим образом: создание производной модели Ω в результате деформации некоторой среднестатистической базовой модели Ψ , согласно набору заданных параметров ρ ; таким образом, новая производная модель Ω после деформации будет удовлетворять набору параметров ρ [5].

Дополнительные возможности параметрического моделирования приобретает с развитием средств

трехмерного сканирования и распространением баз данных сканов реальных объектов [6]. Модели, полученные сканированием, содержат обильные эмпирические данные, они могут использоваться, как для выявления закономерностей изменения форм объектов определенного класса (по статистическим данным, собранным с базы данных сканов), так и для создания базовой модели. Подобные решения требуют методов и средств извлечения семантической информации. В связи с этим сегодня актуально решение задач параметризации сложных объектов и генерации новых моделей на основе семантической информации.

Целью настоящей работы является разработка представления и параметризации базовой модели на основе данных о форме объекта, обладающих следующими свойствами:

- наилучшим образом отражающие соответствие формы базовой модели и объекта-оригинала;
- обладающие возможностью деформации при изменении параметров для проектирования производных моделей;
- ограниченные разумным количеством геометрических параметров.

2. МЕТОД МОДЕЛИРОВАНИЯ

На текущий момент разработано большое количество методов представления поверхности человеческого тела, из которых можно выделить три основных направления: представление в виде слоев, сеток и патчей. Послойное представление [2, 6] определяет модель множеством слоев (горизонтальных сечений), высоты слоев обуславливаются расположением характерных для предметной области параметров или точек. Сеточное представление [7, 8] строится по набору кривых линий, интерполирующих входную модель. Зачастую для моделирования человеческого тела используются четырехугольные сетки, при этом горизонтальные линии равноотстоят друг от друга, а вертикальные пересекают характерные для формы точки. Представление патчами несет в себе больше возможностей для передачи базовой модели особенностей формы исходного объекта, чем точки и кривые. При таком представлении поверхность манекена описывается некоторой сеточной структурой, каждая ячейка которой аппроксимируется гладким поверхностным патчем – трехмерной ограниченной поверхностью [8, 9].

Идея метода, излагаемого в работе, состоит в разделении предметного [10] и геометрического уровней параметризации и построении базовой модели на основании двух геометрических характеристик: точек и уровней, которые содержат в себе информацию о форме объекта. При этом построение базовой модели подразумевает выполнение следующих шагов:

- 1) приведение входной модели к формализованному виду;
- 2) выявление геометрических характеристик формы;

- 3) построение на основе геометрических характеристик четырехугольной сеточной структуры (каркаса);
- 4) аппроксимация ребер каркаса полиномами третьей степени;
- 5) аппроксимация ячеек каркаса линейными поверхностями Кунса.

Геометрическими параметрами в таком представлении выступают координаты характерных точек и значения коэффициентов полиномов. Деформация модели осуществляется путем изменения положения характерных точек, в результате чего происходит перерасчет кривых каркаса и патчей. Обеспечение схожести формы модели с оригинальным объектом достигается использованием при моделировании производных моделей значений коэффициентов полиномов базовой модели. Поверхность Кунса является одной из разновидностей поверхности натяжения и не требует для своего построения параметров, кроме ограничивающих ее кривых.

2.1. Представление входной модели

Исходными данными для построения базовой модели является 3D-модель женского манекена стандартной формы, полученная трехмерным сканированием. Входная модель имеет полигональную структуру: множество точек $t_{ij}(x_{ij}, y_{ij}, z_{ij})$, соединенных в треугольные полигоны $p_i = (t_{i1}, t_{i2}, t_{i3})$, $i = \overline{1, N}$, $j = \overline{1, 3}$, где N – количество полигонов. Модель не имеет внутренних полигонов, пересечений полигонов и отверстий. Задача ориентации модели в пространстве не рассматривается, модель ориентирована известным образом [11]. Входная модель разбивается на множество равноотстоящих друг от друга горизонтальных сечений h_i , каждое сечение из h_i содержит одинаковое количество точек $\{v_{ij}\}$, отсортированных по полярному углу и также равноотстоящих друг от друга $i = \overline{1, N}$, $j = \overline{1, M}$, где N – количество сечений, M – количество точек в сечениях. Также сечения h_i подвергаются симметризации – определению средних значений между левыми и правыми половинами относительно плоскости xOz и сглаживанию – интерполяции сплайнами третьего порядка [11].

2.2. Определение геометрических характеристик

Множество сечений h_i используется для анализа формы с целью выявления геометрических характеристик – характерных точек (ХТ) и характерных уровней (ХУ). Под характерными понимаются точки i -го сечения, наилучшим образом, с точки зрения определенного критерия, характеризующие его форму.

Мощным аппаратом для описания характеристик формы является дифференциальная геометрия. В работе используются два схожих между собой критерия: равенство нулю первой производной; равенство нулю второй производной.

С целью снижения вычислительной нагрузки расчеты выполняются для множества полусечений h'_i , $i = \overline{1, n}$, где n – количество полусечений. Каждое i -е полусечение рассматривается как кусочно-линейная функция $y(x)$ и выражается в параметризованном виде функциями $x(s)$ и $y(s)$. Параметром s выступает накопленная (суммарная) длина, то есть при $s_0 = 0$

$$s_j = s_{j-1} + \sqrt{(x_j - x_{j-1})^2 + (y_j - y_{j-1})^2}, \quad j = \overline{1, m},$$

где m – количество точек в i -м полусечении.

Для $x(s)$ и $y(s)$ определяются пары точек $f_a(s_a)$ и $f_b(s_b)$, между которыми первая производная:

$$f'_j(s) = (f(s_{j+1}) - f(s_j)) / (s_{j+1} - s_j), \quad j = \overline{2, m-1},$$

меняет знак, далее рассчитывается значение параметра s , при котором первая производная равна нулю: $(f_a s_b - s_a f_b) / (f_a - f_b)$. Аналогично определяются точки равенства нулю второй производной. ХТ также признаются точки начала и конца i -го полусечения, располагающиеся на оси симметрии манекена, соответственно.

В качестве третьего критерия определения ХТ используются так называемые доминантные точки, применяемые во многих приложениях машинного зрения, обработке изображений и распознавания образов. Кратко алгоритм их определения [12] для точек i -го полусечения из h'_i можно описать следующим образом:

1) рассчитывается величина:

$$b_{jk_j} = \max(|x_{j-k_j} - 2x_j + x_{j+k_j}|, |y_{j-k_j} - 2y_j + y_{j+k_j}|),$$

$j = \overline{2, m-1}$; при этом k_j определяется итерационно, начиная с 1, увеличиваясь на 1, пока $b_{ik} \leq b_{ik+1}$;

2) определяется величина: $bv_j = 1/k_j \sum_{i=1}^{k_j} b_{ji}$;

3) выполняется проверка условий: $bv_j < \varepsilon$, $bv_j < bv_{j-1}$, $bv_j < bv_{j+1}$, $bv_i = bv_{i-1}$, при $k_i < k_{i-1}$, $bv_i = bv_{i+1}$, при $k_i \leq k_{i+1}$.

Если хотя бы одно из условий выполняется, то точка не является доминантной. Для ε в данной работе, как и в [12], используется значение 0,25.

На рис. 1 представлено полусечение на уровне груди с отмеченными ХТ, определенными по второму и третьему критерию.

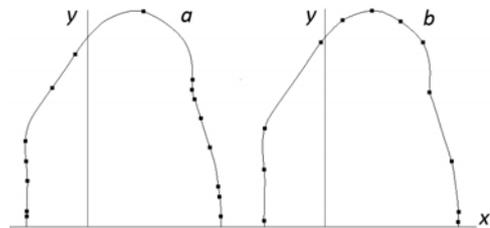


Рис. 1. Полусечение на уровне груди с отмеченными ХТ, определенными с использованием критерия:
(а) равенства нулю второй производной;
(б) доминантных точек

Исходя из основных особенностей формы манекена – симметричности и протяженности вдоль оси Oz – модели достаточно хорошо описываются множеством горизонтальных параллельных сечений, которые, в свою очередь, представляют собой гладкие, симметричные фигуры. Поэтому для достижения большей схожести модели с объектом-оригиналом к множеству h_i добавляются сечения, расположенные на ХУ – уровнях, наилучшим образом, с точки зрения определенного признака, описывающих форму манекена. В работе используются два критерия для определения ХУ: равенство нулю первой производной функции обхвата, равенство нулю первой производной функций контуров.

ХУ определяются по первому критерию путем исследования на экстремум кусочно-линейной функции $l(z)$, где аргументом является высота уровня, а значением – периметр минимальной выпуклой оболочки (обхвата) точек

сечения на уровне z . ХУ второго критерия также определяются исследованием на экстремум кусочно-линейных функций $f(z)$, $b(z)$, $s(z)$, определяющих зависимость от высоты сечения z координат переднего, заднего и бокового контуров, соответственно. Сечения, расположенные на ХУ, добавляются к множеству сечений h_i , на них также определяются ХТ.

2.3. Построение аппроксимирующей поверхности

Под алгоритмом построения каркаса подразумевается соединение точек сечений таким образом, чтобы они образовывали четырехугольную сеточную структуру, в узлах которой находятся ХТ. На вход алгоритма поступает множество полусечений h_i по m точек, при этом $\{p_{ij}\}$ из них характерные. Построение каркаса осуществляется в три этапа: соединение точек трех контуров, построение вертикальных ребер патчей, построение горизонтальных ребер патчей.

Соединение точек контуров происходит вертикальными линиями, при этом точки переднего и заднего контуров определяются как начало и конец полусечений; точки бокового контура определяются, как ХТ, имеющие наибольшую координату по оси y и x -координату в интервале $[-L_1, L_1]$, где L_1 – эмпирически определяемое ограничение.

Построение вертикальных ребер патчей происходит путем соединения ХТ в соседних полусечениях. Если ХТ $a \in \{p_{ij}\}$ имеет в соседнем полусечении ближайшую к ней ХТ $b \in \{p_{i+1j}\}$ и евклидово расстояние $E(a,b)$ между ними удовлетворяет условию $E < L_2$, то a и b соединяются между собой, где L_2 – эмпирически определяемое ограничение.

Построение горизонтальных ребер патчей выполняется за счет соединения ХТ по горизонтали по линиям соответствующих полусечений. При этом ХТ p_{ij} соединяется с соседними ХТ p_{ij+1} и p_{ij-1} , если выполняется одно из двух условий: p_{ij} имеет одно вертикальное соединение (снизу или сверху), p_{ij} вообще не имеет вертикальных соединений.

Линии каркаса аппроксимируются полиномами третьей степени, при этом каждая линия описывает ломаную p в пространстве. Вектор-функция, описывающая аппроксимирующую кривую, имеет вид [13]:

$$r(s) = p_0 + q_0s + as^2 + bs^3,$$

$$a = \frac{\Delta p - q_0}{s_n} - bs_n, b = \frac{q_0 + q_n - 2\Delta p}{s_n^2}, \Delta p = \frac{p_n - p_0}{s_n},$$

где p_0 и p_n – радиус-вектора точек начала и конца ломаной p (являющиеся ХТ); q_0, q_n – касательные к ломаной p в точках p_0 и p_n , соответственно; s – накопленная длина ломаной p ; s_n – длина ломаной p .

Касательные q_0 и q_n определяются с учетом линий каркаса, примыкающих в точках p_0 и p_n . Расчет касательной q_0 происходит в несколько этапов:

- 1) определяются векторы k_0 в точке p_0 по формуле: $k_0 = (p_1 - p_0) / (s_1 - s_0)$;
- 2) рассчитываются векторы k_1, \dots, k_r для всех r линий, примыкающих к точке p_0 по формуле: $k_i = (p_0 - p^{(i)}) / (s^{(i)} - s_0)$, $i = \overline{1, r}$, где (i) – порядковый индекс линии, примыкающей к точке p_0 ;

3) для всех возможных пар векторов из $\{k_0, \dots, k_r\}$ определяются нормали N_0, \dots, N_r , направленные от модели, где $z = r! / 2!(r-2)!$;

4) находится средняя нормаль $N' = \frac{\sum_{j=1}^r N_j}{\left| \sum_{j=1}^r N_j \right|}$;

5) рассчитывается проекция k_0 на плоскость, нормалью которой является N' по формуле $q_0 = k_1 - N' < k_1, N' >$.

Аналогичным образом определяется вектор q_n . Если линия, образующая границу патча, представляет собой отрезок, соединяющий две ХТ, то касательные вектора будут определяться по формулам: $k_0 = (p_n - p_0) / |p_n - p_0|$, $k_n = (p_0 - p_n) / |p_0 - p_n|$.

На каждой четырехугольной ячейке каркаса строится линейная поверхность Кунса, которая описывается векторной функцией [14]:

$$r(u, v) = (1-v)p^{(1)}(s_1u) + (1-u)p^{(2)}(s_2v) + vp^{(4)}(s_4v) + up^{(3)}(s_3u) - (1-u)(1-v)p_1 - u(1-v)p_2 - (1-u)vp_3 - uvv_4,$$

$$0 \leq u \leq 1, 0 \leq v \leq 1,$$

где p_i – радиус-вектор i -й точки, соединяющей кривые ячейки; $p^{(i)}$ – вектор-функция i -й кривой ячейки; s_i – длина i -й кривой ячейки; $i = \overline{1, 4}$.

3. РЕЗУЛЬТАТЫ

На рис. 2 представлены результаты аппроксимации патчами модели манекена с различными критериями определения ХТ и ХУ.

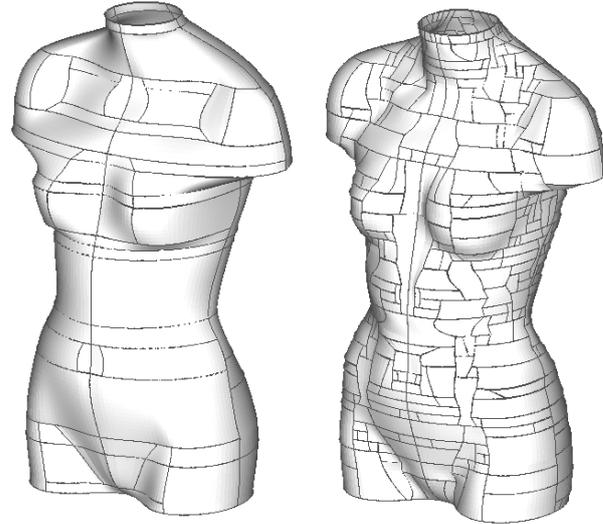


Рис. 2. Аппроксимированная патчами модель с используемыми в качестве критериев определения ХТ и ХУ: (слева) равенство нулю первой производной и обхватов, (справа) равенство нулю второй производной

4. ЗАКЛЮЧЕНИЕ

В статье представлен метод построения и параметризации базовой модели заданной структуры по полигональной модели исходного объекта. Рассмотрена оригинальная параметризация модели трехмерного объекта со сложной структурой, сохраняющая геометрическую информацию об объекте. Рассматриваемые алгоритмы могут быть использованы в процессе параметрического моделирования виртуальных манекенов. При этом создание производных моделей будет происходить за счет изменения координат

узловых точек каркаса с последующим пересчетом линий каркаса и аппроксимирующих поверхностей.

Перспективным развитием рассмотренных решений является:

– разработка процесса параметрического моделирования виртуальных манекенов на основе предложенного представления и геометрической параметризации;

– расширение списка исследуемых критериев для определения характеристик формы модели;

– определение зависимостей предметных и геометрических параметров в контексте моделирования манекенов для целей швейной промышленности.

5. ССЫЛКИ

- [1] Bronsvort, W. F. Developments in Feature Modelling / R. Bidarra, P.J. Nyirenda // Computer-Aided Design & Applications. – 2006. Vol. 2, № 5. – P. 655 – 664.
- [2] Lin, S.-F. Create a Virtual Mannequin Through the 2-D Image based Anthropometric Measurements and Radius Distance Free Form Deformation / S.-C. Chien // International Journal of Advanced Computer Science and Applications (IJACSA). – 2011. – Vol. 2, № 4. – P. 60 – 67.
- [3] Reconstruction and Interpretation of 3D Whole Body Surface Images / B. Buxton, L. Dekker, I. Douros, T. Vassilev // Scanning Proceedings. – 2000. – 17 p.
- [4] Фроловский В.Д. Компьютерное проектирование манекенов и одежды. Трехмерные модели и математические методы. Germany, Saarbrucken: LAMBERT Academic Publishing, 2012. – 269 с.
- [5] The most comprehensive source for body measurement data [Электронный ресурс]. URL: <http://store.sae.org/caesar/> (дата обращения: 30.06.2014).
- [6] Абдулин П.К., Фроловский В.Д. Сжатие геометрической информации сложных объектов на основе порождающих моделей // Труды 15-й Международной конференции по компьютерной графике и ее приложениям "ГРАФИКОН - 2005" (20-24 июня 2005 г.). Новосибирск. Институт вычислительной математики и математической геофизики СО РАН. – С. 175 – 178.
- [7] Kasap M., Magnenat-Thalmann N. Skeleton-aware size variations in digital mannequins // The Visual Computer . – 2011. – Vol. 27. – No. 4. – P. 263 – 274.
- [8] Wang C.C.L. Parameterization and parametric design of mannequins // Computer-Aided Design. – 2005. – Vol. 37. – No. 1. – P. 83 – 98.
- [9] Wu, L. A Parameterized Mannequin for Apparel Design / X. Zhang // Journal of Fiber Bioengineering and Informatics (JFBI). – 2008. – Vol. 1, № 2. – P. 117 – 124.
- [10] Грудинин С.Н., Фроловский В.Д. Предметная параметризация виртуальных манекенов // Автоматика и программная инженерия. – 2014. – № 1(7). – С. 53 – 56.
- [11] Грудинин С.Н., Фроловский В.Д. Параметрическое моделирование и оценка близости виртуальных манекенов // докл. Академии наук высшей школы Российской Федерации. – 2014. – No. 1(22). – С. 62 – 72.

[12] Wu W.-Y. Dominant point detection using adaptive bending value // Image and Vision Computing. – 2003. – Vol. 21. – P. 517 – 525.

[13] Ильин М.Е. Аппроксимация и интерполяция. Методы и приложения: учеб. пос. – Рязань, 2010. – 56 с.

[14] Голованов Н.Н. Геометрическое моделирование. – М.: Изд-во: Физико-математическая литература, 2002. – 472 с.

Об авторах

Сергей Грудинин, аспирант АВТФ НГТУ.

Его адрес: sn_grudinin@mail.ru

Формирование и использование изображений-анаглифов трехмерных моделей местности для рационального природопользования речных бассейнов

А. Коротин, Е. Попов

Нижегородский государственный архитектурно-строительный университет, Нижний Новгород, Россия

Аннотация

Наибольшей эффективностью и наглядностью обладают системы визуализации 3D-моделей с использованием стереоэффекта. В данной работе на базе доступной исходной информации продемонстрирован разработанный подход создания изображений-анаглифов на примере бассейна реки Кудьма Нижегородской области.

Ключевые слова: изображения-анаглифы, 3D модели, рациональное природопользование.

1. ВВЕДЕНИЕ

Выживание человека, как естественного биологического организма, определяется соответствием его действий основным законам биосферы и определяется естественным зонированием территорий, которое формируется на протяжении миллионов лет. Благодаря развитой гидрологической сети природная кластеризация естественно зонированных территорий, образующих единую экосистему, представлена в виде совокупности речных бассейнов. Во второй половине XX и начале XXI вв. обострилась необходимость составления прогнозных оценок экологического ущерба от антропогенного воздействия и сохранения эндемичности экосистемы. Геоинформационные системы позволяют осуществлять комплексный геоэкологический анализ и оценку состояния, использования и охраны земель и гидрологических сетей. В основе геоинформационных моделей предпочтительным является использование трехмерных моделей рельефа, восстановленных по данным дистанционного зондирования Земли.

Как известно трехмерные модели рельефа позволяют вычленить основные топографические единицы: тальвеги и водоразделы, формирующие основную пластику поверхности [1]. При этом именно данные элементы местности образуют характерный контур эндемичной замкнутой экосистемы в виде бассейнов рек. На основе построенных поверхностей, ограниченных топографическими элементами, можно осуществлять различные виды оценок состояния территорий.

2. ОСНОВНАЯ ЧАСТЬ

Главное условие, вытекающее из особенностей бинокулярного зрения, при создании изображений-анаглифов – достижение стереоскопического эффекта за счет цветового сдвига двух изображений (принцип спектрального разделения), полученных из двух точек пространства. Способы создания пространственного изображения объекта в

результате сложения или вычитания цветов были обоснованы еще в середине XIX в. Джозефом Чарльзом д'Альмейда и Вильгельмом Роллманом [2].

Изображения-анаглифы – самый дешевый способ пространственной визуализации стереоскопической информации, так как основан на принципе цветового сдвига и не требует использования сложных фотограмметрических технологий. Как и другие способы формирования изображений с объемным эффектом, анаглиф позволяет обеспечить раздельное изображение для каждого глаза. Отличие между двумя видами позволяет зрительной системе человека генерировать стереоскопическое представление сцены [3]. Анаглиф-изображение формируется путем наложения двух изображений (то есть, левое и правое изображения стереопары) в различных цветах. При просмотре анаглифа через цветные очки каждый глаз будет обозрывать только одно (свое) изображение. Однако при использовании подобного принципа существуют и недостатки, главным из которых является потеря цветовой адекватности при цветопередаче. Математически синтезированный цвет для цветовоспроизведения пикселя двух изображений (левое – L и правое – R) можно представить в виде матрицы суммарного преобразования аддитивной цветовой модели (RGB) [4]:

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} lk_r^1 & lk_r^2 & lk_r^3 \\ lk_g^1 & lk_g^2 & lk_g^3 \\ lk_b^1 & lk_b^2 & lk_b^3 \end{pmatrix} \times \begin{pmatrix} L_r \\ L_g \\ L_b \end{pmatrix} + \begin{pmatrix} rk_r^1 & rk_r^2 & rk_r^3 \\ rk_g^1 & rk_g^2 & rk_g^3 \\ rk_b^1 & rk_b^2 & rk_b^3 \end{pmatrix} \times \begin{pmatrix} R_r \\ R_g \\ R_b \end{pmatrix}$$

$$\text{при условии, что} \\ \sum_{i=1}^{i=3} (lk_r^i, rk_r^i) \cong 1; \sum_{i=1}^{i=3} (lk_g^i, rk_g^i) \cong 1; \sum_{i=1}^{i=3} (lk_b^i, rk_b^i) \cong 1$$

$$\text{Rank} \begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix} = 3$$

$$0 \leq L_{r,g,b} \leq 255; 0 \leq R_{r,g,b} \leq 255$$

Решение задачи создания изображения-анаглифа складывается из последовательности трех действий: получение двух изображений со смещениями (для определения глубины картины), применение цветового фильтра к каждому из изображений, совмещение в единую картину.

3. ПРОЕКТИРОВАНИЕ

Основным условием при создании стереоизображений является пара картин. При создании анаглифов используются

два изображения одного и того же объекта, смещенные вдоль одного пространственного базиса в горизонтальной плоскости. Непосредственной основой для составления изображений-анаглифов является «объемная» модель, при этом она не обязательно должна существовать материально, а может быть представлена в виде 3D изображения, полученного с применением компьютерных технологий.

Восстановление трехмерной модели местности бассейна р. Кудьмы производилось в ГИС MapInfo Professional на основе материалов дистанционного зондирования Земли (спутниковая радиолокационная съемка Земли – SRTM). При этом использовались бесплатные программные приложения и системы (3DEM, QuantumGIS, “P2P”, “3Dview”) [1].

После получения 3D поверхности, используя стандартные функции визуализации и операции оверлей, была получена цифровая модель (ЦМ) бассейна р. Кудьмы и окрестностей (рис. 1).

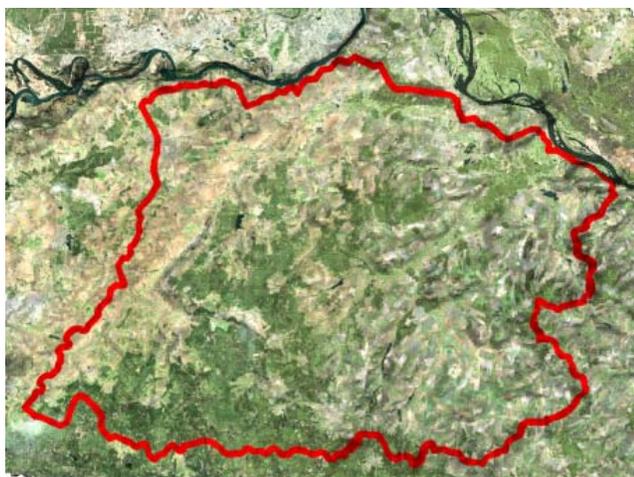


Рис. 1. ЦМ бассейна р. Кудьма

Для создания стереоэффекта необходимо иметь пару изображений, расположенных в одной сферической плоскости (VSI – Virtual Surface Image) с бесконечно большим радиусом. Учитывая использование «искусственного» объекта (трехмерная модель), каждый элемент пары L_{img} и R_{img} (рис. 2) по отношению к пользователю должен быть сдвинут на угол конвергенции (γ), с учетом остроты зрения (рис. 3) [5].

С использованием стандартных визуализаторов MapInfo Professional был получен ряд из 24 картин (12 пар), смещенных относительно друг друга на величину стандартного угла конвергенции. Расчет угла конвергенции производился на основании установленных правил: базис между лучами фокусировки зрения (B_f) составляет 65 мм, расстояние до изображения (картинной плоскости) анаглифа составляет 750 мм.

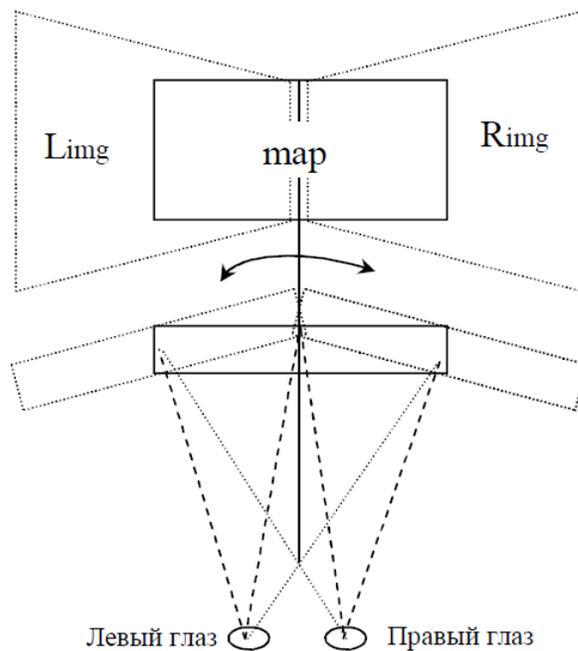


Рис. 2. Программное получение пары изображений «искусственных» объектов

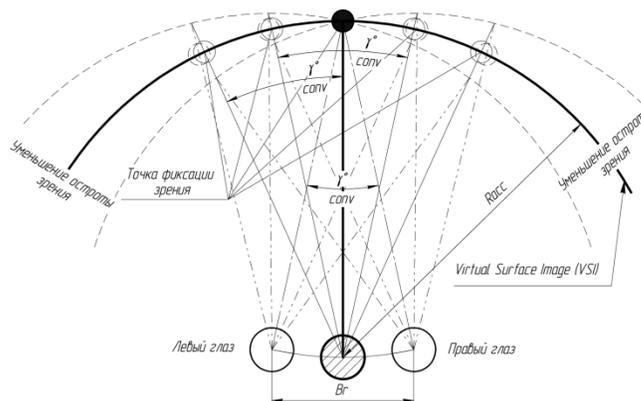


Рис. 3. Трехмерное восприятие изображения при бинокулярном зрении

Для применения цветного фильтра и совмещения пары изображений использовалась бесплатная программа «Anaglyph Maker» (Ver 1.08), разработанная японским фотографом Такаши Секитани (Takashi Sekitani) в 2004 г. [6]. Используя алгоритм данной программы, был получен ряд изображений-анаглифов с различными углами конвергенции (рис. 4).

Из полученных вариантов был выбран один с наибольшей информативностью и наименьшими негативными последствиями для зрения человека (исключить напряжение на глаза и излишнюю фокусировку).

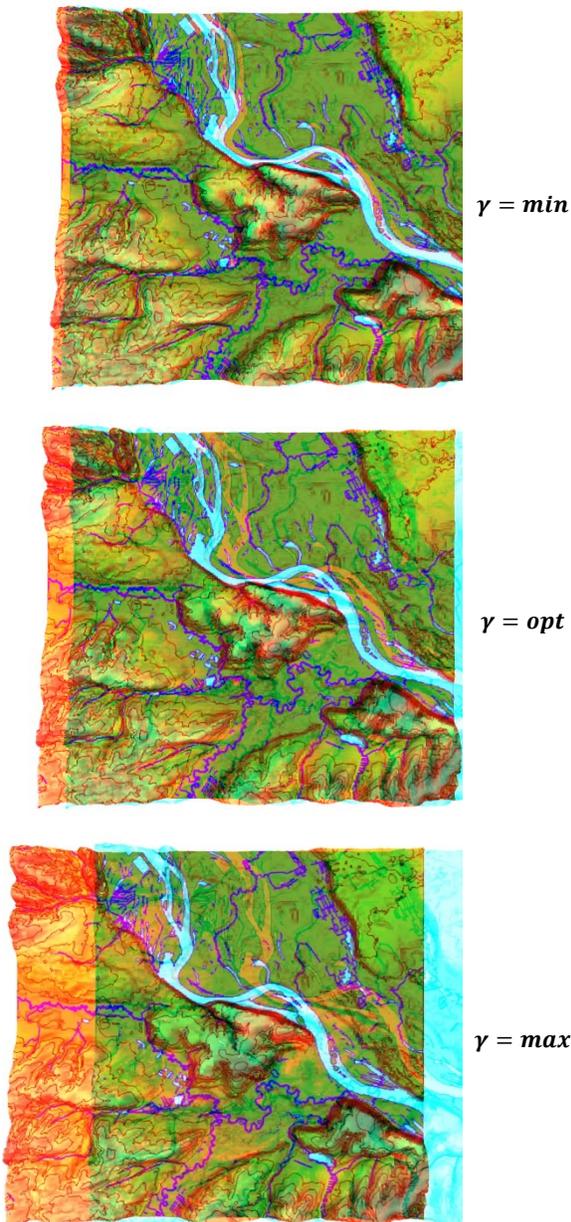


Рис. 4. Примеры изображений-анаглифов, полученных с использованием программы «Anaglyph Maker» при разных углах конвергенции на участок впадения р. Кудьма в р.Волга (Чебоксарское водохранилище).

Изображения-анаглифы могут использоваться как источники при создании гипсометрических и орографических карт, по которым наглядно можно определять элементы пластики рельефа местности (впадины, хребты, их сочленения и т.д.). Одно из возможно основных применений – определение или уточнение границ речных бассейнов (крупных, средних и малых речных систем) и эрозионного расчленения территории визуальным методом [7]. Сопутствующее использование – это прогнозирование, планирование и организация территории и использования земельных

ресурсов. Данный способ создания изображений-анаглифов позволяет сэкономить огромные денежные средства, так как не требует крупных вложений и основан на использовании общедоступной информации.

Подобные изображения-анаглифы могут служить картографической основой для визуальной оценки подготовительных материалов при землеустройстве муниципального района или подготовки схем территориального планирования. Кроме того, данные изображения могут быть использованы при представлении инвестиционных проектов. Ниже (рис. 5) приведены тематические изображения-анаглифы на территорию бассейна р. Кудьма Нижегородской области.

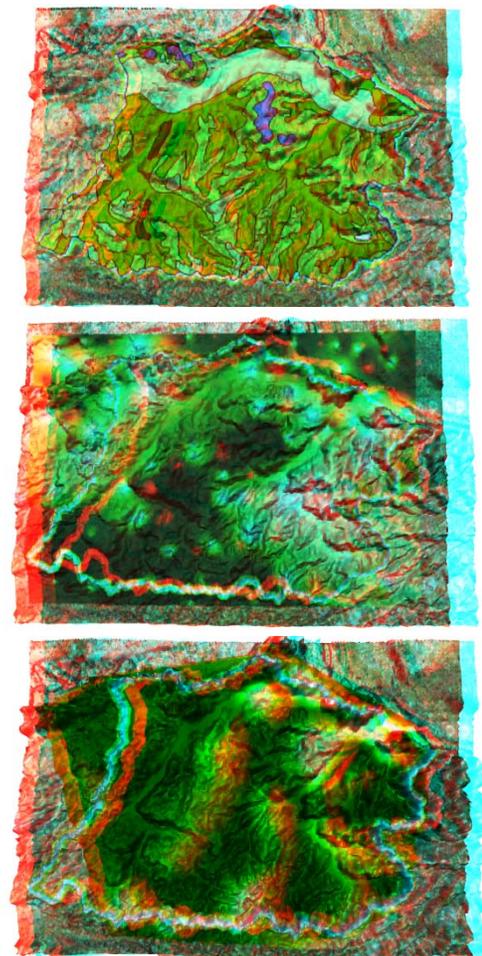


Рис. 5. Тематические изображения-анаглифы бассейна р. Кудьма Нижегородской области (верхнее – геологическое строение бассейна; среднее – залесенность бассейна; нижнее – плотность распределения ООПТ бассейна)

Применение изображений-анаглифов, достоинства которых (генерализация изображений, пространственная визуализация, простота получения) обусловило возможность визуального изучения объектов и характеристик, иначе подававшихся исследованию при двумерном отображении,

открывает для географических (геоинформационных в том числе) пространственных исследований, непосредственно не используя постоянно электронные вычислительные ресурсы и сложное техническое и программное обеспечение [8].

4. ЗАКЛЮЧЕНИЕ

В процессе работы был получен ряд тематических изображений-анаглифов, отображающих особенности геоморфологии территории, распределения лесных массивов, плотности особо охраняемых природных территорий и т.д., которые использовались при формировании предложений по территориальному планированию бассейна и рациональному природопользованию внутри него. Несмотря на многие положительные качества, изображения-анаглифы имеют существенные недостатки в цветности из-за использования спектрального сдвига (при создании тематических изображений необходимо, как можно больше, использовать оттенки цвета в зеленом спектре) и значительной нагрузки на зрение человека.

Применение подобных материалов для анализа территорий (объектов) имеет большие преимущества по сравнению с двумерными изображениями благодаря стереоскопическому эффекту и большей наглядности. Широкое применение подобные изображения могут найти в картографической отрасли для создания карт-анаглифов.

5. ССЫЛКИ

- [1] Коротин, А.С. Восстановление трехмерных моделей рельефа местности на основе материалов дистанционного зондирования Земли / А.С. Коротин, Е.В. Попов // Приволжский научный журнал. – Н. Новгород: ННГАСУ, 2014. - №2(30). – С. 29 - 34.
- [2] Rollmann, W., Zwei Neue stereoskopische Methoden / Wilhelm Rollmann (1853) // Annalen der Physik, vol.- WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim, 2006 - 166, Issue 9. – P.186 - 187.
- [3] Tovee, M.J. An Introduction to the Visual System / Martin J. Tovee // Cambridge University Press, 2008 – 2nd edition. – 222 p.
- [4] Songnan Li, Anaglyph Image Generation by Matching Color Appearance Attributes / Songnan Li, Lin Ma, King N. Ngan // Signal Processing: Image Communication, 2013 – vol. 28, no. 6. – P. 597 - 607.
- [5] Назаров, А.С. Фотограмметрия: учебное пос. для студентов вузов / А.С. Назаров. – Минск: ТетраСистем, 2006. – 368 с.: ил.
- [6] Электронный ресурс: <http://www.stereoeye.jp>
- [7] Лабутина, И.А. Дешифрирование аэрокосмических снимков: учебное пос. для студентов вузов / И.А. Лабутина. – М.: Аспект Пресс, 2004. – 184 с., 8 с. цв. вкл.
- [8] Кравцова, В.И. Космические методы исследования почв: учебное пос. для студентов вузов / В.И. Кравцова. – М.: Аспект Пресс, 2005. – 190 с., 8 с. цв. вкл.

Сведения об авторах

Коротин Антон – соискатель уч. степ. канд. наук кафедры инженерной геометрии, компьютерной графики и автоматизированного проектирования, инженер учебно-научно-производственного центра «Кадастр» управления научных исследований, инноваций и проектных работ,

Нижегородский государственный архитектурно-строительный университет.

Россия, 603950, г. Н. Новгород, ул. Ильинская, д. 65.

Тел.: +7 (831) 430-87-16; факс: +7 (831) 430-87-16;

эл. почта: kadastr-nngasu@mail.ru

Попов Евгений – д-р техн. наук, проф. кафедры инженерной геометрии, компьютерной графики и автоматизированного проектирования, Нижегородский государственный архитектурно-строительный университет.

Россия, 603950, г. Н. Новгород, ул. Ильинская, д. 65.

Тел.: +7 (831) 430-54-00; факс: (831) 434-10-34;

эл. почта: popov@sandy.ru

Автоматизация процесса чтения чертежа с использованием системы Компас-3D

Н. Жилина, ННГАСУ, zhilina@nngasu.ru
М. Лагунова, ННГАСУ, mvlnn@rambler.ru
Т. Мошкова, ННГАСУ, ng_forever.mtv@gmail.com
С. Ротков, ННГАСУ, rotkov@nngasu.ru
В. Тюрина, ННГАСУ, 55555_73@mail.ru

Аннотация

В статье описывается алгоритм автоматизированного построения 3D-модели составного твердого тела по ортогональным проекциям на базе системы геометрического моделирования Компас-3D. Алгоритм основан на анализе ортогональных проекций твердого тела в соответствии с законами начертательной геометрии, а также на эвристическом методе распознавания характерных признаков формообразующих операций, используемых в системе Компас-3D для построения 3D-моделей. Данный алгоритм имеет ограничения на входящую информацию, а именно: не рассматриваются разрезы и сечения. Результатом применения алгоритма является преобразование двумерных данных об объекте в трехмерные данные (так называемый переход из 2D в 3D). Рассмотренная проблема относится к сложно формализуемой задаче «автоматического чтения чертежа». Предложенный в статье алгоритм предназначен для автоматизации процесса решения обратной задачи начертательной геометрии.

Ключевые слова: ортогональные проекции, формообразующие операции, обратная задача начертательной геометрии, автоматизированное чтение чертежа, эвристический метод, преобразование данных.

Сущность проблемы автоматизации чтения чертежа состоит в возможности заменить человека в процессе распознавания формы и положения трехмерного объекта по его графическим изображениям. Термин «автоматизация чтения чертежа» был введен профессором В.С. Полозовым [1, 2].

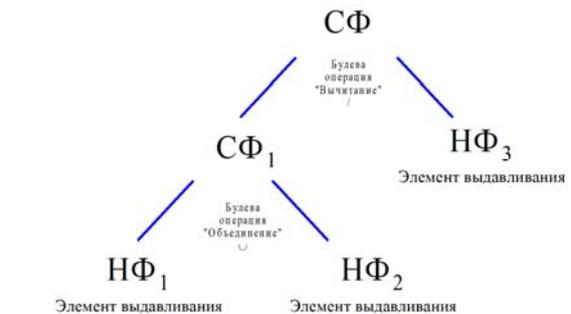
Существуют теоретические, алгоритмические и программные попытки решения этой проблемы [1, 2, 3, 4, 5, 6], но в полной мере она не решена. Предлагаемый в статье алгоритм предназначен для частичного решения этой задачи на базе системы трехмерного геометрического моделирования Компас-3D. Предполагается использовать данный алгоритм для разработки приложения Компас-3D, которое представляет интерес, как в теоретическом плане – в качестве попытки автоматизировать решение обратной задачи начертательной геометрии, так и в практическом – для использования этого приложения в учебном процессе по дисциплинам «Начертательная геометрия» и «Компьютерная графика».

В основу разработки алгоритма положен эвристический подход к проблеме [1]. Анализ мыслительной деятельности человека в процессе чтения чертежа, в частности, при рассмотрении ортогональных видов реального трехмерного объекта, позволяет выделить ключевые моменты, на основе которых человек получает представление о форме и расположении объекта относительно некоторой системы отсчета. В данной работе ставилась цель составить алгоритм действий человека (студента, конструктора) при построении 3D-

модели твердого тела с помощью системы Компас-3D. Входящей информацией являются ортогональные проекции объекта, включающие в себя невидимые линии (сечения и разрезы в данном случае не используются). Используются два ортогональных вида, построенные по законам начертательной геометрии, т.е. обратимый чертеж объекта.

При анализе проекций происходит мысленное разбиение составного объекта или фигуры (далее СФ) на простейшие или «непроизводные» фигуры (далее НФ). Будем считать «непроизводной» фигурой такой объект, который может быть построен в системе Компас-3D с помощью любой из четырех, имеющихся в арсенале этой системы, формообразующих операций. А именно, следующих операций: «Выдавливание», «Вращение», «Кинематическая», «По сечениям». Дальнейший анализ изображений позволяет определить связи между этими простейшими объектами. Эти связи реализуются применением к этим объектам, как к множествам, булевых операций («объединение», «пересечение», «вычитание»). В системе Компас-3D эти операции реализованы в виде возможности «приклеивания» либо «вырезания» элементов с помощью любой из четырех уже упоминавшихся выше формообразующих операций. Человек, анализирующий проекции, определяет необходимость применения той или иной операции по характерным признакам, которые он выделяет или «распознает» на чертеже.

Пример бинарного дерева для некоторого случая



показан на рис. 1.

Таким образом, поставленная задача сводится к выделению, классификации этих признаков и автоматизации их распознавания. остановимся на проблеме выделения признаков формообразующих операций.

Формообразующие операции имеют приоритеты, т.е. рассматриваются в порядке простоты их использования. Так как операции «Выдавливание» и «Вращение» являются частными случаями операции «Кинематическая» и реализуют кинематический закон образования поверхностей, то признаки этих операций выявляются на чертеже в первую очередь. Причем, из этих двух операций первой рассматривается операция «Выдавливание» как имеющая приоритет в системе Компас-3D, а затем операция «Вращение». После того,

как тот или иной признак позволил выявить трехмерный примитив, относящиеся к нему линии удаляются со всех проекций чертежа. Каждая последующая итерация рассматривает уже «облегченный» чертеж. Следующей по приоритетам является «Кинематическая» операция – общий случай кинематического закона образования поверхностей. После распознавания и удаления с чертежа кинематических элементов на чертеже выявляются признаки операции «По сечениям» и формируются соответствующие непроизводные фигуры (НФ). Полученный набор НФ разных типов используется далее для формирования графа сборки конструкции (бинарного дерева).

Следует иметь в виду, что достаточно часто трехмерные примитивы могут быть получены разными способами, т.е. с использованием не единственно возможной формообразующей операции. Здесь свободный выбор пользователя приходится заменить установлением приоритетов операций.

Приведем примеры некоторых признаков:

1. Для возможного применения операции «Выдавливание» (без уклона) должно выполняться следующее условие: замкнутому контуру, выделенному на одной из проекций, на другой проекции соответствует, т.е. расположен в проекционной связи, прямоугольный замкнутый контур (рис. 2).

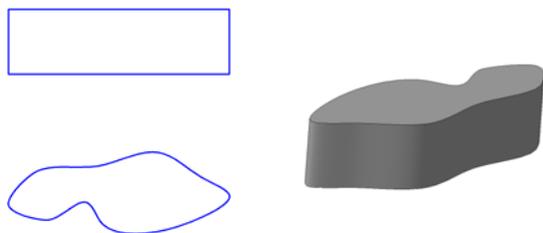


Рис. 2

2. Для возможного применения операции «Вращение» должны одновременно выполняться следующие условия (рис. 3):

- а) на одной из проекций присутствуют контуры, имеющие форму концентрических окружностей;
- б) на другой проекции присутствует замкнутый контур, находящийся в проекционной связи с окружностями и имеющий ось симметрии, проходящую через центр этих окружностей.

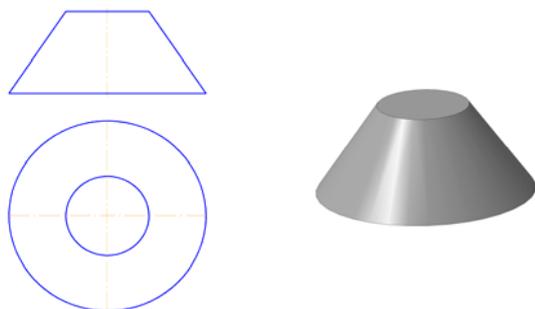


Рис. 3

3. Для возможного применения операции «Кинематическая» должны одновременно выполняться следующие условия (рис. 4):

- а) на проекциях нет признаков операций «Выдавливание» и «Вращение»;
- б) на одной проекции присутствуют конгруэнтные замкнутые контуры, связанные между собой криволинейными или прямолинейными ребрами;

- в) на другой проекции им соответствуют замкнутые контуры, также связанные криволинейными или прямолинейными ребрами;
- г) обе проекции должны иметь внешний замкнутый контур (представляющий собой очерк описываемой непроизводной фигуры).

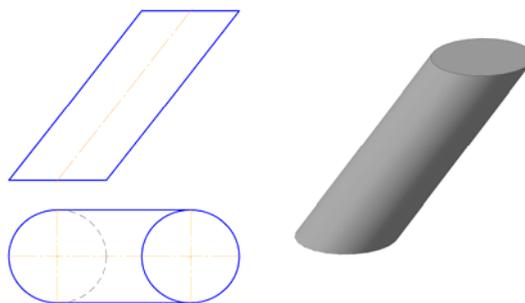


Рис. 4

4. Для возможного применения операции «По сечениям» (созданных в плоскостях, параллельных одной из координатных плоскостей) должны одновременно выполняться следующие условия (рис. 5):

- а) на проекциях нет признаков операций «Выдавливание», «Вращение» и «Кинематическая»;
- б) на одной проекции присутствуют замкнутые контуры, связанные между собой внешним контуром (очерком);
- в) на другой проекции им соответствует замкнутый контур, имеющий форму трапеции, который может содержать внутренние криволинейные или прямолинейные ребра, имеющие соответствующие им ребра на другой проекции.

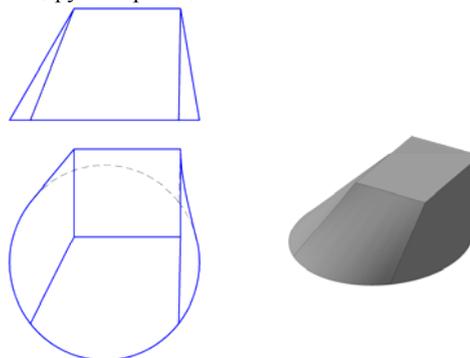


Рис. 5

Для разных вариантов применения формообразующих операций можно выделить разнообразные группы характерных признаков.

Общая схема действия алгоритма:

1. Анализ на симметрию и выделение групп осевых линий, которые используются при выявлении признаков формообразующих операций.
2. Анализ на возможность применения операции «Выдавливание». Формируется массив $НФ_i$ – простейших непроизводных фигур – элементов выдавливания.
3. После выделения каждой $НФ_i$ с проекций удаляются соответствующие им линии.
4. Измененные проекции анализируются на возможность применения операции «Вращение». Формируется массив $НФ_k$ – простейших непроизводных фигур – элементов вращения.
5. После выделения каждой $НФ_k$ с проекций удаляются соответствующие им линии.
6. Измененные проекции анализируются на возможность применения операции «Кинематическая». Формируется

массив $НФ_g$ – простейших непроезводимых фигур – кинематических элементов.

7. После выделения каждой $НФ_g$ с проекций удаляются соответствующие им линии.

8. Измененные проекции анализируются на возможность применения операции «По сечениям». Формируется массив $НФ_f$ – простейших непроезводимых фигур – элементов по сечениям.

9. Первичные проекции анализируются на необходимость применения булевых операций.

10. Формируется бинарное дерево (граф сборки конструкции), соответствующее составной фигуре, изображенной на исходных проекциях.

11. Формируется программный макрос выполнения сборки в соответствии с полученным бинарным деревом.

На рис. 6 показан пример составной фигуры, в которой можно выделить шесть непроезводимых фигур. Бинарное дерево или граф сборки конструкции, полученный с помощью описанного выше алгоритма, позволяет задать управляющую программу и получить 3D-модель данного объекта в автоматическом режиме, без участия пользователя.

На рис. 7 показано бинарное дерево для рассмотренной составной фигуры.

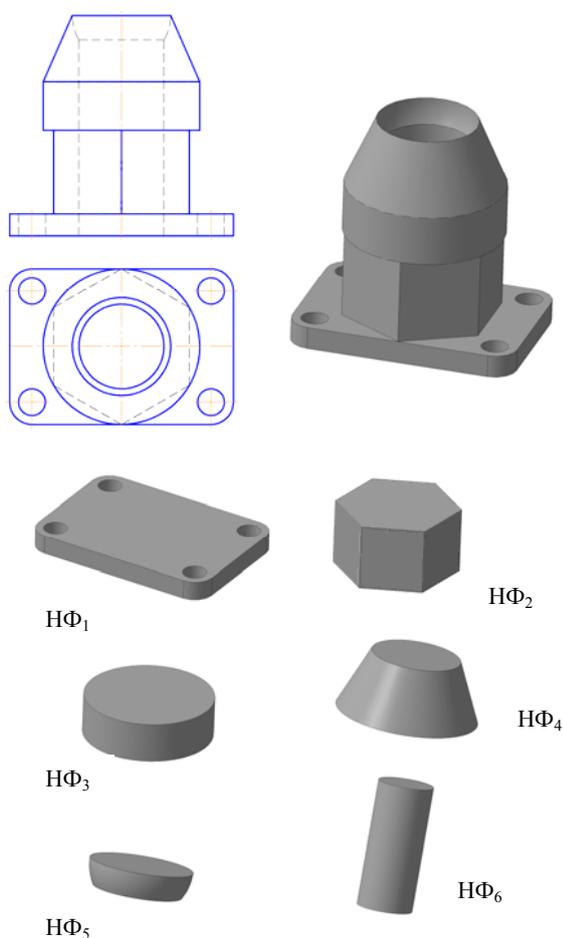


Рис. 6

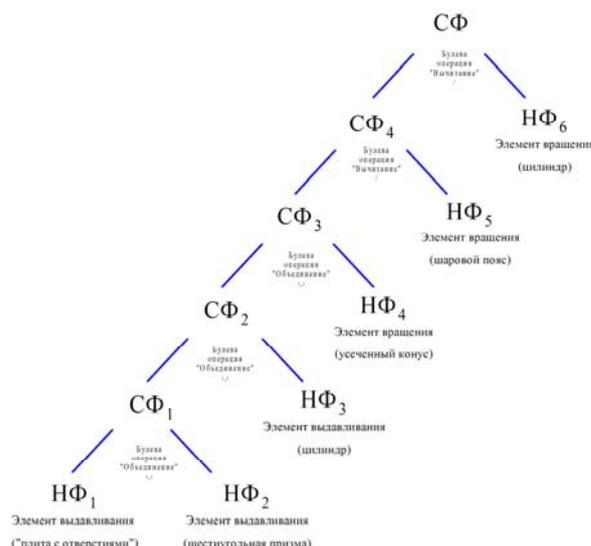


Рис. 7

ССЫЛКИ

1. Котов И.И., Полозов В.С., Широкова Л.В. Алгоритмы машинной графики. М., Машиностроение, 1977. – 250 с.
2. Полозов В.С., Ротков С.И., Широкова Л.В., Будеков О.А. Автоматизированное проектирование. Геометрические и графические задачи. – М., Машиностроение, 1983. – 280 с.
3. Ротков С.И. Разработка средств геометрического моделирования и компьютерной графики пространственных объектов для CALS-технологий. Дисс. докт. техн. наук, 05.01.01. – Нижний Новгород: ННГАСУ, 1999.
4. Тюрина В.А. Разработка методов преобразований каркасной модели в задаче синтеза образа 3D объекта по его проекциям. Дисс. канд. техн. наук, 05.01.01. – Нижний Новгород: ННГАСУ, 2003.
5. Кучуганов В.Н. Автоматический анализ машиностроительных чертежей. – Иркутск: Изд-во Иркутского университета, 1985. – 112 с.
6. Осипов А.А. Машинные методы проектирования непрерывно-каркасных поверхностей, М., Машиностроение, 1979. – 250 с.

АВТОМАТИЧЕСКОЕ ОБНАРУЖЕНИЕ ГЕОМЕТРИЧЕСКИХ ОШИБОК НА МАШИНОСТРОИТЕЛЬНЫХ 2D-ЧЕРТЕЖАХ

Д. Васин¹, С. Ротков²

¹ НИИ прикладной математики и кибернетики

Нижегородского государственного университета им. Н.И. Лобачевского

² Нижегородский государственный архитектурно-строительный университет

Аннотация

В статье рассмотрены алгоритмы автоматического обнаружения наиболее распространенных геометрических ошибок на машиностроительных 2D-чертежах.

Современные системы САПР широко представлены средствами, обеспечивающими моделирование объектов различной сложности в разных отраслях деятельности. Эти программные комплексы упрощают работу проектировщиков и систематизируют процесс создания цифровой модели изделия. Но при всей своей функциональности современное программное обеспечение САПР не учитывает один немаловажный факт. В этих системах предполагается, что разработчик должен быть квалифицированным и не допускать ошибки построения чертежей. Фактически же системой может пользоваться практически любой пользователь, не имеющий представления о требованиях ГОСТов к чертежам. С другой стороны, даже профессиональные разработчики допускают ошибки. Поэтому достаточно актуальной представляется задача автоматизации контроля метрической информации на машиностроительных чертежах.

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).

Описание структуры хранения векторных данных. Для хранения векторных данных и выполнения над ними различных операций предлагается следующий набор геометрических примитивов (ГП):

Примитив Point – точка: Point (int x, int y), где x, y – координаты в декартовой системе координат, связанной с чертежом.

Примитив Line – отрезок: Line (Point t1, Point t2, long Color, int Width, int View), где t1, t2 – координаты начала и конца отрезка; Color – цвет; Width – толщина; View – вид (пунктирная, сплошная и т.д.).

Примитив Polyline – ломаная линия: PolyLine (int N, Point point [N]), где N – число точек ломаной; point [N] – координаты узлов ломаной линии.

Примитив Circle – окружность: Circle (Point C, int R), где C – координаты центра окружности; R – радиус окружности.

Примитив Arc – дуга окружности: Arc (Point C, int R, float Angle1, float Angle2), где Point C – координаты центра окружности, которой принадлежит дуга; R – радиус дуги; Angle1, Angle2 – начальный и конечный углы дуги.

В настоящее время выделено три вида ошибочных ситуаций, наиболее распространенных при создании чертежа.

1. Ситуации совпадения / наложения

- 1.1. Ошибочное наложение отрезков.
- 1.2. Ошибочное наложение ломаных линий.
- 1.3. Ошибочное наложение дуг.
- 1.4. Ошибочное наложение окружностей.
- 1.5. Ошибочное наложение ломаной линии и отрезка.
- 1.6. Ошибочное наложение окружности и дуги.
- 1.7. Ошибочное наложение / переплетение контуров.

2. Ситуации угловых ошибок

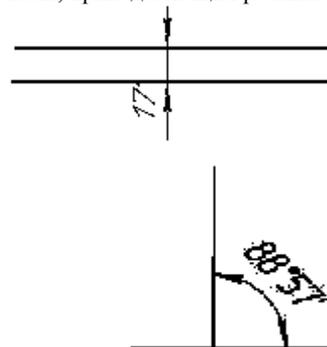
- 2.1. Нарушение ортогональности отрезков / ломаных линий.
- 2.2. Нарушение параллельности отрезков / ломаных линий.

3. Ситуации нарушения топологии элементов чертежа

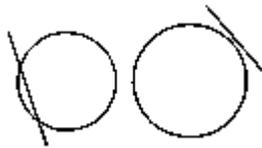
- 3.1. Нарушение топологии линейных ГО: отсутствие необходимого примыкания (пересечения) линейных элементов / ошибочное примыкание (пересечение) линейных элементов.
- 3.2. Нарушения топологии окружности и отрезков прямых.
- 3.3. Разрывы в контуре.

Ошибки 1-го вида возникают для всех представленных ГП. В процессе автоматического контроля необходимо учитывать, что наложение друг на друга одинаковых примитивов не всегда является ошибочным, а наложение различных ГП в большинстве случаев будет являться ошибкой, при этом для линейных ГП в зависимости от типа линии устанавливается факт допустимости наложения.

Ошибки 2-го вида возникают только для ГП Line и ГП Polyline. Они возникают не только между одинаковыми примитивами, но и между различными примитивами. Наиболее часто наблюдаются небольшие отклонения от состояний ортогональности или параллельности между двумя отрезками, принадлежащими разным линиям.



Ошибки 3-го вида возникают для всех ГП. Наиболее распространены ошибки отсутствия необходимой общей точки между различными ГП либо, наоборот, наличие недопустимой общей точки.



Существует вероятность неточного совмещения концевых точек линий. Линии должны образовывать замкнутый контур, то есть начальная и конечная точки, определяющие линию, обязательно должны накладываться на начальные и конечные точки другой линии или же должны накладываться на другую линию (принадлежать ей).



Рассмотрим алгоритмы обнаружения указанных ошибочных метрических состояний.

Алгоритм ProcessLineLine – находит и фиксирует ошибки 1.1, 2.1, 2.2, 3.1, возникающие с ГП «отрезок». Входные параметры алгоритма: Line1, Line2 – анализируемые ГП; tolerance – допустимая погрешность в градусах, для ортогональности \ параллельности ГП; tolerance_lenght – допустимое расстояние, для выявления ошибок типа 3.1; tolerance_lenght1 – допустимое расстояние для фильтрации независимых отрезков. Выходные параметры алгоритма: все анализируемые недопустимые состояния геометрии ГП помечаются спецсимволом для дальнейшего визуального анализа и исправления.

Задаем массив length [4] и заполняем его взаимными декартовыми расстояниями от концов одного отрезка до концов другого. Определяем минимальное (MnLength) из найденных расстояний и, сравнивая его с tolerance_lenght1, исключаем отрезки, лежащие далеко друг от друга. Ищем среди length[i] максимальное – MxLength. Определяем длины L₁ и L₂ обоих отрезков. Рассчитываем углы наклона первого и второго отрезков к оси Oх, и по разности этих углов находим угол между отрезками U.

Если $(90 - tolerance < U < 90 + tolerance) \vee (270 - tolerance < U < 270 + tolerance)$, то установлен факт ошибки ортогональности анализируемых отрезков; если $(0 - tolerance < U < 0 + tolerance) \vee (180 - tolerance < U < 180 + tolerance)$, то установлен факт ошибки параллельности анализируемых отрезков.

Если анализируемые отрезки лежат на одной прямой и при этом:

если $(L1 = MxLength \wedge L2 = MxLength)$, то установлен факт полного наложения отрезков, без учета их направления;

если $(U = 0 \wedge$ концы обоих отрезков совпадают по координатам), то установлен факт полного наложения сонаправленных отрезков;

если $(U = 180 \wedge$ начало одного отрезка является концом другого), то установлен факт полного наложения противоположно направленных отрезков.

Определяем взаимное расположение отрезков, лежащих на одной прямой:

если $(MxLength < L_1 + L_2 - tolerance)$, то установлен факт наложения отрезков;

если $(MxLength < (L_1 + L_2) \wedge MxLength \geq (L_1 + L_2) - tolerance)$, то установлен факт наложения отрезков, возможна ошибка типа ошибочное примыкание (пересечение) линейных элементов;

если $(MxLength = (L_1 + L_2))$, то отрезки имеют общую вершину;

если $(MxLength \leq (L_1 + L_2 + tolerance) \wedge MxLength > (L_1 + L_2))$, то отрезки независимы, возможна ошибка отсутствия необходимого примыкания (пересечения) линейных элементов;

если $(MxLength > (L_1 + L_2 + tolerance))$, то установлен факт независимости анализируемых отрезков.

В случае если анализируемые отрезки не лежат на одной прямой, находим t_ins – точку пересечения прямых, которым принадлежат отрезки, и вычисляем Q1_min, Q1_max – наименьшее и наибольшее расстояния от концов первого отрезка до t_ins, Q2_min, Q2_max – наименьшее и наибольшее расстояния от вершин второго отрезка до t_ins. Тогда:

если $((Q1_min+Q1_max) = 0)$, то флаг принадлежности Flag1 = истина, т.е. t_ins ∈ Line1;

если $((Q2_min+Q2_max) = 0)$, то флаг принадлежности Flag2 = истина, т.е. t_ins ∈ Line2;

если $(Flag1 = истина \wedge Flag2 = истина)$, то t_ins ∈ Line1 \wedge t_ins ∈ Line2.

Далее возможны 4 варианта ошибок:

если $(Flag1 = истина \wedge Flag2 = истина \wedge (Q1_min < tolerance_lenght \vee Q2_min < tolerance_lenght \vee MnLength < tolerance_lenght))$, то возможно ошибочное примыкание (пересечение) линейных элементов;

если $(Flag1 = истина \wedge Flag2 = лож \wedge (Q2_min < tolerance_lenght \vee MnLength < tolerance_lenght))$, то возможно отсутствие необходимого примыкания (пересечения) линейных элементов;

если $(Flag1 = лож \wedge Flag2 = истина \wedge (Q1_min < tolerance_lenght \vee MnLength < tolerance_lenght))$, то возможно отсутствие необходимого примыкания (пересечения) линейных элементов;

если $(Flag1 = лож \wedge Flag2 = лож \wedge (Q1_min < tolerance_lenght \vee Q2_min < tolerance_lenght \vee MnLength < tolerance_lenght))$, то возможно отсутствие необходимого примыкания (пересечения) линейных элементов.

Алгоритм ProcessCircleLine находит и фиксирует ошибку 3.2, возникающую между ГП «отрезок» и ГП «окружность». Входные параметры алгоритма: Line, Circle – анализируемые структурные элементы; tolerance – допустимая погрешность определения точки касания отрезка и окружности. Выходные параметры алгоритма: все анализируемые недопустимые состояния геометрии ГП помечаются спецсимволом для дальнейшего визуального анализа и исправления.

Будем считать, что Point C – координаты центра ГП «окружность», R – ее радиус, а Point t1, t2 – координаты ГП «отрезок» на плоскости. Тогда уравнение прямой, проходящей через точки t1 и t2, имеет вид $A*x - B*y + C = 0$, где $A = (t1.y - t2.y)$, $B = (t1.x - t2.x)$, $C = (t1.x*t2.y - t2.x*t1.y)$. Вектор с координатами N(A,B) – нормальный, тогда уравнение прямой, проходящей через центр окружности C и перпендикулярной к прямой, проходящей

$$\text{через ГП «отрезок»}: \frac{y - C.y}{t1.y - t2.y} = \frac{x - C.x}{t2.x - t1.x}.$$

Решая систему уравнений:

$$\begin{cases} (t1.y - t2.y) * x - (t1.x - t2.x) * y + (t1.x * t2.y - t2.x * t1.y) = 0 \\ \frac{y - C.y}{t1.y - t2.y} = \frac{x - C.x}{t2.x - t1.x} \end{cases}$$

находим точку t_{ins} пересечения прямой и перпендикуляра к ней, проходящего через центр окружности C .

Ищем расстояние от точки пересечения до центра

$$\text{окружности: } D = \sqrt{(C.x - t_{ins}.x)^2 + (C.y - t_{ins}.y)^2}$$

Тогда, если $((D - R > 0 \vee D - R < 0) \wedge (D - R \leq \text{tolerance}))$, то установлен факт ошибки.

Если t_{ins} не принадлежит отрезку, то определяем расстояния от центра окружности до вершин отрезка:

$$D1 = \sqrt{(C.x - t1.x)^2 + (C.y - t1.y)^2};$$

$$D2 = \sqrt{(C.x - t2.x)^2 + (C.y - t2.y)^2}.$$

Если $(D1 < D < D2) \vee (D2 < D < D1)$, то установлен факт пересечения отрезка и окружности.

Алгоритм ProcessCircleCircle находит и фиксирует ошибку 1.4, возникающую между ГП «окружность». Входные параметры алгоритма: Circle1, Circle2 – анализируемые ГП «окружность», Tolerance_center – допустимое расстояние между центрами окружностей Circle1 и Circle2, Tolerance_radius – допустимое значение, определяющее разницу радиусов окружностей Circle1 и Circle2. Выходные параметры алгоритма: все анализируемые недопустимые состояния геометрии ГП помечаются спецсимволом для дальнейшего визуального анализа и исправления.

Если $(\text{Circle1.C} = \text{Circle2.C} \wedge |\text{Circle1.R} - \text{Circle2.R}| < \text{Tolerance_radius})$, то установлен факт концентрического вложения окружностей.

Если $(\text{Circle1.C} = \text{Circle2.C} \wedge \text{Circle1.R} = \text{Circle2.R})$, то установлен факт полного совпадения окружностей.

$$D = \sqrt{(\text{Circle2.C.x} - \text{Circle1.C.x})^2 + (\text{Circle2.C.y} - \text{Circle1.C.y})^2}$$

– декартово расстояние между центрами окружностей. Если $(\text{Circle1.C} \neq \text{Circle2.C} \wedge D < \text{Tolerance_center})$, то установлен факт подозрительной близости центров окружностей.

Алгоритм ProcessArcArc – находит и фиксирует ошибки 1.3, возникающие с ГП «дуга», и является модификацией алгоритма ProcessCircleCircle. Входные параметры алгоритма: Arc1, Arc2 – анализируемые ГП «дуга»; Tolerance_center – допустимое расстояние между центрами дуг; Tolerance_radius – допустимое значение, определяющее разницу радиусов дуг. Выходные параметры алгоритма: все анализируемые недопустимые состояния геометрии ГП помечаются спецсимволом для дальнейшего визуального анализа и исправления. Положим, что $\text{Arc1.Angle1} \leq \text{Arc1.Angle2}$, а $\text{Arc2.Angle1} \leq \text{Arc2.Angle2}$

Если $(\text{Arc1.C} = \text{Arc2.C} \wedge |\text{Arc1.R} - \text{Arc2.R}| < \text{Tolerance_radius})$, то установлен факт возможной ошибки наложения окружностей.

Если $(\text{Arc1.R} = \text{Arc2.R} \wedge ((\text{Arc2.Angle1} \leq \text{Arc1.Angle1} \leq \text{Arc2.Angle2}) \vee (\text{Arc2.Angle1} \leq \text{Arc1.Angle2} \leq \text{Arc2.Angle2}) \vee (\text{Arc1.Angle1} \leq \text{Arc2.Angle1} \leq \text{Arc1.Angle2}) \vee (\text{Arc1.Angle1} \leq \text{Arc2.Angle2} \leq \text{Arc1.Angle2}))$, то установлен факт возможной ошибки наложения дуг.

Пусть D – расстояние между центрами окружностей, тогда, если $(\text{Arc1.R} \neq \text{Arc2.R} \wedge D < \text{Tolerance_center})$, то установлен факт подозрительной близости центров окружностей.

Алгоритм ProcessPolyline – находит и фиксирует ошибки типа 1.2, 1.7, возникающие с ГП «ломаная линия». Входные параметры алгоритма: Polyline1, Polyline2 – анализируемые ГП «ломаная линия». Выходные

параметры алгоритма: Alone – счетчик наложений точек, все анализируемые недопустимые состояния геометрии ГП помечаются спецсимволом для дальнейшего визуального анализа и исправления.

Осуществляя полный перебор точек обеих ломаных, выполняем:

если $(\text{Polyline1.point}[i] = \text{Polyline2.point}[j])$, то установлен факт совпадения точек ломаных,

если $(\text{Polyline1.point}[i-1] = \text{Polyline2.point}[j-1] \vee$

$\text{Polyline1.point}[i-1] = \text{Polyline2.point}[j+1] \vee$

$\text{Polyline1.point}[i+1] = \text{Polyline2.point}[j-1] \vee$

$\text{Polyline1.point}[i+1] = \text{Polyline2.point}[j+1])$, то установлен факт полного совпадения обеих ГП.

Иначе увеличение счетчика Alone.

Практическая проверка указанных алгоритмов на реальных машиностроительных чертежах выявила, что при соответствующем выборе входных параметров алгоритмами выявляется от 98 до 100% ошибочных состояний.

Физически корректный рендеринг прозрачных оптически активных изотропных кристаллов

Д. Козлов, В. Дебелов

Лаборатория численного анализа и машинной графики

Институт вычислительной математики и математической геофизики СО РАН,

Новосибирск, Россия

{kozlov, debelov}@oapmg.ssc.ru

Аннотация

Данная работа посвящена проблеме физически корректного рендеринга оптически активных кристаллов. Оптическая активность – это способность среды (кристалла) поворачивать плоскость поляризации, распространяющегося в ней луча по мере его прохождения в среде.

Одним из наиболее известных представителей класса оптически активных кристаллов является кварц, который проявляет оптическую активность будучи одноосным кристаллом. Однако существуют и изотропные кристаллы, которые являются оптически активными, например, бромат натрия.

К настоящему моменту данное оптическое явление находилось за рамками работ, посвящённых фотореалистическому рендерингу кристаллов. В данной статье предложен метод расчёта поворота плоскости поляризации линейно поляризованного луча света в прозрачных оптически активных изотропных средах. Разработанный метод предназначен для использования в алгоритмах лучевой трассировки, позволяет их авторам добавлять физически корректный учёт явления оптической активности.

Ключевые слова: фотореалистический рендеринг, физически корректный рендеринг, кристаллы, поляризация света, оптическая активность, дисперсия света.

1. ВВЕДЕНИЕ

В изотропных и анизотропных кристаллах наблюдается множество оптических эффектов: дисперсия света, двойное лучепреломление, коническая рефракция и плеохроизм. Все эти явления в настоящее время так или иначе рассмотрены в работах по фотореалистическому рендерингу кристаллов [1], [7], [9] и [12].

Оптическая активность также является одним из эффектов, проявляющихся в кристаллах, однако она не рассмотрена ни в одной из работ по фотореалистическому рендерингу кристаллов.

Среда называется оптически активной, если она обладает способностью поворачивать плоскость поляризации линейно поляризованного луча света, распространяющегося в ней, по мере его прохождения в среде.

Существуют работы, посвящённые моделированию оптической активности, например, [10], но эта работа относится, скорее, к области оптики, а не компьютерной графики, каких-либо изображений в ней не рассчитано.

2. ОПТИЧЕСКАЯ АКТИВНОСТЬ

Обычно оптическая активность возникает в двойко преломляющих кристаллах, например, кварц, киноварь, а также в жидкостях, например, скипидар или раствор сахара. Но существуют и изотропные оптически активные кристаллы, например, хлорат натрия, бромат натрия и др. [4].

Угол поворота ϕ плоскости поляризации луча света линейно зависит от расстояния, пройденного лучом [3]:

$$\phi = \alpha \cdot d, \quad (1)$$

где d – это расстояние пройденное лучом, а α – *постоянная вращения*, численно равная углу поворота плоскости поляризации луча света слоем вещества единичной толщины. Если при распространении луча в среде плоскость поляризации поворачивается по часовой стрелке (если смотреть навстречу лучу), то такая среда называется правовращающей, в противном случае – левовращающей.

Будем использовать для правовращающей среды положительные значения α , а для левовращающей – отрицательные.

Постоянная вращения существенно зависит от длины волны. В спектральных диапазонах, где среда прозрачна, постоянная вращения обратно пропорциональна квадрату длины волны (закон Био) [3]:

$$\alpha \sim 1 / \lambda^2.$$

Такая зависимость называется *вращательной дисперсией*, она описывается *спектром оптического вращения*. Если постоянная вращения увеличивается с уменьшением длины волны, то такая вращательная дисперсия называется нормальной, она наблюдается в областях спектра, где отсутствует поглощение. В областях поглощения зависимость обратная, такая вращательная дисперсия называется аномальной [5].

В общем случае значение постоянной вращения может быть аппроксимировано формулой Друде [3]:

$$\alpha = \sum_i \frac{A_i}{\lambda^2 - \lambda_i^2},$$

где константы A_i соответствуют длинам волн λ_i полос поглощения $i = 1, 2, 3, \dots$

3. МЕТОД ВИЗУАЛИЗАЦИИ

В сценах с оптически активными кристаллами поворот плоскости поляризации можно наблюдать при размещении кристалла между двумя поляризаторами.

Для моделирования поляризатора в данной работе используется прозрачный одноосный монокристалл кальцита с плоскими параллельными гранями, расположенный в вакууме (среда, заполняющая сцену).

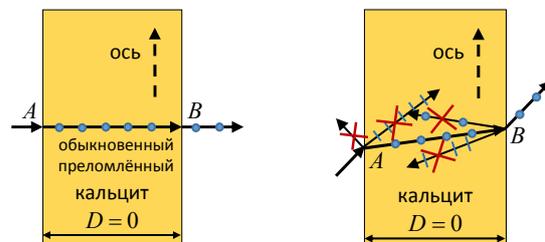


Рис. 1. Схема моделирования поляризатора

На рис. 1 приведена схема моделирования поляризатора. На схеме слева монокристалл падает неполяризованный луч. В результате взаимодействия с границей сред вакуум – кристалл в вакууме образуется отражённый частично поляризованный луч, а в кристалле из-за двойного лучепрелом-

ления – два отражённых: обыкновенный и необыкновенный. Последние два – линейно поляризованы. Как показано на рис. 1, оптическая ось кристалла (обозначена на рисунке как «ось») и падающий луч лежат в плоскости рисунка, а значит, обыкновенный луч будет поляризован перпендикулярно плоскости рисунка. Для обозначения плоскости поляризации линейно поляризованных лучей на рис. 1 использованы общепринятые обозначения кружками (точками) и штрихами. При расчёте образовавшиеся отражённый и необыкновенный преломлённый лучи отбрасываются (обозначены на схеме справа крестиками) и не участвуют в дальнейших вычислениях. Обыкновенный луч взаимодействует со второй границей сред кристалл – вакуум. При этом образуются два отражённых луча (обыкновенный и необыкновенный) и один преломлённый луч. Преломлённый луч будет параллелен исходному лучу, падающему на кристалл, и поляризован в плоскости перпендикулярной оптической оси и плоскости рисунка. Преломлённый луч выходит из поляризатора.

При моделировании поляризатора в рамках нашего эксперимента толщина кристалла D полагается равной 0, то есть точки A и B совпадут (рис. 1), а луч не смещается при взаимодействии с поляризатором, но при этом становится линейно поляризованным.

Такая схема позволяет корректно моделировать плоскость поляризации луча при прохождении поляризатора под углом, а также различную интенсивность поляризованного луча при падении луча на поляризатор под различными углами.

Для визуализации оптической активности рассмотрим следующую трёхмерную сцену. Сцена содержит прозрачный оптически активный изотропный монокристалл. Первый поляризатор наложен на плоский квадратный источник света, в результате он излучает линейно поляризованный свет. Второй поляризатор является поляризационным фильтром, который «устанавливается» на камеру-обскуру.

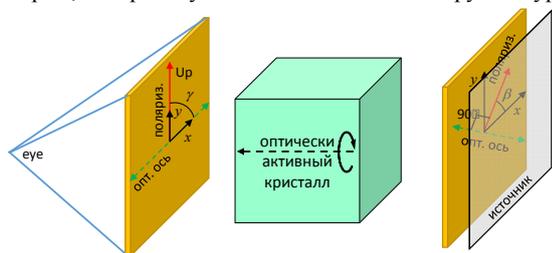


Рис. 2. Положение камеры № 1

Плоскость поляризации каждого фильтра задаётся углом относительно оси x декартовой системы координат, заданной в плоскости поляризатора (рис. 2), который будем называть *углом поляризации*. Для поляризатора, наложенного на камеру, ось y системы координат совпадает с U_p вектором камеры, а ось x перпендикулярна ей. Для поляризатора, наложенного на источник света, оси системы координат параллельны сторонам источника. Угол поляризации поляризатора на источнике света обозначим β , а угол поляризации поляризационного фильтра камеры обозначим γ .

Таким образом, для положения камеры № 1 (см. рис. 2) условие $|\beta - \gamma| = 90^\circ$ соответствует скрещенным поляризаторам, при этом свет, излучённый источником и непрошедший через кристалл, будет полностью погашен поляризационным фильтром на камере. Условие $\beta = \gamma$ соответствует сонаправленным поляризаторам; при этом свет, излучённый источником и непрошедший через кристалл, не будет гаситься поляризационным фильтром камеры. Ин-

тенсивность луча света, прошедшего через кристалл, в обоих случаях будет определяться пройденным лучом расстоянием в кристалле, так как длина пути в кристалле определяет угол поворота плоскости поляризации луча света, согласно формуле (1).

В численных экспериментах, проведённых в данной работе $\beta = \gamma = 90^\circ$, поэтому источник света виден на рассчитанных изображениях. Оптические оси поляризаторов на рис. 2 и 3 обозначены зелёными стрелками, а направления поляризации – красными.

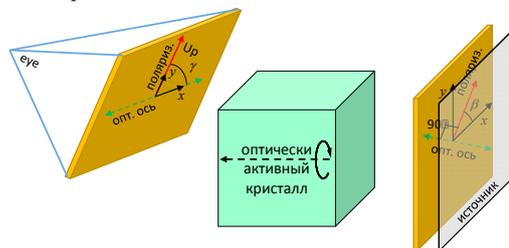


Рис. 3. Положение камеры № 2

Если плоскости поляризационного фильтра, поставленного на камеру, и поляризатора, установленного на источник, не параллельны, что соответствует положению камеры № 2, приведённого на рис. 3, то источник света не будет гаситься полностью даже при условии $|\beta - \gamma| = 90^\circ$. Тем не менее, при небольших отклонениях от ракурса № 1 гашение будет приемлемым для проведения численных экспериментов. В данной работе рассматривается именно такое небольшое отклонение ракурса камеры.

4. ЧИСЛЕННЫЕ ЭКСПЕРИМЕНТЫ

Для проведения численных экспериментов использовалась библиотека LIAC [2], предназначенная для физически корректного расчёта взаимодействия луча света с изотропными и анизотропными кристаллами.

Геометрия кристалла – это куб с длиной ребра 1.6 см. Куб стоит на источнике белого света CIE D65, то есть со спектром излучения абсолютно чёрного тела с температурой 6500K. Внешнее освещение имеет спектр, равный 40% от CIE D65. На источник света виртуально (нет зазора, толщина поляризатора равна 0) наложен поляризатор в соответствии со схемой на рис. 2 и 3, поэтому источник света излучает поляризованный свет. На рис. 4 приведены изображения модели сцены с двух ракурсов. Слева для положения камеры № 1, справа для положения камеры № 2.

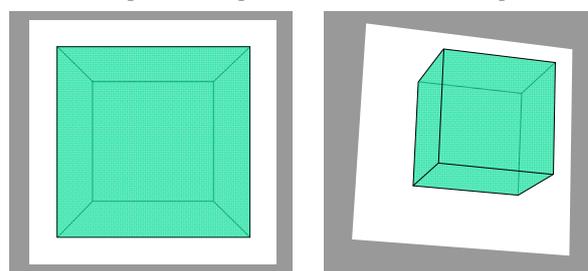


Рис. 4. Схематические изображения модели сцены для визуализации явления оптической активности

Для проведения экспериментов используется спектр оптического вращения кристалла кварца, который хорошо аппроксимируется формулой, предложенной в работе [6]:

$$\alpha = \frac{k\lambda^2}{(\lambda^2 - \lambda_0^2)^2},$$

где $k = 7.19$ и $\lambda_0 = 0.0926283$ мкм, а значение α – в градусах на миллиметр.

В качестве спектра преломления используется спектр преломления кварца, который может быть рассчитан по формуле Селлмейера с коэффициентами из работы [8]:

$$n_{o,e}^2(\lambda) = A_{o,e} + \frac{B_{o,e}\lambda^2}{\lambda^2 - C_{o,e}} + \frac{D_{o,e}\lambda^2}{\lambda^2 - E_{o,e}},$$

где значение длины волны λ в микрометрах, подстрочный индекс o соответствует обыкновенной волне, а индекс e – необыкновенной, значения констант следующие:

$$A_o = 1.28604141, B_o = 1.07044083, C_o = 1.00585997 \cdot 10^{-2},$$

$$D_o = 1.10202242, E_o = 100,$$

$$A_e = 1.28851804, B_e = 1.09509924, C_e = 1.02101864 \cdot 10^{-2},$$

$$D_e = 1.15662475, E_e = 100.$$

Поскольку в данной работе рассматриваются изотропные кристаллы, а кристалл кварца является одноосным, то коэффициенты преломления n для изотропного кристалла рассчитываются как среднее значение между коэффициентами преломления кварца для обыкновенной n_o и необыкновенной n_e волны:

$$n(\lambda) = (n_o(\lambda) + n_e(\lambda)) / 2.$$

Для изотропной прозрачной оптически активной среды введём следующее описание:

$$\text{Medium} = \{SP_{ref}, SP_{rot}\},$$

где SP_{ref} – спектр преломления; SP_{rot} – спектр оптического вращения, спектры заданы на видимом диапазоне длин волн от 380 до 780 нм.

Состояние поляризации луча описывается матрицей когерентности и системой координат луча, в которой она задана. Для пересчёта матрицы когерентности при повороте плоскости поляризации на угол ϕ применяется следующая матрица модификации когерентности из работы [11]:

$$M_{rotate} = \begin{pmatrix} \cos \phi & \sin \phi \\ -\sin \phi & \cos \phi \end{pmatrix}.$$

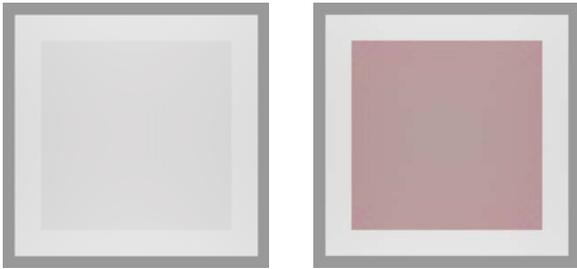


Рис. 5. Рассчитанные изображения изотропного оптически неактивного (слева) и оптически активного монокристалла

На рис. 5 приведены результаты первого численного эксперимента – рассчитанные изображения двух прозрачных кристаллов с ракурса камеры № 1. Слева приведено изображение оптически неактивного кристалла, а справа – оптически активного кристалла. Оба кристалла имеют одинаковые спектры преломления. Хорошо заметно, что оптически активный кристалл при наблюдении в поляризованном свете приобрёл красную окраску, а оптически неактивный остался бесцветным.

На рис. 6 показаны результаты второго эксперимента: изменение окраски кристалла, возникающей из-за оптической активности, в зависимости от размеров кристалла. Слева на рис. 6 приведено изображение описанного выше оптического активного кристалла с ракурса камеры № 2. Справа изображен кристалл с теми же оптическими характери-

ками, но вдвое меньших размеров. Центры обоих кристаллов расположены в одной точке. На изображениях хорошо заметно различие окрасок этих кристаллов.

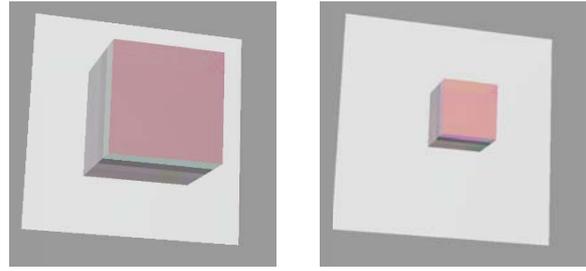


Рис. 6. Различие окрасок, вызванное оптической активностью, для различных размеров кристалла

Третий численный эксперимент моделирует ситуацию, когда кристалл либо очень крупный, либо его постоянная вращения очень большая. В данном случае значительно увеличен спектр оптического вращения, он равен спектру оптического вращения кристалла кварца, умноженному на 75. При таком спектре оптического вращения плоскость поляризации линейно поляризованного монохроматического луча с длиной волны 580 нм при прохождении им в кристалле расстояния, равного длине ребра кристалла, поворачивается на угол, примерно равный 300×90 градусов.

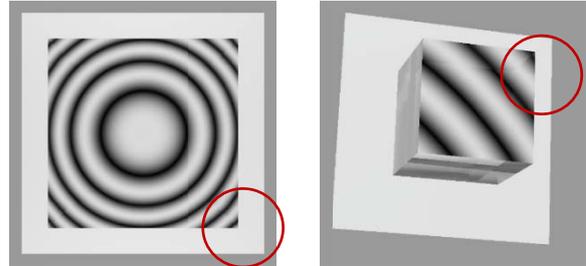


Рис. 7. Рассчитанные изображения изотропного оптически активного монокристалла в монохроматическом свете с длиной волны 580 нм

Рассчитанные изображения такого кристалла с ракурсов камер № 1 и 2 в монохроматическом свете с длиной волны 580 нм приведены на рис. 7. На изображениях хорошо заметны чередующиеся светлые и тёмные полосы, которые образуются вследствие различной длины пробега луча света в кристалле для направлений, соответствующим различным пикселям изображения.

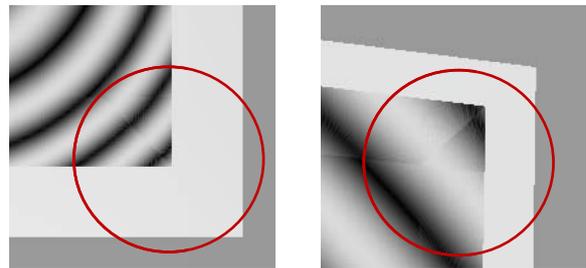


Рис. 8. Фрагменты рассчитанных изображений изотропного оптически активного монокристалла

Небольшие сдвиги частей полос (на рис. 7 и 8 отмечены красным кругом, на рис. 8 они показан крупнее) относительно друг друга в углах изображений кристаллов объясняются разным изменением поляризации лучей света при отражении от разных боковых граней (см. схемы сцены на рис. 4).

На рис. 9 приведены полноцветные изображения прозрачного изотропного кристалла, рассчитанные на видимом диапазоне спектра от 380 до 780 нм с шагом 20 нм.

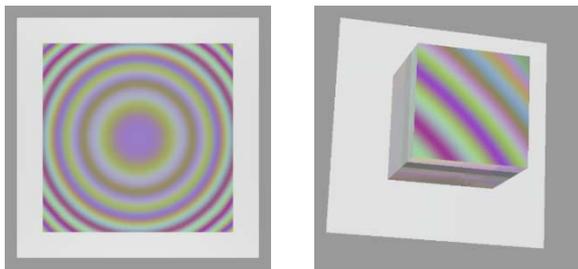


Рис. 9. Полноцветные изображения изотропного оптически активного монокристалла

На изображениях хорошо заметны переливы цветов, возникающие в основном из-за дисперсии вращения, так как на источнике света отсутствуют какие-либо контрастные переходы цветов, необходимые для наблюдения обычной (линейной) дисперсии света (см. рис. 5, слева).

5. ЗАКЛЮЧЕНИЕ

В данной работе выполнено моделирование оптической активности в прозрачных изотропных кристаллах. Рассчитаны изображения, которые продемонстрировали существенное влияние оптической активности на фотореалистические изображения кристаллов, в сценах, содержащих источники поляризованного света и в случае присутствия на камере поляризационного фильтра.

Отметим, что поляризованный свет в сцене может возникать и по естественным причинам, например, анизотропные кристаллы являются естественными поляризаторами, а поляризационным фильтром может служить какой-либо кристалл, расположенный между камерой и сценой.

На рассчитанных изображениях были выявлены особенности (см. рис. 8), полезные для проверки корректности расчёта изменения поляризации луча света при отражении и преломлении на границах кристаллов.

6. БЛАГОДАРНОСТИ

Данная работа поддержана РФФИ по грантам № 12-07-00386 и № 12-07-00391.

7. ССЫЛКИ

- [1] Дебелов В.А., Козлов Д.С. Физически корректный алгоритм рендеринга анизотропных поглощающих кристаллов // Труды 23-й Международной конференции по компьютерной графике и зрению ГрафиКон-2013, 16-20 сентября 2013, ИАиПУ ДВО РАН, ДВФУ. – Владивосток, Россия. – С. 191 – 194.
- [2] Козлов Д.С. Библиотека LIAC: расчёт взаимодействия луча света с кристаллами, ФАП СО РАН, рег. PR14011, 2014. – <http://fap.sbras.ru/node/4089>.
- [3] Ландсберг Г.С. Оптика: учебное пос. для вузов. – 6-е изд. – М: ФИЗМАТЛИТ, – 2003.
- [4] Работа 8. Вращение плоскости поляризации. <http://physfak.org/radio/optics/8.pdf>
- [5] Чугаев Л.А. Избранные труды. Т. 2. Об аномальной вращательной дисперсии (предварительное сообщение) (1909). – М: Издательство АН СССР, 1955.
- [6] Chandrasekhar S. The optical rotatory dispersion of quartz. // Proceedings of the Indian Academy of Sciences - Section A, – Vol. 45, – No. 3, – 1957. – P. 147 – 160.
- [7] Debelov V.A., Kozlov D.S. A Local Model of Light Interaction with Transparent Crystalline Media. // IEEE Transactions on visualization and computer graphics. – Vol. 19, No. 8. – 2013. – P. 1274 – 1287.

- [8] Ghosh G., Dispersion-equation coefficients for the refractive index and birefringence of calcite and quartz crystals. // Optics Communications. – Vol. 163, – No. 1–3, – May 1999. – P. 95 – 102.
- [9] Guy S., Soler C. Fast and Physically-Based Rendering of Gemstones. // Comp. Graphics (SIGGRAPH '03 Proceedings), ACM Press.
- [10] McClain S. C., Chipman R.A. Polarization ray tracing in anisotropic optically active media II, theory and physics // Applied Optics, vol. 10, no. 11. – P. 2383 – 2393 (Nov. 1993).
- [11] Tannenbaum, D.C., Tannenbaum T., Wozny M.J. Polarization and birefringency considerations in rendering // Comp. Graphics. – 1994. – P. 221 – 222.
- [12] Weidlich A., Wilkie A. Realistic Rendering of Birefringency in Uniaxial Crystals // ACM Transactions on Graphics, Vol. 27, No. 1, 2008. – P. 6:1 – 6:12.

Об авторах

Козлов Дмитрий – инженер лаборатории численного анализа и машинной графики Института вычислительной математики и математической геофизики СО РАН.

E-mail: kozlov@oapmg.sscc.ru.

Дебелов Виктор – ведущий научный сотрудник лаборатории численного анализа и машинной графики Института вычислительной математики и математической геофизики СО РАН.

E-mail: debelov@oapmg.sscc.ru.

This paper is devoted to the problem of photorealistic rendering of optically active crystals. Optical activity (rotation) is turning of plane of linearly polarized light while it travels through an optically active medium.

Quartz is one of well-known optically active crystals. It demonstrates optically activity being an optically uniaxial crystal. However, there are optically active isotropic crystals, e.g. sodium bromate.

Up to this paper, the phenomenon was out of scope of papers devoted to photorealistic rendering of crystals. In this paper, we present a method for physically correct calculation of turning of linearly polarized light polarization plane in transparent optically active isotropic crystals. The method is primarily intended to be used in ray tracing algorithms allows to their authors computing optical rotation phenomenon in physically correct way.

Keywords: *Photorealistic rendering, physically based rendering, crystals, light polarization, optical rotation, optical dispersion.*

Моделирование и визуализация 3D виртуальных пультов управления в тренажерных системах

М. Михайлюк, М. Торгашев

Научно-исследовательский институт системных исследований РАН, Москва, Россия

mix@niisi.ras.ru, mtorg@mail.ru

Аннотация

В статье рассматривается технология моделирования и визуализации трехмерных виртуальных пультов с использованием элементов виртуальной реальности. Технология включает в себя создание пультов с помощью системы трехмерного моделирования 3D MAX, визуализацию в стерео режиме, моделирование воздействия пользователя на элементы управления, моделирование движения этих элементов в результате воздействия, а также создание и расчет функциональной схемы пульта управления.

Ключевые слова: виртуальные пульты управления, визуализация, виртуальная реальность, тренажеры

1. ВВЕДЕНИЕ

Пульты управления используются во многих научных и производственных комплексах и процессах. К их числу относятся электростанции, летательные аппараты, производственные технологические линии, автоматизированные системы управления, станки, различные приборы (измерительные, диагностические, медицинские и др.) и т.д. При имитационном моделировании таких комплексов возникает задача моделирования входящих в них пультов управления. Очень часто для обучения персонала работе на сложных комплексах возникает необходимость создания соответствующих тренажерных систем. При этом во многих областях, связанных с опасной окружающей средой (космос, радиоактивно зараженные местности и т.д.), приходится использовать видео тренажеры, в которых обстановка и управляемые объекты моделируются и визуализируются на компьютерах и соответствующих средствах отображения. Поэтому во многих тренажерах также приходится моделировать пульты управления. Наконец, при разработке новых пультов управления возникают задачи эргономики – удобства пользования этим пультом, его достаточная информативность и т.д.

Для всех задач такого рода предлагается использовать виртуальные пульты управления, т.е. пульты, которые визуализируются на экране монитора, но являются интерактивными, т.е. пользователь может не только изучить их назначение и структуру, но и воздействовать на их элементы управления и осуществлять, таким образом, управление виртуальными или реальными объектами или процессами.

Двумерные виртуальные пульты управления достаточно распространены. Это виртуальные микшерные пульты для работы с музыкой, виртуальные пульты электронных очередей в банках и социальных службах, виртуальные пульты на планшетах или мобильных телефонах для управления телевизором или другими бытовыми приборами и т.д. Двумерные пульты используются также в компьютерных симуляторах, обучающих и тренажерных системах. Примерами могут служить авиа симуляторы, в которых

кабина летчика содержит виртуальный пульт, тренажер оператора прокатного стана, созданный ООО «Большие системы» с использованием движка Unity3D и др. Исследования в области 3D интерфейса также активно ведутся в настоящее время. Одним из типов таких интерфейсов являются трехмерные пульты управления, которые предполагают использование технологий виртуальной реальности. В НИИСИ РАН разработан комплекс моделирования и визуализации 2D и 3D виртуальных пультов управления. Двумерные пульты описаны в [2], в настоящей статье мы предлагаем технологию работы с 3D виртуальными пультами управления.

2. СОЗДАНИЕ И ВИЗУАЛИЗАЦИЯ ТРЕХМЕРНЫХ ПУЛЬТОВ

В общем случае пульт управления содержит подложку, управляющие элементы различных типов (кнопки, тумблеры, джойстики, переключатели, регуляторы и т.д.), сигнальные лампочки, информационные транспаранты и другие элементы. Трехмерные пульты управления можно создавать в любой системе трехмерного моделирования. Мы выбрали для этой цели систему 3D MAX, однако это не существенно, так как из любой другой системы виртуальную сцену можно конвертировать в систему 3D MAX. Если моделируется реальный пульт управления, то для построения модели естественно использовать текстуры, полученные из фотографий пульта и его отдельных элементов. Если

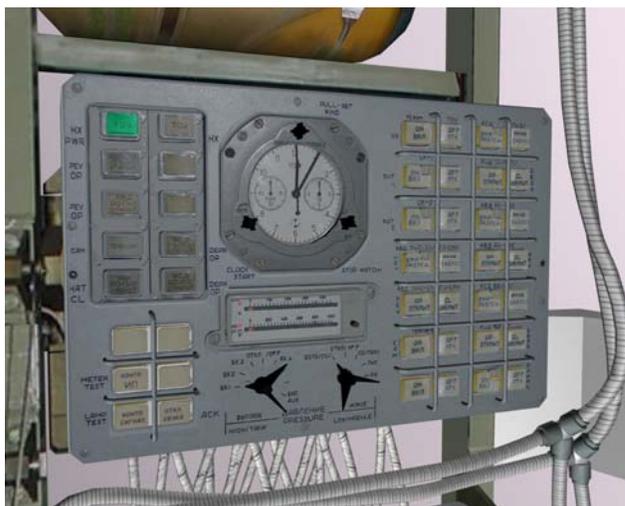


Рис. 1. Модель пульта управления модуля СО-1

создается новый пульт, то его состав и оформление определяется задачами, для которых он предназначен. При этом возможна постановка задачи исследования эргономики пульта. Например, при нажатии на кнопку рука оператора может закрывать соответствующую сигнальную лампочку, что неудобно. Естественно, элементы управления

необходимо создавать как отдельные трехмерные объекты, которые могут осуществлять движение при воздействии на них.

Визуализация 3D пультов осуществляется с помощью системы визуализации трехмерных виртуальных сцен. Мы используем для этой цели разработанную в НИИСИ РАН систему GLView. На рис. 1 показан пример визуализации пульта управления космического модуля СО-1.

3. МОДЕЛИРОВАНИЕ ИНТЕРАКТИВНОГО ВОЗДЕЙСТВИЯ НА ЭЛЕМЕНТЫ УПРАВЛЕНИЯ

Воздействие на элементы управления 3D пульта можно осуществлять несколькими способами.

Самый простой способ – с помощью компьютерной мыши. При наведении курсора мыши на объект-элемент управления можно определить, что это за объект, а затем по направлению движения мыши и способу нажатия ее кнопки (одиночное, двойное, нажатие с перемещением и т.д.) понять тип воздействия [2].

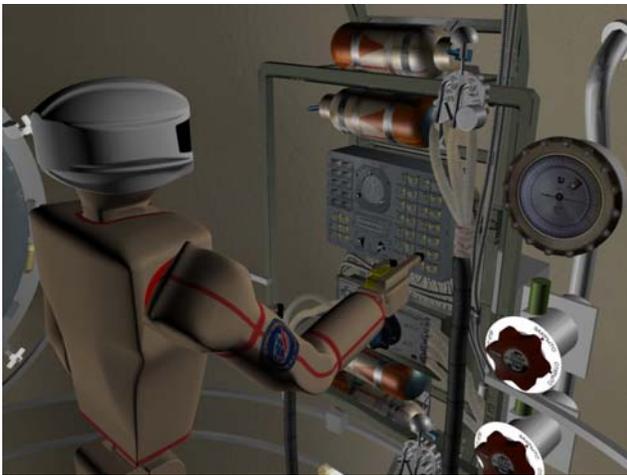


Рис. 2. Нажатие на кнопку пульта управления

Другим способом является воздействие на виртуальный элемент управления с помощью другого виртуального объекта. Обычно этим объектом является виртуальная рука оператора, манипулятора или робота. На рис. 2 показан пример нажатия на кнопку пальцем робота, управляемого оператором с помощью экзоскелета [1]. В этом случае для обработки воздействия на элемент управления необходимо использовать обработку коллизий, т.е. пересечений виртуального пальца с виртуальным элементом управления. Для обеспечения реального масштаба времени обычно вместо этих объектов рассматривают аппроксимирующие параллелепипеды (боксы) и определяют пересечение этих боксов (рис. 3).

Третий способ воздействия на элементы управления использует технологию виртуальной реальности, включающую в себя стерео режим визуализации, систему трекинга и компьютерные перчатки. В стерео режиме с использованием стерео очков оператору кажется, что пульт управления находится между ним и экраном. Компьютерная перчатка отслеживает положение руки и пальцев оператора в своей системе координат. Совмещая систему координат виртуальной сцены (включающей пульт управления) с системой координат перчатки и учитывая стерео режим,

можно вычислить взаимное положение пальца оператора и элемента управления. Далее, аналогично предыдущему случаю определяется направление движения пальца и факт его коллизии с управляющим элементом. Естественно, что при перемещении головы оператора изображение

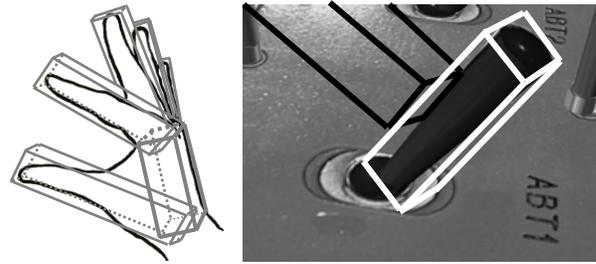


Рис. 3. Ограничивающие параллелепипеды пальцев руки и тумблера

виртуальной сцены также должно меняться. Для этого используется система трекинга головы, которая в масштабе реального времени определяет положение и ориентацию головы оператора и передает эти данные в систему визуализации. В НИИСИ РАН создана оригинальная оптическая система трекинга, основанная на использовании одной камеры и нескольких светодиодных маркеров, которые крепятся на голове оператора. Выделяя в каждом кадре визуализации маркеры и определяя их взаимное расположение на изображении, можно однозначно восстановить их расположение в трехмерном пространстве, а значит определить положение и ориентацию головы оператора. На рис. 4 показан пример использования этой технологии.

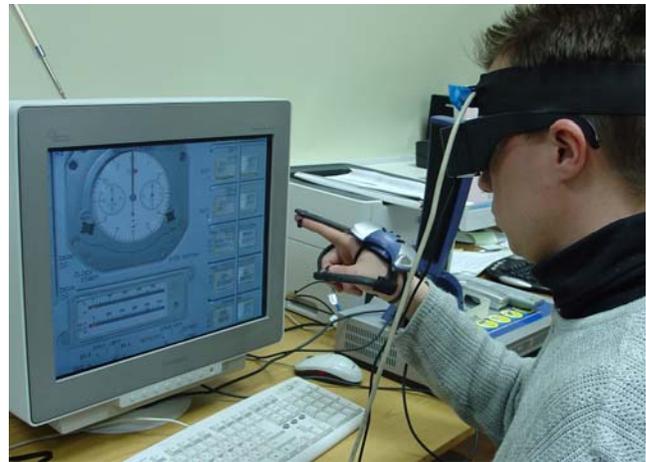


Рис. 4. Оператор нажимает рукой на виртуальные кнопки

При воздействии на элемент управления последний должен двигаться в соответствии с движением руки оператора и в соответствии со своими свойствами. Например, при нажатии на однопозиционную кнопку она должна утапливаться, а при отпуске руки – возвращаться в первоначальное положение. При этом если нажатие было проведено до конца, то кнопка срабатывает (т.е. включает или выключает некоторое действие). Если же было осуществлено легкое нажатие, то кнопка возвращается без срабатывания.

Моделирование динамики движения элементов управления также разработано в рамках данного комплекса.

Следующим важным шагом является обеспечение обратной связи, когда оператор «чувствует», что дотронулся до элемента управления и даже приложил некоторое усилие для его перемещения. Кроме того, он каким-то образом должен быть лишен возможности продолжать движение пальца, если элемент управления достиг своего крайнего положения. Для этого можно использовать компьютерные перчатки с обратной связью, однако в данной работе эти вопросы не рассматриваются.

4. ВЫЧИСЛЕНИЕ УПРАВЛЯЮЩИХ СИГНАЛОВ

Результатом работы виртуального пульта в каждый момент времени является вектор управляющих сигналов, которые затем можно передать (через файл или по сети) в объекты управления. Например, при управлении электрическим двигателем с помощью джойстика управляющим сигналом может быть величина напряжения, подаваемая на этот двигатель. Для вычисления управляющих сигналов пульта управления используются так называемые функциональные схемы. Функциональная схема состоит из функциональных блоков, имеющих некоторое число входов и некоторое число выходов. Каждый блок реализует одну или несколько функций. Это могут быть арифметические функции (сумматор, произведение, квадратный корень и т.д.), логические (конъюнкция, дизъюнкция, сравнение и т.д.), тригонометрические (синус, арктангенс и т.д.), цифровые (триггеры, мультиплексоры и т.д.), формователи (генераторы сигналов, мультивибраторы, таймеры и т.д.), динамические (дифференциатор, интегратор, апериодическое звено, задержка, нелинейный элемент и т.д.) и другие функции. Для создания функциональной схемы предназначен специально разработанный интерактивный редактор. В его поле редактирования можно перетаскивать пиктограммы функциональных блоков из обширной библиотеки и соединять их входы и выходы. Некоторые оставшиеся свободными входы присоединяются к элементам управления пульта, некоторые свободные выходы являются выходами функциональной схемы. На рис. 5 показан пример создания функциональной схемы в редакторе.

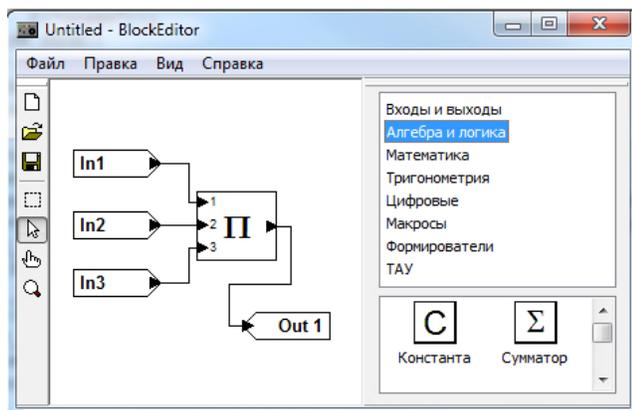


Рис. 5. Редактор функциональных схем

Находясь в каком-либо состоянии, каждый элемент управления генерирует управляющий сигнал. Например, включенная кнопка может генерировать целочисленную единицу, а выключенная – целочисленный ноль. Ползунок или регулятор могут генерировать вещественное число от 0

до 1, показывающее положение ползунка (регулятора) относительно начальной позиции. Управляющие сигналы от всех элементов управления попадают на входы функциональной схемы, которая производит вычисление значений на своих выходах. Эти значения (как было отмечено выше) передаются в подсистему динамики для управления динамическими объектами. Вычисления в схеме должны выполняться в масштабе реального времени, поэтому для увеличения скорости на каждом шаге производится пересчет только тех частей схемы, в которых входы изменили свое значение.

5. ЗАКЛЮЧЕНИЕ

В работе предложена технология создания и работы с 3D виртуальными пультами управления, включающая в себя создание виртуального пульта с помощью системы моделирования 3D MAX, визуализацию пульта в стерео режиме, моделирование воздействия оператора (с помощью компьютерной перчатки и системы трекинга) на элементы управления, создание функциональной схемы пульта и ее расчет (вычисление управляющих сигналов) в масштабе реального времени. Данная технология реализована в виде программных модулей, встроенных в комплекс тренажерных средств управления сложными динамическими системами, разработанными в НИИСИ РАН. Данный комплекс с успехом используется в ФГБУ «НИИ ЦПК им. Ю.А. Гагарина» для тренировок космонавтов.

6. БЛАГОДАРНОСТИ

Данная работа выполнена в рамках Программы фундаментальных исследований ОНИТ 1, проект № 2.9.

7. ССЫЛКИ

- [1] Бурдин Б.В., Михайлюк М.В., Сохин И.Г., Торгашев М.А. Использование виртуальных 3D-моделей для экспериментальной отработки бортовых полетных операций, выполняемых с помощью антропоморфных роботов // Робототехника и техническая кибернетика, 2013. – № 1. – С. 42 - 46.
- [2] Михайлюк М.В. Двумерные виртуальные пульта управления в тренажерных комплексах // Программная инженерия, 2014. – № 5. – С. 20 - 25.

Сведения об авторах

Михайлюк Михаил – д. ф.м. н., профессор, Научно-исследовательский институт системных исследований РАН, Email: mix@niisi.ras.ru.

Торгашев Михаил – зав. сектором, Научно-исследовательский институт системных исследований РАН, Email: mtorg@mail.ru.

Эффективная реализация HDR визуализации на GPU

П. Тимохин, М. Торгашев

Научно-исследовательский институт системных исследований РАН, Москва, Россия
webpismo@yahoo.de, mtorg@mail.ru

Аннотация

В статье рассматривается задача синтеза изображений виртуальных сцен в расширенном диапазоне в системах визуализации для имитационно-тренажерных комплексов и обучающих систем. Для ее решения предложен новый эффективный метод для реализации HDR технологии, использующий параллельные вычисления на GPU и обеспечивающий визуализацию в режиме реального времени. Технология использует модифицированный глобальный тональный оператор Рейнхарда с оригинальным алгоритмом вычисления глобальных яркостей кадра и обеспечивает моделирование эффекта «заплывания» ярких областей изображения.

Ключевые слова: визуализация, расширенный диапазон, тональный оператор, видеотренажерные системы, параллельные вычисления, GPU.

1. ВВЕДЕНИЕ

Одной из основных задач систем визуализации является расчет освещенности виртуальных объектов от источников света. В системах визуализации реального времени наблюдается тенденция к использованию более сложных моделей освещения, основанных на реальных физических законах, например BRDF. Предпочтительным вариантом является использование реальных фотометрических параметров для источников света и материалов. Освещенность реальных сцен может иметь огромный динамический диапазон. Стандартный подход, при котором расчет освещенности выполняется в диапазоне с глубиной представления цвета 8 бит на каждый цветовой канал (Low Dynamic Range, LDR), не позволяет получить приемлемый результат для подобных сцен. В последние годы на рынке начали появляться HDR (High Dynamic Range) мониторы, которые имеют больший динамический диапазон и поддерживают HDR форматы передачи с большей глубиной представления цвета, однако распространение таких мониторов до настоящего момента очень ограничено. В связи с этим все более актуальной задачей становится преобразование изображения из расширенного диапазона HDR в диапазон LDR. Такое преобразование имеет название тонального оператора, задачей которого является приведение изображения из расширенного диапазона в стандартный таким образом, чтобы добиться наиболее естественного отображения, близкого к тому, которое наблюдается в реальной обстановке. При реализации тонального оператора в системе визуализации для тренажерных и обучающих систем ключевым требованием является соблюдение режима реального времени синтеза изображений (с частотой не менее 25 кадров в секунду). Это ограничивает возможность использования сложных тональных операторов и требует максимально эффективного использования возможностей современных аппаратных средств, в частности параллельных вычислений на GPU. Одним из хорошо зарекомендовавших себя тональных операторов, широко используемым в системах визуализации, является глобальный оператор

Рейнхарда [1]. В исследованиях, оценивающих качество работы различных операторов, этот алгоритм занимает достойное место и в ряде случаев имеет преимущества в субъективном качестве получаемых изображений над другими, в том числе более сложными операторами. Одной из востребованных задач визуализации в расширенном диапазоне также является синтез эффектов, возникающих в средствах наблюдения, таких, как «заплывание» изображения ярких областей, блики и ореолы. Эффект «заплывания» (Bloom) изображения моделирует реальный физический процесс, при котором области изображения с избыточной яркостью засвечивают окружающие участки изображения. Например, для фото и видео камер физическая природа эффекта состоит в том, что заряд перетекает с пересвеченной ячейки светочувствительной матрицы на соседние, в результате чего яркие области «заплывают».

В последнее время возможность HDR визуализации активно интегрируется, как в приложения для построения фотореалистичных изображений (см., например, [3]), так и в графические движки и приложения виртуальной реальности (см., например, [2]). В данной работе предлагается новая реализация HDR технологии в системе визуализации с использованием параллельных вычислений на GPU. Технология реализована на базе библиотеки OpenGL и шейдерного языка GLSL, использует глобальный тональный оператор Рейнхарда с модифицированным алгоритмом вычисления глобальных яркостей кадра и обеспечивает моделирование эффекта «заплывания» ярких областей изображения. В сравнении с существующими подходами предложена модификация тонального оператора, обеспечивающая лучшее воспроизведение изображений с большой площадью темных участков, а также эффективный алгоритм расчета глобальных яркостей кадра на GPU.

2. ТЕХНОЛОГИЯ СИНТЕЗА ИЗОБРАЖЕНИЙ ВИРТУАЛЬНЫХ СЦЕН В HDR РЕЖИМЕ

В данной работе HDR режим визуализации реализуется путем поэтапной попиксельной обработки изображения сцены, синтезированного в расширенном диапазоне (высокодиапазонное изображение). Предлагаемая реализация основана на синтезе и обработке текстур на многоядерном графическом процессоре (GPU) с помощью разработанных шейдерных программ (см. схему). Рассмотрим основные этапы предлагаемой технологии.

Подготовка к визуализации. Для обеспечения расширенного диапазона яркостей в виртуальной сцене у источников освещения устанавливаются интенсивности, соответствующие их реальным прототипам, а перед визуализацией сцены в графическом конвейере отключается автоматическое отсечение компонент цвета по границам диапазона [0,1].

Синтез высокодиапазонного изображения сцены. Синтез высокодиапазонного изображения сцены выполняется путем визуализации виртуальной сцены в текстуру rgb-формата, каждый цветовой канал которой задается 32-битным

вещественным числом. Для этого используются две такие текстуры размера, совпадающего с размером кадра, в одной из которых для каждого тексела хранятся сэмплы. Виртуальная сцена визуализируется в первую текстуру (с сэмплами), из которой изображение копируется во вторую текстуру так, что цвет C_{HDR} каждого тексела второй текстуры вычисляется путем интерполяции соответствующих ему сэмплов первой текстуры.



Схема визуализации сцены в HDR режиме

Вычисление глобальных яркостей изображения. На основе синтезированного высокодиапазонного изображения сцены вычисляются средняя L_{cp} , минимальная L_{min} и максимальная L_{max} яркость изображения. В данной работе в качестве яркости пиксела берется Y -компонента цвета C_{HDR} , вычисленного в эталонном цветовом пространстве $CIE XYZ$, которая вычисляется как

$$L_{HDR} = 0.2126 \cdot r + 0.7152 \cdot g + 0.0722 \cdot b, \quad (1)$$

где r, g, b - компоненты цвета C_{HDR} . В данной работе L_{cp} вычисляется как среднее геометрическое взвешенное

$$L_{cp} = \exp\left(\frac{\sum w \cdot \ln(L_{HDR} + \delta)}{\sum w}\right), \quad (2)$$

где $w = 1 - \exp(-L_{HDR})$ - весовой коэффициент; а δ - малая константа для обработки случая $L_{HDR} = 0$. В отличие от существующих реализаций, где используется среднее геометрическое [1, 2], применение предложенных весовых коэффициентов позволяет уменьшить чувствительность средней яркости к областям изображения, яркость пикселей которых близка к нулю. Это позволяет добиться сбалансированного изменения яркости изображения от кадра к кадру и устранить пересвечивание объектов на кадрах с преобладающими темными областями (рис. 1). В разделе 3 описана предложенная реализация, позволяющая вычислить L_{cp} , L_{min} и L_{max} с помощью геометрического шейдера за один проход, учитывая яркость каждого пиксела кадра.

Моделирование эффекта «заплывания». Эффект «заплывания» моделируется путем коррекции цвета для каждого пиксела изображения сцены. Для этого синтезируется текстура эффекта, которая смешивается с исходным изображением сцены. В данной работе текстура эффекта синтезируется на основе изображения в расширенном диапазоне, а смешивание выполняется после его тонального преобразования (в стандартном диапазоне), что позволяет уменьшить вычислительную сложность эффекта, сохраняя при этом визуальную привлекательность результирующего изображения.

Моделирование текстуры эффекта выполняется с помощью двух этапов. На первом этапе выполняется выделение областей исходного кадра с избыточной яркостью, а на втором этапе - размытие изображения с помощью гауссовского фильтра. Поскольку эффект сам по себе является размытием, то его пространственное разрешение может быть уменьшено, что позволяет значительно сократить вычислительную сложность и расход памяти без большого ущерба качеству изображения. В данной работе размытие выполняется в половинном разрешении по горизонтали и вертикали.

Выделение ярких областей выполняется отсечением пикселей с яркостью L_{HDR} (см. (1)), меньшей порогового значения L_T , с одновременным линейным приведением цвета ярких областей в стандартный диапазон на основе максимальной яркости L_{max} (см. раздел 3):

$$C_{Bright} = \begin{cases} C_{HDR} \cdot \frac{L_{HDR}}{L_{max}}, & L_{HDR} > L_T \\ 0, & L_{HDR} \leq L_T \end{cases}$$

Пороговая яркость L_T является регулируемым параметром, значение которого можно выбирать, исходя из светочувствительности матрицы или других соображений.

Наиболее качественное и близкое к реальности размытие обеспечивает гауссовский фильтр. Известно, что этот фильтр обладает свойством сепарабельности, означаящим, что свертка с гауссианом в двумерном пространстве может быть заменена на две последовательных одномерных свертки по осям X и Y :

$$C_{Bloom} = C_{Bright} \otimes G_N(x, y) = (C_{Bright} \otimes G_N(x)) \otimes G_N(y).$$

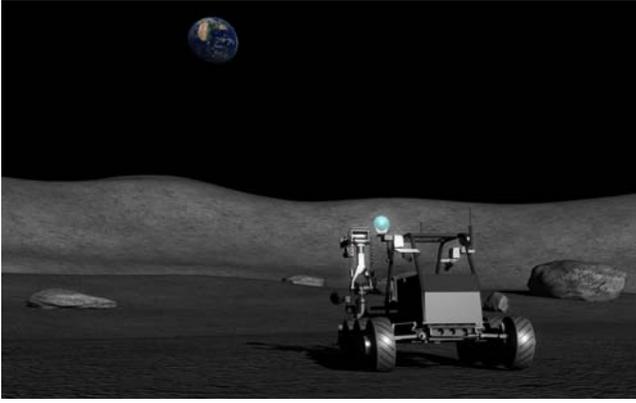


Рис. 1. Сравнение результатов работы модифицированной (слева) и исходной версии (справа) глобального оператора Рейнхарда на изображении со значительной площадью темных участков

Благодаря этому, вычислительная сложность преобразования существенно уменьшается (с $O(N^2)$ до $O(N)$, где N - размер фильтра). Размер фильтра задается пользователем и регулирует степень размытия.

Тональное отображение. На данном этапе высокодиапазонное изображение сцены приводится к стандартному диапазону визуализируемого кадра. С помощью фрагментного шейдера для каждого пиксела кадра вычисляется цвет C_{tone} . Для этого вначале шейдер вычисляет отмасштабированную яркость L_{scaled} вида

$$L_{scaled} = a \cdot \frac{L_{HDR}}{L_{cp}} \quad (3)$$

где L_{HDR} - яркость пиксела, вычисленная согласно (1); a - ключевой параметр (аналог экспозиции у фотокамер). Параметр a вычисляется для текущего кадра с помощью следующей формулы [1]:

$$a = 0.18 \cdot 2^{2(B-A)/(A+B)},$$

где $A = L_{max} - L_{cp}$; $B = L_{cp} - L_{min}$. Полученная в (3) яркость L_{scaled} затем отображается в стандартный диапазон с помощью глобального оператора Рейнхарда [1]:

$$L_{LDR} = \frac{L_{scaled} \cdot \left(1 + \frac{L_{scaled}}{L_{white}^2}\right)}{1 + L_{scaled}},$$

где L_{white} - яркость, которая соответствует белому цвету; в данной работе в качестве L_{white} используется максимальная яркость L_{max} , отмасштабированная аналогично (3). Результирующий цвет C_{tone} пиксела после тонального отображения вычисляется путем обработки цвета C_{HDR} вида

$$C_{tone} = \frac{L_{LDR}}{L_{HDR}} \cdot C_{HDR}.$$

Данное выражение получено на основе комбинации прямого и обратного преобразования из цветового пространства RGB в CIE xuY (производное от CIE XYZ).

Смешивание тонированного изображения и эффекта. На заключительном этапе выполняется смешивание изображения эффекта с изображением после тонального преобразования:

$$C_{LDR} = C_{tone} + k_{EFF} \cdot C_{Bloom},$$

где k_{EFF} - пользовательский параметр, регулирующий «мощность» эффекта.

3. РЕАЛИЗАЦИЯ ВЫЧИСЛЕНИЯ MAX, MIN И СРЕДНЕЙ ЯРКОСТИ НА GPU

В описанной технологии визуализации вычисление L_{cp} , L_{min} , L_{max} обладает высокой вычислительной сложностью, так как, по определению, вычисление каждой такой яркости включает в себя обработку всех пикселей кадра. Для эффективного вычисления таких глобальных яркостей в режиме реального времени предлагается следующая схема их распределенного вычисления на множестве ядер графического процессора. Текстура с изображением виртуальной сцены разбивается логически на блоки текселов одинакового размера. Для каждого такого блока ядро GPU вычисляет минимальную и максимальную яркость блока ($L_{min,block}$ и $L_{max,block}$), а также сумму S взвешенных логарифмов яркостей пикселей блока и сумму W весов пикселей блока; см. выражение (2). Минимальная яркость L_{min} находится как минимальная из всех вычисленных $L_{min,block}$, а L_{max} - как максимальная из всех $L_{max,block}$. Для получения средней яркости L_{cp} вычисляется сумма всех S и сумма всех W , которые подставляются, соответственно, в числитель и знаменатель дроби в выражении (2).

Предлагаемая реализация описанной схемы вычислений основана на возможности графического ускорителя записывать в один тексел результаты параллельной обработки дублирующихся вершин, сгенерированных геометрическим шейдером. Для выполнения такой обработки в видеопамати создается массив вершин, имеющий размер, равный количеству блоков. Для каждой вершины в массив записываются координаты x , y крайнего левого нижнего тексела соответствующего блока. Также в видеопамати создается текстура из двух текселов (такого же формата, что и текстура с изображением сцены), $rgba$ -каналы которой проинициализированы значениями 0 , $-L_s$, 0 , 0 , соответственно, где L_s - некоторая большая величина, заведомо большая максимально возможной яркости освещения в сцене. Предварительно установив в графическом конвейере режим смешивания, при котором в rg,b -каналы

записываются максимальные значения, а в альфа-канал – сумма значений, мы визуализируем массив вершин в текстуру с помощью разработанной шейдерной программы, выполняющей следующий параллельный

Алгоритм вычисления L_{\min} , L_{\max} , L_{cp}

1. Передаем из вершинного шейдера в геометрический шейдер вершину с координатами (x, y) .

2. В геометрическом шейдере выполняем

$$L_{\min, block} = L_S, L_{\max, block} = 0, S = 0, W = 0;$$

Цикл по всем текстелам блока, заданного (x, y) .

Вычисляем L_{HDR} , согласно (1);

$$w = 1 - \exp(-L_{HDR});$$

$$S = S + w \cdot \ln(L_{HDR} + \delta);$$

$$W = W + w;$$

$$L_{\min, block} = \min(L_{\min, block}, L_{HDR});$$

$$L_{\max, block} = \max(L_{\max, block}, L_{HDR});$$

Конец цикла.

Генерируем вершину V_0 с координатами $x = -0.5$ и $y = 0.0$ и цветом C_0 с компонентами $r = L_{\max, block}$, $g = -L_{\max, block}$, $a = S$.

Генерируем вершину V_1 с координатами $x = 0.5$ и $y = 0.0$ и цветом C_1 с компонентой $a = W$.

3. С помощью фрагментного шейдера добавляем каждый цвет C_0 и C_1 в 0-й и 1-й текстел текстуры соответственно.

В результате такой визуализации в r и g каналах 0-го текстела текстуры записываются величины L_{\max} и $-L_{\min}$, а альфа-канале 0-го и 1-го текстела формируется, соответственно, числитель и знаменатель дроби из выражения (2), по которым (предварительно выгрузив текстуру в оперативную память) легко вычисляется величина L_{cp} .

4. ЗАКЛЮЧЕНИЕ

В работе предложена технология реализации режима HDR визуализации виртуальных сцен в реальном времени, адаптированная для выполнения на графическом процессоре.

В технологии реализовано сглаживание синтезируемых изображений, моделирование эффекта «заплывания» пересвеченных областей и тональное отображение с помощью модифицированной версии глобального тонального оператора Рейнхарда. В разработанной модификации для вычисления средней яркости кадра используются взвешенные значения яркости пикселей, что позволяет корректно визуализировать кадры с существенным процентом темных областей (см. рис. 1) и обеспечивает сбалансированное изменение яркости синтезируемых изображений от кадра к кадру. Представлен эффективный алгоритм для быстрого вычисления глобальных яркостей, обеспечивающий обработку всех пикселей кадра с использованием параллельных вычислений на GPU. Использование данного алгоритма позволяет выполнять HDR визуализацию

виртуальных сцен в режиме реального времени на средствах отображения с высоким разрешением (HD, Full HD). Разработанные алгоритмы, методы и подходы были интегрированы в программный комплекс визуализации [4] и прошли успешную апробацию на ряде сцен для имитационно-тренажерных комплексов. На рис. 2 представлен пример кадра визуализации сцены со сложной моделью освещения, включающей в себя сумеречное освещение, атмосферу и освещение фарой, установленной на мобильном роботе.

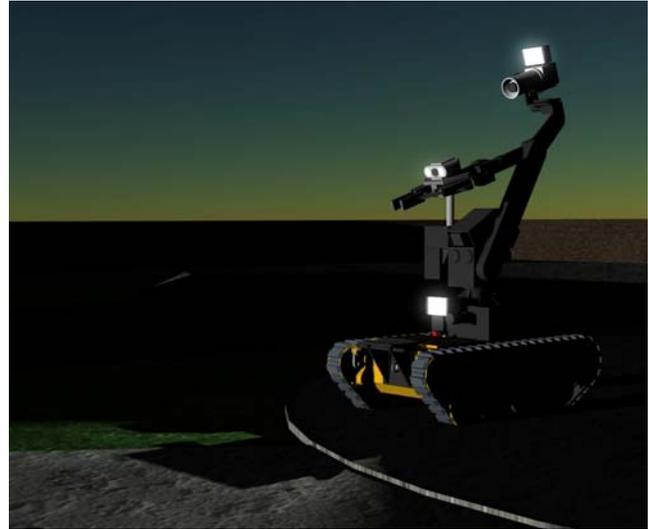


Рис. 2. Пример кадра визуализации сцены со сложным освещением

5. БЛАГОДАРНОСТИ

Работа была выполнена при финансовой поддержке РФФИ в рамках гранта №13-07-00674.

6. ССЫЛКИ

- [1] Kiser C., Reinhard E., Tocci M., Tocci N. Real time automated tone mapping system for HDR video. In IEEE International Conference on Image Processing, 2012. – P. 2749 - 2752.
- [2] Luksh, C.: Realtime HDR rendering. Tech. Rep., Institute of Computer Graphics and Algorithms, TU Vienna (2007).
- [3] Барладян Б.Х., Волобой А.Г., Галактионов В.А., Копылов Э.А.. Эффективный оператор сжатия динамического диапазона яркостей // Программирование, 2004. – № 5. – С. 35 - 42.
- [4] Михайлюк М.В., Торгашев М.А.. Система «GLVIEW» визуализации для моделирующих комплексов и систем виртуальной реальности // Вестник Российской академии естественных наук, 2011. – № 2. – С. 20 - 28.

Сведения об авторах

Тимохин Петр – научный сотрудник, Научно-исследовательский институт системных исследований РАН, webpismo@yahoo.de.

Торгашев Михаил – зав. сектором, Научно-исследовательский институт системных исследований РАН, mtorg@mail.ru.

Using dual quaternions for Control Cluster Method

A. Bukatov

SFU Computer Center
Bolshaya Sadovaya, 105, 344000, Rostov-on-Don, Russia

baa@sfedu.ru

E. Gridchina

Informatics and Computational Experiment Department, SFU, Bolshaya Sadovaya, 105, 344000, Rostov-on-Don, Russia

helen.gridchina@gmail.ru

D. Zastavnoy

Informatics and Computational Experiment Department, SFU,
Bolshaya Sadovaya, 105, 344000, Rostov-on-Don, Russia

dzast@sfedu.ru

Abstract

Realistic character mesh deformation has been one of the key issues in different applications of 3D computer graphics as video games or crowd simulation. Geometric techniques such as skinning are considered an intuitive way to achieve visually plausible animation of the human-like characters despite of having certain shortcomings. Control cluster method is an alternative to tackle those drawbacks of the traditional techniques. In this paper a modification of control cluster method using dual quaternions and Bezier patch meshes is presented. Control cluster method tends to split all the vertices of the polygonal model mesh into two groups: cluster and free vertices. Free vertices position is calculated based on cluster vertices position in a non-linear fashion. Dual quaternion skinning application to cluster vertices calculation is discussed.

Keywords: *Skinning, Control Cluster, Non-linear Deformation, Polygonal Mesh Deformation.*

1. INTRODUCTION

Realistic mesh deformation has been one of the key issues in different applications of 3D computer graphics as video games or crowd simulation [1 - 3].

This research focuses on non-linear geometric approach to deformation because most of the geometric algorithms proposed demonstrate shortcomings connected with their linear approach.

Previously an approach based on a control clusters technique for correcting 2D models skinning deformation was described by us in [4]. Then a generalized and more versatile approach to deforming 3D polygonal models – Control Cluster Method (CCM) was proposed in [5]. The generalized approach could be implemented in a wide variety of applications including physical processes simulation, such as ecological processes (e.g., computational domain flooding and dewatering as a result of wind upsurge-downsurge; fire spreading; prevalence of air and water pollution). It makes use of control clusters to facilitate the work of the animator in creating realistic animation of the character without involving much user-input data. The method proposed provides intuitive control and it is easy to use because it allows the user to influence only a small group of vertices leaving the non-linear deformation of the rest of the vertices to the algorithm.

In this paper a modification of control cluster method using dual quaternions and Bezier patch meshes is presented. Control cluster method tends to split all the vertices of the polygonal model mesh into two groups: cluster and free vertices. Free vertices position is calculated in a non-linear fashion based on cluster vertices position. Using dual quaternion skinning as a way to deform cluster vertices is discussed.

2. RELATED WORK

A lot of research on 3D object deformation from different perspectives has been done, but we focus on recent development applicable to skeletal animation.

Linear Blend Skinning (LBS) is considered the most widely used technique for real-time animation in spite of its well-known shortcomings [6] thanks to its versatility, efficiency and straightforward GPU implementation [7].

Recent skinning techniques can be divided into geometric and example-based techniques. Geometric techniques use the single input mesh, as for the example-based ones, they require using sample meshes to correct or train the weights of the model mesh.

Dual-quaternion skinning is a relatively recent geometric skeletal animation technique [8], describing both rotations and translations using quaternions. Having no shortcomings of LBS, dual-quaternion skinning demonstrates some new limitations connected with too much volume (see Fig. 1).

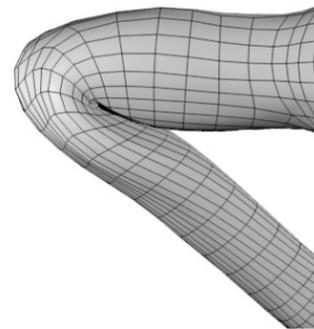


Fig. 1. Bending the arm model with Dual Quaternion Skinning (courtesy of [9])

To avoid notorious artifacts Pose Space Deformation proposed in [10] takes advantage of using sample shapes of the model, the technique requiring much input from the user [11].

It was also proposed in [12] to pre-compute optimized skinning weights for linear and dual-quaternion skinning at joints to approximate the skin transformations produced by nonlinear variational deformation methods.

Research [9] exploits the advanced compositions mechanisms of volumetric implicit representations for correcting the results of geometric skinning techniques.

Some other techniques proposed such as [13] modify the basic LBS formulation by generalizing the weight functions to matrix-valued or vector-valued forms.

3. CONTROL CLUSTER METHOD

3.1 Approach description

The key idea of Control Cluster Method consists in all the vertices of the model being split into two types: cluster vertices and so-called free vertices. The position of the cluster vertices is to be set using some input data. The key areas of the mesh defined by cluster vertices form the shape of the model, thus cluster vertices control the deformation of the model. The position of free vertices is calculated automatically by a non-linear algorithm.

3.2 Cluster vertices transformation

Cluster vertices can be transformed in a great variety of ways, either by direct mesh editing or by applying some deformers including lattices, cage or skeleton. Cluster vertices transformation could be also defined by data ranges from some type of measuring equipment, for example water level measuring device or motion capture data.

The key cluster vertices before were deformed with the help of Linear Blend Skinning, so vertex $\vec{v} \in V_p$ is associated with a set of weights $\{w_i\}$, $\sum_i w_i = 1$, where w_i denotes the weight of the bone \vec{b}_i . Weight w_i defines the extent to which the vertex position is influenced by the bone \vec{b}_i . Let $\{B_i\}$ be the skeleton configuration in the bind pose. The skeleton being in an arbitrary pose W_i , the transformed position of vertex v' is calculated according to the formula.

$$\vec{v}' = LBS(\vec{v}) = \sum_i w_i W_i B_i^{-1} \vec{v} \quad (1)$$

But as deformation is inherently spherical we seek non-linear blending of rigid transformations. As for LBS, the component-wise interpolation of matrices results in not preserving the orthogonality of the rotational part of the matrix. Even in the proximity of singular configuration, the matrix being regular involves a non-uniform scaling, which ends in the loss of volume of the deformed character.

3.3 Free vertices transformation

If a vertex does not belong to the cluster, it is considered a free one and its position is recalculated based on its position in the model topology and its nearest cluster vertices position. Free vertices position can be defined in several non-linear ways so that it corresponds to the position of the cluster vertices.

In previous works on Control Cluster Method [4, 5] Catmull-Rom splines [14] or cubic cardinal splines generalizing cubic Catmull-Rom splines were used as a non-linear item of the method. Using interpolation, C^1 -continuity and local control, cardinal splines are an acceptable way to solve the task. The spline remaining flexible, smooth natural look of the character is provided by the C^1 -continuity of the spline. Direct control over the points of the curve is achieved with the spline interpolating its control points. With local control the spline has every control vertex provide a slight impact on the overall look, so the details of the model remain preserved.

The problem is that using splines requires special topology of the model and is not always suitable. So a more general approach would be a welcome alternative.

3.4 Fields of application

Control cluster method could be used in different spheres of visualization ranging from animation to ecological processes simulation.

The cluster vertices deformation can be set using user-input data, including different deformers such as a skeleton to produce a character animation or some certain pose of the model.

Provided cluster vertices positions are defined by some scientific equipment, a wind upsurge-downsurge flooding and dewatering simulation could be achieved with the help of control cluster method. Among possible spheres of control cluster method application in ecological processes simulation fire spreading and prevalence of air and water pollution can be mentioned.

In this paper application of the method to animation of human-like character models is presented.

4. MODIFICATION OF THE METHOD AND ITS SKINNING APPLICATION

4.1 Limitations of Control Cluster Method skinning

Skinning application of CCM can be used for creating character animation. The existing modification of Control Cluster Method using Linear Blend Skinning and cardinal splines as cluster and free vertices deformers respectively is characterized by some limitations. On one hand, capable of preserving the level of detail, it gives satisfying results without demanding much user input (see Fig. 2); on the other hand, splines cannot meet every possible topology of human-like models. Therefore the new modification is to deal with an arbitrary topology models.

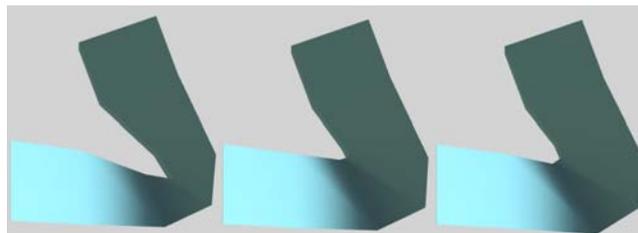


Fig. 2. Side view of the simplified two bone hand model, frames of the child bone rotation animation. From left to right: LBS, PSD, CCM

4.2 Cluster vertices transformation for skinning

Generally speaking, cluster vertices transformations can be defined in different ways that are not strictly set. To demonstrate it we consider three different ways for cluster vertices transformation in creating arm rotation animation: Dual Quaternion Skinning (DQS), Pose Space Deformation (PSD) and procedural dependence.

4.2.1 DQS deformed cluster vertices

The most efficient, versatile and wide-spread way of creating a character animation, Linear Blend Skinning, is challenged by Dual Quaternion Skinning [8] that is available in many of 3D editors of nowadays. The advantages of using DQS include allowing for spherical nature of skeletal deformations, efficiency and simple implementation. Moreover, the technique as well as LBS does not require the modification of existing models. Therefore, base cluster vertices can be deformed with DQS.

The first step is to convert skinning matrices W_1, \dots, W_p to unit dual quaternions $\hat{q}_1, \dots, \hat{q}_p$. This operations doesn't take much time because the number of the joints p is usually rather small and the conversion itself needs only one quaternion multiplication. Presented as 2×4 matrices, the dual quaternions $\hat{q}_1, \dots, \hat{q}_p$ are sent to GPU.

Then the linear combination is calculated for every vertex v according to the following formula:

$$\hat{\mathbf{b}} = \sum_i w_i \hat{\mathbf{q}}_{ji} \quad (2)$$

But to get a unit dual quaternion we need normalization

$$\hat{\mathbf{b}}' = \frac{\hat{\mathbf{b}}}{\|\hat{\mathbf{b}}\|}.$$

Then after the optimization where $\hat{\mathbf{b}} = \hat{\mathbf{b}}_0 + \varepsilon \hat{\mathbf{b}}_\varepsilon$ and

w_k, x_k, y_k, z_k denote the components of $\frac{\hat{\mathbf{b}}_k}{\|\hat{\mathbf{b}}_k\|}$,

the resulting unit dual quaternion is converted into matrix M as follows:

$$M = \begin{pmatrix} 1-2y_0^2-2z_0^2 & 2x_0y_0-2w_0z_0 & 2x_0z_0+2w_0y_0 & t_0 \\ 2x_0y_0+2w_0z_0 & 1-2x_0^2-2z_0^2 & 2y_0z_0-2w_0x_0 & t_1 \\ 2x_0z_0-2w_0y_0 & 2y_0z_0+2w_0x_0 & 1-2x_0^2-2y_0^2 & t_2 \end{pmatrix} \quad (3)$$

where $t_0 = 2(-w_\varepsilon x_0 + x_\varepsilon w_0 - y_\varepsilon z_0 + z_\varepsilon y_0)$;

$t_1 = 2(-w_\varepsilon y_0 + x_\varepsilon z_0 - y_\varepsilon w_0 + z_\varepsilon x_0)$;

$t_2 = 2(-w_\varepsilon z_0 + x_\varepsilon y_0 - y_\varepsilon x_0 + z_\varepsilon w_0)$;

So $v' = Mv$.

Due to coordinate-invariance the operation produces the shortest path screw motion. And though the motion speed is not constant it is close to constant [8]. Therefore no visible drawback in skinning is present.

4.2.2 PSD deformed cluster vertices

DQS exceeds LBS in many points but it is considered to have too much volume, producing a non-natural smooth outgrowth (see Fig. 1). To correct those undesired artifacts PSD can be used.

It requires using sample pairs $\langle X, S \rangle$, where X is a user-input sample shape of the model corresponding to skeleton configuration S . With the flexibility of control cluster method it is possible to avoid defining sample shapes of the whole model. Instead only the problem vertices of the elbow area of the model can be defined as cluster ones and only one sample pair is used for deforming those vertices likewise PSD thus achieving the necessary amount of volume.

So cluster vertices of the elbow problem area undergo additional transformations with the help of the displacements. Let us assume that v is a vertex position in the sample pose of cluster vertices in the pose X^i , v^0 is vertex position in the bind pose B . Then displacement d_j is calculated to compensate the deformation.

If the current pose is $X=X^i$, the current cluster vertex position $v' = DQS(B_j^{-1}v + d_j)$. For arbitrary pose calculation radial basis functions are used to interpolate between different samples X^i .

4.2.3 Procedurally deformed cluster vertices

Another welcome feature for realistic animation is muscle bulging. Both LBS and DQS lack such a feature. Yet it could be achieved using PSD but it would require for the user to create some sample pairs to imitate realistic muscle bulging. As an alternative way cluster vertices corresponding to biceps area of the upper arm can be deformed procedurally depending on the angle between upper arm and lower arm bones.

4.2.4 Free vertices transformation for skinning

For free vertices transformation Bezier patch meshes are used as they allow working with the models of arbitrary topology.

5. RESULTS

As an example arm model with rigging is used (see Fig. 3). All the vertices are split into 4 groups: 3 cluster groups and free vertices.

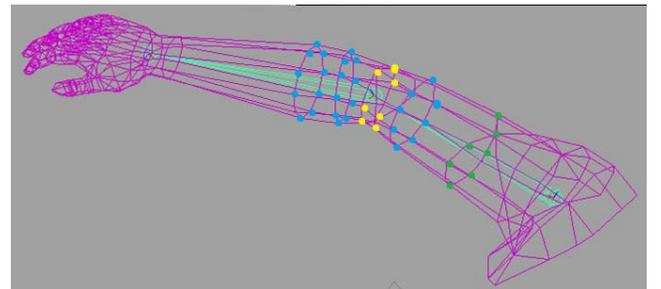


Fig. 3. The model of an arm with a two bone skeleton. Blue – free vertices; yellow – cluster likewise PSD deformed vertices; green – cluster procedurally deformed vertices; not marked – DQS deformed vertices.

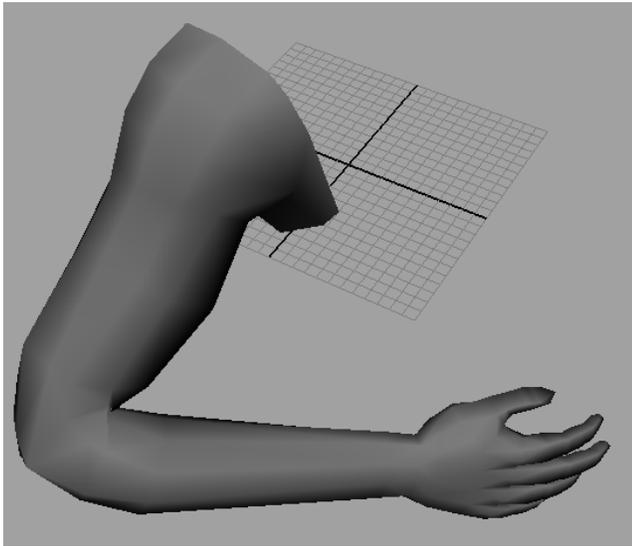


Fig. 4. CCM rotation animation of the model with a slight muscle bulging effect

The base cluster vertices of the model are deformed with help of DQS, they define the overall shape of the model. Cluster vertices of the elbow problem area of the model are transformed using one sample pair for the four cluster vertices, the sample pair being extracted from PSD sample shape of the whole model. To create biceps bulging effect corresponding cluster vertices position is calculated procedurally based on the angle between the two bones. Comparing the three methods (LBS, DQS, PSD and CCM) in a rotation animation it is possible to say that yielding close results neither PSD, nor DQS or CCM demonstrates the volume loss common for LBS. As far as user input data is concerned, PSD requires to create at least one sample pair of the whole model for the animation, while it is enough to store only several vertices sample for CCM. Both LBS and DQS do not require any samples but it is very difficult to adjust the weights properly and moreover one can't achieve all possible shapes adjusting weights. Therefore some sample data seems to be an inevitable part of the user guided animation.

6. CONCLUSION

Control Cluster Method is an approach for geometric non-linear deformation of the model in a labor-saving way. In this paper its modification in application to skinning is described.

The advantage of CCM is that it captures the designed shape, including effects like muscle bulging as well as exemplified based methods. The drawback is the necessity of acquiring the example vertices positions. CCM inherits some versatility from geometric methods as examples are needed only for several vertices of the model, the rest of the vertices calculated with the non-linear approach automatically. The proposed algorithm has a limitation. The CCM modification is limited to rigid transformation and therefore no models that have got parts that can scale or shear can be skinned in an adequate way.

Applications of the approach to other fields as physical processes simulation is in the scope of the future research.

7. REFERENCES

- [1] Hidalgo M.G., Torres A.M., Gómez J.V. Deformation Models: Tracking, Animation and Applications, Lecture Notes in Computational Vision and Biomechanics. Springer, 2012.
- [2] Botsch M., Kobbelt L., Pauly M., Alliez P., Lévy Bruno Polygon Mesh Processing, AK Peters / CRC Press, Sept, 2010.
- [3] Dunyach M., Vanderhaeghe D., Barthe L., Botsch M. Adaptive remeshing for real-time mesh deformation, 2013, Eurographics Short Papers – P. 29 – 32.
- [4] Bukatov A., Gridchina E., Zastavnoy D. A spline-based approach to control cluster deformation // World Applied Sciences Journal, vol. 26, no. 6, 2013. – P. 724 – 727,
- [5] Bukatov A., Gridchina E., Zastavnoy D. A control cluster approach to non-linear deformation, in Poster Paper Proceedings of WSCG, Vaclav Skala, Ed., Plzen, Czech Republic, 2014, to be published.
- [6] Weber J. Run-time skin deformation, in Intel Architecture Labs. Proceedings of Game Developers Conference, March 2000.
- [7] Lee M.. Seven ways to skin a mesh: Character skinning revisited for modern GPUs, in "Gamefest Unplugged (Europe)", 2007.
- [8] Kavan L., Collins S. Z'ara, Ji'ri, O'Sullivan Carol. Skinning with dual quaternions, in Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games, New York, NY, USA, 2007, I3D '07. – P. 39 – 46, ACM.
- [9] Rodolphe Vaillant, Loïc Barthe, Gaël Guennebaud, Marie-Paule Cani, Damien Rohmer, Brian Wyvill, Olivier Gourmel, and Mathias Paulin, "Implicit skinning: Real-time skin deformation with contact modeling," ACM Trans. Graph., vol. 32, no. 4 – P. 125:1–125:12, July 2013.
- [10] Lewis J.P., Cordner M., Fong N. Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation, in Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, New York, NY, USA, 2000, SIGGRAPH '00. – P. 165 – 172, ACM Press/Addison-Wesley Publishing Co.
- [11] Gene S. Lee, Hanner F. Practical experiences with pose space deformation, in ACM SIGGRAPH ASIA 2009 Sketches, New York, NY, USA, 2009, SIGGRAPH ASIA '09. – P. 43:1–43:1, ACM.
- [12] Kavan L., Sorkine O. Elasticity-inspired deformer for character articulation, ACM Trans. Graph., vol. 31, no. 6 – P. 196:1. – 196:8, Nov. 2012.
- [13] Jacobson A., Baran I., Popović J., Sorkine O. Bounded biharmonic weights for real-time deformation, ACM Trans. Graph., vol. 30, no. 4 – P. 78:1– 78:8, July 2011.
- [14] Catmull E., Rom R. A class of local interpolating splines, in Computer Aided Geometric Design, R. Barnhill and R. Riesenfeld, Eds. – P. 317 – 326. Academic Press, 1974.

ФОТОРЕАЛИСТИЧНАЯ МОДЕЛЬ ОБЪЕМНОГО РАССЕИВАНИЯ В ЗАДАЧЕ ДВУНАПРАВЛЕННОЙ СТОХАСТИЧЕСКОЙ ТРАССИРОВКИ ЛУЧЕЙ

Д. Жданов

Санкт-Петербургский национальный исследовательский университет ИТМО, Санкт-Петербург

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

Государственный оптический институт им. С.И. Вавилова, Санкт-Петербург

А. Гарбуль

Государственный оптический институт им. С.И. Вавилова, Санкт-Петербург

А. Волобой, В. Галактионов, С. Ершов, И. Потемин, В. Соколов

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

Аннотация

Данная статья посвящена разработке физически корректной модели рендеринга сцен, содержащих объекты с объемным рассеиванием и объекты, находящиеся в рассеивающей среде. В качестве базового подхода решения уравнения рендеринга использовался метод двунаправленной стохастической трассировки лучей. Было разработано два подхода, отличающихся эффективностью решения для различных параметров рассеивающей среды, как правило, концентрации рассеивающих частиц. Приведенные примеры иллюстрируют возможность применения разработанных моделей, как для фотореалистичной визуализации сцен с объектами и в объектах объемного рассеивания, так и для моделирования осветительных устройств, построенных на основе эффектов объемного рассеивания.

Ключевые слова: Сцена, объемное рассеивание, рендеринг, двунаправленная трассировка лучей, стохастическая трассировка лучей, фотореалистичное изображение.

1. ВВЕДЕНИЕ

На настоящий момент наиболее эффективным решением уравнения рендеринга для задачи построения фотореалистичного изображения является двунаправленная стохастическая трассировка лучей. Метод двунаправленной трассировки лучей имеет большое количество алгоритмических модификаций [1 – 3]. Основная вариация алгоритмов сводится к порядку расчета трасс лучей (сначала трассируются прямые лучи, а потом обратные или наоборот), типам фотонных карт, которые сохраняются в результате расчета (сохраняются карты только прямых лучей, только обратных лучей или карты всех лучей) и способу хранения и обработки фотонных карт. Алгоритмы также варьируются по глубине интегрирования первичной и каустической яркости вдоль трассы луча и учету физических явлений, возникающих в процессе взаимодействия луча с объектами сцены (учет спектральных и поляризационных характеристик излучения и оптических свойств объектов сцены). При этом первичная яркость вычисляется случайным «сканированием» источников света в точке наблюдения, а вторичная яркость – интегрированием световых фотонов, попавших в область видения лучей, испущенных из камеры, или интегрированием обратных лучей, попавших в область фотонов, испущенных источниками света. Приведенные модели хорошо адаптированы для вычисления яркости поверхностей объектов сцены. Однако в случае рассеивания света внутри

материалов сцены задача вычисления яркости сильно усложняется. Во-первых, физически аккуратное моделирование эффекта объемного рассеивания требует применения спектральных методов трассировки лучей и, соответственно, создания, хранения и использования спектральных фотонных карт. Хранение фотонных карт в RGB пространстве (для экономии оперативной памяти) приводит к потере точности спектрального моделирования и, следовательно, непригодно. Вторая серьезная проблема моделирования эффекта объемного рассеивания – это «разреженность» фотонных карт. Если в случае поверхностного рассеивания фотоны концентрируются на поверхностях сцены и их плотность высока, то в случае объемного рассеивания, например, тумана, фотоны рассеиваются в объеме и их плотность становится на порядок ниже. В результате, вероятность того, что события экстинкции прямого и обратного луча произойдут в пределах одной небольшой сферической области (источника формирования вторичной яркости рассеивающей среды) крайне мала, и большая часть лучей трассируется вхолостую. Все это приводит к появлению значительного случайного шума на изображении сцены. Поэтому для уменьшения шума изображения сцены, необходимо повысить количество независимых локальных «источников» формирования вторичной яркости, приходящихся на одну трассу луча.

В рамках данной работы рассматриваются общие принципы построения модели фотореалистичной визуализации объектов объемного рассеивания и решения, позволяющие значительно снизить шум изображения, возникающий на объектах объемного рассеивания. В рамках разработанной модели возможна, как визуализация собственно материалов с объемным рассеиванием, так и визуализация объектов, находящихся в среде объемного рассеивания, например, в тумане.

2. ПРИМЕНЕНИЕ ДВУНАПРАВЛЕННОЙ СТОХАСТИЧЕСКОЙ ТРАССИРОВКИ ЛУЧЕЙ ДЛЯ РЕШЕНИЯ УРАВНЕНИЯ РЕНДЕРИНГА

Двунаправленная стохастическая трассировка лучей является одним из способов решения уравнения рендеринга [4], которое определяет яркость цветовой компоненты объекта c в точке поверхности \vec{p} с локальной нормалью \vec{n} и в направлении \vec{v} . Для статических сцен уравнение рендеринга можно записать следующим образом:

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

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

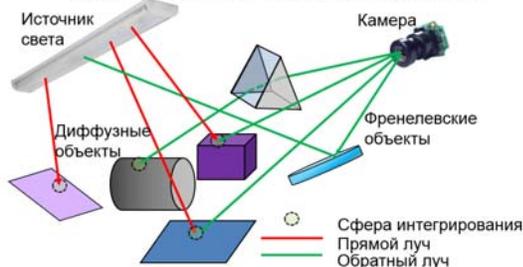
$\tau(\vec{p}, \vec{v}, c)$ – пропускание (прозрачность) среды между наблюдателем и точкой наблюдения;

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

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

Уравнение рендеринга (1) – это уравнение с бесконечной рекурсией, а двунаправленная стохастическая трассировка луча позволяет физически корректно выполнить бесконечное интегрирование наиболее эффективным образом. 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. При этом глубина диффузной трассировки лучей и, соответственно, глубина сохранения фотонных карт видимости сцены может быть переменной, что в предельном случае (бесконечная глубина трассировки обратного луча) может свести данный алгоритм к методу обратной стохастической трассировки лучей. На второй стадии трассируются прямые лучи, и в местах пересечения прямых лучей с фотонными картами (сферами, в пределах которых происходит интегрирование вторичной яркости) видимости сцены вычисляются вторичные и каустические яркости. Рис. 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

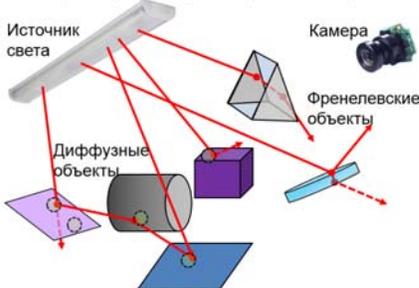


Рис. 1. Двунаправленная стохастическая трассировка лучей

Для расчета вторичной яркости с использованием фотонных карт уравнение рендеринга (1) более удобно записать в виде суммы яркостей, создаваемых всеми фотонами, попадающими в сферу интегрирования луча, выпущенного из камеры:

$$L(\vec{p}, \vec{v}, c) = \tau(\vec{p}, \vec{v}, c) \frac{1}{\pi} \sum_i^{all\ photons} BPDF(\vec{p}, \vec{v}, \vec{v}', c) E_i(\vec{p}, \vec{v}', c), \quad (2)$$

$$E_i(\vec{p}, \vec{v}', c) = \frac{F_i(\vec{p}, c)}{\pi R^2},$$

где $E_i(\vec{p}, \vec{v}, c)$ – освещенность сферы интегрирования радиуса R , создаваемая потоком i -го прямого луча $F_i(\vec{p}, c)$ в точке \vec{p} .

Очевидно, что в данном виде решение применимо только к объектам поверхностного рассеяния. Случай объемного рассеяния отличается тем, что как таковая поверхность объемного рассеяния отсутствует и рассеивание света (и, следовательно, источник вторичной яркости) формируется на виртуальной частице. При этом размеры, форма, относительный показатель преломления частицы и концентрация частиц в объеме сводятся к фазовой функции (аналог ДФР поверхности) и сечениям рассеяния и поглощения частиц [5]. Попадание луча (прямого и обратного) на частицу и последующее рассеивание света частицей моделируется случайным образом. Во-первых, формируется длина свободного прямолинейного распространения света в среде объемного рассеяния. Данная длина определяется случайным образом в соответствии с сечением экстинкции (сумма сечений поглощения и рассеяния). На конце пути свободного пробега происходит случайный выбор между событиями рассеяния и поглощения (в соответствии с сечениями рассеяния и поглощения). В случае рассеяния изменяется цвет луча (в соответствии со спектральными характеристиками объемлющей среды) и собственно случайное рассеивание луча (в соответствии с фазовой функцией рассеяния частицы).

Алгоритмы стохастической трассировки лучей между поверхностями сцены и между виртуальными рассеивающими частицами имеют много общего, поэтому адаптация уравнения рендеринга (1), приведенная в следующей главе, может позволить физически корректно вычислять яркость рассеивающей среды методами двунаправленной стохастической трассировки лучей.

3. ПРИМЕНЕНИЕ ДВУНАПРАВЛЕННОЙ СТОХАСТИЧЕСКОЙ ТРАССИРОВКИ ЛУЧЕЙ ДЛЯ РЕШЕНИЯ УРАВНЕНИЯ РЕНДЕРИНГА В СРЕДЕ ОБЪЕМНОГО РАССЕИВАНИЯ

Очевидно, что точки излома (события экстинкции) направления луча в среде объемного рассеяния являются аналогами рассеивающих поверхностей сцены и формируют яркость внутри данной среды. По аналогии с уравнением (2) можно написать уравнение вычисления вторичной яркости внутри рассеивающей среды [5]. Основные отличительные особенности данного уравнения – это то, что двунаправленная функция рассеяния поверхности заменяется фазовой функцией рассеивающей среды, а освещенность поверхности – объемной плотностью излучения.

$$L(\vec{p}, \vec{v}, c) = \frac{3\tau(\vec{p}, \vec{v}, c)}{4\pi^2 R^3} \sum_i^{all\ photons} P(\vec{p}, \vec{v}, \vec{v}', c) F_i(c) S_{C_i}(\vec{p}, \vec{v}) S_{L_i}(\vec{p}, \vec{v}),$$

$$S_{C_i}(\vec{p}, \vec{v}) = \frac{\sum_c \tau(\vec{p}, \vec{v}, c)}{\sum_c \tau(\vec{p}, \vec{v}, c) \epsilon(\vec{p}, \vec{v}, c)},$$

$$S_{L_i}(\vec{p}, \vec{v}) = \frac{\sum_c F_i(\vec{p}, c)}{\sum_c F_i(\vec{p}, c) \epsilon(\vec{p}, \vec{v}, c)},$$

(3)

где $P(\vec{p}, \vec{v}, \vec{v}', c)$ – спектральная фазовая функция рассеивающей среды для цветовой компоненты c в точке \vec{p} и направлениях наблюдения и освещения \vec{v} и \vec{v}' , соответственно;

$\epsilon(\vec{p}, \vec{v}, c)$ – спектральное сечение экстинкции рассеивающей среды.

Рис. 2 иллюстрирует принцип работы алгоритма стохастической двунаправленной трассировки лучей в рассеивающей среде, когда события экстинкции прямого и обратного луча совмещены внутри одной сферы интегрирования.

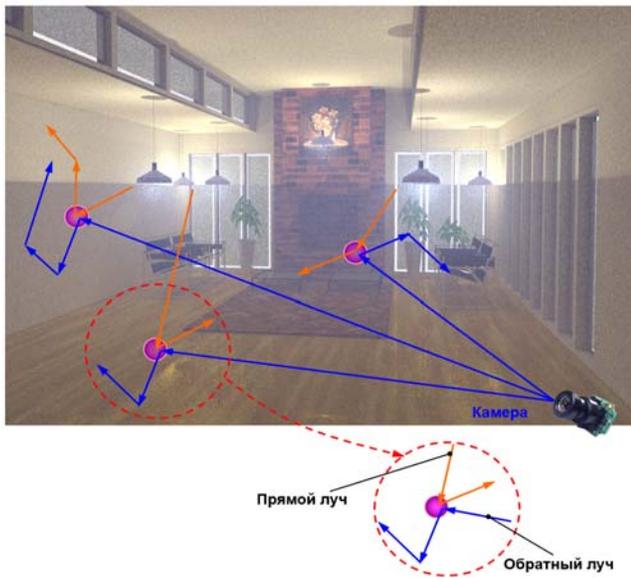


Рис. 2: Двунаправленная стохастическая трассировка лучей в среде объемного рассеивания (события экстинкции прямого и обратного лучей происходит в одной сфере интегрирования)

Для вычисления яркости среды объемного рассеивания достаточно ограничиться моделью вторичного освещения с нулевой глубиной диффузной трассировки лучей, поскольку первичная яркость, как правило, отсутствует, а резкие тени внутри среды не возникают.

Рис. 3 демонстрирует изображение комнаты, заполненной туманом с концентрацией капель водяного пара, равной 0.01%. Время расчета составило 8 часов.



Рис. 3. Модель изображения комнаты в тумане (концентрация капель водяного пара равна 0.01%)

Очевидно, что основным недостатком полученного изображения – это высокий уровень шума, который обусловлен низкой вероятностью совпадения позиций экстинкций прямых и обратных лучей. Такая ситуация возникает при малом значении сечений экстинкции рассеивающей среды (например, низкой концентрации рассеивающих частиц).

Для уменьшения уровня шума изображения в средах с малыми значениями сечений экстинкции был предложен альтернативный метод вычисления вторичной яркости рассеивающей среды. Его суть сводится к накоплению яркости от лучей, которые пересекают сферы интегрирования обратных лучей. Естественно, что лучей, которые пересекают сферу интегрирования, значительно больше, чем лучей, чьи события экстинкции происходят внутри данной сферы. При этом уравнение вторичной яркости приобретает следующий вид:

$$L(\vec{p}, \vec{v}, c) = \frac{\tau(\vec{p}, \vec{v}, c)}{\pi^2 R^2} \sum_i^{all\ photons} P(\vec{p}, \vec{v}, \vec{v}', c) F_i(c) S_{C_i}(\vec{p}, \vec{v}).$$

(4)

Рис. 4 иллюстрирует принцип работы алгоритма стохастической двунаправленной трассировки лучей в рассеивающей среде, когда сфера интегрирования собирает световую энергию всех лучей, пересекающих ее.

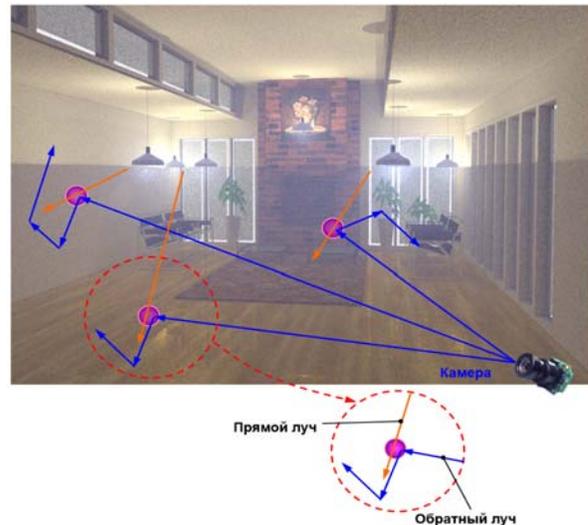


Рис. 4. Двунаправленная стохастическая трассировка лучей в среде объемного рассеивания (прямой луч пересекает сферу интегрирования обратного луча)

Рис. 5 демонстрирует изображение комнаты, полученное альтернативным методом, за время расчета, равное 8 часам.



Рис. 5. Альтернативная модель изображения комнаты в тумане (концентрация капель водяного пара равна 0.01%)

4. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ФОТОРЕАЛИСТИЧНОЙ МОДЕЛИ ОБЪЕМНОГО РАССЕИВАНИЯ

Два подхода, основанные на решениях уравнения рендеринга (3) и (4), были реализованы в программной модели стохастической двунаправленной трассировки лучей программного комплекса Lumiscept [6]. Данные решения могут быть использованы для построения фотореалистичных изображений объектов объемного рассеивания (а также объектов, находящихся в рассеивающей среде). Практическое применение разработанных решений показало, что альтернативный подход, как правило, более эффективен для сред с низкой концентрацией рассеивающих частиц.

Рис. 6 демонстрирует изображение комнаты, заполненной туманом с концентрацией капель водяного пара (PVC), варьирующейся от 0.01 до 0.1%.

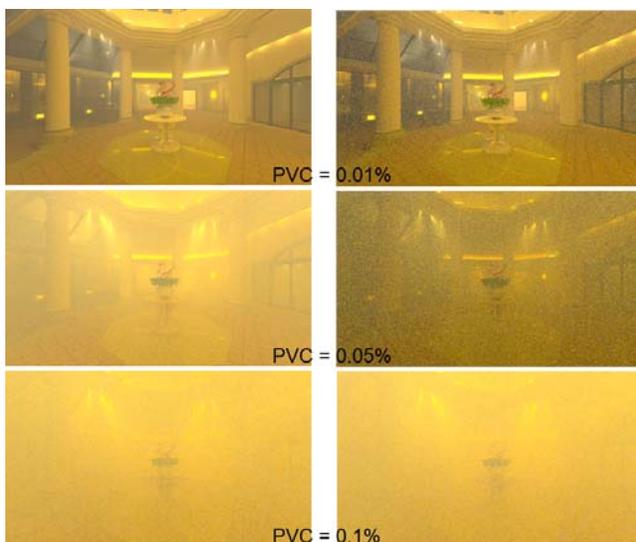


Рис. 6. Модель изображения комнаты в тумане (справа базовая модель, слева альтернативная)

Видно, что в случае высокой концентрации рассеивающих частиц эффективность обоих решений практически одинакова, однако при низкой концентрации частиц альтернативный метод выигрывает. Кроме того, при сильном уровне шума изображение кажется более темным, что объясняется наличием небольшого количества сверхярких точек.

Разработанные программные решения нашли применение для моделирования оптических устройств, например, диффузных световодов с диодной подсветкой, показанных на рис.7 и 8. В данном случае концентрация частиц в среде световода достаточно высока и базовый метод расчета обеспечивает более высокое качество изображения сцены.

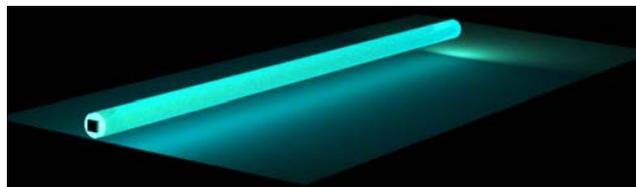


Рис. 7. Модель диффузного световода с диодной подсветкой (базовая модель)

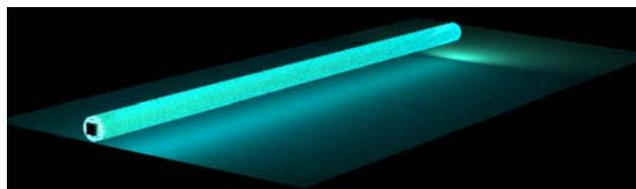


Рис. 8. Модель диффузного световода с диодной подсветкой (альтернативная модель)

5. ЗАКЛЮЧЕНИЕ

В рамках программного комплекса Lumiscept [6] были реализованы два физически корректных программных подхода к решению уравнения рендеринга для вычисления значения яркости рассеивающей среды. Разработанные программные модели позволяют осуществлять фотореалистичную визуализацию, как собственно материалов с объемным рассеиванием, так и объектов, находящихся в среде объемного рассеивания. Кроме того, данные программные модели позволяют осуществлять анализ и проектирование осветительных устройств, основанных на использовании объектов с объемным рассеиванием.

Модель рассеивающей среды была реализована в спектральном пространстве, а поляризационные эффекты были поддержаны для прямой трассировки лучей, что обеспечивает физически корректный результат наблюдения в неполяризованном пространстве.

6. БЛАГОДАРНОСТИ

Работа выполнена при государственной финансовой поддержке ведущих университетов Российской Федерации (субсидия 074-U01), грантов РФФИ № 12-01-00560 и 13-01-00454, а также компании Integra Inc.

7. ССЫЛКИ

- [1] Henrik Wann Jensen, Per Christensen. High quality rendering using ray tracing and photon mapping /SIGGRAPH '07.
- [2] Matt Pharr, Greg Humphreys. Physically Based Rendering - From Theory to Implementation // Morgan Kaufmann, 2004.
- [3] Toshiya Nishisaka and Henrik Wann Jensen. Stochastic progressive photon mapping. ACM Trans. Graph., 28(5): 1 – 8, 2009.
- [4] Kajiya, J.T. The rendering equation / Siggraph 1986: 143.
- [5] Борен К., Хафман Д.. Поглощение и рассеяние света малыми частицами. – М.: Мир, 1986.
- [6] <http://www.integra.jp/en>

Abstract

The article is devoted to design of physically correct rendering model of scenes with and inside of volume scattering objects. Base solution of the rendering equation was Monte-Carlo bidirectional ray tracing. Two effective solutions of the rendering equation for different parameter of the volume scattering was developed. As an example, the article is illustrated with rendering results of the scene with different fog parameters and LCD based light guide with volume scattering.

Keywords: *Scene, volume scattering, rendering, bidirectional ray tracing, stochastic ray tracing, photorealistic visualization.*

Сравнительный анализ современных рендер-систем для 3ds Max

В. Фролов, В. Галактионов, М. Трофимов

Институт прикладной математики имени М.В. Келдыша, Москва, Россия
vova@frolov.pp.ru, vlgal@gin.keldysh.ru, trofimovmax@mail.ru

Аннотация

В данной статье представлен сравнительный анализ производительности современных рендер-систем. Нами было проведено сравнение среди 7 наиболее популярными продуктов для 3D Studio Max (Mental Ray, IRay, VRay3, VRayRT3, Corona, Octane, Hydra) на скорость интегрирования освещения на 7 различных сценариях. В нашей работе нас, прежде всего, интересовало сравнение законченных реализаций в целом в промышленном решении в противовес сравнению отдельных алгоритмов в одной и той же системе. Мы постарались ответить на некоторые важные вопросы: что на практике дают последние достижения в области алгоритмов, такие, как SPPM [1] и VCM [2]? Есть ли выигрыш от GPU? Смещенные и несмещенные рендер-системы – что и когда лучше? Какие алгоритмы несут наибольшую практическую пользу?

На основе проведенного исследования нами были сделаны заключения о современном положении дел в области рендер-систем, преимуществах и недостатках CPU и GPU реализаций, а также о достоинствах и недостатках различных алгоритмов, применяемых в исследуемых системах и возможных векторах развития. Рендер-системы были ранжированы по очкам относительной производительности для каждого из сценариев и в сумме. Наши тестовые сцены и данные находятся в открытом доступе.

Ключевые слова: сравнение, рендер-системы, CPU, GPU.

1. ВВЕДЕНИЕ

Сравнение современных рендер-систем – нетривиальная задача, встречающая определенный ряд сложностей.

Во-первых, недостаточно сравнить системы на одной или двух сценах, поскольку существует достаточно большое число причин, по которым расчет освещения может стать медленным. Необходимо покрытие достаточно широкого спектра сценариев. Во-вторых, во время проведения сравнения необходимо добиваться максимально-возможного совпадения условий освещения, поскольку даже незначительное различие в условиях освещения может серьезно влиять на производительность. Например, еле заметное для глаза (из-за гамма-коррекции) увеличение яркости источника освещения в 1.5 раза может вызывать увеличение времени синтеза изображения до 2 раз, поскольку скорость сходимости метода Монте-Карло пропорциональна квадрату времени расчета, а яркость источника напрямую влияет на абсолютное значение ошибки метода Монте-Карло. В-третьих, системы, использующие смещенное решение (такие, как Mental Ray, VRay и Hydra), не всегда позволяют напрямую регулировать время синтеза и точность получаемого изображения, что чрезвычайно усложняет процесс получения изображения с заданным качеством или временем. Огромное число настроек различных алгоритмов в этих системах создает простор для экспериментов, и опытный пользователь может добиться более быстрого и точного синтеза, чем новичок. Наконец, рендер-системы не

идентичны по набору функциональности, моделям материалов и источников света. Например, источник типа 'Небесный Портал' (Sky Portal), позволяющий значительно ускорить расчет дневного освещения в помещениях корректно, реализован не во всех системах.

2. РЕНДЕР-СИСТЕМЫ

2.1. VRay и Mental Ray

Mental Ray – один из пионеров среди систем рендеринга со смещенным решением [3]. Со временем система не потеряла популярность и продолжает активно развиваться. Используется преимущественно в мультипликации. Ключевой особенностью системы является высокая степень гибкости, позволяющая дизайнеру настроить расчет в точности, так как нужно. Ключевыми алгоритмами расчета являются кэш освещенности в сочетании с финальным сбором, использующим фотонные карты [3]. Алгоритм интерполяции освещенности интересен тем, что использует не строго ближайшие точки, как в классической реализации кэша освещенности, а фильтрует освещение, учитывая многие точки. Mental также может использовать фотонные карты для расчета каустиков.

VRay [4] – одна из наиболее популярных рендер-систем в архитектуре. VRay – CPU рендер со смещенным решением. Ключевыми алгоритмами являются кэш освещенности (называемый в VRay термином 'Irradiance Maps') и карты светимости (называемые в VRay термином 'Light Cache'). Каустики в VRay могут рассчитываться при помощи фотонных карт. Для сравнения была использована новейшая версия VRay 3.0.

2.2. IRay, VRayRT, Octane

IRay является гибридной CPU/GPU рендер-системой с несмещенным решением, частично совместимой с Mental Ray. IRay использует GPU для ускорения трассировки лучей [5]. На основе документации в IRay реализован один из вариантов алгоритма переноса света Метрополиса. VRayRT – частично совместимая с VRay GPU рендер-система с несмещенным решением. В системе реализована только простая Монте-Карло трассировка путей. Эффективный расчет каустик невозможен. Система Octane [6] является одной из первых рендер-систем с несмещенным решением, реализованная полностью на GPU. В Octane реализован метод PMC [7].

2.3. Corona и Hydra

Corona [8] – дебютированная в 2012 г. CPU рендер-система со смещенным и несмещенным решениями. В отличие от VRay и Mental Ray, Corona использует кэш освещенности только для третьего переотражения, выполняя, таким образом, попиксельно финальный сбор. Отличительной чертой системы Corona является алгоритм Vertex Connection Merging (VCM) [2], значительно ускоряющий расчет каустик. Кроме этого, в системе Corona реализована двунаправленная трассировка лучей (BDPT) и двунаправленные прогрессивные фотонные карты (BDPM). Hydra [9] – дебютированная также в 2012 г. GPU рендер-система Hydra является системой со смещенным и

несмещенным решениями. Ключевыми алгоритмами являются кэш освещенности и стохастические прогрессивные фотонные карты (SPPM) [1] для расчета каустик.

3. МЕТОДИКА СРАВНЕНИЯ

Для того чтобы решить указанные во введении проблемы, применялась следующая методика:

1. Для сравнения была выбрана одна среда (3D Studio Max), что позволило добиться геометрического совпадения элементов сцены, включая положения источников света и камеры.
2. Для каждой системы и каждого сценария рассчитывался свой эталон, с которым впоследствии производилось сравнение. Это позволяет исключить различия реализации отдельных незначительных моментов и сосредоточиться на сравнении скорости интегрирования.
3. Для всех систем сравнивались два типа. Сначала фиксировалась небольшая ошибка (насколько это возможно) и измерялось время синтеза изображения. Так, сравнивалось время синтеза качественного изображения. Затем фиксировалось время синтеза изображения (обычно небольшое, в пределах 1 минуты) и измерялась ошибка. Сравнивалось качество полученных изображений в условиях фиксированного времени. Из двух сравнений для систем со смещенным решением в результирующий график входил средний индекс производительности. Такой способ позволяет частично решить проблему оптимизации настроек для систем со смещенным решением, поскольку на итоговый результат влияют оба типа оптимизаций (ориентированные на качество и скорость). Для систем с несмещенным решением соотношение скорость/качество для обоих типов сравнений совпадает.

3.1. Индекс производительности

Для того чтобы оценить производительность рендер-систем на различных сценах и сопоставить их друг с другом, введем абсолютный индекс производительности (1).

$$P_t = \frac{1}{MSE * MSE * t} \quad (1)$$

Здесь MSE - квадратичная ошибка (Mean Square Error), вычисляемая при помощи программы 'The Compressorator', а t - время в секундах. Если сцена, условия освещения и оборудование фиксированы, отношение индексов производительности для двух систем будет адекватно отражать отношение производительности этих систем. Однако зависимость сравнения от оборудования и абсолютные значения индекса снижают наглядность сравнения. По этой причине введем относительный индекс производительности (2).

$$P_{I(Rel)} = 100 * \frac{P_t}{MAX(P_{I(VRay)}, P_{I(MentalRay)}, P_{I(IRay)}, \dots)} \quad (2)$$

Относительный индекс производительности будет равен 100 баллам для системы, которая является на данной сцене самой быстрой. Остальные системы получают баллы пропорционально тому, насколько они медленнее. Таким образом, относительный индекс производительности не зависит от сложности сцены, и его можно усреднить по всем тестовым сценам, оценив, насколько система проявила себя в комплексе.

4. ЭКСПЕРИМЕНТЫ

Все эксперименты производились на машине с CPU Core i7 3770 3.4 Ghz, и GPU Nvidia GTX680. Память типа DDR3, 16 Gb. В силу различия архитектуры CPU и GPU мы опирались на компромиссное решение, основанное на выборе оборудования одного ценового диапазона. Все ошибки были посчитаны при помощи программы 'The Compressorator' [10].

4.1. Тестовые сценарии (рис. 1)

Сценарий номер 1 – Cornell Box с зеркальным чайником. Несмотря на свою простоту в данном сценарии присутствуют практически все основные эффекты трехмерной компьютерной графики: шумное первичное освещение и мягкие тени, зеркальные блики от источника освещения, отраженные каустики. Являясь геометрически простой, сцена в некоторой степени амортизирует стандартные потери производительности GPU трассировщиков лучей на ветвлениях, а CPU трассировщиков лучей – на кэш промахах.

Сценарий номер 2 – уличная (outdoor) сцена. Данная сцена, являясь геометрически сложной (из-за травы), тем не менее, с точки зрения освещения достаточно проста – однородная панорама и один относительно неяркий источник освещения. Время синтеза изображения на такой сцене должно в основном быть обусловлено скоростью трассировки лучей.

Сценарий номер 3 – верхний коридор Crytek Sponza. В данном сценарии практически все видимое освещение вторичное. Силен вклад от второго и третьего диффузных переотражений, поэтому финальный сбор должен давать значительное ускорение. Указанный сценарий наиболее точно отражает скорость расчета вторичного освещения. Сценарий нельзя назвать 'сложным для расчета' (hard sampling), поскольку яркость солнца относительно невелика и солнцем освещена значительная часть поверхности сцены.

Сценарий номер 4 – зал конференций. Сложность расчета освещения в данной сцене заключается в большом числе источников освещения. Под потолком находится 20 источников прожекторного типа. Тем не менее, каждый из источников светит на относительно небольшой участок сцены, поэтому, если в рендер-системе реализована эффективная схема сэмплирования источников, в каждой конкретной точке сцены большинство источников должны быть эффективно отброшены или учитываться относительно нечасто.

Сценарий номер 5 – освещение Небесными Порталами. Данная сцена демонстрирует достаточно типичный сценарий расчета дневного освещения в помещении. При таком сценарии напротив окон ставят источники света, имитирующие свет от внешнего окружения, проникающий через окно. Такой источник называется Небесным Порталом (Sky Portal). При правильной реализации он является полностью корректным решением. Такой источник света служит средством расчета освещения от окружения при помощи явной стратегии (стратегии сэмплирования источников света). Иными словами, Небесный Портал является не самостоятельным источником света, а всего лишь подсказкой для Монте-Карло трассировки лучей, позволяющей вычислять освещение от окружения более эффективно в тех случаях, когда изнутри помещения видима относительно небольшая часть окружения. Тем не менее, при использовании небесных порталов расчет первичного освещения в

определенной степени осложняется по двум причинам. Во-первых, небесные порталы могут иметь значительные размеры, в результате чего замедляется расчет мягких теней. Во-вторых, число источников этого типа может быть достаточно большим (5 – 10), что еще более осложняет вычисление первичного освещения. На данной сцене были использованы различные комбинации методов для различных систем, наилучшим образом показавшие себя.

Сценарий номер 6 – тест на MLT (Metropolis Light Transport). В данном сценарии небольшой участок сцены освещается исключительно ярким направленным источником света, имитирующим солнце. Вторичное освещение, вызванное таким освещением, является трудным для расчета (hard sampling). Фотонные карты в сочетании с финальным сбором (как и карты светимости) амортизируют увеличение времени расчета только за счет ускорения вычисления компоненты от третьего и более переотражений. Однако вычисление компоненты от второго переотражения света данным методом не ускоряется. С другой стороны, алгоритм переноса света Метрополиса (MLT) и аналогичные алгоритмы при условии корректной реализации их в той или иной системе должны давать на данной сцене значительное ускорение по сравнению с традиционным методом Монте-Карло и финальным сбором.

Сценарий номер 7 – водные каустики. В данной сцене присутствуют отраженные каустики и подводные каустики SDS типа (Specular Diffuse Specular), являющиеся сложными для расчета. IRay и VRayRT не были способны корректно рассчитать данный тип каустиков за приемлемое время, хотя система IRay способна эффективно считать каустики, видимые напрямую (ESDL).

5. ВЫВОДЫ

Анализируя графики на рис. 1 и 2, можно заключить, что несмещенные рендер-системы на GPU (IRay, VRayRT, Octane) обеспечивают в лучшем случае паритет с Mental Ray и VRay, а в худшем – проигрывают последним в 1.5 – 2 раза. Выигрыш этих систем (IRay, VRayRT, Octane) на открытой сцене (сценарий 2) можно объяснить простой формой освещения, при которой время рендера зависит только от скорости трассировки лучей. С другой стороны, GPU и рендер-система на основе алгоритмов со смещенным решением Hydra показала значительный отрыв по скорости (в 4 – 5 раз) при вычислении вторичного освещения (сценарий 3) и каустиков (сценарий 7). На каустиках второе место занимает Corona с алгоритмом VCM. Несмотря на такое же соотношение скорости у VRay, в области вычисления каустиков мы отдадим ей третье место, поскольку в VRay сложнее добиться каустиков высокого качества, так как в этой системе фотонные карты не прогрессивные. Однако на вторичном освещении VRay прочно обосновалась на втором месте. Рендер-система Corona лучше всего себя показала при расчете освещения от небесных порталов (сценарий 6) и в условиях сложного диффузного освещения (сценарий 7). Неплохие результаты в среднем могут говорить о том, что классический подход вычисления освещения при помощи финального сбора и карт светимости и сейчас является весьма практичным решением.

Сравнивая скорость интегрирования каустиков в CPU рендерах VRay и Corona, мы приходим к выводу о том, что внимание, прикованное к алгоритму VCM в последние годы, возможно, завышено. Реализация этого алгоритма на практике в системе Corona дала выигрыш только на два

сценариях (1 и 7) и только за счет каустик. При этом каустики могут быть столь же эффективно вычислены при помощи более простого в реализации алгоритма SPPM, а во всех остальных случаях VCM был значительно медленнее (до 10 раз), чем стандартный вариант Corona с попиксельным финальным сбором. Что касается MLT, ни Octane, ни IRay, предполагающие реализацию идей MLT, не показали ожидаемого для этого алгоритма увеличения производительности по сравнению со стандартной Монте-Карло трассировкой. Метод PMC в системе Octane позволил за приемлемое время рассчитать каустики в бассейне, однако он приблизительно на два порядка проигрывает алгоритму SPPM, реализованному в системе Hydra.

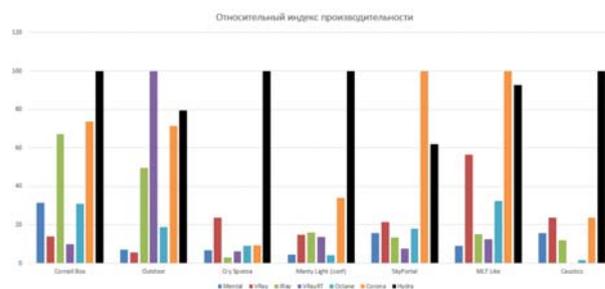


Рис. 1. Сравнение рендер-систем по относительному индексу производительности (на различных сценах)
http://ray-tracing.ru/upload/free/hydra/comparison_final.pdf

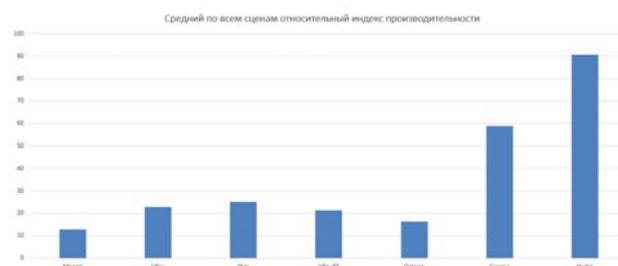


Рис. 2. Сравнение рендер-систем по относительному индексу производительности (среднее по всем сценам)
http://ray-tracing.ru/upload/free/hydra/comparison_final.pdf

6. ССЫЛКИ

- [1] Hachisuka T., Jensen H.W. Stochastic progressive photon mapping, ACM Trans. Graph., vol. 28, no. 5, dec. 2009. – P. 141:1 – 141:8
- [2] Georgiev I, Krivanek J, Slusallek Ph. Bidirectional light transport with vertex merging in SIGGRAPH Asia, 2011 Sketches, New York, NY, USA, 2011, SA '11. – P. 27:1 – 27:2, ACM.
- [3] NVIDIA, “Mental ray”, 2014.
- [4] Chaos Group, “Vray,” 2014.
- [5] Nvidia, “Nvidia iray faq,” 2014.
- [6] OTOY, “Octane renderer,” 2014.
- [7] Yu-Chi Lai, Shaohua Fan, Stephen Chenney, and Charcle Dyer, “Photorealistic image rendering with population monte carlo energy redistribution,” Eurographics Symposium on Rendering, 2007.
- [8] Ondra Karlik, Adam Hotovy, Jaroslav Krivanek, Jakub Konecny, and Ludvik Koutny, “Corona render,” 2014.
- [9] Ray Tracing Systems, “Hydra renderer,” 2014.
- [10] AMD, “The compressor,” 2014.

Благодарности

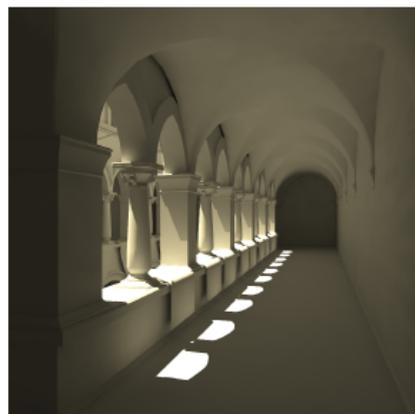
Работа поддержана грантами РФФИ № 12-01-00560, № 13-01-00454 и стипендией президента РФ СП-4053.2013.5.



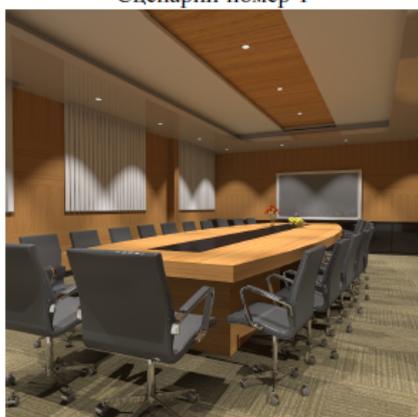
Сценарий номер 1



Сценарий номер 2



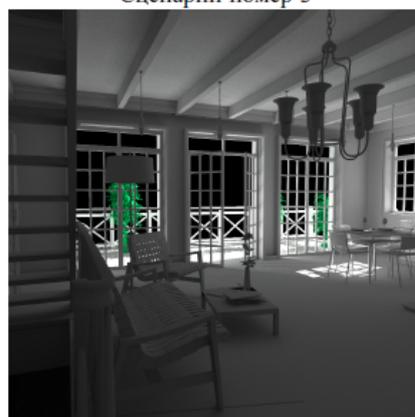
Сценарий номер 3



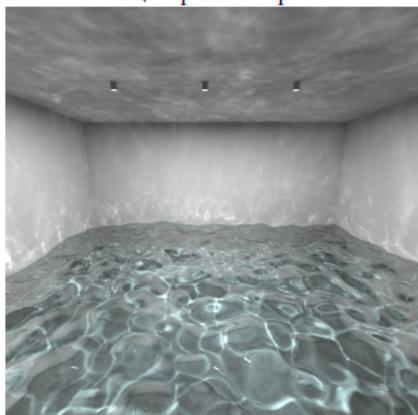
Сценарий номер 4



Сценарий номер 5



Сценарий номер 6



Сценарий номер 7

Рис. 3. Изображения сцен, на которых проводились сравнения. Достигнута высокая степень совпадения для всех рендер-систем. Полные данные доступны по адресу: http://ray-tracing.ru/upload/free/hydra/comparison_final.pdf

Практичный подход ускорения синтеза изображений методом трассировки лучей Монте-Карло

А. Груздев, В.Фролов, А. Игнатенко

Московский государственный университет им. М.В. Ломоносова,
Факультет вычислительной математики и кибернетики, Москва, Россия
aleks-gruzdev92@yandex.ru, vfrolov@graphics.cs.msu.ru, ignatenko@graphics.cs.msu.ru

Аннотация

Данная работа предлагает новый высококачественный метод ускорения синтеза изображений методом трассировки путей Монте-Карло. Ключевой идеей данного метода является выделение и обработка (фильтрация) падающего освещения в пространстве экрана на основе дополнительной информации о пикселях (глубина, нормаль и т.п.), что позволяет добиться высокого качества фильтрации и избежать размытия границ. Не накладывая никаких требований на алгоритм трассировки лучей, предложенный метод позволяет ускорить построение изображений методом трассировки путей Монте-Карло на 1 – 2 порядка.

Keywords: трассировка путей Монте-Карло, многомерная фильтрация, интенсивность освещения

1. ВВЕДЕНИЕ

Алгоритм трассировки путей Монте-Карло [Ritschel et al 2011] – физически корректный и высококачественный алгоритм синтеза. Но его существенным недостатком является медленная сходимости.

Возможно ускорение синтеза в результате трассировки небольшого числа лучей, а затем удаления оставшегося шума с помощью фильтрации изображения. В работе предлагается алгоритм, превосходящий аналогичные методы по качеству фильтрации и позволяющий существенно сократить время синтеза изображения.

2. ОБЗОР СУЩЕСТВУЮЩИХ МЕТОДОВ

2.1. Кэш освещенности

Кэш освещенности [Pharr and Humphreys 2004] является одним из способов ускорения вычисления интеграла освещенности на множестве точек. Метод предназначен для работы с диффузной составляющей освещения. Компонента вторичного диффузного освещения на изображении меняется плавно, поэтому ее можно вычислить каким-либо из методов лишь в очень небольшом числе точек, а в остальных – интерполировать.

В работе [Kontkanen et al 2006] рассматривается метод ускорения синтеза методом трассировки путей Монте-Карло или фотонных карт с помощью объединения алгоритма кэша освещенности [Pharr and Humphreys 2004] и фильтрации. Метод позволяет значительно сократить количество выборок, необходимых для формирования кэша, путем фильтрации на этапе выборки значений из кэша. Для фотонных карт фильтрация применяется на этапе финального сбора. Веса фильтра подбираются индивидуально для каждой точки кэша, в зависимости от сложности геометрии в этой точке. Алгоритм позволяет синтезировать изображение в 5 раз быстрее, чем с помощью кэша освещенности. Метод позволяет избежать артефактов, характерных для кэша освещенности, но обладает артефактами, вызванными размытием, и усложняет операцию выборки из кэша освещенности.

2.2. Адаптивная фильтрация

Алгоритм [Suykens and Willems 2000] ускоряет синтез изображений прогрессивным методом трассировки лучей Монте-Карло за счёт фильтрации результата каждой итерации. Авторы предлагают использовать фильтр с ядром Епанечникова [Silverman 1986], размер которого варьируется в зависимости от плотности распределения выборок Монте-Карло по эмпирическому закону. Учитывая лишь распределение выборок, а не особенности сцены, алгоритм плохо сохраняет границы на изображениях.

2.3. Многомерная фильтрация

Авторами статьи [Gastal and Oliveira 2012] предложен многомерный фильтр для изображений, применимый к обработке произвольных сигналов. Рассмотренный фильтр позволяет учесть большое количество измерений, помимо цвета, содержащих дополнительную информацию. Показывая высококачественный результат, сравнимый с билатеральной фильтрацией [Kornprobst and Tumblin 2009], метод обладает линейной сложностью и высокой скоростью работы.

2.4. Фильтрация с wavelet-преобразованием

В [Dammertz et al 2010] предлагается алгоритм для фильтрации освещения в пространстве экрана в реальном времени. Статья предлагает усовершенствование билатерального фильтра с wavelet-преобразованием A-Trous [Holschneider et al 1989], добавляя к преобразованию дополнительную весовую функцию для определения границ на изображении. Для вычисления функции используется дополнительная информация о пикселях: направление нормали и координаты в мировом пространстве. Дополнительный вес в wavelet-преобразовании определяется на основе близости пикселей по нормали и координате. Метод хорошо сохраняет границы на изображениях, но фильтрация страдает наличием артефактов в виде цветных пятен на однородных поверхностях. Реализация на графическом процессоре позволяет обрабатывать изображения 0.25 мегапикселя со скоростью примерно 4 кадра в секунду.

2.5. Адаптивное распределение выборок Монте-Карло.

В статье [Nachisuka et al 2008] рассматривается метод адаптивного распределения Монте-Карло выборок по изображению. Каждый луч рассматривается как точка многомерного пространства (например, учитываются координаты на линзе, время), тем самым задаётся многомерная функция в пространстве изображения. Многомерная функция анизотропно интерполируется, и для следующих выборок определяются позиции ближе к резким изменениям многомерной функции, которые соответствуют тем областям на изображении, где необходимо больше выборок: границы, мягкие тени, области вне фокуса камеры.

Минимизация MSE. Работа [Rousselle et al 2011] описывает итеративный метод расчёта Монте-Карло выборок и синтеза изображений. На каждой итерации метод выбирает наиболее

оптимальное распределение некоторого фиксированного числа Монте-Карло выборок в пространстве изображения, которое минимизирует ошибку по метрике MSE [Lehmann and Casella 1998]. Так как невозможно вычислить MSE без эталонного изображения, то используется аппроксимация [Silverman 1986]. Затем реконструируется изображение по вычисленным выборкам: производится размытие рассчитанного изображения предопределённым набором фильтров (размытие по Гауссу с разным размером ядра), а затем для каждого пикселя выбирается значение, которое минимизирует ошибку по MSE. Оценка ошибки текущей итерации учитывается в следующей итерации для выбора распределения.

Использование нелокальной фильтрации.
Итеративный метод расчёта и фильтрации выборок Монте-

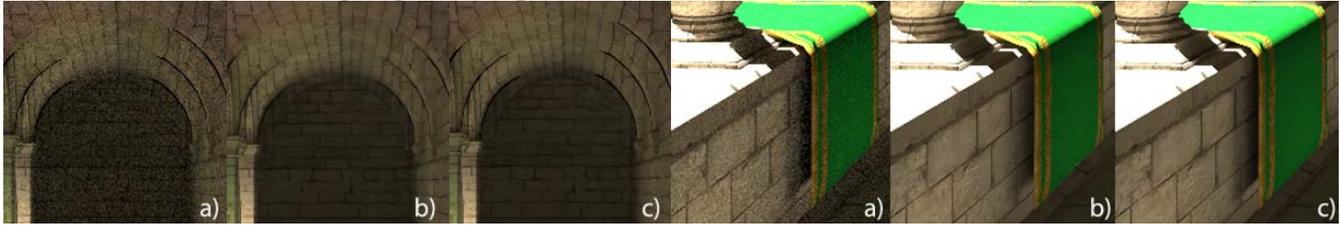


Рис. 1. Пример работы предложенного метода: (а) – без фильтрации, (б) – предложенный алгоритм (~1 мин), (с) – эталон (~50)

2.6. Guided-фильтрация

[He et al 2010] предлагает guided-фильтр, обрабатывающий изображение, руководствуясь дополнительной информацией о каждом пикселе. При удачном выборе дополнительной информации guided-фильтр превосходит по качеству билатеральную фильтрацию. В [Bauszat et al 2011] рассматривается применение guided-фильтрации в трассировке лучей Монте-Карло. В рассмотренном методе освещение разделяется на первичное и вторичное. Вторичное освещение обрабатывается с помощью guided-фильтра, где в качестве дополнительной информации используются карта нормалей, карта глубины и дополнительный буфер цвета, полученный расчётом лучей, проходящих через центр пикселя. Предложенный подход демонстрирует лучшие результаты и по качеству, и по времени по сравнению с кросс-билатеральной фильтрацией [Kornprobst and Tumblin 2009] и A-Trous фильтрацией [Dammertz et al 2010].

2.7. Вероятностная фильтрация

Рассматривая процесс трассировки лучей Монте-Карло как Пуассоновский процесс, авторы [Doidge and Jones 2013] предлагают эффективный способ фильтрации выборок. Для каждого пикселя изображения все выборки кластеризуются по глубине луча, и, исходя из предположения о Пуассоновском процессе, возможно предположение о том, что распределение выборок в кластере одного пикселя может сравниться с распределением в кластерах соседних пикселей, что позволяет фильтру переиспользовать освещение с соседних пикселей. Также фильтр использует информацию о нормальных и глубине для обнаружения границ. Рассмотренный алгоритм позволяет получать изображение при малом количестве выборок на пиксель, по качеству превосходящее кросс-билатеральную фильтрацию [Kornprobst and Tumblin 2009].

В статье [Sen and Darabi 2012] авторы представили метод многомерной фильтрации, учитывающий стохастическую зависимость между случайными величинами в методе

Карло [Rousselle et al 2012] существенно ускоряет процесс трассировки лучей Монте-Карло и превосходит по качеству метод [Rousselle et al 2011]. Каждая итерация делится на три шага. На первом шаге производится расчёт фиксированного числа лучей, и результат записывается в буфер. Затем, на втором шаге, выполняется нелокальная фильтрация буфера методом non-local means denoising [Buades et al 2005]. Третий шаг вычисляет попиксельную невязку и значения невязки, затем используются на первом шаге следующей итерации для распределения лучей в пространстве изображения. Для первой итерации используется равномерное распределение. Быстродействие метода ограничивается вычислительно сложным алгоритмом нелокальной фильтрации, синтез 1 мегапикселя занимает 8 – 16 секунд с ускорением на графическом процессоре.

трассировки лучей Монте-Карло и значениями в полученном семпле. Для каждого пикселя рассматривается вектор, включающий в себя набор случайных величин и дополнительную информацию о сцене (например, распределение точек на линзе). Если шум на изображении вызван случайным распределением точек прохождения луча через линзу, то между распределением точек на линзе и между шумом изображения существует некоторая функциональная зависимость. Наличие такой информации позволяет снизить вклад в ядро фильтра тех пикселей, на цвет которых повлиял шум. В алгоритме предлагается использование билатеральной фильтрации [Kornprobst and Tumblin 2009], параметры которой вычисляются на основе взаимной информации между распределением дополнительной информации в точке и случайных величин.

2.8. Выводы

Кэш освещенности [Pharr and Humphreys 2004], [Kontkanen et al 2006] решает задачу ускорения синтеза изображений, но обладает артефактами и требует существенного изменения алгоритма синтеза.

Работы [Hachisuka et al 2008], [Rousselle et al 2011] и [Rousselle et al 2012] описывают использование более эффективного распределения выборок в пространстве экрана, чем равномерное, однако эти подходы требуют модификации трассировщика лучей.

Методы фильтрация позволяют обойтись меньшими изменениями алгоритма синтеза. Методы [Suykens and Willems 2000], [Kornprobst and Tumblin 2009], [Dammertz et al 2010] обрабатывают изображения, не имея информации о геометрии, и, следовательно, плохо сохраняют границы изображений. Аналогично подходам [Gastal and Oliveira 2012], [Dammertz et al 2010], [He et al 2010], [Doidge and Jones 2013], [Sen and Darabi 2012] предлагается использовать дополнительную информацию о сцене для улучшения качества фильтрации на границах. Метод [Gastal and Oliveira 2012] позволяет объединить идею guided-фильтрации [He et al 2010] для

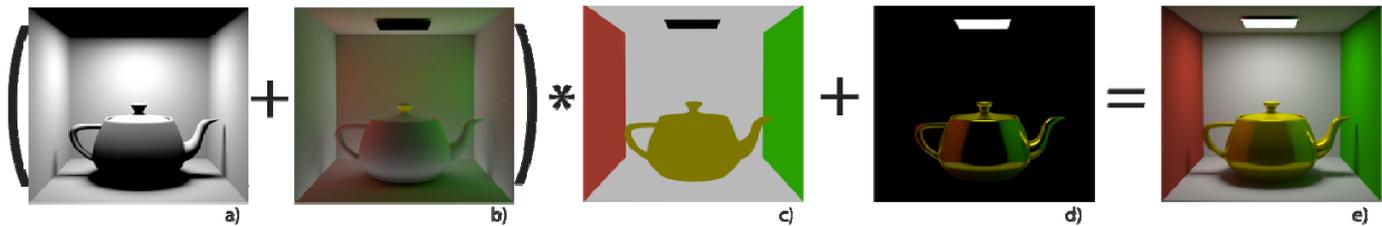


Рис. 2. Вычисление цвета финального изображения: а) интенсивность первичного освещения; б) интенсивность вторичного освещения; в) цвет текстуры; д) отражённое освещение; е) финальное изображение

сохранения границ и нелокальной фильтрации [Buades et al 2005], улучшающей обработку однородных областей.

3. ПРЕДЛОЖЕННЫЙ АЛГОРИТМ

В данной работе предлагается применение идей, описанных в статье [Gastal and Oliveira 2012], к задаче синтеза изображений методом трассировки путей Моне-Карло. Метод основан на предположении, что изображение сцены F может быть представлено в виде следующей формулы:

$$F = (D + I) * T + R, \quad (1)$$

где D – интенсивность первичного освещения; I – интенсивность вторичного освещения; T – цвет текстуры; R – освещение, отражённое от зеркальных поверхностей (рис. 2). Трассировщик лучей, таким образом, должен иметь возможность отдельно рассчитывать указанные изображения. Ключевой идеей такого подхода является отделение интенсивности падающего освещения от цвета пикселя. Такой подход позволяет отделить фильтрацию зашумлённой интенсивности падающего освещения от рисунка текстуры, т.е. переиспользовать значения соседних пикселей с одинаковой освещённостью, но разным цветом текстуры. В результате чего в предложенном подходе фильтр не размывает текстуру в отличие от [Gastal and Oliveira 2012].

Предложенный алгоритм можно разделить на три шага:

1. Предобработка освещения: перевод пикселей в non-local-means space [Buades et al 2005], [Gastal and Oliveira 2012].
2. Фильтрация освещения. Применяется сепарабельный фильтр, описанный в [Gastal and Oliveira 2012] использующий в качестве дополнительной информации освещение в non-local-means space, карту нормалей и карту глубины.
3. Суммирование освещения и синтез финального изображения по формуле (1).

Первые два шага применяются одинаковым образом к первичному (D) и вторичному (I) освещению. Затем выполняется третий шаг, использующий результаты

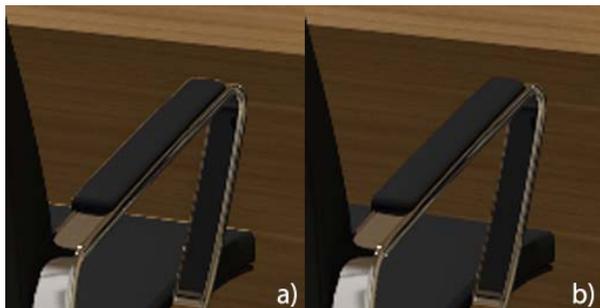


Рис. 3. Устранение артефактов на границах

предыдущих шагов. При расчёте изображения по формуле (1) могут возникать артефакты на границах (рис. 3) из-за того, что на границах объектов изображения не соблюдается введенное предположение (формула 1), поскольку цвет пикселей, принадлежащих граничным объектам, складывается из освещения различных поверхностей, и в этом случае формула 1 не выполняется.

3.1. Устранение ступенчатости

Для устранения артефактов используется аналогичная MSAA техника (рис. 3). [Lelegr 1980]. Для того чтобы устранить артефакт ступенчатости, фильтр на основе дополнительной информации выделяет множество граничных пикселей G , не похожих по нормали и глубине на соседние пиксели. Для этих пикселей трассировщик лучей сохраняет дополнительно 16 суб-пикселей (4x4), содержащих нормаль и глубину. Далее если пиксел принадлежит множеству G , фильтрация для него выполняется 16 раз с различной доп. информацией суб-пикселей, а результат усредняется.

3.2. Отраженное освещение

Отражённое освещение R не обрабатывается фильтром, так как для него нужен другой набор дополнительной информации. Однако возможно рекурсивное применение метода по глубине переотражения лучей. Т.е. алгоритм полностью повторяется для отражённого освещения: сохраняется интенсивность освещения, вычисляется дополнительная информация для отражённых лучей. Затем в формуле (1) более высокого уровня используется отфильтрованное освещение более низкого уровня.

4. РЕЗУЛЬТАТЫ

Для проведения экспериментальных исследований производился синтез изображений в разрешении 1024x768. Для синтеза входных данных фильтрации рассчитывалось 256 путей на пиксель, для эталонного изображения – 4096 путей на пиксель.

MSE	Adaptive manifolds filtering	Adobe Photoshop denoising	Предложенный алгоритм	Коэф. освещённости
«Переговорная комната»	7.73	7.15	1.50	2.05
«Cornel box»	8.12	7.05	1.64	1.58
«Спальня»	6.73	6.11	3.12	2.65

Таблица 1. Сравнение качества фильтрации с альтернативными методами



(a) Adobe Photoshop

(b) Adaptive Manifolds

(c) Предложенный метод

(d) Кэш освещённости

Рис. 4. Сравнение предложенного метода с альтернативными методами: а) Adobe Photoshop (реализация билатеральной фильтрации); б) Adaptive Manifolds [Gastal and Oliveira 2012]; в) Предложенный метод; г) Кэш освещённости [Pharr and Humphreys 2004]

Время синтеза (сек)	Синтез входной информации	Фильтрация на CPU	Суммарное время на CPU	Фильтрация на GPU	Суммарное время на GPU	Кэш освещённости	Синтез эталона
«Cornel box»	53,40	6,42	59,82	0,16	53,56	24,02	611,13
«Переговорная комната»	56,83	6,40	63,23	0,16	59,99	98,23	629,04
«Спальня»	70,21	6,53	76,74	0,17	70,38	111,41	716,22
«Спонза»	69,04	6,59	75,63	0,16	69,20	26,83	643,65

Таблица 2. Сравнение скорости фильтрации с альтернативными методами

По результатам оценки качества фильтрации (табл. 1 и 2) видно, что предложенный метод обладает серьёзным преимуществом перед Adaptive Manifold filtering [Gastal and Oliveira 2012] и Adobe Photoshop denoising, но остаётся наравне с кэшем освещённости [Pharr and Humphreys 2004] (рис. 4). Тем не менее, предложенный метод лишен характерных артефактов в кэша освещённости в углах и является более простым в реализации.

5. ССЫЛКИ

[1] [Bauszat et al 2011] Bauszat, P., Eisemann, M., Magnor, M.: *Guided Image Filtering for Interactive High-quality Global Illumination*. In Computer Graphics Forum, Vol. 30, No. 4. – P. 1361 - 1368. Blackwell Publishing Ltd, June, 2011.

[2] [Buades et al 2005] Buades, A., Coll, B., Morel, J.M.: *A non-local algorithm for image denoising*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2005.

[3] [Dammertz et al 2010] Dammertz, H., Sewtz, D., Hanika, J., Lensch H.: *Edge-Avoiding A-Trous Wavelet Transform for fast Global Illumination Filterin*. In High Performance Graphics, 2010.

[4] [Doidge and Jones 2013] Doidge, I. Jones. – M.: *Probabilistic illumination-aware filtering for Monte Carlo rendering*. In The Visual Computer, 29(6 - 8). – P. 707 - 716, 2013.

[5] [Gastal and Oliveira 2012] Gastal, E., and Oliveira. – M.: *Adaptive manifolds for real-time high-dimensional filtering*. In ACM Transactions on Graphics (TOG), 31(4), 33, 2012.

[6] [Gorban et al 2008] Gorban, A., Kégl, B., Wunsch, D., and Zinovyev, A.: *Principal manifolds for data visualization and dimension reduction*. ISBN 978-3-540-73750-6, 2008.

[7] [Hachisuka et al 2008] Hachisuka, T., Wojciech, J., Weistroffer, R., Dale, K., Humphreys, G., Zwicker, M., Jensen, H.: *Multidimensional Adaptive Sampling and Reconstruction for Ray Tracing*. In ACM Transactions on Graphics, Vol. 27, No. 3, Article 33, August, 2008.

[8] [He et al 2010] He, K., Sun, J. and Tang, X.: *Guided image filtering*. In Computer Vision–ECCV 2010. – P. 1 - 14. Springer Berlin Heidelberg, 2010.

[9] [Holschneider et al 1989] Holschneider, M., Kronald-Martinet, R., Morlet, J., Tchamitchian, P.: *A real-time algorithm for signal analysis with the help of the wavelet transform*. Springer-Verlag, 1989.

[10] [Kontkanen et al 2006] Kontkanen, J., Räsänen, J. and Keller, A.: *Irradiance filtering for monte carlo ray tracing*. In “Monte Carlo and Quasi-Monte Carlo Methods 2004” – P. 259 - 272. Springer Berlin Heidelberg, 2006.

[11] [Kornprobst and Tumblin 2009] Kornprobst, P., and Tumblin, J.: *Bilateral filtering: Theory and applications*. Now Publishers Inc, 2009.

[12] [Lehmann and Casella 1998] Lehmann, E. and Casella, G.: *Theory of point estimation*. ISBN 0-387-98502-6, 1998.

[13] [Leler 1980] Leler, W.: *Human Vision, Anti-aliasing, and the Cheap 4000 Line Display*. In ACM SIGGRAPH Computer Graphics 14 (3): 308 – 313, July, 1980.

[14] [Pharr and Humphreys 2004] Pharr, M. and Humphreys, G.: *Physically based rendering: from theory to implementation*. Morgan Kaufmann Publishers Inc., 2004.

[15] [Ritschel et al 2011] Ritschel, T., Dashesbacher, C., Grosch, T., Kautz, J.: *The state of the art in interactive global illumination*. In Computer Graphics Forum, vol 31. – P. 160 - 188, February 2012.

[16] [Rousselle et al 2012] Rousselle, F., Knaus, C. and Zwicker, M.: *Adaptive Rendering with Non-Local Means Filtering*. In ACM Transactions on Graphics, Vol. 31, No. 6, November 2012

[17] [Rousselle et al 2011] Rousselle, F., Knaus, C. and Zwicker, M.: *Adaptive sampling and reconstruction using greedy error minimization*. In Proceedings of the 2011 SIGGRAPH Asia Conference, ACM, New York, NY, USA, 159:1 – 159:12, 2011.

[18] [Sen and Darabi 2012] Sen, P., and Darabi, S.: *On filtering the noise from the random parameters in Monte-Carlo rendering*. In ACM Trans. Graph. 31, 3, June, 2012.

[19] [Silverman 1986] Silverman, B.: *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, New York, 1986.

[20] [Suykens and Willems 2000] Suykens, F. and Willems, Y.: *Adaptive filtering for progressive monte carlo image rendering*. In WSCG, February, 2000.

Интерактивный визуальный анализ многомерных данных

О. Масленников*, И. Мильман*, А. Сафиуллин*, А. Бондарев**, Ш. Низаметдинов*, В. Пилогин*

НИЯУ МИФИ*

Институт прикладной математики им. М.В. Келдыша РАН**

Москва, Россия

maslolpavl@gmail.com; igalush@gmail.com; amir147@rambler.ru; bond@keldysh.ru; sh_nizam@mail.ru; pilyugin@sv-journal.com

Аннотация

Работа посвящена вопросам разработки интерактивной системы, предназначенной для решения задач визуального анализа многомерных данных. Рассматриваемые в примерах и иллюстрациях задачи относятся к области визуальной аналитики. Система позволяет непосредственную работу пользователя с визуальными представлениями многомерного облака данных в пространствах меньшей размерности, выдвижение и проверку гипотез о строении и характере изучаемых данных с помощью геометрических построений в интерактивном режиме.

Ключевые слова: визуальная аналитика, анализ многомерных данных, интерактивный интерфейс

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

Основные методы, подходы и алгоритмы визуальной аналитики описаны в работах [1 – 5]. В этих же работах приведен ряд примеров современного применения визуальной аналитики в различных сферах человеческой деятельности, а также приведены описания ряда программных продуктов, построенных на основе визуальной аналитики.

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 отображения результатов применения методов Data Analysis.

Данная работа представляет разрабатываемую интерактивную систему визуального анализа многомерных данных. В рамках разрабатываемой системы

рассматриваются классические задачи анализа многомерных данных, такие, как: построение кластеров и их оболочек 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:

- 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;

- строить оболочки обнаруженных кластеров, максимально приближенные к данным, в системе координат главных компонент;

- 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 [6 - 9].

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 их оболочек различными способами.

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).

Оптическая модель сцены предполагает присвоение цилиндрам цвета, отвечающего расстоянию между точками. Чем ближе расстояние к d , тем синее цилиндр.

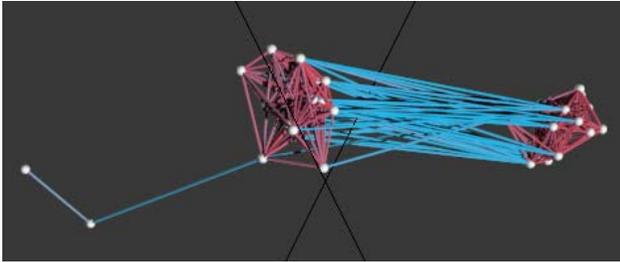


Рис. 1. Отображение множества точек

Пользователю предлагается проводить анализ разбиения на кластеры в зависимости от параметра d . Предусматривается два метода перехода – последовательный просмотр при задаваемых пользователем d или задание двух значений параметра и построение видеоряда.

Анализ формы кластера предлагается методом построения оболочки в трехмерном пространстве. Построение оболочки возможно несколькими методами – построение прямоугольного параллелепипеда методом пересечения сфер и смешанным методом.

В первом методе прямоугольный параллелепипед строится на осях, полученных с помощью метода главных компонент. Данное построение гарантирует хорошее приближение параллелепипеда к кластеру (рис. 2).

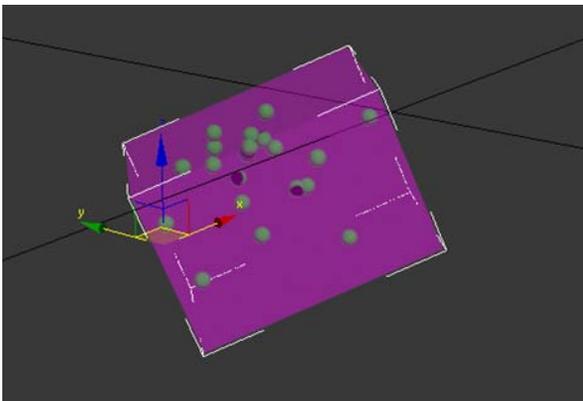


Рис. 2. Оболочка в виде прямоугольного параллелепипеда для кластера

Во втором методе строятся сферы с центром в каждой точке кластера и радиусом, равном максимальному расстоянию от этой точки до точек кластера (оно заведомо меньше либо равно d). А затем строится пересечение сфер. Полученная оболочка, очевидно, включает в себя все точки и дает достаточно хорошее приближение (рис.3). Данная оболочка лучше предыдущей при относительно одинаковых собственных значениях матрицы корреляций внутри кластера.

Смешанный метод построения оболочки предполагает построения оболочек по двум описанным методам и дальнейшее пересечение двух оболочек.

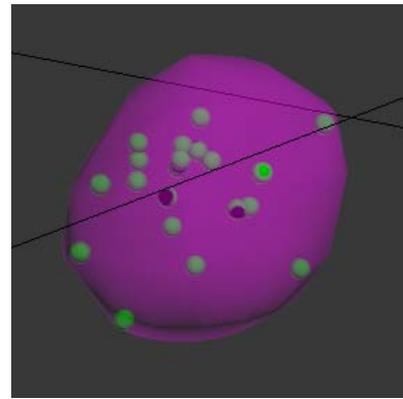


Рис. 3. Оболочка, построенная с помощью пересечения сфер

Решение задачи дискриминантного анализа 2D и 3D проекционным методом

Решение данной задачи предоставляет пользователю возможность построения плоскости, разделяющей классы точек, при работе с проекциями исходного многомерного объема данных на двумерные и трехмерные подпространства, образованные из исходных координат.

Основным предположением дискриминантного анализа является то, что существуют две или более группы, которые по некоторым переменным отличаются от других групп, причем такие переменные 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.

Основным методом решения задачи дискриминантного анализа является метод нахождения коэффициентов гиперплоскости Фишера [9]. В результате исследования методов решения задачи в качестве метода решения был предложен метод построения разделяющей гиперплоскости при помощи метода последовательных проекционных изображений. Суть метода заключается в том, что если мы можем в проекции построить разделяющую плоскость, то при переходе к пространству с размерностью на единицу больше, данная плоскость будет являться также разделяющей. В качестве алгоритма решения был предложен последовательный просмотр и анализ 2х и 3х-мерных проекций с целью нахождения разделяющей плоскости или системы плоскостей (рис. 4).

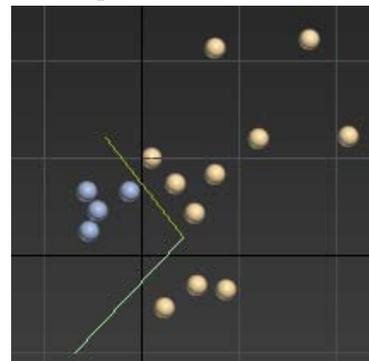


Рис. 4. Система разделяющих прямых для двух групп точек

После нахождения такой системы пользователю предоставляется возможность проведения верификации построенной формальной системы решающих правил. После проведения верификации пользователь может решать задачу классификации новых объектов, добавляемых к исходному многомерному объему данных (рис. 5).

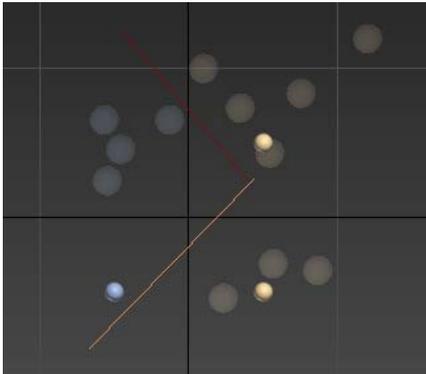


Рис. 5. Решение задачи классификации новых объектов

Данный метод позволяет принимать решения о возможности построения решающих правил для задач классификации новых объектов.

Решение задачи выделения кластеров 2D проекционным методом

Цель решения данной задачи в разрабатываемой интерактивной системе – предоставление пользователю возможности одновременной работы со всеми проекциями исходного многомерного объема на двумерные подпространства, образованные из исходных координат. Исходя из соображения, что точки, близкие во всех двумерных проекциях, будут близки и в исходном многомерном пространстве, пользователь интерактивной системы может выделять близкие точки на двумерной проекции, маркировать их цветом, вносить или удалять точки. Все действия пользователя отображаются одновременно на всех двумерных проекциях.

Алгоритм решения задачи:

Этап 1. Точки исходного многомерного пространства проецируются на двумерные подпространства, образованные из исходных координат. Таким образом, получается матрица проекций (рис. 6).

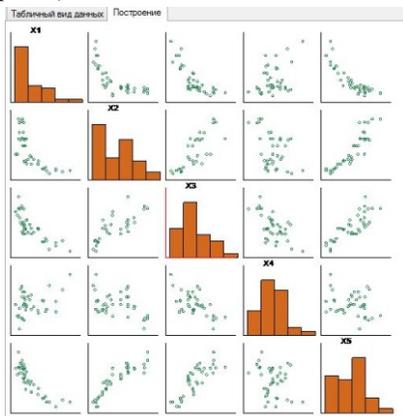


Рис. 6. Матрица проекций

Этап 2. На одной из проекций выделяются характерные образования – кандидаты на сгустки.

Этап 3. Проводится анализ всех остальных проекций, при обнаружении точек, лежащих далеко от сгустка, данные точки исключаются из рассмотрения.

Этап 4. Оставшиеся выделенные точки помечаются как кластер и исключаются из дальнейшего рассмотрения. Если не осталось сгустков точек (одиночные точки, либо все точки помечены как кластер), то переходим к этапу 5, иначе – к этапу 2.

Этап 5. В результате пользователем получена первичная картина кластерного разбиения изучаемого многомерного объема данных. Для дальнейшего улучшения картины в системе реализован алгоритм к-средних.

Для наглядности представления результатов кластеризации в системе реализовано построение профильной диаграммы (рис. 7) как способа двумерного представления объектов кластеризации.

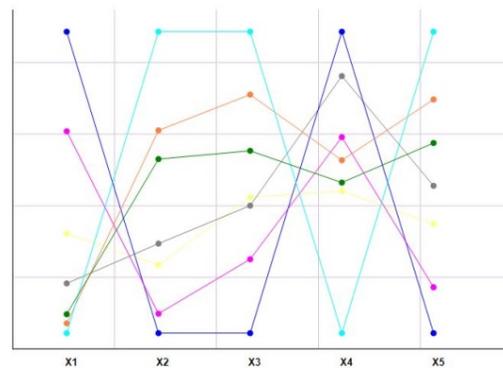


Рис. 7. Профильная диаграмма

В результате работы с интерактивной системой пользователь может также определить, какие объекты являются спорными, т.е. могут принадлежать одновременно к нескольким кластерам.

3. ЗАКЛЮЧЕНИЕ

В данном докладе представлен ряд реализованных задач, относящихся к разрабатываемой 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.

4. ССЫЛКИ

[1] Thomas J., Cook K. Cook, Illuminating the Path: Research and Development Agenda for Visual Analytics. IEEE-Press, 2005.

[2] Keim D.A, Mansmann F, Schneidewind J, Thomas J, Ziegler H: Visual analytics: Scope and challenges, Visual Data Mining: 2008. – P. 82.

- [3] 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. – P. 154 – 175, 2008. Springer-Verlag Berlin Heidelberg 2008.
- [4] 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.
- [5] Keim D., Kohlhammer J., Ellis G. and Mansmann F. (Eds.), Mastering the Information Age – Solving Problems with Visual Analytics, Eurographics Association, 2010.
- [6] Пилюгин В.В., Маликова Е.Е., Пасько А.А., Аджиев В.Д. *Научная визуализация как метод анализа научных данных / Научная визуализация. Т.4, № 4. – С .8 – 25, 2012, URL: <http://sv-journal.com/2012-4/06.php?lang=ru>*
- [7] Бондарев А.Е., Галактионов В.А. *Анализ многомерных данных в задачах многопараметрической оптимизации с применением методов визуализации / Научная визуализация. Т.4, № 2. – С. 1 – 13, 2012, URL: <http://sv-journal.com/2012-2/01.php?lang=ru>*
- [8] *Основы научной визуализации [сайт]. URL: <http://edu-cons.net/unl/> (дата обращения: 10.05.2014)*
- [9] Низаметдинов Ш.У. *Анализ данных. – М.: МИФИ, 2006.*

Об авторах

Масленников Олег – студент НИЯУ МИФИ.

E-mail: maslolpavl@gmail.com

Сафиуллин Амир – студент НИЯУ МИФИ.

E-mail: amir147@rambler.ru

Мильман Игаль – студент НИЯУ МИФИ.

E-mail:

igalush@gmail.com

Бондарев Александр – к.ф.-м. н., старший научный сотрудник ИПМ им. М.В. Келдыша РАН.

E-mail: bond@keldysh.ru

Низаметдинов Шамиль – к.т.н., доцент кафедры системного анализа НИЯУ МИФИ.

E-mail: sh_nizam@mail.ru

Пилюгин Виктор – к.т.н., профессор, заведующий лабораторией «Научная визуализация» НИЯУ МИФИ.

E-mail: pilyugin@sv-journal.com

INTERACTIVE VISUAL ANALYZING OF MULTIDIMENSIONAL DATA

Abstract

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 his hypotheses about types of data inside the volume by interactive geometrical constructions.

Keywords: *visual analytics, multidimensional data, interactive system*

Постобработка и анализ синтезированных реалистичных изображений

Б. Барладян, Е. Бирюков, И. Валиев, А. Волобой, Л. Шапиро
Институт прикладной математики им. М.В. Келдыша РАН, Москва

АННОТАЦИЯ

В работе сформулированы требования к системам анализа изображений, получаемых в системах реалистичной визуализации в результате физически корректного моделирования распространения света. Рассмотрены детали реализации программы постобработки и анализа полученных синтезированных изображений.

Ключевые слова: системы реалистичной визуализации, анализ результатов моделирования, изображения с высоким динамическим диапазоном, регрессионное тестирование, операторы сжатия динамического диапазона яркости

1. ВВЕДЕНИЕ

Типичным результатом работы систем оптического моделирования и реалистичной компьютерной графики являются изображения. Эти синтезированные изображения следует разделить на два класса:

- изображения, предназначенные для восприятия человеком;
- результаты оптического моделирования, полученные с помощью виртуальных сенсоров и представляющие распределение физических значений на плоскости (в заданном прямоугольнике) или в пространстве направлений (отображение гониодиаграммы).

В разрабатываемых нами системах оба типа изображений получаются в результате моделирования в физических единицах в динамическом диапазоне чисел с плавающей точкой, т.е. являются изображениями с высоким динамическим диапазоном, в отличие от обычных изображений на экране монитора, диапазон значений которых мал.

Для отображения результатов на графическом мониторе изображения должны преобразовываться в RGB пространство монитора с учетом особенностей человеческого зрения. Функционально это преобразование близко к возможностям, предоставляемым в системах обработки фотоизображений для профессиональных фотографов, таких, как Lightroom [1], Apple Aperture [2], Phase One Capture One [3]. Однако требования к преобразованиям изображений, полученных в системах реалистичной визуализации, и изображений, полученных с помощью фотоаппарата, имеют существенные различия. Основной задачей программ обработки фотоизображений является обеспечение пользователя широким спектром возможностей целенаправленного редактирования фотоизображений, полученных в RAW [4] формате, для максимального проявления и усиления художественных достоинств фотографии или создания различных спецэффектов. Динамический диапазон фотоизображений в RAW формате расширен до 12 – 14 бит на пиксел, но истинные значения яркостей, как правило, неизвестны. В то же время для обоих типов изображений, получаемых в системах реалистичной компьютерной графики и оптического моделирования, известны и сохраняются в соответствующих файлах значения «истинных» яркостей и

освещенностей, полученные в результате моделирования. Соответственно, важной задачей при обработке и анализе синтезированных изображений является получение итоговых фотореалистичных изображений с учетом специфики человеческого зрения и анализ различных физических характеристик, сохраненных в полученных изображениях. Пользователи могут интересоваться значениями яркости или освещенности для различных сечений изображения, различные статистические характеристики для выделенных областей. При спектральном моделировании пользователя может также интересовать спектральный состав яркости или освещенности в различных точках и областях изображения.

Анализ изображений, представляющих результаты моделирования на виртуальных сенсорах, помимо удобного визуального представления, требует также специальных средств, учитывающих специфику данного сенсора. Оси координат у таких изображений могут быть как в линейных, так и в угловых единицах. Изображение может представлять различные физические единицы – яркость, освещенность, интенсивность и т.д.

2. ТРЕБОВАНИЯ К ПОСТОБРАБОТКЕ И АНАЛИЗУ ПОСТРОЕННЫХ ИЗОБРАЖЕНИЙ

Исходя из опыта практического использования систем реалистичной компьютерной графики и оптического моделирования, можно сформулировать следующие требования к подсистеме постобработки и анализа результатов работы таких систем:

1. Подсистема должна обеспечивать загрузку изображений в различных форматах. В первую очередь должны быть поддержаны форматы всех типов изображений и результирующих данных, создаваемых базовой системой, включая спектральные изображения и изображения, формируемые на виртуальных измерительных приборах.

2. Подсистема должна обеспечивать преобразование исходного изображения, заданного в физических единицах в динамическом диапазоне чисел с плавающей точкой, в RGB пространство монитора с ограниченным динамическим диапазоном. Должен быть обеспечен выбор хотя бы из нескольких типовых операторов сжатия динамического диапазона, удобное и эффективное варьирование их параметров для получения изображений необходимого качества.

3. Подсистема должна также обеспечивать визуализацию изображений в искусственных цветах (color contours visualization) для представления распределения значений физических величин на изображении в виде, удобном для восприятия человеком. Здесь также необходимо обеспечить гибкий контроль параметров визуализации. Должна быть обеспечена возможность установления соответствия физических величин и цветов для их представления, как автоматически, так и по явному заданию пользователя. Необходимо обеспечивать и дискретный, и плавный переходы между цветами.

4. Изображения, получаемые в системе моделирования, могут также являться сферическими панорамами или быть результатами моделирования, полученными на виртуальных измерительных приборах в полярных координатах. В силу этих причин подсистема должна поддерживать визуализацию изображений, как в декартовой, так и полярной системе координат, а также визуализировать панорамные изображения в виде, удобном для восприятия человеком.

5. Необходимо обеспечить экспорт постпроцессированных (итоговых) изображений в форматы, необходимые для создания различного рода презентаций.

6. Подсистема также должна предоставлять возможность экспорта физических данных, таких, как, например, яркость, освещенность, интенсивность, цветность (x, y chromaticity), в текстовые форматы, пригодные для загрузки в текстовые процессоры для непосредственного анализа человеком или как входные данные для других программ.

7. Подсистема должна поддерживать работу с изображениями большой размерности практически без ограничений по используемой памяти.

8. Для обеспечения регрессионных тестов подсистема предоставляет возможности для анализа сравнения двух и, возможно, более изображений.

9. Для каждого пикселя изображения, полученного из системы моделирования освещенности, необходимо иметь возможность представить соответствующее ему физическое значение, как в фотометрических, так и в радиометрических единицах и в различных цветовых пространствах (RGB, спектральном, HSV и т.д.). Для изображений, сохраненных в спектральном пространстве, должна визуализироваться спектральная плотность анализируемой величины. А для величин, полученных с помощью моделирования стохастическими методами, желательно показывать оценку точности вычисления данного пикселя.

10. Подсистема должна обеспечивать визуализацию физических величин в различных сечениях изображения в графическом и табличном виде.

3. РЕАЛИЗАЦИЯ ПОДСИСТЕМЫ АНАЛИЗА ИЗОБРАЖЕНИЙ

Разработанная нами подсистема постобработки и анализа изображений используется в различных системах реалистичной компьютерной графики. Она может использоваться автономно для поддерживаемых форматов входных данных (изображений). Рассмотрим детали реализации требований, сформулированных в предыдущем разделе.

3.1. Поддерживаемые входные форматы

В настоящее время подсистема поддерживает следующие форматы изображений:

1. Radiance RGBE формат, предложенный G. Ward в [6]. Этот формат использует для представления данных в RGB пространстве 4 байта на пиксель и обеспечивает динамический диапазон ~76 порядков и относительную точность ~1% [7].

2. Integra LUX/NIT – формат [5] для сохранения изображений, описывающих освещенность или яркость в RGB пространстве, в динамическом диапазоне плавающих чисел (12 байт на пиксел).

3. Integra Radiance/Iradiance – формат [5] для сохранения изображений, описывающих освещенность или яркость в

спектральном пространстве, в динамическом диапазоне плавающих чисел (4 байта для каждой длины волны на пиксел).

4. Adobe Digital Negative (DNG) – открытый формат упаковки файлов необработанных данных, сформированных с помощью цифровых камер [8].

Поддержка Adobe DNG формата была реализована с помощью открытых кодов из [9].

3.2. Сжатие динамического диапазона

Подсистема обеспечивает преобразование исходного изображения, представленного в физических единицах в динамическом диапазоне чисел с плавающей точкой, в RGB пространство монитора с ограниченным динамическим диапазоном.

В системе реализованы следующие три варианта операторов сжатия динамического диапазона:

1. Нелинейный оператор сжатия динамического диапазона из работы [10].

2. Простой линейный оператор сжатия, контролируемый единственным параметром – максимальным значением величины яркости или освещенности, которые не будут обрезаны.

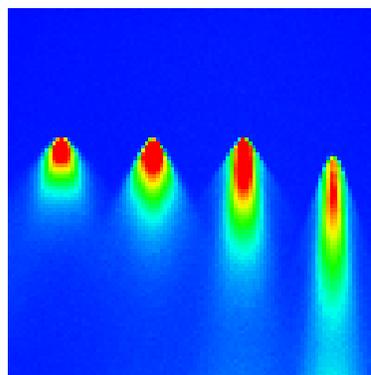
3. Табулированный оператор сжатия, задаваемый таблицей значений, редактируемой пользователем.

В случае использования нелинейного оператора сжатия управление осуществляется либо с помощью максимального значения величины яркости или освещенности, которые не будут обрезаны, либо с помощью адаптивной яркости сцены.

Линейный оператор сжатия управляется единственным параметром – максимальным значением величины яркости или освещенности. Этот параметр может задаваться, как непосредственно, так и через набор фотографических параметров: время экспозиции, относительное отверстие (диафрагменное число) объектива и чувствительность камеры.

3.3. Визуализация в искусственных цветах

Визуализация изображений в искусственных цветах, пример которой показан на рис. 1, предназначена для представления значений физических величин в виде, удобном



для восприятия человеком. Интерфейс управляющих параметров в этом случае позволяет контролировать, как цвета визуализации, так и значения яркостей, освещенностей или точности вычислений, соответствующих этим цветам.

Рис. 1. Изображение в искусственных цветах

3.4. Способы визуализации

Изображения, получаемые в базовой системе моделирования, могут быть не только обычными изображениями, но и являться результатами моделирования на виртуальных измерительных приборах (сенсорах) в полярных и декартовых координатах. Также изображения могут быть сферическими панорамами. Для работы с такими изображениями подсистема способна визуализировать панорамные изображения и поддерживает, как декартову, так и полярную систему координат.

Для визуализации сферической панорамы в нашей подсистеме был реализован специальный режим, показывающий соответствующие изображения с помощью перспективной проекции. Камера находится в центре сферы и может поворачиваться для визуализации требуемой части панорамы. Также можно варьировать угол зрения камеры. Ориентация камеры изменяется движением мыши при нажатой левой клавише. Для изменения угла поворота камеры вокруг ее направления зрения необходимо одновременно с левой клавишей мыши удерживать нажатую клавишу Ctrl. Угол зрения камеры варьируется путем вращения колеса мыши.

Визуализация углового распределения интенсивности на виртуальном измерительном приборе «гонометре» может быть также представлена в искусственных цветах, как это показано на рис. 2.

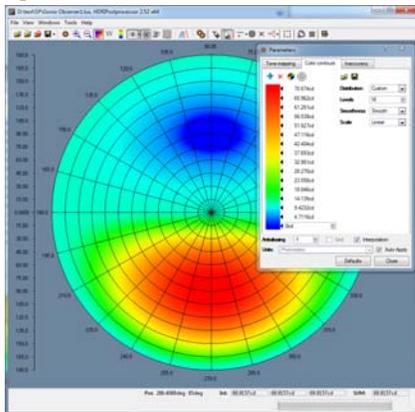


Рис. 2. Представление распределения интенсивности света на виртуальном гониометре

3.5. Экспорт изображений

Подсистема поддерживает сохранение итоговых изображений в таких стандартных форматах, как PNG, JPEG, TIFF и BMP.

Также можно экспортировать загруженные изображения в известный формат HDR [6] и Integra Nit/Lux, Radiance/Irradiance форматы.

3.6. Регрессионный анализ

Для того чтобы была возможность провести регрессионный анализ, подсистема позволяет загрузить два изображения одинакового типа и разрешения. Система вычисляет их разность, и для нее поддерживаются все функциональности подсистемы, что и для одиночного изображения. Регрессионный анализ позволяет отследить изменения функциональности системы моделирования от версии к версии, а также влияние на результат моделирования таких параметров, как, например, количество использованных лучей в трассировке методом Монте-Карло, изменения

алгоритмов генерации лучей и их взаимодействия с поверхностью и средой распространения.

Размер изображений, анализируемых подсистемой, в 64-битной версии программы ограничивается только объемом доступной памяти.

3.7. Пиксельный анализ изображений

Специальное диалоговое окно позволяет получать детальную информацию о каждом пикселе изображения. Цветность пикселя может быть представлена в CIE xy координатах, RGB, HCL и спектральном пространстве. HCL является фактически улучшением известных цветовых пространств HSV и HSL. CIE xy координаты отображаются на цветовом треугольнике графического монитора, что позволяет сразу распознать случаи, когда цвет исследуемого пикселя находится за пределами допустимой палитры, как это видно на примере, показанном на рис. 3.

Кроме цвета пикселя показывается также величина оценки точности вычислений и значение физической величины яркости или освещенности, представленной данным изображением.

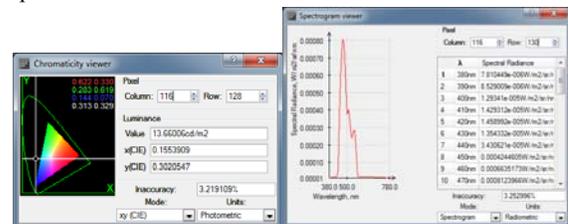


Рис. 3. Анализ цвета точки изображения

Анализируемый пиксель выбирается непосредственно в окне изображения с помощью мыши. В этом случае координаты пикселя отображаются в ячейках Столбец (Column) и Строка (Row). Однако координаты могут быть также заданы непосредственно с клавиатуры.

Информация о пикселе представляется в фотометрических или в радиометрических единицах.

3.8. Анализ сечений и областей

Подсистема обеспечивает визуализацию графиков изменения физической величины вдоль горизонтальных и вертикальных сечений изображения. Сечения добавляются с помощью специальной кнопки на панели инструментов. После добавления положение сечения может изменяться с помощью мыши в пределах окна визуализации. Несколько сечений могут существовать одновременно, их графики визуализируются в дополнительном окне (рис. 4).

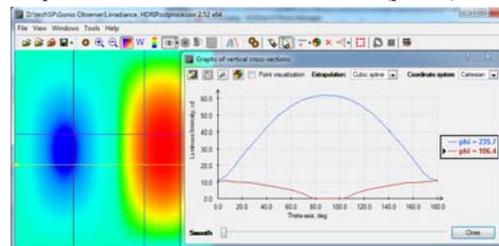


Рис. 4. Графики значений физических величин (яркости)

Численные значения величин в сечениях можно также вывести в виде таблицы, которая может быть сохранена в текстовом файле. Подсистема также обеспечивает анализ прямоугольных и эллиптических областей изображения. Примеры этого приведены на рис. 5.

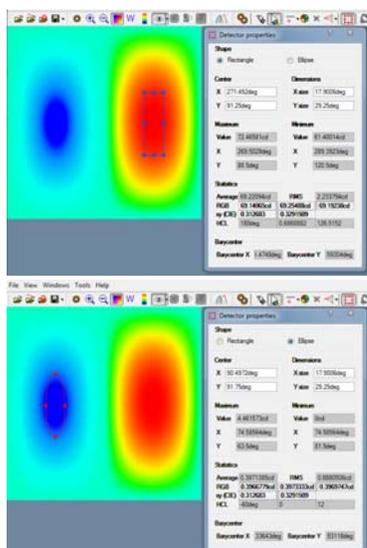


Рис. 5. Анализ значений физических величин

Форма и положение области на изображении может задаваться произвольно, как прямым изменением числовых параметров в диалоговом окне, так и мышью непосредственно в окне изображения с помощью “горячих” точек. В диалоговом окне при этом будут отображаться различные статистики для выбранной области – минимальные и максимальные значения, среднее значение и среднеквадратичное отклонение от среднего значения. Среднее значение отображается при этом одновременно в пространствах RGB, xy (CIE) и HCL.

3.9. Экспорт текстовых данных

Вся информация, рассчитанная системой оптического моделирования и содержащаяся в изображении, может быть экспортирована в текстовый файл в txt или csv формат. Последний формат удобен для последующего ввода данных в табличные процессоры (например, Microsoft Excel). Кроме непосредственно значений физических величин, эти файлы содержат также дополнительную статистическую информацию, зависящую от типа файла. Общими для всех файлов являются среднее, минимальное и максимальное значения данной физической величины. Для изображений, являющихся результатом регистрации на виртуальных измерительных приборах, выводится также полная величина зарегистрированного потока в радиометрических и фотометрических единицах.

Для исследуемого изображения может выводиться не только непосредственно физическая величина в каждом пикселе, но и ее производные. Так, например, для спектрального изображения, зарегистрированного на виртуальном измерительном приборе в полярных координатах, могут выводиться файлы, содержащие для каждого пикселя интенсивность, цветность, спектрограмму, оценку ошибки вычислений и количество лучей.

4. РЕЗУЛЬТАТЫ

Предложенная подсистема постобработки и анализа изображений, сгенерированных системами реалистичной компьютерной графики, постоянно используется для анализа полученных результатов, отладки алгоритмов моделирования распространения света, получения итоговых изображений в пространстве графического монитора. Эта система

удовлетворяет большинству требований, сформулированных в разделе 2.

Система поставляется с несколькими системами реалистичной визуализации и оптического моделирования. Однако может использоваться и автономно. Для использования ее вместе с новыми системами может потребоваться расширение списка поддерживаемых входных форматов изображения. Возможно также и добавление специфических методов анализа.

Работа поддержана грантами РФФИ № 13-01-00454, 12-01-00560, а также фирмой Integra Inc. (Япония).

5. ССЫЛКИ

- [1] <http://adobe-photoshop-lightroom.ru.softonic.com/>
- [2] <http://www.apple.com/ru/aperture/>
- [3] <http://www.phaseone.com/>
- [4] Фрейзер Брюс. Реальный мир Camera Raw и Adobe Photoshop CS2 = World Camera Raw with Adobe Photoshop CS2. – «Вильямс», 2007. – Р. 304.
- [5] Жданов Д.Д., Потемин И.С., Галактионов В.А., Барладян Б.Х., Востряков К.А., Шапиро Л.З. Спектральная трассировка лучей в задачах построения фотореалистичных изображений // Программирование, 2011. – № 5. – С. 13 – 26.
- [6] G. Ward, "Real Pixels", Graphics Gems II, Ed. by J. Arvo, Academic Press, 1992.
- [7] <http://www.cg.tuwien.ac.at/research/theses/matkovic/node84.html>
- [8] <http://www.adobe.com/ru/products/photoshop/extend.display.Tab2.html>
- [9] <http://www.cybercom.net/~dcoffin/dcrawl/>
- [10] Барладян Б.Х., Волобой А.Г., Галактионов В.А., Копылов Э.А. Эффективный оператор сжатия динамического диапазона яркостей // Программирование, 2004. – № 5. – С. 35 – 42.

Abstract

Requirements to the programs for postprocessing and analysis of images generated in the realistic rendering systems are formulated in the paper. It is implied that realistic rendering is based on physically accurate simulation of light propagation. So the results of simulation can be interrogated by proposed program and converted into usual images. Details of the program implementation are also present.

Keywords: realistic rendering software, analysis of simulation results, HDRI, regression testing, tone mapping operators

Об авторах

Барладян Борис – к.т.н., с.н.с.,

ИПМ им. М.В. Келдыша РАН. E-mail: obb@gin.keldysh.ru

Бирюков Елисей – м.н.с.,

ИПМ им. М.В. Келдыша РАН. E-mail: peb@gin.keldysh.ru

Валиев Ильдар – н.с., ИПМ им. М.В. Келдыша РАН.

Волобой Алексей – д.ф.-м.н., в.н.с.,

ИПМ им. М.В. Келдыша РАН. E-mail: voloboy@gin.keldysh.ru

Шапиро Лев – к.ф.-м.н., с.н.с.,

ИПМ им. М.В. Келдыша РАН. E-mail: pls@gin.keldysh.ru

Графический инструментарий стоматолога-гнатолога на основе DVR

А. Белокаменская, М. Новожилов, В. Турлапов

Факультет вычислительной математики и кибернетики

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

belokamenskaya@vnmk.unn.ru

Аннотация

Разработан программный комплекс, в существенной мере разрешающий проблему несовершенства трехмерных инструментов стоматолога-гнатолога в работе с височно-нижнечелюстным суставом (ВНЧС). Инструментальная поддержка построена на использовании трехмерных исходных данных, которые предоставляет томография. Для визуальной 3D реконструкции используется метод объемного рендеринга (DVR). Для измерений используется геометрическая реконструкция изоповерхностей методом марширующих кубов. На основе указанных методов реконструкции реализован ряд инструментов, обеспечивающих двумерные и трехмерные методики измерений в практике стоматолога-гнатолога.

Ключевые слова: 3D визуализация, цефалометрия, стоматология

1. ВВЕДЕНИЕ

Патология височно-нижнечелюстного сустава (ВНЧС) затрагивает значительную часть населения, хотя и не рассматривается как проблема здравоохранения. Между тем, 3 - 7% населения пытается вылечить боль и дисфункции сустава или связанных с ним структур. Согласно данным литературы, распространенность клинических симптомов у населения разнится от 6 до 93%, вероятно, в результате использования различных клинических критериев.

В Национальном обзоре стоматологического здоровья, проведенном в Испании в 1994 г., в соответствии с критериями для эпидемиологических исследований здоровья полости рта Всемирной организации здравоохранения (ВОЗ), было выяснено, что в 12 лет 6,3% населения наблюдает щелчки в суставе, и эта цифра увеличивается до 9,4% в возрастной группе от 15 лет, достигает 14,70% в 35 - 44 лет и 23% - в 65 - 74 лет. Блокирование ВНЧС наблюдается у 2,2% населения в 12 лет, 4,5% - в интервале 35 - 44 лет и 3,5% - в возрастной группе от 65 до 74 лет. Боль, в свою очередь, наблюдается у 0,2% населения в возрасте 15 лет, 3,4% - в 35 - 44 лет и у 1,3% - у лиц в возрасте 65 - 74 лет.

В следующем исследовании, проведенном на национальном уровне в 2000 г., было видно, что 17,6% населения в возрасте 35 - 44 лет ощущает щелчки в суставе, в то время как 1,8% чувствуют боль при пальпации, и у 1,8% ограничена подвижность челюсти. Симптомы были обнаружены у 10,8% населения. В группе 65 - 74 лет щелчки присутствовали у 15,5%, боль при пальпации - у 2,5%, а также снижение подвижности - в 2,9% случаев. Симптомы присутствовали у 11,2% населения [6].

Согласно монографии 2009 г. Сёмкина и Рабухиной, анализ клинических и рентгенологических проявлений у больных с функциональной патологией ВНЧС позволяет констатировать, что дисфункции количественно преобладают среди всех заболеваний ВНЧС и встречаются более чем у 90,7% от всех пациентов с жалобами на зоны сочленений [3].

Исследования последних двух десятилетий констатируют примерно постоянный процент людей, наблюдающих симптомы дисфункции ВНЧС, что говорит о том, что пока не существует массовых эффективных методов лечения этого заболевания.

Важнейшим инструментом диагностики и оценки результатов ортодонтического лечения является цефалометрия. Цефалометрический анализ - это оценка размеров и пространственного взаимоотношения зубов, челюстей и лицевой части черепа в целом. Например, в 2002 г. 90% ортодонт в США регулярно применяли цефалометрический анализ [4]. Долгое время цефалометрический анализ проводили, используя боковые и фронтальные рентгенограммы. С распространением компьютерной томографии (КТ) и конусно-лучевой компьютерной томографии (КЛКТ или СВСТ) стало возможным проводить цефалометрический анализ, используя трехмерные данные. Благодаря системам КЛКТ стоматологи могут получить трехмерные данные с очень низкой дозой излучения, в 15 раз меньшей, чем обычные сканеры КТ. Доза облучения при КЛКТ составляет дозу в среднем 12 панорамных рентгенограмм.

Современные информационные стоматологические системы позволяют загружать, хранить и использовать различные источники данных: рентгенограммы, фотографии, панорамные снимки зубов, оцифрованные слепки. Наблюдается дрейф в сторону использования трёхмерных данных - оцифрованных слепков и данных КЛКТ в формате DICOM для проведения диагностики и ведения лечения.

На сегодняшний момент существует множество коммерческих программ для 3D визуализации в медицине, и более десятка предназначено для визуализации в стоматологии. Самый большой недостаток этих программ - отсутствие валидации результата [1]. Из всего многообразия можно выделить 6 наиболее серьёзных и развитых систем, предназначенных для ортодонт.

Модуль цефалометрии системы Dolphin Imaging создаёт двумерную картинку из трехмерного объема - латеральную, панорамную или фронтальную без перспективных искажений, либо с симуляцией перспективных искажений, как в реальной рентгеновской установке. После к изображению можно применить фильтры, улучшающие контраст. Далее можно воспользоваться предустановленными двумерными цефалометрическими анализами. Для более детального анализа ВНЧС выделяется бокс, содержащий сустав, по нему строится 7 сечений сустава.

В программе InVivo Dental фирмы Anatomage - самый полный модуль для цефалометрического анализа, как в 2D, так и в 3D. Ортодонт может сам построить новую трехмерную методику анализа и диагностики, задав собственные измерения и нормы, или воспользоваться предустановленными двумерными исследованиями. Anatomage предлагает также довольно мощный инструмент автоматической 3D идентификации скелетных точек, позволяющий в значительной мере снять проблемы 3D манипуляций на плоском экране.

OnyxSerh позволяет хранить в базе данных клинические фотографии, внутривидеосъемки, телерентгено-

граммы, фотографии стоматологических слепков и т.д. Можно воспользоваться более 50 анализами латеральной и фронтальной телерентгенограмм.

ITK_SNAP – единственная система с открытым кодом (www.itknap.org) обладает модулями визуализации и сегментации трехмерных данных. Инструментов для проведения цефалометрического анализа нет.

Компания Sirona планирует в скором времени выпустить на рынок модуль SiCAT, соединяющий снимок Galileos с оптическим слепком Ceges и запись движения нижнечелюстного сустава. Программа будет визуализировать жевательные движения и позволит заказать точную окклюзионную капу. Для ортодонт Sirona уже сейчас предоставляет возможность провести двумерный цефалометрический анализ на латеральном и заднепереднем снимках.

В системе Maxilim (<http://www.medicim.com/en/products/3d-cerphalometry>) есть модуль для проведения трехмерного цефалометрического анализа.

2. РЕАЛИЗАЦИЯ МЕТОДОВ АНАЛИЗА КЛКТ ТОМОГРАММ

Для трехмерной визуализации компьютерной томограммы использовалась техника прямого объёмного рендеринга (Direct Volume Rendering, DVR) (рис. 1).



Рис. 1. Визуализация методом DVR

Метод прямого объёмного рендеринга вычисляет интеграл объёмной визуализации [2]:

$$I = \int_0^D \text{intensity}(\mathbf{x}(t)) \times \text{extinction}(\mathbf{x}(t)) \times \exp\left(-\int_0^t \text{extinction}(\mathbf{x}(t')) dt'\right) dt$$

Луч $\mathbf{x}(t)$ параметризован расстоянием t от объекта виртуальной камеры, а излучаемая яркость $\text{intensity}(\mathbf{x})$ и коэффициент затухания $\text{extinction}(\mathbf{x})$ могут быть вычислены в любой точке \mathbf{x} пространства, используя передаточную функцию. Интеграл объёмной визуализации описывает процесс накопления цвета вдоль луча с учетом затухания по пути следования.

Через D обозначено расстояние, на которое луч проникает в объем (в большинстве случаев определяется временем выхода из ограничивающей оболочки).

Для устранения артефактов постклассификации, которые заключаются в искажениях цвета, хорошо видимых глазом в виде колец или полос, реализован метод предынтегрированного объёмного рендеринга.

Для более точной постановки диагноза и определения содержания лечения стоматологу необходимо всесторонне изучить зубочелюстную систему пациента. Сечение в произвольной плоскости (рис. 2) позволяет подробно рассмотреть зону поражения в наиболее представительном разрезе. Реализовано сечение цилиндрической поверхностью с криволинейной направляющей, которое позволяет точно отследить существенную для постановки диагноза линию исследуемой части зубочелюстной системы.

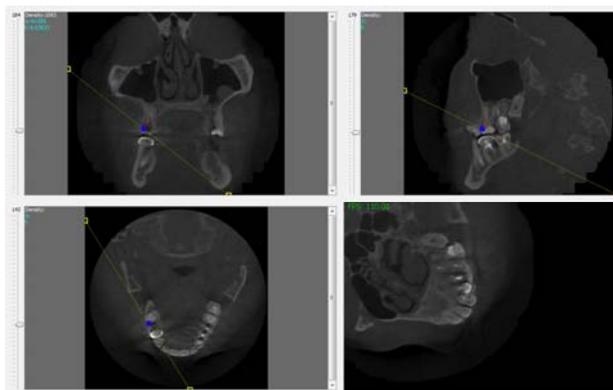


Рис. 2. Сечение томограммы произвольной плоскостью

Эмулирован рентгенологический метод исследования ортопантомографии, дающий обзорный панорамный снимок верхней и нижней челюстей (рис. 3). Благодаря такому снимку можно получить картину состояния каждого отдельного зуба, костной ткани челюстей и пародонта, составить представление о состоянии височно-нижнечелюстных суставов и гайморовых пазух.

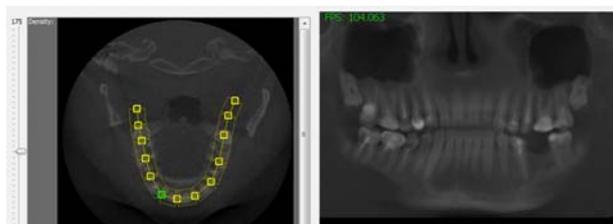


Рис. 3. Имитация панорамного снимка по данным томограммы

Программа позволяет построить панорамный снимок для различных эквидистант. Есть возможность варьировать толщину слоя, принимаемого во внимание при построении панорамного снимка.

В программе реализована методика анализа профильных телерентгенограмм, предложенная Р.А. Фаддеевым и А.В. Кузаковой в учебном пособии «Клиническая цефалометрия» [3]. Реализован режим телерентгенограммы. В этом режиме ортодонт наносит цефалометрические точки с помощью «мышь»: 16 скелетных антропометрических точек, 8 зубных, 7 кожных. После чего программа автоматически строит 10 цефалометрических плоскостей и рассчитывает 46 цефалометрических параметров (рис. 4). Вычисленные значения выводятся в сводную таблицу вместе с нормальными значениями и компенсаторными оценками. Если отклонение параметра от нормального значения не превышает норму, то результат выводится зелёным цветом, если превышает, – то красным.

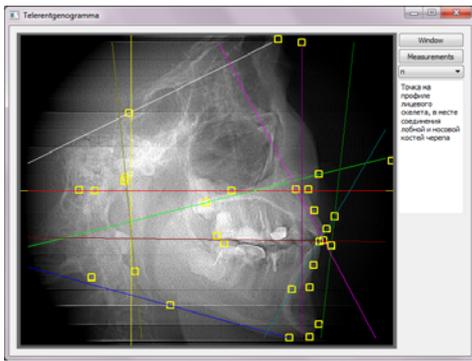


Рис. 4. Боковая двумерная цефалометрия

В качестве задела для построения инструментов трехмерной цефалометрии реализована возможность построить трёхмерные объекты: точку, линию, плоскость и отобразить их вместе с объёмными данными (рис. 5).

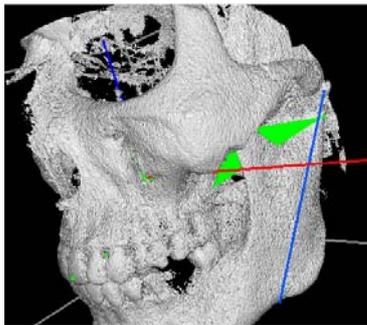


Рис. 5. Геометрия и изоповерхность кости

Одним из реализованных инструментов является геометрическая реконструкция челюстно-лицевой системы (рис. 6). Можно выделить зону интереса, построить полигональную модель и оперировать с ее отдельными элементами. Для построения полигональной модели используется метод марширующих кубов. Есть возможность проводить измерения, например, можно измерить кратчайшее расстояние в суставе (рис. 7). Программа написана в среде MS Visual Studio на языках C++ , OpenCL и GLSL, с использованием библиотеки Qt.

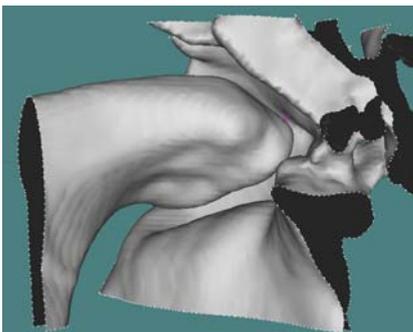


Рис. 6. Геометрическая реконструкция ВНЧС

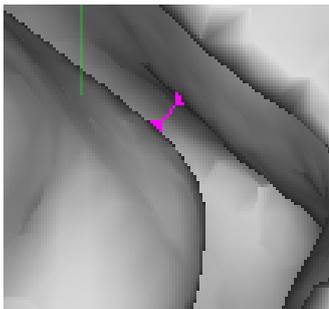


Рис. 7. Измерение зазора суставных поверхностей

3. ЗАКЛЮЧЕНИЕ

Разработан программный комплекс, в существенной мере разрешающий проблему несовершенства трехмерных инструментов стоматолога-гнатолога в работе с височно-нижнечелюстным суставом (ВНЧС). Инструментальная поддержка построена на использовании трехмерных исходных данных, которые предоставляет томография. Для визуальной 3D реконструкции используются методы объемного рендеринга (DVR). Для измерений используется геометрическая реконструкция полигональной модели изоповерхностей методом марширующих кубов. На основе указанных методов реконструкции реализован ряд инструментов, обеспечивающих двумерные и трехмерные методики измерений в практике стоматолога-гнатолога, такие, как: измерения на сечении томограммы произвольной плоскостью; инструменты для проведения двумерного цефалометрического анализа и построение таблицы компенсаторных оценок; диалоговый режим 3D редактирования и дополнения цефалометрических ориентиров; измерение расстояний между элементами челюстно-лицевой системы. Для томограмм поддерживается входной формат DICOM.

4. БЛАГОДАРНОСТИ

Работа была выполнена при поддержке программы «УМНИК».

5. ССЫЛКИ

- [1] Becker Otávio Emmel, et al. Three-dimensional Planning in Orthognathic Surgery using Cone-beam Computed Tomography and Computer Software. Computer Science Systems Biology. 6.6.2013 – P. 311 - 316.
- [2] Боголепов Д.К., Бугаев И., Белокаменская А.А., Турлапов В.Е. Полноэкранное сглаживание в реализации прерывного рендеринга для визуализации трехмерных скалярных полей на GPU Научная визуализация, 2012. -Кв.4. Т.4. - № 4. – С. 2 - 16.
- [3] Фадеев Р.А., Кузакова А.В. Клиническая цефалометрия. Учебное пособие по диагностике в ортодонтии / под ред. д.м.н. Р.А.Фадеева – СПб.: ООО "МЕДИ издательство", 2009. - 64 с.
- [4] Nijkamp Peter G, at al. The influence of cephalometrics on orthodontic treatment planning. The European Journal of Orthodontics. 3.11.2008. – 6 p.
- [5] Рабухина Н.А. и Сёмкин В.А. Дисфункция височно-нижнечелюстных суставов (клиника, диагностика и лечение). – М.: ЗАО Редакция журнала «Новое в стоматологии», 2000. – 56 с.
- [6] Roda Rafael Poveda, at al. Review of temporomandibular joint pathology. Part I: Classification, epidemiology and risk factors. 2007, Med Oral Patol Oral Cir Bucal – P. 292 - 298.

Об авторах

Белокаменская Александра – аспирант ВМК ННГУ им. Н.И. Лобачевского, belokamenskaya@vmk.unn.ru
 Новожилов Михаил – студент ВМК ННГУ им. Н.И. Лобачевского, michael.novogilov@gmail.com
 Турлапов Вадим – профессор каф. МО ЭВМ ВМК ННГУ им. Н.И. Лобачевского, vadim.turlapov@cs.vmk.unn.ru

РАСПОЗНАВАНИЕ СИМВОЛОВ НА БАЗЕ НИЗКОУРОВНЕВЫХ МОДЕЛЕЙ ОПИСАНИЯ ГРАФИЧЕСКИХ ИЗОБРАЖЕНИЙ

Д. Васин¹, М. Ершов¹

¹ НИИ Прикладной математики и кибернетики
Нижегородского государственного университета им. Н.И. Лобачевского

Аннотация

В докладе рассмотрен алгоритм формирования эффективных признаков и решающих правил распознавания символов на низкоуровневой структурированной модели описания бинарного растрового изображения

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 произвольной ориентации по отношению к горизонту, на низкоуровневой структурированной модели описания бинарного растрового изображения (БРИ) [1].

Низкоуровневая структурированная модель описания БРИ.

Непроизводным элементом модели является штрих. Формально штрих – одномерный кластер связанных пикселей заданного цвета, т.е. представляется набором чисел $S=(t, N, K, C)$, где t – номер строки растра; N, K – координата начала и конца совокупности подряд идущих пикселей одного цвета вдоль линии растра; C – код цвета связанных пикселей, образующих штрих.

Для бинарных изображений штрих описывается тройкой типа $S=(t, N, K)$, где t – номер строки растра; N, K – координаты начала и конца штриха.

Совокупность штрихов для заданной строки растра t определяет строку штрихов $SH=\{S_i\}(i=1,2,\dots,m)$, где S_i – штрих.

Описание растра с помощью набора строк штрихов является однозначным. Отметим, что штриховой формат представления и хранения растровых данных является в 10 – 20 раз более компактным, по сравнению с пиксельным.

Топологические свойства штриха $S=(t, N, K)$ определяются через его связность со штрихами смежных строк [1]. Два штриха S_t и S_{t-1} смежных строк t и $t-1$ связаны, если $(N_{t-1} \geq N_t) \wedge (N_{t-1} \leq K_t) \vee (K_{t-1} \geq N_t) \wedge (K_{t-1} \leq K_t)$. Связный штрих – это набор $H=\{S, sw_p, sw_s\}$, sw_p, sw_s – количество связанных штрихов со штрихами смежных строк, а t – номер текущей строки штрихов растра. Тогда $sw_p=0$, если в предыдущей строке $p=t-1$ нет ни одного штриха, для которого выполняется условие $(N_p \geq N_t) \wedge (N_p \leq K_t) \vee (K_p \geq N_t) \wedge (K_p \leq K_t)$,
(1)

иначе, $sw_p = \mu_p$, μ_p – кратность выполнения условия (1), т.е. количество связанных штрихов предыдущей строки.

Аналогично, $sw_s=0$, если в следующей строке $s=t+1$ нет ни одного штриха, для которого выполняется условие: $N_s \geq N_t \wedge N_s \leq K_t \vee K_s \geq N_t \wedge K_s \leq K_t$,
(2)

иначе, $sw_s = \mu_s$, μ_s – кратность выполнения условия (2), т.е. количество связанных штрихов последующей строки.

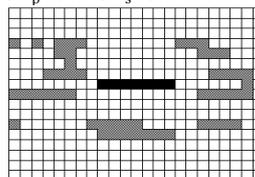
Здесь N, K – координаты начала и конца исследуемого на связность штриха, N_t, K_t – координаты начала и конца текущего штриха предыдущей смежной строки штрихов,

N_s, K_s – координаты начала и конца текущего штриха следующей смежной строки штрихов.

По суперпозиции значений sw_p и sw_s можно ввести следующую классификацию графических ситуаций. В текущей строке:

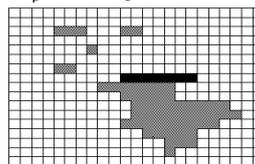
Штрих изолированный (ШИ):

$$sw_p=0 \wedge sw_s=0.$$



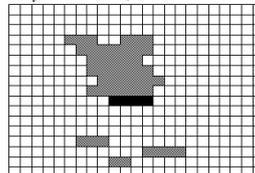
Штрих начала растрового объекта (ШН):

$$sw_p=0 \wedge sw_s=1$$



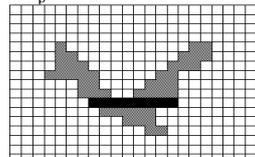
Штрих конца растрового объекта (ШК):

$$sw_p=1 \wedge sw_s=0$$

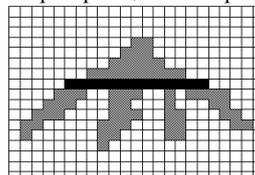


Штрих слияния растровых объектов (ШС):

$$sw_p > 1$$

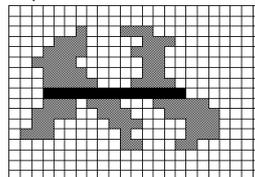


Штрих расщепления растровых объектов (Шр): $sw_s > 1$



Штрих слияния и расщепления растровых объектов (Шср):

$$sw_p > 1 \wedge sw_s > 1.$$



Растровый простой объект РПО – кластер связанных штрихов, не содержащий графических ситуаций типа слияния и расщепления, то есть для любого штриха кластера выполняется условие

$$sw_p < 2 \wedge sw_s < 2. \quad (3)$$

Растровый составной объект РСО – кластер связанных штрихов, для каждого штриха которого выполняется условие

$$sw_p > 0 \vee sw_s > 0, \quad (4)$$

и для любых двух элементов (штрихов или пикселей) найдется хотя бы одна соединяющая их пиксельная траектория, состоящая из связанных смежных элементов.

Предложенная иерархия моделей описания РИГД на нижних уровнях позволила перейти к структурированному описанию РИГД. При этом РИ представляется совокупностью связанных растровых компонент (СРК) – РПО, РСО, узловых штрихов (Шр, Шс) и штрихов начала и конца объекта (Шн, Шк).

Алгоритм построения эффективных признаков и решающих правил классификации символов

В качестве рассматриваемых символов были выбраны заглавные буквы русского алфавита А – Я, цифры 0 – 9 и знаки препинания – точка, запятая, прямой и обратный слеш.

Примеры обучающих символов:

АБВГДЕЁЖЗИЙКЛ
МНОПРСТУФХЦЧ
ШЩЪЫЬЭЮЯ
1234567890,./\

Для каждого символа алфавита была сформирована матрица связанных штрихов и фиксировались следующие признаки:

1. Число внутренних областей (ВО).
2. Отношение аспекта (отношение ширины к высоте) (Е).
3. РПО, для которых вычисляются:
 - количество штрихов, входящих в РПО;
 - средняя длина штриха.
4. Процент заполненности – соотношение пиксельной площади символа к площади описывающего прямоугольника.
5. Число “вертикальных” РПО (ВРПО). РПО признается “вертикальным”, если средняя длина штриха более 70% высоты символа. Уровень 70% высоты символа как признак вертикальности выбран на основе эмпирических наблюдений.
6. Особые штрихи (Шн, Шк, Шр, Шс).

Приведем фрагмент автоматически сформированной таблицы значений вышеуказанных признаков (см. таблицу).

Символ	ВО	Е	Число ВРПО	% заполн.	Число Шс/Шр
7	0	0.52-0.65	0	21.5-26.6	1/0
3	0	0.59-0.64	0		2/1
Д	1	0.71-0.89	1		2/2
Ю	1	1.35-1.45	1		2/2
Ф	2	1-1.12	1		2/2
В	2	0.8-0.92	1		3/2

Алгоритм подсчета числа внутренних областей

Пусть имеется изображение с объектом, представленным единственной связной компонентой. В процессе работы на изображении находятся “особые” точки, хранящиеся в списке S.

1. Просматривая штриховой файл, находим штрихи расщепления.
2. Пусть штрих расщепления находится в i -столбце. Просматриваем все связанные с ним штрихи в столбце ($i+1$), для каждого $H=(t, N, K)$ добавляем в список S особые точки ($i+1, N-1$), ($i+1, K+1$).
3. Просматриваем все особые точки (x, y) из списка.
 - 3.1. Если координаты точки выходят за рамки матрицы P, то удаляем её из списка как несуществующую
 - 3.2. Запускаем поиск в ширину со стартом в (x, y) . При этом *цвет 1* – белый, *цвет 2* – любой дополнительный цвет (отличный от черного и белого).
 - 3.3. Проверяем все особые точки из S, следующие по списку за точкой (x, y) . Если они окрашены в *цвет 2*, то удаляем их как лишние.
 - 3.4. Просматриваем граничные точки всего изображения $[P_{ij}], [P_{Nj}], [P_{i1}], [P_{iM}]$ ($i = 1, \dots, N; j = 1, \dots, M$). Если хотя бы одна из них будет закрашена в *цвет 2*, то значит точка (x, y) не является точкой, принадлежащей внутренней области объекта (замкнутая область белого цвета). Удаляем её из списка S.
4. Количество элементов в списке S равно количеству внутренних областей в объекте.

Алгоритм автоматического распознавания символов

Разработан на основе иерархической модели, в узлах которой производится вычисление указанных признаков.

Приведем примеры решающих правил для некоторых символов шрифта «Times New Roman». В записи решающих правил выражение вида $\text{Ш}i = \xi$ означает, что рассматриваемый символ содержит ξ штрихов типа Шi $\in \{\text{Шн}, \text{Шк}, \text{Шр}, \text{Шс}\}$.

- 1) $\text{ВО} = 2$, это “В” ∨ “Ф”. Если $E < 0.95$, то это “В”, иначе “Ф”.
- 2) $\text{ВО} = 1 \wedge E \in [0.0-0.66]$, это “0” ∨ “4” ∨ “6” ∨ “9”: если $\text{Шс} = 1 \wedge \text{Шр} = 2$ – это “6”, если $\text{Шс} = 2, \wedge \text{Шр} = 1$ – это “9”. Пусть площадь внутренней области S_0 , а площадь описанного вокруг буквы прямоугольника S_n . Если $(S_0/S_n) < 0.2$, то это “4”, иначе “0”.
- 3) Если $\text{ВО} = 1 \wedge E \in [0.71-0.89] \wedge \text{Шс}=2 \wedge \text{Шр}=1$, это “Б” ∨ “Р” ∨ “Ъ”. Для их распознавания поворачиваем символ на 90 градусов. Далее: если $\text{Шс} = 2 \wedge \text{Шр} = 1$, это “Б”; если $\text{Шс} = 1 \wedge \text{Шр} = 1$, это “Р” ∨ “Ъ”. Если Шс находится в первой трети повернутого символа, то это “Ъ”, если в последней трети – то это “Р”.
- 4) Если $\text{ВО} = 0 \wedge E \in [0.65-0.87] \wedge \text{ВРПО}=1$, это “Г” ∨ “Е” ∨ “Т” ∨ “Ч”: если ВРПО находится во второй половине символа, то это “Ч”, если в первой половине символа, то поворачиваем символ на 90 градусов, и если $(\text{Шс}+\text{Шр})=1$, это “Г”, если $(\text{Шс}+\text{Шр})>1$, это “Е”, если ВРПО расположен в середине символа, то это “Т”.

В ходе анализа работы распознавания с помощью введенных выше решающих правил было отмечено следующее:

- исходные эталоны были расположены горизонтально с минимальным углом отклонения от горизонта, имели обычную ориентацию (не были повернуты на 90 или 270 градусов, не были перевернуты на 180 градусов);

- при малом угле наклона текста по отношению к горизонтальной линии (до 2°) значения признаков (особые штрихи, вертикальные ПРО, аспект и т.д.) не претерпевают значительных изменений. Измеряемые значения находятся в допустимых рамках, принимаемых решающими правилами;

- при большем наклоне текста (ближе к 5° и более) качество распознавания резко падает. Наиболее чувствительными к повороту являются признаки количества вертикальных ПРО, количества штрихов слияния/расщепления, взаимное расположение особых штрихов, отношение аспекта. Наиболее устойчивым, ожидаемо, является число внутренних областей у символа;
- при наклоне даже эталонных символов на $\pm 5^\circ$ правильно распознано было около трети эталонов.

В связи с этим возникает необходимость корректировки наклона текста.

Определим основные требования к алгоритму определения угла наклона текста к горизонту:

- быстродействие (время работы алгоритма не должно зависеть от контента изображения);
- точность (малая величина систематической ошибки);
- корректная работа с полутоновыми и, в большинстве случаев, с цветными графическими документами;
- угол наклона может быть определён, как локально – для некоторой части изображения, так и для всего документа в целом;
- вместе с углом наклона, метод должен выдавать степень доверия к своему ответу, либо оценивать погрешности измерения.

В обзоре [2] предлагается несколько различных методов определения угла наклона документа. Разработанный алгоритм комбинирует некоторые из них и адаптирован к штриховому формату, что уменьшает количество необходимых вычислений. Для оценки угла наклона текста используем вертикальные проекционные гистограммы [3], представляющие собой одномерный массив размерности, равной числу строк растра N . Каждая ячейка гистограммы хранит количество чёрных пикселей в соответствующей строке изображения.

Пусть имеется бинарное растровое изображение P , размером $N * M$ пикселей.

$$P[x, y] = \begin{cases} 1, & \text{если пиксель } (x, y) - \text{чёрный} \\ 0, & \text{если пиксель } (x, y) - \text{белый} \end{cases}$$

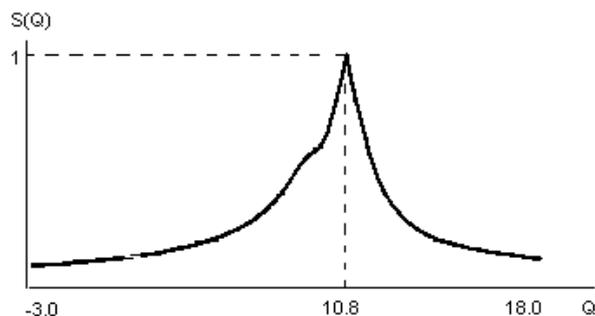
Пусть $S(i, Q)$ – суммарное количество чёрных пикселей в строке растра с номером i при повороте исходного

$$\text{изображения на угол } Q: S(i, Q) = \sum_{j=0}^{j=M(Q)} P(i, j).$$

Вычисляем суммарную функцию угла наклона

$$Q: S(Q) = \sum_{i=0}^{i=N(Q)} S^2(i, Q)$$

Пример нормированной по максимальному значению функции $S(Q)$, полученной на реальном графическом документе, повернутом на угол около 10° .



На базе разработанных алгоритмов создано оригинальное программное обеспечение.

В результате практической апробации созданного программного обеспечения было выявлено, что при горизонтальном расположении текста правильно распознается 96 – 98% символов, однако по мере увеличения угла поворота снижается до 90 – 93%.

Работа выполнена при финансовой поддержке РФФИ – проекты №13-07-00521 и № 13-07-12211

ССЫЛКИ

1. Васин Д.Ю., Васин Ю.Г., Громов В.П. Структурное описание растровых данных // Методы и средства обработки сложной графической информации: 6 Всероссийская конференция с участием стран СНГ, 25 – 27 сентября 2001 г. – Н. Новгород – С. 21 – 23.
2. Hull J. Document Image Skew Detection: Survey and Annotated Bibliography, Document Analysis Systems II, World Scientific, 1998. – P. 40 – 64.
3. Dan S. Bloomberg, Gary E. Kopec and Lakshmi Dasari. Measuring document image skew and orientation. // Xerox Palo Alto Research Center. – Режим доступа: <http://www.leptonica.com/papers/skew-measurement.pdf>, свободный.

Визуализация результатов историко-теоретических исследований архитектурного наследия Северного Кавказа

О. Иевлева, Т. Карпюк, А. Кошевой
Академия архитектуры и искусств
Южный федеральный университет
ijevleva@sfnedu.ru

Аннотация

В докладе рассматривается один из возможных вариантов визуализации результатов историко-архитектурных исследований на примере архитектурного наследия Северного Кавказа

Ключевые слова: Визуализация, историко-теоретические исследования, архитектурное наследие Северного Кавказа

1. ВВЕДЕНИЕ

Историко-теоретические исследования архитектурного наследия представляют собой объемную, кропотливую работу, связанную с архивными исследованиями, работой с литературой, осуществлением фотофиксаций, обмеров памятников, составлением кроков и выполнением по ним чертежей, выявлением объектов-аналогов, графо-аналитическим анализом, а часто и виртуальной реконструкцией памятников архитектуры и культуры. Большая часть этих работ связана с графическим 2D и 3D представлением процесса исследования и его результата.

Традиционно, если исследуется один памятник, то результаты работы, как правило, представляются в виде текста и последовательности графических изображений. Если исследуется несколько памятников, то осуществляется их классификация, выявляются их классификационные признаки, проводится не только их графо-аналитический анализ, но и визуальное сравнение, выявление общих и индивидуальных архитектурно-планировочных, пространственных, конструктивных характеристик, декоративных элементов и пр. Здесь в качестве результатов исследований, обычно, показывают не только графические изображения отдельных памятников, но и требуют совместного представления однотипных видов (планов, фасадов, объемных изображений) различных объектов.

Если же исследуются памятники культурного наследия определенного временного периода целого региона, то в качестве результатов исследования могут потребоваться не только фотографии и рисунки объектов в процессе их жизненного цикла, чертежи ситуационных планов местности, планов и разрезов, конструкций и деталей объектов, изображения, позволяющие демонстрировать сравнение объектов по определенным классификационным признакам, видео фиксации, а иногда 3D модели виртуальной реконструкции и чертежи проекта реконструкции, но и карты регионов размещения памятников, как в настоящее время, так и в более ранние временные периоды. Визуализация результатов такого сложного и объемного научного исследования, связанного с архитектурным наследием Северного Кавказа X – XVII вв., представлена в настоящей работе в виде информационно-поисковой системы, содержащей графическую и текстовую информацию о памятниках и позволяющей осуществлять поиск различных результатов исследования по целому ряду поисковых признаков.

2. СТРУКТУРА СИСТЕМЫ

Разработанная система состоит из управляющей программы, базы графических и текстовых данных, блока заполнения и блока просмотра информации (рис. 1).

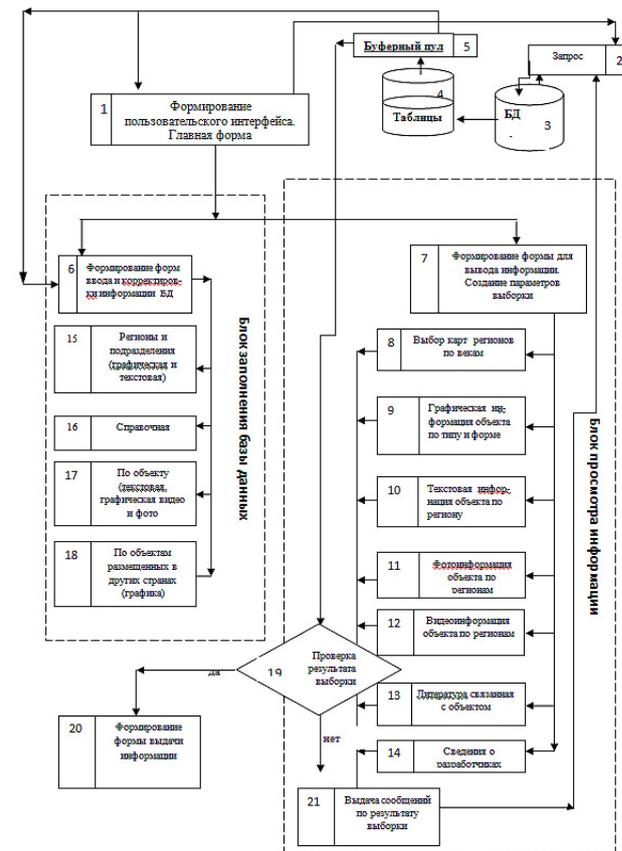


Рис. 1. Укрупненная блок-схема системы

На схеме управляющей программе соответствуют блоки 1...5. Здесь осуществляется формирование пользовательского интерфейса, создание главной экранной формы, меню для формирования запросов, устанавливается связь компонентов управления с базой данных и таблицами, осуществляется анализ соединения, проверка на наличие информации в БД, анализируется вид запроса и выполняется формирование информации из БД для создания форм блока заполнения БД или блока просмотра информации.

Заполнение и корректировка базы данных результатами исследований осуществляется в блоках 6, 15...18.

В связи с тем, что исследованиями охватывается довольно большая территория, в состав которой входят несколько областей и республик, то для идентификации места расположения памятника используется блок 15, в котором содержатся сведения о регионах и подразделениях, позволяющие получить необходимую карту. Под

регионами в настоящей работе понимается республика, область или край, а под подразделением – поселение или город, в котором находится рассматриваемый памятник. Блок 16 содержит список «направлений Зодчества» Северного Кавказа. В результате исследования все рассмотренные объекты культурного наследия были разделены на три типа «направлений Зодчества»: Христианское, Исламское, Гражданское. В соответствии с проведенными историко-архитектурными изысканиями объекты каждого направления могут подразделяться по пространственно-временным характеристикам, т.е. по временному историческому периоду и месту расположения на древних картах (Древних названий стран расположения). Для этого предусмотрен справочник исторических видов памятников, сложившихся в разных регионах, например, Христианские церкви Двалетии XI в. или Храмы Дигории XIII в. Блок 17 содержит таблицы сведений об отдельных памятниках. Эти таблицы содержат информацию о названии объекта, имеют поля для подгрузки текстовых описаний, графических изображений, фотоизображений местности, где располагается объект, видео изображений, планов объектов. Блок 18 содержит сведения об объектах, расположенных в других регионах и странах, имеющих аналогичные памятники, и с которыми производилось сравнение в процессе исследования. Эти таблицы содержат информацию о названии объекта, изображении его плана, фотофиксацию и описание объекта.

Для создания запроса на поиск памятника создается специальная форма (блок 7), которая содержит следующие данные: а) список регионов; б) карту региона с координатами всех населенных пунктов (подразделений) региона (карта региона загружается в графическом поле при выборе региона из списка регионов); в) список исторических периодов развития региона.

Поиск информации, выборка ее в соответствии с запросом и просмотр информации осуществляется в блоках 8...21.

Для анализа географического развития региона предусмотрен блок 8, который позволяет рассмотреть исторические карты различных временных периодов. Выбор карт осуществляется в соответствии с периодом развития региона, который пользователь выбирает в блоке 7.

При выборе региона, его подразделений (населенных пунктов) и периода развития региона в графическое поле экранной формы загружаются изображения по всем найденным объектам (блок 9), а после выбора памятника – подгружается текстовая информация (блок 10).

Если число выбранных объектов больше одного, то формируется список объектов и графическое меню с изображением каждого из них.

Для каждого объекта, в зависимости от географического расположения и периода развития региона, предусмотрен просмотр фотоальбома с изображением местности, где расположен объект, его фотоизображения (блок 11) и видео изображения (блок 12), а также приводится список литературы, связанной с этим объектом (блок 13).

Предложенная блок-схема реализована на языке Borland Builder C++; для организации символьной информации БД использовано СУБД MS Access; для организации фото и видео изображений в БД использована стандартная СУБД Database Desktop, встроенная в Borland Builder C++.

3. АЛГОРИТМ РАБОТЫ СИСТЕМЫ

Работа в системе визуализации результатов историко-теоретических исследований объектов культурного наследия Северного Кавказа [1, 2] начинается с заглавной страницы, позволяющей выбрать режим работы (просмотр или занесение информации в базу данных) или осуществить выход из системы. Первая страница системы в режиме «Просмотр» позволяет начать формирование запроса для поиска информации о памятнике (рис. 2).

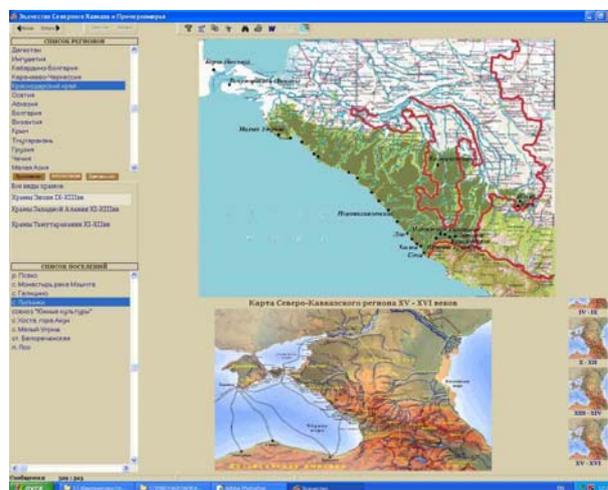


Рис. 2. Составление запроса для поиска памятника

На этой странице в «Списке регионов» можно выбрать республику, область или край, в котором находится памятник. В результате в верхней части графического поля страницы появится современная карта региона. Для уточнения поиска можно указать в списке «Все виды храмов» исторический период создания памятника. После чего в поле «Список поселений» появится перечень населенных пунктов этого региона, в которых находятся памятники. Для получения списка памятников конкретного населенного пункта можно выбрать его в «Списке поселений» или щелкнуть на изображении современной карты, после чего последняя появится в новом окне в увеличенном масштабе, где можно указать курсором место расположения памятника (рис. 3)

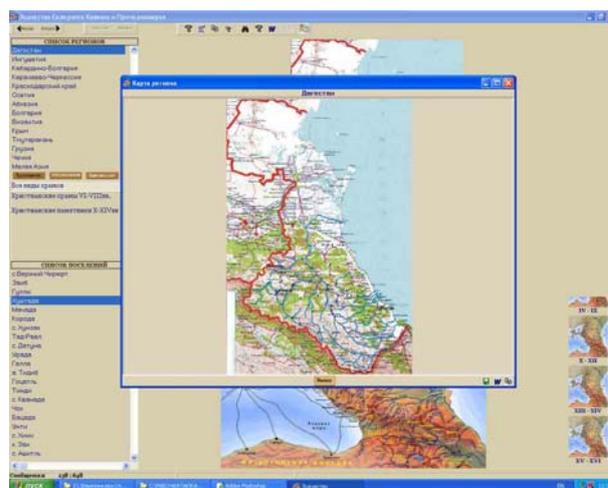


Рис. 3. Графический выбор объекта на карте

Можно также в нижней части графического поля выбрать временной период существования памятника и посмотреть древнюю карту этого периода.

Для получения сведений о конкретном объекте с левой стороны экрана содержится список памятников выбранного региона и населенного пункта. Для выбранного памятника в текстовом поле окна содержатся текстовые сведения о месте расположения объекта, а в нижнем графическом поле – наиболее характерные его изображения. Причем, в левой части графического поля находится графическое меню, из которого можно выбрать изображение для просмотра (рис. 4).



Рис. 4. Информация об основных характеристиках объекта

На отдельной странице можно просмотреть фотоизображения памятника с различных точек обзора и в различных временные периоды. Для этого предусмотрена кнопка «Фото», расположенная в верхней части графического экрана. Все имеющиеся фотографии, как экстерьерера, так и интерьера приведены в графическом меню внизу экрана, и выбранное изображение появляется в увеличенном масштабе (рис. 5).

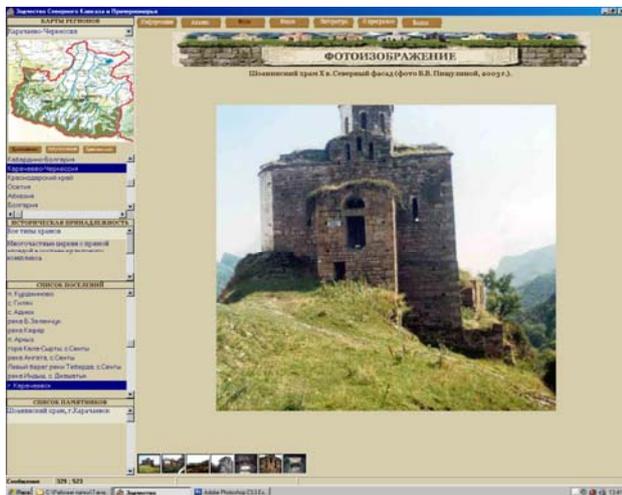


Рис. 5. Просмотр фотофиксаций

Для просмотра видео клипов, а также анимированных моделей памятников предусмотрена кнопка «Видео». На странице появится кадр из первого видео клипа и под ним его название. Список всех имеющихся клипов от конкретного объекта размещается под названием выбранного клипа. Для управления видеоизображением с правой стороны от названия клипа располагается панель управления просмотром файла (рис. 6).

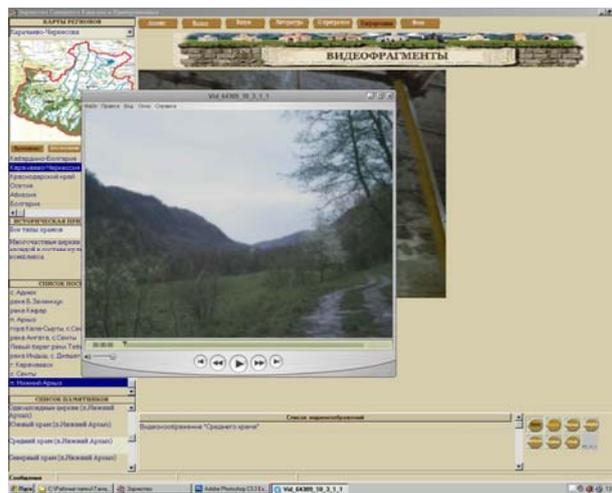


Рис. 6. Просмотр видео фиксаций

В случае если в результате исследования осуществлялась виртуальная реконструкция, представленная в виде анимированной 3D модели, то она представляется также с помощью кнопки «Видео» (рис. 7).

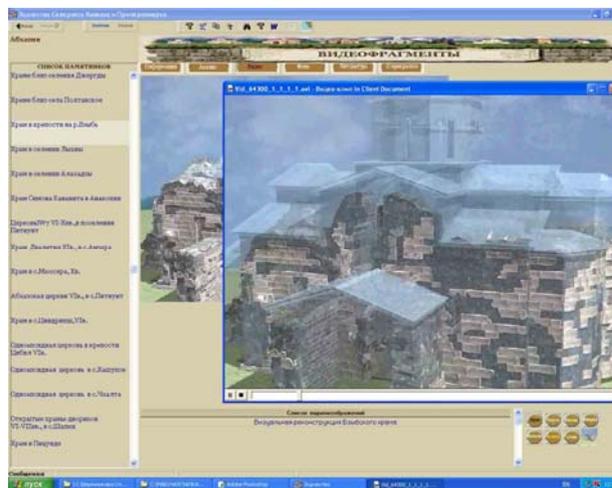


Рис. 7. Представление виртуальной реконструкции

4. ЗАКЛЮЧЕНИЕ

В представленном докладе предложен один из возможных вариантов визуального представления результатов масштабных историко-теоретических исследований архитектурного наследия региона на примере памятников архитектуры Северного Кавказа X – XVII вв.

Предложенная система позволяет занести в базу данных результаты проведенных исследований и осуществить их визуализацию по определенным поисковым признакам в соответствии с полученными выводами. Система состоит из управляющей программы, базы данных, блока занесения информации и блока визуализации. Разработанная система может быть предложена в качестве программной оболочки для визуализации аналогичных исследований

5. БЛАГОДАРНОСТИ

Работа была выполнена при частичной поддержке гранта Российского научного фонда №14-18-00595.

6. ССЫЛКИ

- [1] Иевлева О.Т. Особенности использования средств и методов информационных технологий для сохранения сведений о культурном наследии //Архитектура и искусство в контексте культуры: сб. материалов международной научно-практической конференции. – Ростов-на-Дону: ААИ ЮФУ, 2014. – С. 88 - 90.
- [2] Иевлева О.Т., Карпюк Т.А., Кошевой А.И. Концептуальная модель информационно-поисковой системы «Средневековая архитектура Северного Кавказа» // «Современные информационные технологии: тенденции и перспективы развития»: Материалы XX конференции. Южный федеральный университет, Ростов-на-Дону 24 -26 апреля 2013 г. – Ростов-на-Дону: Изд-во УФУ.– С. 184- 186. ISBN 978-5-9275-1094-8
- [3] Пищулина В.В. и др. Гражданское зодчество Северного Кавказа X – XVII вв. – Ростов-на-Дону, ЮФУ-РГСУ, 2014. - 388 с.
- [4] Пищулина В.В. Христианское храмовое зодчество Северного Кавказа VI – XVI вв. – Ростов-на-Дону: ЮФУ, 2012. – 375 с.

Об авторах

Иевлева Ольга – заведующая кафедрой графики и информационных технологий архитектурного проектирования Южного федерального университета.
E-mail: ijevleva@sfedu.ru.

Карпюк Тамара – ведущий программист кафедры графики и информационных технологий архитектурного проектирования Южного федерального университета.
E-mail: karpuk@raai.sfedu.ru.

Кошевой Алексей – старший преподаватель кафедры графики и информационных технологий архитектурного проектирования Южного федерального университета.
E-mail: ox.oxo@ya.ru

Комбинированный метод коррекции последствий неравномерности чувствительности датчиков спутниковой съемочной системы сканового типа

В. Гусев

Московский авиационный институт (НИУ), Москва, Россия
gusev3@mail.ru

Аннотация

Радиометрическая коррекция первичных спутниковых изображений является весьма актуальной задачей. К этому классу относится коррекция вертикальных полос на спутниковых изображениях, возникающих ввиду деградации характеристик датчиков. Существующие методы выравнивания яркости по столбцам изображения не идеальны, и не всегда способны учитывать сильные естественные изменения яркости на изображении из-за рельефа местности или облачности. Предлагается комбинированный метод коррекции, который сводится к корректровке перепада яркости по двумерной гистограмме яркостей пар пикселей между столбцами, либо к анализу особенностей изображения с помощью рассмотрения локальных статистических характеристик разностей столбцов изображения. Подобный анализ позволил определять одноименные гладкие участки, по которым можно точнее определить искусственный скачок яркости.

Ключевые слова: обработка спутниковых изображений, радиометрическая коррекция, яркостные искажения, структурные искажения, космические изображения, удаление вертикальных полос.

1. ВВЕДЕНИЕ

Особенностью рассматриваемой спутниковой съемки является ее скановый принцип. Для регистрации изображения используется линейка из ПЗС матриц, расположенных в два ряда с перекрытиями. Из-за рассогласования чувствительности датчиков появляются вертикальные полосы различной ширины на изображениях.

Коррекцию подобных искажений можно выполнять с помощью двух технологий: по данным бортовой калибровки датчиков и путем статистического анализа изображения, искаженного структурным шумом [1]. Однако технология коррекции изображений по данным бортовой калибровки требует периодического, довольно частого проведения сеансов калибровки, так как характеристики датчиков могут значительно изменяться за несколько дней [1]. Также не всегда бывают доступны данные бортовой калибровки. Поэтому требуются надежные статистические методы. Сложность создания таких методов заключается в том, что при определении искусственного перепада яркости между столбцами изображения могут оказывать большее влияние естественные перепады, вызванные изменениями в характере местности (например, при появлении облаков). Требуется определенный способ селекции и обработки данных о яркостях пикселей для правильного выравнивания вертикальных полос.

2. ПОСТАНОВКА ЗАДАЧИ

Модель искажений представляется в виде

$$g(x,y) = a(x) \cdot f(x,y) + b(x), \quad (1)$$

где $f(x,y)$ – изображение без помех высотой h и шириной w пикселей; $g(x,y)$ – изображение с искажениями; $a(x)$, $b(x)$ – коэффициенты, зависящие от номера столбца. Задачей является получение приближенного неискаженного изображения. На изображении искажения представляются в виде вертикальных полос по всей высоте. На рис. 1 приводится пример изображения, полученного спутником Ресурс-ДК, с искажениями яркости.

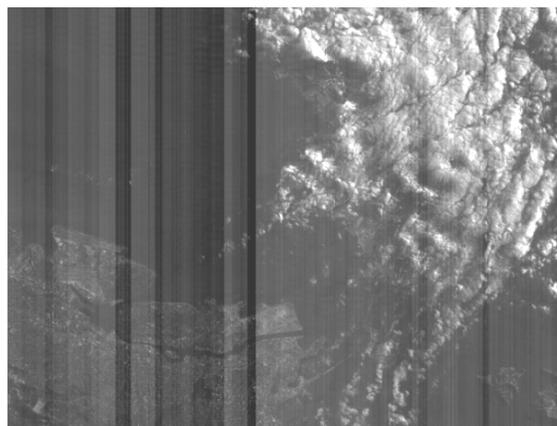


Рис. 1. Исходное изображение с искажениями

3. МЕТОД КОРРЕКЦИИ

Используется известное предположение [2], что разность яркостей соседних пикселей описывается с помощью функции плотности вероятности

$$p_{LG}(v; \theta) = Ap_L(v; s) + (1-A)p_G(v; \sigma^2), \quad (2)$$

где вектор параметров $\theta = [s; \sigma^2; A]$; v – разность яркостей, а p_L и p_G – следующие функции распределения вероятностей Лапласа и Гаусса:

$$p_L(v; s) = \frac{1}{2s} \exp\left(-\frac{|v|}{s}\right), \quad p_G(v; \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{v^2}{2\sigma^2}\right). \quad (3)$$

В случае искажений по формуле (1), при рассмотрении разностей пар пикселей соседних столбцов с фиксированным значением яркости пикселей одного из столбцов, разности будут описываться по подобному закону со смещением математического ожидания (МО). Тогда, отдельно для каждого уровня яркости пикселей первого столбца по максимуму гистограммы можно определить соответствующий уровень яркости во втором столбце.

Обработка ведется по столбцам. В каждый момент времени рассматриваются два очередных соседних столбца. Обозначим как p_1 и p_2 два соседних столбца пикселей эталонного изображения $f(x,y)$, а \bar{p}_1 и \bar{p}_2 – соответствующие столбцы из $g(x,y)$. Будем считать, что пиксели \bar{p}_1 имеют эталонный уровень яркости, а пиксели из \bar{p}_2 искажены $\bar{p}_2 = a_2 p_2 + b_2$.

Далее строится двумерная гистограмма H_{12} яркостей для \bar{p}_2 и \bar{p}_1 . Значение H_{12} в некоторой позиции (v_1, v_2) соответствует числу соседних пар пикселей, для которых яркость первого пикселя v_1 , второго – v_2 .

Для уровней яркости $q = \overline{0, V_{max} - 1}$ пикселей первого столбца, где V_{max} – число градаций яркости, находится соответствующий уровень яркости пикселей второго столбца $m(q) = \operatorname{argmax}_j [H_{12}(q, j)]$. Данный процесс соотносится с нахождением пика распределения и оценки МО, рассмотренного ранее распределения.

Задача – определить оценки коэффициентов \hat{a}_2 и \hat{b}_2 линейного преобразования яркостей. Имея массив найденных соответствий яркостей q и $m(q)$, применяя модель линейной регрессии и метод наименьших квадратов, находим данные коэффициенты.

Практическая реализация требует учета различных факторов, вводится ряд дополнительных условий. Так, например, вследствие обработки по столбцам, когда все предыдущие коэффициенты учитываются в общей формуле

$$a_{общ.} := a_2 \cdot a_{общ.}, b_{общ.} := a_2 \cdot b_{общ.} + b_2, \quad (4)$$

то для предотвращения накопления ошибок важно надежное вычисление коэффициентов. Для этого производится фильтрация исходных пар пикселей и вводится ряд ограничений: на допустимые значения коэффициентов a_2 и b_2 , на минимальные учитываемые значения в гистограмме H_{12} , на минимальную ширину диапазона значений при использовании линейной регрессии и т.д. Другим способом снижения вероятности ошибок является уменьшение количества раз использования формулы (4). Если величины \hat{a}_2 и \hat{b}_2 такие, что $\hat{a}_2 p_2 + \hat{b}_2$ практически не отличается от p_2 , то можно оставить коэффициенты $a_{общ.}$ и $b_{общ.}$ без изменений для текущего столбца. Так, например, для $|\hat{b}_2|$ можно задать порог, равный 1, только после которого будут учитываться значения корректировок. Полосы с перепадом менее чем в 1 значение яркости остаются практически незаметны глазу.

Если из-за ограничений вычисление корректировки невозможно, то применяется альтернативный метод коррекции.

Суммарно алгоритм с псевдокодом:

1. Сглаживание $g(x, y)$ вертикальной усредняющей маской.
2. Устанавливаются пороги: `porog_sko`, `porog_num`, `porog_range`, `porog_a`, `porog_b`.
3. $a_{общ.} = 1$, $b_{общ.} = 0$.
4. For $j=2$ to w .
5. Получаем два соседних столбца \bar{p}_1 и \bar{p}_2 , соответствующих позициям $j-1$ и j .
6. Вычисляем МО и СКО (среднеквадратическое отклонение) для разностей пикселей \bar{p}_2 и \bar{p}_1 .
7. Инициализируем массив двумерной гистограммы H_{12} и массивы m_1 и m_2 для линейной регрессии, $k=0$;
8. For $i=1$ to h .

9. if $(\bar{p}_1(i) > 0$ and $\bar{p}_1(i) < V_{max} - 1$ and $\bar{p}_2(i) > 0$ and $\bar{p}_2(i) < V_{max} - 1$ and $|\bar{p}_2 - \bar{p}_1 - MO| < CKO \cdot porog_sko$).
10. $H_{12}(\bar{p}_1(i), \bar{p}_2(i)) = H_{12}(\bar{p}_1(i), \bar{p}_2(i)) + 1$.
11. For $i=0$ to $V_{max} - 1$.
12. $v = \operatorname{argmax}_q [H_{12}(i, q)]$, $num = H_{12}(i, v)$.
13. if $(num > porog_num)$ $k=k+1$, $m_1(k)=i$, $m_2(k)=v_{2m}$ End.
14. End.
15. if (массивы m_1 и m_2 не пустые).
16. if $(\max(m_1) - \min(m_1) > porog_range)$ // (для надежного вычисления диапазон не должен быть узким).
17. По МНК для модели линейной регрессии по массивам m_1 и m_2 находим коэффициенты a_2 и b_2 .
18. if $(|a_2 - 1| > porog_a)$ $a_2=1$, $b_2 = \frac{\sum_{i=1}^k (m_2(i) - m_1(i))}{k}$
19. End // (ограничение на значения a_2 , сильно отличающиеся от 1).
20. Else.
21. $a_2=1$, $b_2 = \frac{\sum_{i=1}^k (m_2(i) - m_1(i))}{k}$.
22. End.
23. Else.
24. Альтернативный метод коррекции.
25. End.
26. if $(|b_2| > porog_b)$ $a_{общ.} = a_2 \cdot a_{общ.}$, $b_{общ.} = a_2 \cdot b_{общ.} + b_2$.
27. End // (корректируем, если превышено минимальное значение яркостного сдвига).
28. $\hat{f}(j, y) = \frac{g(j, y) - b_{общ.}}{a_{общ.}}$ для всех y .
29. End.
30. Постобработка.
31. Поправка общего уровня яркости $\hat{f}(x, y) := \hat{f}(x, y) + \frac{\sum_{j=1}^w \sum_{i=1}^h (g(j, i) - \hat{f}(j, i))}{hw}$.

4. АЛЬТЕРНАТИВНЫЙ МЕТОД КОРРЕКЦИИ

Если из-за каких-то введенных ограничений невозможно применение основного алгоритма коррекции для текущего столбца, то для него используется альтернативный метод на основе анализа локальных характеристик [3]. В данном случае осуществляется корректировка только яркостного сдвига. Используется стратегия, заключающаяся в том, что искусственный перепад яркости следует определять на одноименных объектах изображения, тогда будет убрано влияние естественных изменений яркости.

В каждой точке y двух столбцов p_1, p_2 вычисляется локальная дисперсия разностей яркостей пикселей между столбцами

$$D_{p_1, p_2}(y) = \frac{\sum_{i=y-a_h}^{y+a_h} (p_2(i) - p_1(i) - E_{p_1, p_2}(y))^2}{2a_h + 1}, \text{ где локальное среднее}$$

$$E_{p_1, p_2}(y) = \frac{\sum_{i=y-a_h}^{y+a_h} (p_2(i) - p_1(i))}{2a_h + 1}, \quad y = \overline{a_h + 1, h - a_h}.$$

Точки с граничными значениями яркостей 0 и $V_{max} - 1$ не учитываются. Такие области игнорируются, поскольку в них истинное значение яркости может выходить за пределы диапазона.

Сначала происходит сортировка массива E_{p_1, p_2} . Получается отсортированная последовательность E_{p_1, p_2}^S и соответствующе упорядоченная последовательность D_{p_1, p_2}^S . Данные массивы укорачиваются с обоих концов в равных долях, пока диапазон значений для E_{p_1, p_2}^S не будет меньше допустимой величины η_1 . Получаем новые соответствующие массивы $E_{p_1, p_2}^C, D_{p_1, p_2}^C$. Из элементов E_{p_1, p_2}^C выбираются те,

которым соответствует маленькая локальная дисперсия $D_{p_1, p_2}^c(i) < p_D$, и по ним, как среднее, вычисляется корректировка \hat{b}_2 . Это позволяет более точно определять величину коррекции. Идея состоит в том, что если локальные участки столбцов принадлежат одноименным объектам, то соответствующие локальные дисперсии разностей яркостей пикселей будут иметь малые значения.

Предварительная фильтрация связанных массивов D_{p_1, p_2} и E_{p_1, p_2} по массиву E_{p_1, p_2} приводит к более стабильным результатам. Это объясняется тем, что медианное значение разностей пикселей соседних столбцов будет оценкой смещения яркости при достаточно большой выборке, стремящейся к бесконечности с учетом функции распределения разностей (2) и модели искажения яркости в виде сдвига. Давая допуск в районе медианы на отбор нужных участков по дополнительным эвристическим критериям, удалось повысить качество обработки, когда выборка не является столь большой.

Кроме этого, введен минимальный порог на количество отбираемых из массива E_{p_1, p_2}^c значений, по которым вычисляется средняя разность (не меньше $\eta_2=10\%$ от общего числа для столбца), в противном случае корректировка \hat{b}_2 вычисляется по всем пикселям $\hat{b}_2 = \frac{\sum_{i=1}^h (p_2(i) - p_1(i))}{h}$.

5. ПОСТОБРАБОТКА

Предлагается этап постобработки. Производится повторная более грубая корректировка $g(x, y)$ с удалением только явно выделяющихся полос (к примеру, если перепад средних значений яркости более 10), получаем изображение $\bar{f}(x, y)$. По значениям $\sum_{i=1}^h \bar{f}(x, i)$ для всех x строится массив значений A_1 скользящего среднего с довольно большим окном. Задание большого окна позволяет учесть на изображении общую информацию о яркости и сгладить влияние оставшихся полос. Далее подобное скользящее среднее A_2 строится по $\bar{f}(x, y)$, и затем значения $\hat{f}(x, y)$ обновляются $\hat{f}(x, y) := \bar{f}(x, y) + A_1(x) - A_2(x)$. Это во многом решает проблемы неустойчивой коррекции, возникающие в процессе обработки изображений предложенными методами в случае накопления ошибок.

На рис. 2. приведен пример обработки неоднородного по текстуре изображения, представленного ранее. Видно, что коррекция на сложных участках выполняется верно.

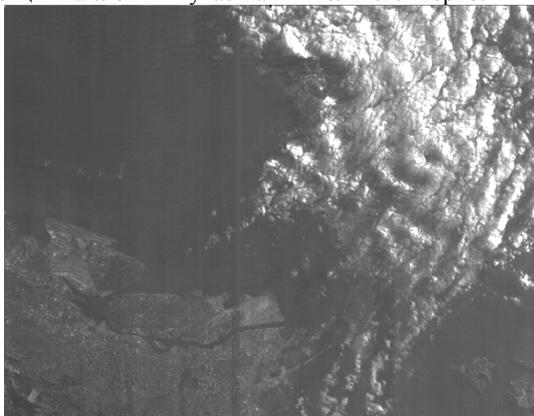


Рис. 2. Результат обработки изображения

6. ЗАКЛЮЧЕНИЕ

Рассмотрена проблема радиометрической коррекции спутниковых изображений. Предложен метод удаления вертикальных полос на изображениях, отличающийся повышенной надежностью благодаря комбинированному подходу, дополнительному контролю и улучшению корректировок яркостей. Используется вычисление корректировок с применением двумерной гистограммы яркостей пикселей соседних столбцов, либо локальный метод на основе рассмотрения статистических характеристик фрагментов разностей столбцов изображения. Подобный локальный анализ позволил определять одноименные гладкие участки столбцов изображения, по которым можно корректнее определить искусственный скачок яркости, свободный от влияния естественных изменений сюжета изображения в строчном направлении. Дополнительный контроль с использованием скользящего среднего по средним яркостям столбцов для сопоставления исходного и скорректированного изображения позволяет исправить возможную неправильную коррекцию в глобальном масштабе. Все это позволило обрабатывать изображения с существенными нерегулярными перепадами яркости, характерными для спутниковых изображений. Также метод является достаточно быстрым и нетребовательным к размеру оперативной памяти, поскольку в каждый момент времени требуются только соседние столбцы изображения.

7. ССЫЛКИ

1. Злобин В.К., Еремеев В.В. Обработка аэрокосмических изображений. – М.: ФИЗМАТЛИТ, 2006. – 288 с.
2. Odoma D., Milanfar P. Modeling Multiscale Differential Pixel Statistics. Electrical Engineering Department, University of California, Santa Cruz CA. 95064 USA.
3. Гусев В.Ю. К вопросу фильтрации структурных искажений на космических оптико-электронных изображениях // Геодезия и картография, 2012. – № 12. – М.: ФГУП “Государственный картографический и геодезический центр”. – С. 111 - 114.

Об авторах

Гусев Владимир – аспирант Московского авиационного института (НИУ).

E-mail: gusev3@mail.ru

НЕЛИНЕЙНАЯ ОРТОГОНАЛЬНАЯ ФИЛЬТРАЦИЯ ЦИФРОВЫХ ИЗОБРАЖЕНИЙ

В. Сазонов, М. Щербаков

Пензенский государственный университет, Пенза, Россия

Аннотация

Современные системы контроля и наблюдения позволяют получать оперативную информацию о параметрах технологического процесса, что в значительной степени повышает надежность и эффективность управления производственными процессами. Однако подобные системы управления и контроля распределенными объектами постоянно сталкиваются с проблемой качества и достоверности исходной информации. Для устранения или минимизации дестабилизирующего воздействия шумов и помех широко используют различные методы и алгоритмы предварительной обработки информации, в частности, процедуры цифровой фильтрации сигналов и изображений.

Изложена процедура построения нелинейного SVD-фильтра с адаптацией к локальным свойствам изображения. Приведен пример эффективной нелинейной «сингулярной» фильтрации подавления помех

Ключевые слова: нелинейная фильтрация, цифровая обработка изображений, фильтр Винера-Колмогорова, сингулярное разложение матриц, аппроксимация матриц.

1. ВВЕДЕНИЕ

В общем случае решение задачи восстановления стационарного случайного сигнала или процесса на фоне случайных шумов и помех предполагает выбор одной из двух (как минимум) гипотез: определена или нет априорная модель исходного сигнала. Это принципиальный момент, во многом определяющий дальнейшее решение поставленной задачи.

В первом случае наблюдаемый сигнал целенаправленно подгоняется под выбранную модель; при этом помеха рассматривается как пассивный параметр оценки несоответствия между моделью и сигналом.

Принято считать, что наилучшее восстановление для класса стационарных гауссовых сигналов достигается линейной системой, а ее оптимальной реализацией является фильтр Винера-Колмогорова [1, 2], который может быть представлен в следующем виде:

$$\hat{x}_{i,j} = \frac{\sigma_x^2}{\sigma_x^2 + \sigma_\eta^2} y_{i,j}, \quad (1)$$

где σ_x^2 - дисперсия полезного сигнала; σ_η^2 - дисперсия шума; $\hat{x}_{i,j}$ - восстановленный элемент исходного изображения;

$y_{i,j}$ - элемент наблюдаемого изображения.

Соответственно, эффективность работы фильтра Винера в данных условиях во многом определяется точностью количественной оценки шумовой компоненты. Однако данный подход обладает некоторыми недостатками, в том числе:

- оптимальность достигается интегрально по всей совокупности анализируемого процесса [1];

- возможная неустойчивость алгоритмов фильтрации [2].

Эти недостатки хорошо известны, они привели к появлению целого ряда достаточно успешных модификаций фильтра Винера, например, или в виде апертурных (масочных) алгоритмов, работающих в окрестности локальных точек [1, 2], или в виде робастных (устойчивых) алгоритмов [3].

В целом же, следует признать, что эти подходы зачастую противоречат друг другу, поскольку переход к локальной оптимизации в силу естественного ограничения объема исходных данных может приводить к практической невозможности построения статистически устойчивых алгоритмов.

2. ЛОКАЛЬНАЯ ФИЛЬТРАЦИЯ ИЗОБРАЖЕНИЙ НА ОСНОВЕ СИНГУЛЯРНОГО РАЗЛОЖЕНИЯ

Пусть $x_{i,j}$ - значение яркости полезного изображения на пересечении i -й строки и j -го столбца, а наблюдаемое изображение описывается моделью:

$$y_{i,j} = x_{i,j} + \eta_{i,j}, \quad i = \overline{0, I-1}, \quad j = \overline{0, J-1}. \quad (2)$$

Здесь $\eta_{i,j}$ - значение помехи в точке с координатами (i, j) ;

$f(\cdot)$ - функция, описывающая взаимодействие сигнала и помехи, а I и J - соответственно, число строк и столбцов изображения.

Рассмотрим некоторую k -окрестность точки наблюдаемого изображения $y_{m,m \in 2k-1}(i, j)$, где k - апертура пространственного фильтра.

Такое представление элемента изображения позволяет сформировать подматрицу наблюдаемых данных \mathbf{A} размером $(2k-1) \times (2k-1)$, подлежащую дальнейшей обработке (без потери общности будем считать данную область локально стационарной). Одним из наиболее эффективных методов статистического анализа матриц является ее SVD-разложение с целью приведения к каноническому виду [3, 4].

Пусть матрица $\mathbf{A} \in \mathbf{R}^{m \times n}$ имеет m столбцов и n строк, причем $m > n$. Такая матрица может быть представлена в виде разложения:

$$\mathbf{A} = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^T = \sum_{i=1}^n \sigma_i \mathbf{u}_i \mathbf{v}_i^T, \quad (3)$$

где \mathbf{u}_i и \mathbf{v}_i - левый и правый сингулярные векторы матрицы \mathbf{A} , являющиеся ортонормированными столбцами матриц $\mathbf{U} \in \mathbf{R}^{m \times m}$ и $\mathbf{V} \in \mathbf{R}^{n \times n}$, соответственно; $\sigma_i \geq 0$ - диагональные элементы матрицы \mathbf{S} , называемые сингулярными числами матрицы \mathbf{A} .

В случае если соблюдается соотношение $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \geq 0$, где $\sigma_n > 0$, и матрица \mathbf{A} имеет полный ранг, то ее можно представить в виде разложения

$$\mathbf{A} = \sigma_1 \cdot \mathbf{A}_1 + \sigma_2 \cdot \mathbf{A}_2 + \dots + \sigma_n \cdot \mathbf{A}_n = \sum_{i=1}^n \sigma_i \cdot \mathbf{A}_i, \quad (4)$$

где $\mathbf{A}_i = \mathbf{u}_i \cdot \mathbf{v}_i^T$ - внешнее произведение столбца унитарной матрицы \mathbf{U} и соответствующего столбца унитарной матрицы \mathbf{V} .

В терминах матричного анализа SVD-разложение (3, 4) предполагает возможность аппроксимации матрицы исходных данных матрицей более низкого ранга, что при восстановлении изображений, искаженных аддитивными некоррелированными шумами (2), позволяет разделить наблюдаемую матрицу \mathbf{A} на две компоненты: «полезное» изображение и шум. В качестве критерия эффективности матричной аппроксимации можно использовать критерий вида:

$$\lambda(p) = \frac{\sum_{i=1}^p \sigma_i^2}{\sum_{i=1}^p \sigma_i^2 + \sum_{j=p+1}^n \sigma_j^2} \leq \lambda_{\text{опт}}, \quad (5)$$

где $\lambda_{\text{опт}}$ - эффективный порог аппроксимации, позволяющий оценить «мощность» шумов $\sum_{j=p+1}^n \sigma_j^2$, удаляемых из рассмотренной окрестности матрицы исходных данных.

Можно заметить, что критерий (5) матричной SVD-аппроксимации полностью повторяет и структуру, и физический смысл классического фильтра Винера-Колмогорова (2). Важнейшим свойством предложенного подхода является робастность данной SVD-фильтрации, так как любое пренебрежение малое (даже равное нулю) значение сингулярного числа автоматически относится к шумовой составляющей, что полностью гарантирует устойчивость предложенного алгоритма.

3. НЕЛИНЕЙНЫЙ SVD-ФИЛЬТР, АДАПТИРОВАННЫЙ К ЛОКАЛЬНЫМ СВОЙСТВАМ ИЗОБРАЖЕНИЯ

Следует отметить, что в восстановленном с помощью SVD-фильтра изображении практически отсутствуют искаженные шумом области, однако его контрастность по отношению к исходному изображению в определенной степени ухудши-

лась. Это вполне ожидаемый результат, так как и фильтр Винера и SVD-фильтр Винера-Колмогорова [4] относятся к классу линейных НЧ-фильтров.

Для устранения указанного недостатка в [5] был предложен новый подход к проектированию нелинейных фильтров с адаптацией к локальным свойствам изображений. На его основе можно построить различные структуры цифровых нелинейных фильтров, отличающихся видом оценок, используемых для анализа локальных свойств сигналов, и параметрами адаптации.

В частности, входной процесс $x(n)$ может быть представлен в виде низкочастотной $x_L(n)$ и высокочастотной $x_H(n)$ составляющих, а выходной сигнал $y(n)$ фильтра формируется как сумма

$$y(\mathbf{n}) = x_L(\mathbf{n}) + \alpha x_H(\mathbf{n}), \quad (6)$$

где $\mathbf{n} = [n1, n2]$ - вектор, определяющий координаты точки (пикселя) изображения, а параметр α адаптации определяется локальным значением критерия (5).

В зависимости от локальных свойств изображения (фон или перепад) параметр α должен усиливать либо ослаблять вклад нелинейной составляющей фильтра.

Таким образом, поведение данного нелинейного фильтра будет иметь адаптивный характер, изменяясь в зависимости от локальных свойств входного сигнала. Исходя из выбранной модели (6), была разработана нелинейная модификация SVD-фильтра, сохраняющая контрастные свойства изображений.

Будем формировать выходной сигнал фильтра, согласно следующему выражению:

$$y(n) = (1 - \alpha_n)x(n) + \alpha_n \hat{x}(n), \quad (7)$$

где α_n - параметр адаптации; $-1 \leq \alpha_n \leq 1$, а $\hat{x}(n)$ - выходной сигнал линейного SVD-фильтра.

Если параметр $\alpha_n = 1$, то выражение (7) будет соответствовать линейному фильтру нижних частот, при $\alpha_n = -1$ - фильтру верхних частот, а при $\alpha_n = 0$ входной сигнал будет передаваться без изменения. Определяя соответствующим образом параметр α адаптации, можно изменять поведение фильтра в зависимости от локальных свойств входного сигнала.



а



б



в

Результаты нелинейной SVD-фильтрации изображения:

а – исходное изображение; б – изображение, искаженное гауссовым шумом; в – результат адаптивной нелинейной фильтрации

Допустим, требуется обеспечить фильтрацию широкополосного шума без искажения фронтов. В этом случае фильтр должен изменять свое поведение, проявляя низкочастотные свойства на пологих участках изменения входного сигнала и высокочастотные – при обнаружении перепадов.

Одним из вариантов преобразования критерия λ_n в параметр адаптации α_n является кусочно-линейная функция

$$\alpha_n = \begin{cases} 1, & \lambda_n < a; \\ \frac{2\lambda_n - a - b}{b - a}, & a \leq \lambda_n \leq b; \\ -1, & \lambda_n > b, \end{cases} \quad (8)$$

где предварительная настройка фильтра на заданный динамический диапазон осуществляется изменением величин порогов a и b .

Пример фильтрации зашумленного изображения приведен на рисунке. В качестве входного использовалось изображение «Замок» (рис., а), искаженное гауссовым шумом с дисперсией $\sigma_n=0.01$ (рис., б). Результат адаптивной нелинейной фильтрации вида (7) показан на рис., в. Параметр адаптации формировался согласно выражениям (5) и (8). Из приведенных результатов видно, что качество изображения после нелинейной обработки отличается высокой степенью подавления шума и четкостью деталей изображения.

4. ВЫВОДЫ И РЕКОМЕНДАЦИИ

- ❖ Показано использование сингулярного разложения матрицы исходных данных для оперативного восстановления изображений, искаженных помехами различной природы.
- ❖ Предложен подход к построению нелинейных SVD-фильтров с адаптацией к локальным свойствам изображений.
- ❖ Приведенный пример иллюстрирует потенциальные возможности нелинейной SVD-фильтрации при восстановлении изображений.
- ❖ Анализ влияния размера апертуры и выбора порога эффективности SVD-аппроксимации, а также параметров адаптации на качество восстановления является отдельной задачей и требует дополнительных исследований.

5. ССЫЛКИ

- [1] Гонсалес Р., Вудс Р. Цифровая обработка изображений. – М.: Техносфера, 2005. – 1072 с.
- [2] Прэтт У. Цифровая обработка изображений; пер. с англ. – М.: Мир, 1982. – Кн.1 – 312 с.
- [3] Голуб Дж., Ван Лоун Ч. Матричные вычисления; пер. с англ. – М.: Мир, 1999. – 548 с.
- [4] Сазонов В.В., Щербаков М.А. Применение сингулярного фильтра Винера-Колмогорова при восстановлении изображений // Инновационные информационные технологии: Труды междунар. науч.-практ. конф. – М.: МИЭМ, 2012. – С. 309 – 312.
- [5] Щербаков М.А., Сазонов В.В. Проектирование нелинейных фильтров с адаптацией к локальным свойствам изображений // Проблемы автоматизации и управления в технических системах: Труды международного симпозиума. – Пенза: Пенз. гос. ун-т. – 2013. – С. 185 – 191.

Об авторах

Сазонов Владимир – докторант, Пензенский государственный университет.

E-mail: sazonov@inbox.ru

Щербаков Михаил – доктор технических наук, профессор, заслуженный деятель науки РФ, заведующий кафедрой автоматизации и телемеханики, Пензенский государственный университет.

E-mail: avitel@pnzgu.ru

Об оптимальном выделении субполосных компонент изображений

Андрей Черноморец

Белгородский государственный национальный исследовательский университет, Белгород, Россия
chernomorets@bsu.edu.ru

Аннотация

Рассматривается задача оптимальной субполосной фильтрации изображений в применении к проблеме удаления узкополосного шума на изображениях. Предложено решение оптимизационной задачи фильтрации на основе кронекеровского произведения субполосных матриц. Показано преимущество данного подхода по сравнению с известными методами.

Ключевые слова: изображение, оптимальная фильтрация, подобласть пространственных частот, оптимальные субполосные компоненты

In this article we consider the task of optimal subband images filtering in application to a problem of deleting narrow-band noise on images. The solution of the optimization task of filtering on the basis of kroneker product of subband matrixes is proposed. Advantage of this approach in comparison with known methods is shown.

Keywords: image, optimal filtering, subarea of spatial frequencies, optimal subband components

1. ВВЕДЕНИЕ

Одной из существенных проблем, возникающих при регистрации и передаче изображений в телекоммуникационных каналах связи, является наличие различного рода помех, вызванных работой аппаратуры. Во многих случаях помехи на изображении имеют узкополосный характер, для удаления которых разработано значительное количество методов фильтрации, однако при их реализации, в большинстве случаев, возникает проблема нежелательного удаления информации об объектах на изображении [1 – 2]. В связи с этим возникает задача разработки методов фильтрации, которые позволяют вносить минимальные искажения в оригинальное изображение после удаления узкополосных шумов.

Важным представляется подход к фильтрации изображений, основанный на вычислении оптимальных значений их отдельных характеристик в области пространственных частот (ПЧ), в первую очередь, энергетических. Для удаления помех на изображении в работе [3] предложен метод оптимальной субполосной фильтрации, исследованию путей компьютерной реализации которого посвящена данная работа. Теоретической основой метода оптимальной субполосной фильтрации являются положения субполосного анализа и синтеза [4 – 5] изображений, базирующиеся на учете энергетических характеристик изображений в отдельных подобластях пространственных частот (ППЧ) V_{sr} , $s = 1, 2, \dots, S$, $v = 1, 2, \dots, R$, вида,

$$V_{sr} = \{ (x, y) | (x \in D_s) \cap (y \in G_r) \}, \quad s = 1, 2, \dots, S, \quad v = 1, 2, \dots, R, \quad (1)$$

$$D_s = \{ x | x \in [-u_{s2}, -u_{s1}] \cup [u_{s1}, u_{s2}] \}, \quad G_r = \{ y | y \in [-v_{r2}, -v_{r1}] \cup [v_{r1}, v_{r2}] \},$$

$$u_{11} = 0, \quad u_{s2} = \pi; \quad v_{11} = 0, \quad v_{r2} = \pi,$$

$$u_{s+1,1} = u_{s2}; \quad v_{r+1,1} = v_{r2}, \quad 0 \leq u_{s1} < u_{s2} < \pi; \quad 0 \leq v_{r1} < v_{r2} < \pi,$$

где x, y – пространственные частоты (ПЧ).

В процессе субполосного анализа-синтеза изображение Φ представлено матрицей $\Phi = \{f_{ik}\}$, $i = 1, 2, \dots, N$, $k = 1, 2, \dots, M$, значений яркости изображения в соответствующих пикселях.

2. ОПТИМАЛЬНАЯ СУБПОЛОСНАЯ ФИЛЬТРАЦИЯ

Постановке задачи оптимальной субполосной фильтрации изображений в выбранной ППЧ вида (1) соответствует задача [3, 6] выделения аддитивной субполосной компоненты $Y = (y_{ik})$, $i = 1, 2, \dots, N$, $k = 1, 2, \dots, M$, изображения Φ на основе оптимизации функционала следующего вида

$$W_{sr}(Y, \gamma) = (1 - \gamma) \iint_{(u,v) \in V_{sr}} |F^\Phi(u, v) - F^Y(u, v)|^2 dudv / 4\pi^2 + \\ + \gamma \iint_{(u,v) \in V_{sr}} |F^Y(u, v)|^2 dudv / 4\pi^2, \quad (2)$$

где $F^\Phi(x, y)$, $F^Y(x, y)$ – трансформанты Фурье изображений Φ и Y ; γ определяет весомость соответствующих интегралов, причем предполагается выполнение неравенства

$$0 < \gamma < 1. \quad (3)$$

Естественным принципом оптимизации является следующее вариационное условие:

$$W_{sr}(Y, \gamma) = \min, \quad (4)$$

где минимум ищется по всем изображениям Y размерности $(M \times N)$ в предположении, что параметр γ является постоянной величиной.

В работе [3] показано, что субполосная компонента $Y = (y_{ik})$ изображения $\Phi = (f_{ik})$, $i = 1, 2, \dots, N$, $k = 1, 2, \dots, M$, соответствующая ППЧ V_{sr} вида (1) и удовлетворяющая оптимизационной задаче (2) – (4) определяется соотношением

$$(\gamma I + (1 - 2\gamma)C^{sr})\bar{w} = (1 - \gamma)C^{sr}\bar{v}, \quad (5)$$

где матрица C^{sr} – кронекеровское произведение

$$C^{sr} = A_s \otimes B_r, \quad (6)$$

субполосных матриц A_s и B_r ($A_s = \{a_{in}^s\}$, $i, n = 1, \dots, N$, $B_r = \{b_{km}^r\}$, $k, m = 1, \dots, M$), соответствующих заданной ППЧ V_{sr} , с элементами вида:

$$a_{in}^s = \int_{x \in D_s} \exp(-jx(i-n))dx / 2\pi; \quad b_{km}^r = \int_{y \in G_r} \exp(-jy(k-m))dy / 2\pi, \quad (7)$$

$\bar{w} = (w_1, \dots, w_{N \cdot M})^T$, $\bar{v} = (v_1, \dots, v_{N \cdot M})^T$ – векторы, компоненты которых формируются на основе строк матриц Y и Φ следующим образом:

$$w_{(i-1) \cdot M + k} = y_{ik}, \quad v_{(i-1) \cdot M + k} = f_{ik}, \quad i = 1, 2, \dots, N, \quad k = 1, 2, \dots, M, \quad (8)$$

I – единичная матрица соответствующей размерности.

Можно показать, что матрица

$$\Gamma^{sr} = \gamma I + (1 - 2\gamma)C^{sr} \quad (9)$$

является неособенной, следовательно, обратная матрица $(\Gamma^{sr})^{-1}$ существует и решение (5) оптимизационной задачи (2) – (4) относительно вектора \bar{w} имеет вид:

$$\bar{w} = (1 - \gamma)(\gamma I + (1 - 2\gamma)C^{sr})^{-1} C^{sr} \bar{v}. \quad (10)$$

Непосредственная реализация соотношения (10) требует применения алгоритмов обращения матриц больших размерностей, что приводит к существенным вычислительным затратам.

3. КОМПЬЮТЕРНАЯ РЕАЛИЗАЦИЯ НА ОСНОВЕ СВОЙСТВ СУБПОЛОСНЫХ МАТРИЦ

В данной работе предложена компьютерная реализация соотношения (10), основанная на использовании свойств субполосных матриц, что позволяет существенно снизить вычислительную сложность метода оптимальной субполосной фильтрации изображений.

Так, можно показать, что матрица C^{sr} является симметрической. Следовательно, справедливо следующее представление:

$$C^{sr} = X^{sr} M^{sr} (X^{sr})^T, \quad (11)$$

где $M^{sr} = \text{diag}(\mu_1^{sr}, \mu_2^{sr}, \dots, \mu_{NM}^{sr})$ – диагональная матрица собственных чисел матрицы C^{sr} , $X^{sr} = (\bar{x}_1^{sr} \dots \bar{x}_{(i-1)M+k}^{sr} \dots \bar{x}_{NM}^{sr})$ – матрица, столбцы которой образованы соответствующими собственными векторами $\bar{x}_1^{sr}, \bar{x}_2^{sr}, \dots, \bar{x}_{NM}^{sr}$ данной матрицы.

Также можно показать, что

$$(\Gamma^{sr})^{-1} = X^{sr} (N^{sr})^{-1} (X^{sr})^T, \quad (12)$$

где N^{sr} – диагональная матрица, образованная на основе множества собственных чисел матрицы Γ^{sr} вида,

$$\eta_n = \gamma + (1 - 2\gamma)\mu_n^{sr}, \quad n = 1, 2, \dots, NM. \quad (13)$$

Подставив соотношения (9), (11), (12) и (13) в выражение (10) и учитывая ортогональность собственных векторов матрицы C^{sr} , получим:

$$\bar{w} = (1 - \gamma)X^{sr} Z^{sr} (X^{sr})^T \bar{v}, \quad (14)$$

где матрица Z^{sr} – диагональная матрица, размерности $NM \times NM$, $Z^{sr} = \text{diag}(z_1^{sr}, \dots, z_n^{sr}, \dots, z_{NM}^{sr})$, с элементами вида

$$z_n^{sr} = \frac{\mu_n^{sr}}{\gamma + (1 - 2\gamma)\mu_n^{sr}}, \quad n = 1, 2, \dots, NM. \quad (15)$$

Следовательно, при компьютерной реализации рассматриваемого метода соотношение для вычисления отдельного элемента w_m вектора \bar{w} имеет вид:

$$w_m = (1 - \gamma) \sum_{n=1}^{NM} v_n \sum_{p=1}^{NM} x_{m,p}^{sr} z_p^{sr} x_{n,p}^{sr}, \quad m = 1, 2, \dots, NM. \quad (16)$$

где

$$z_{(i-1)M+k}^{sr} = \frac{\lambda_i^s \lambda_k^r}{\gamma + (1 - 2\gamma)\lambda_i^s \lambda_k^r}, \quad i = 1, 2, \dots, N, \quad k = 1, 2, \dots, M, \quad (17)$$

$$x_{(j-1)M+l, (i-1)M+k}^{sr} = q_{ji}^s q_{lk}^r, \quad j, i = 1, 2, \dots, N, \quad l, k = 1, 2, \dots, M,$$

q_{ji}^s – j -й элемент собственного вектора \bar{q}_i^s , $i = 1, 2, \dots, N$, матрицы A_s и q_{lk}^r – l -й элемент собственного вектора \bar{q}_k^r , $k = 1, 2, \dots, M$, матрицы B_r .

В выражении (16) можно существенно уменьшить объем вычислений, если использовать только значения z_p^{sr} , соответствующие ненулевым собственным числам матрицы кронекеровского произведения C^{sr} ,

$$\mu_{(i-1)J_s+k}^{sr} = \lambda_i^s \lambda_k^r, \quad i = 1, \dots, J_s, \quad k = 1, \dots, J_r, \quad \lambda_i^s > \varepsilon, \quad \lambda_k^r > \varepsilon, \quad (18)$$

где J_s и J_r – количество ненулевых собственных чисел соответствующих субполосных матриц A_s и B_r ; ε – малое положительное число.

Тогда, если ввести обозначения: Z_1^{sr} – диагональная матрица, составленная из значений вида (15), (18), соответствующих ненулевым собственным числам субполосных матриц A_s и B_r ; X_1^{sr} – матрица, составленная из собственных векторов матрицы кронекеровского произведения C^{sr} , соответствующих ненулевым собственным числам вида (15), (18), то соотношение

$$\bar{w} \approx (1 - \gamma)X_1^{sr} Z_1^{sr} (X_1^{sr})^T \bar{v} \quad (19)$$

следует применять для практической реализации метода оптимальной субполосной фильтрации изображений (решение оптимизационной задачи (2) – (4)), не выполняя обращения возникающих при этом матриц большой размерности. Значения элементов оптимальной субполосной компоненты $Y = (y_{ik})$ изображения $\Phi = (f_{ik})$, $i = 1, 2, \dots, N$, $k = 1, 2, \dots, M$, соответствующей ППЧ V_{sr} , вычисляются на основе следующего соотношения:

$$y_{ik} = w_{(i-1)M+k}, \quad i = 1, 2, \dots, N, \quad k = 1, 2, \dots, M. \quad (20)$$

Использование матричных вычислений в среде Matlab позволяет получать решение на основе соотношения (19) за приемлемое время, при этом существенное снижение требуемого объема оперативной памяти достигается путем реализации соотношения (19) в виде

$$\bar{w} \approx (1 - \gamma)(X_1^{sr} Z_1^{sr})(X_1^{sr})^T \bar{v}. \quad (21)$$

Эффективность применения метода оптимальной субполосной фильтрации изображений для удаления узкополосных шумов объясняется тем, что субполосные компоненты, получаемые на основе соотношений (16), (19) и (20), имеют важное свойство – отсутствие влияния на данные компоненты энергии исходного изображения из смежных подобластей пространственных частот, что обеспечивает высокое качество фильтрации узкополосных помех, и подтверждено многочисленными вычислительными экспериментами с реальными изображениями и наложенным на них аддитивным шумом.

4. ВЫЧИСЛИТЕЛЬНЫЕ ЭКСПЕРИМЕНТЫ

При проведении вычислительных экспериментов для сравнения результатов применения различных методов фильтрации было использовано изображение аэропорта, приведенное на рис. 1, а, являющееся результатом наложения на исходное изображение земной поверхности, размерностью

128x128 пикселей, узкополосного модельного шума, сгенерированного на основе соотношения

$$f_{ik}^{noise} = \sin(0.11 \cdot 2\pi k), \quad i, k = 1, 2, \dots, 128,$$

и затем преобразованного таким образом, чтобы уровень шум/сигнал имел значение 0.6.

На рис. 1, б отображены величины абсолютных значений трансформант Фурье исходного изображения, заданного аддитивного шума и зашумленного изображения, соответствующие выбранной подобласти ПЧ D_0 ,

$$D_0 = \{(u = 0.299\pi) \cap (v \in [0.05\pi, 0.2\pi])\}. \quad (22)$$

Размеры подобласти D_0 выбраны для отображения результатов в соответствии с распределением энергии шума – основная часть энергии шума сосредоточена в ППЧ V_0 ,

$$V_0 = \{(u \in [0, \pi]) \cap (v \in [0.095\pi, 0.125\pi])\}. \quad (23)$$

В ходе вычислительного эксперимента была выделена при значении $\gamma = 0.5$ оптимальная субполосная компонента вида (19) – (20), соответствующая ППЧ V_0 (23), а также были вычислены результаты КИХ-фильтрации (длина импульсной характеристики – 512, ширина переходной полосы $c = 0,005\pi$) в той же подобласти ПЧ. Величины, соответствующие подобласти ПЧ D_0 , абсолютных значений трансформант Фурье зашумленного изображения, выделенной субполосной компоненты и результата КИХ-фильтрации приведены на рис. 2, а.

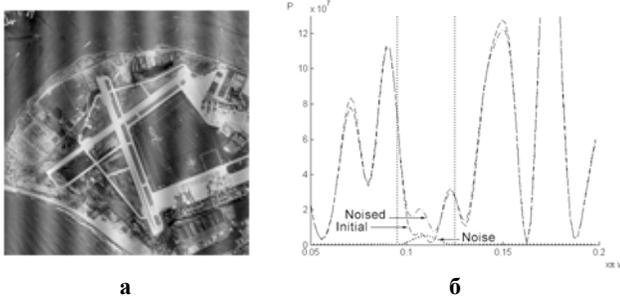


Рис. 1. Зашумленное изображение земной поверхности: а – зашумленное изображение, б – абсолютные значения трансформант Фурье шума, исходного и зашумленного изображений

На основе полученных результатов была выполнена режекторная фильтрация шума. Результаты режекторной фильтрации на основе выделения оптимальной субполосной компоненты и КИХ-фильтрации в подобласти V_0 приведены на рис. 2, в и 2, г. На рис. 2, в (результат субполосной фильтрации) полос, вызванных наличием шума, значительно меньше, чем на рис. 2, г (результат КИХ-фильтрации).

На рис. 2, б отображены абсолютные значения трансформант Фурье исходного изображения и результатов, представленных на рис. 2, в и 2, г.

На рис. 2, а видно, что на трансформанту Фурье результата КИХ-фильтрации оказывает существенное влияние увеличение энергии в смежных с V_0 (слева и справа) подобластях ПЧ. При этом значения трансформанты Фурье выделенной оптимальной субполосной компоненты снижаются за пределами области V_0 .

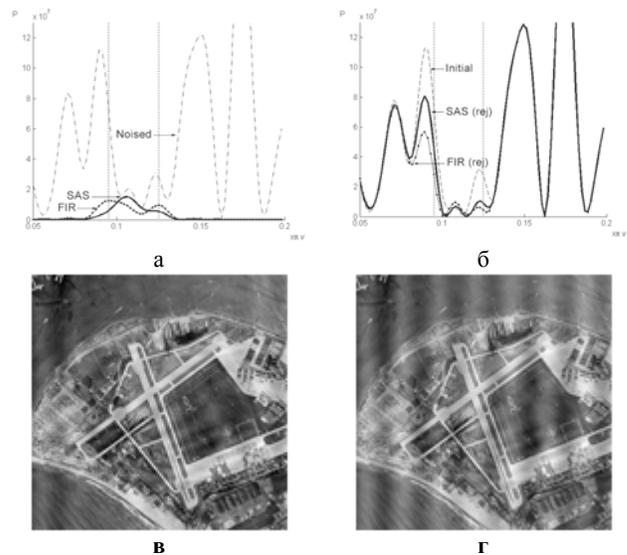


Рис. 2. Результаты фильтрации при $\gamma = 0.5, c = 0,005\pi$: а – абсолютные значения трансформант Фурье зашумленного изображения и результатов субполосной (SAS) и КИХ-фильтрации (FIR); б – абсолютные значения трансформант Фурье исходного изображения и результатов режекторной фильтрации; в – результат режекторной субполосной фильтрации; г – результат режекторной КИХ-фильтрации

Приведенные на рис. 2, б графики демонстрируют, что при применении оптимальной субполосной фильтрации трансформанта Фурье результата режекторной субполосной фильтрации менее отличается вне области фильтрации от трансформанта Фурье исходного изображения, чем трансформанта Фурье результата режекторной КИХ-фильтрации.

СКО в пространственной области результатов режекторной фильтрации относительно исходного изображения – 0.277 и 0.53, соответственно, при применении субполосных компонент и КИХ-фильтрации.

Визуально отличие в погрешности можно наблюдать на рис. 2, в и 2, г – метод субполосной фильтрации позволил удалить шум визуально значительно лучше, чем при использовании КИХ-фильтров.

Аналогичные расчеты результатов фильтрации в подобласти V_0 (рис. 3) были выполнены для выделения оптимальной субполосной компоненты при $\gamma = 0.8$ и при использовании КИХ-фильтра, длина импульсной характеристики которого равна 512 и ширина переходной полосы $c = 0,0005\pi$.

Результаты режекторной фильтрации, представленные на рис. 3, в и 3, г, визуально практически не различаются, однако СКО первого изображения несколько лучше, чем у второго изображения (0.261 и 0.283, соответственно).

Изображения, приведенные на рис. 2 и 3, демонстрируют преимущество применения субполосной фильтрации по сравнению с КИХ-фильтрами.

В ходе эксперимента также были получены результаты режекторной фильтрации в ППЧ V_0 на основе идеального фильтра и фильтра Гаусса (см. таблицу, где ОСФ – оптимальная субполосная фильтрация).

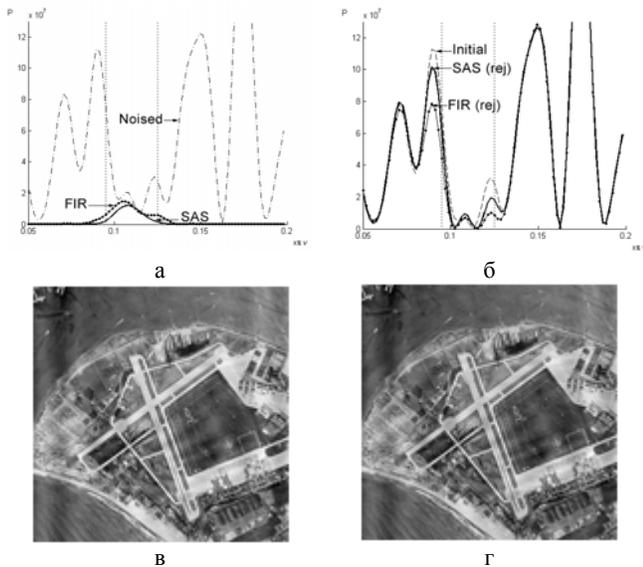


Рис. 3. Результаты фильтрации при $\gamma = 0.8$, $\epsilon = 0.0005\lambda$: а – абсолютные значения трансформант Фурье зашумленного изображения и результатов субполосной (SAS) и КИХ-фильтрации (FIR); б – абсолютные значения трансформант Фурье исходного изображения и результатов режекторной; в – результат режекторной субполосной фильтрации; г – результат режекторной КИХ-фильтрации

Результаты фильтрации

	Доля энергии вне V_0	Режекторная фильтрация		
		СКО в D_0	СКО	PSNR
ОСФ, $\gamma=0.1$	0.091	0.309	0.303	25.371
ОСФ, $\gamma=0.2$	0.068	0.250	0.292	25.695
ОСФ, $\gamma=0.3$	0.055	0.203	0.286	25.882
ОСФ, $\gamma=0.4$	0.047	0.166	0.281	26.029
ОСФ, $\gamma=0.5$	0.041	0.135	0.277	26.176
ОСФ, $\gamma=0.6$	0.036	0.109	0.271	26.338
ОСФ, $\gamma=0.7$	0.032	0.087	0.266	26.517
ОСФ, $\gamma=0.8$	0.028	0.068	0.261	26.682
ОСФ, $\gamma=0.9$	0.025	0.05	0.264	26.579
КИХ, $\epsilon = 0.005$	0.143	0.21	0.53	20.527
КИХ, $\epsilon = 0.0005$	0.043	0.141	0.283	25.977
Идеальный	0.149	0.382	0.332	24.597
Гаусса	0.112	0.327	0.254	26.926

В таблице представлены значения доли энергии [4] результатов фильтрации, полученных указанными в таблице методами, вне подобласти фильтрации V_0 , а также величины СКО и PSNR результатов режекторной фильтрации относительно исходного изображения и СКО абсолютных значений трансформант Фурье (в подобласти D_0) результатов режекторной фильтрации относительно абсолютных значений трансформанты Фурье исходного изображения.

В ходе вычислительных экспериментов (см. таблицу) наименьшая погрешность в пространственной области после удаления шума на изображении была получена при применении метода оптимальной субполосной фильтрации ($\gamma=0.8$).

5. ВЫВОДЫ

В настоящей работе разработаны подходы к практической реализации на вычислительной технике метода оптимальной субполосной фильтрации изображений в заданных подобластях пространственных частот, обеспечивающего вычисление субполосных компонент, на которые не оказывает влияние энергия изображения из смежных подобластей. Полученные в ходе приведенных вычислительных экспериментов результаты демонстрируют преимущество применения оптимальных субполосных компонент в процессе фильтрации узкополосных помех по сравнению с использованием известных фильтров,

6. ДОПОЛНИТЕЛЬНАЯ ИНФОРМАЦИЯ

Работа выполнена в рамках Государственного задания НИУ «БелГУ» на 2014 год (код проекта № 358).

7. ССЫЛКИ

- [1] Гонсалес, Р. Цифровая обработка изображений [Текст] / Р. Гонсалес, Р. Вудс. – М.: Техносфера, 2012. – 1104 с.
- [2] Методы компьютерной обработки изображений [Текст]; под редакцией В.А. Соифера. – М.: ФИЗМАТЛИТ, 2003. – 784 с.
- [3] Жилияков, Е.Г. О субполосном анализе изображений [Текст] / Е.Г. Жилияков, А.А. Черноморец // ГрафиКон'2013: 23-я Международная конференция по компьютерной графике и зрению: 16 – 20 сентября, 2013 г. Владивосток, Институт автоматизации и процессов управления ДВО РАН: Труды конференции. – С. 230 - 233.
- [4] Жилияков, Е.Г. Об эффективности метода оценивания значений долей энергии изображений на основе частотных представлений [Текст] / Е.Г. Жилияков, А.А. Черноморец, А.Н. Заливин // Известия ОрелГТУ. Информационные системы и технологии. – № 2/52 (563) март-апрель. – 2009. – С. 12 - 22.
- [5] Жилияков, Е.Г. О частотном анализе изображений [Текст] / Е.Г. Жилияков, А.А. Черноморец // Вопросы радиоэлектроники. Сер. ЭВТ. – 2010. – Вып. 1. – С. 94 - 103.
- [6] Жилияков, Е.Г. Об оптимальном выделении субполосных компонент изображений [Текст] / Е.Г. Жилияков, А.А. Черноморец // Информационные системы и технологии. – № 1 (75). – 2013. – С. 5 - 11.

Информация об авторах

Черноморец Андрей – к. т. н., доцент, заведующий кафедрой прикладной математики и информатики, Белгородский государственный национальный исследовательский университет.

e-mail: chernomorets@bsu.edu.ru

ИСПОЛЬЗОВАНИЕ КАРТОГРАФИЧЕСКОЙ ОСНОВЫ В ЗАДАЧЕ АНАЛИЗА ЛИНЕЙНОЙ СПЕКТРАЛЬНОЙ СМЕСИ НА ГИПЕРСПЕКТРАЛЬНЫХ ИЗОБРАЖЕНИЯХ

А. Денисова, В. Мясников
Факультет информатики

Самарский государственный аэрокосмический университет им. академика С.П. Королёва
(национальный исследовательский университет), Самара, Россия,

denisova_ay@geosamara.ru, vmyas@smr.ru

Аннотация

В статье предлагается алгоритм с использованием картографической основы для анализа линейной спектральной смеси. Алгоритм выполняет разложение изображения по компонентам линейной спектральной смеси с использованием картографической информации для уточнения коэффициентов смеси в пикселях, соответствующих нескольким областям карты.

Ключевые слова: гиперспектральные изображения, спектральное разложение, гиперспектральный анализ, метод наименьших квадратов, картографическая основа.

1. ВВЕДЕНИЕ

Задача линейного спектрального разложения является одной из основных в круге задач анализа гиперспектральных данных ДЗЗ. Предполагается, что каждый пиксель изображения описывается линейной моделью спектральной смеси некоторых спектральных сигнатур [1 - 3] и задача линейного спектрального разложения состоит в отыскании коэффициентов этой линейной комбинации.

Совокупность методов линейного спектрального анализа (Linear Spectral Analysis – LSMA) была разработана и систематизирована в работах профессора Chang, C.I. [1 - 3]. Методы можно разделить по степени использования априорной информации на *управляемые* (Supervised LSMA - SLSMA), когда известен список сигнатур разложения, и *неуправляемые* (Unsupervised LSMA – ULSMA), когда априорная информация о списке сигнатур отсутствует. Особенность ULSMA заключается в наличии стратегии оценки набора сигнатур, составляющих изображение, некоторым оптимальным способом с последующим применением одного из алгоритмов SLSMA. Алгоритмы SLSMA представляют собой комбинацию методов модификации данных для учёта различной априорной информации о взаимозависимостях каналов и сигнатур и методов линейного спектрального разделения смесей. Последние подразделяются по наличию ограничений на коэффициенты разложения и основываются на соответствующих методах оптимизации.

В настоящей работе рассматривается задача линейного спектрального разложения, но её формулировка пополняется дополнительной информацией, присутствующей на картографической основе в геоинформационных системах. Используя картографическую основу, можно наложить дополнительные ограничения в задаче линейного

спектрального разложения и существенно улучшить решение на границах объектов.

2. ЗАДАЧА РАЗЛОЖЕНИЯ ЛИНЕЙНОЙ СМЕСИ СПЕКТРАЛЬНЫХ СИГНАТУР

Для описания пикселя v изображения используется линейная модель спектральной смеси [1 - 3], когда исходный пиксель представляется в виде линейной комбинации нескольких спектральных сигнатур $M = (\overline{m}_1, \dots, \overline{m}_p)$:

$$v = M\alpha + \pi, \quad (1)$$

где π – ошибка модели и измерений; $\alpha^T = (\alpha_1, \dots, \alpha_p)^T$ – коэффициенты, которые удовлетворяют одному или обоим ограничениям:

$$1) \text{ нормировка } \sum_{j=1}^p \alpha_j = 1; \quad (2)$$

$$2) \text{ неотрицательность: } \alpha_j \geq 0, 0 \leq j \leq p. \quad (3)$$

Поиск коэффициентов производится путём минимизации величины среднеквадратического отклонения линейной смеси сигнатур от истинного значения гиперспектрального пикселя:

$$\varepsilon^2 = (v - M\alpha)^T (v - M\alpha) \rightarrow \min_{\alpha}. \quad (4)$$

Методы и алгоритмы решения задачи (4) представлены, как в указанных обзорных работах [1 - 3], так и в специализированных работах [4, 6, 7].

3. АЛГОРИТМ СПЕКТРАЛЬНОГО РАЗЛОЖЕНИЯ С ИСПОЛЬЗОВАНИЕМ КАРТОГРАФИЧЕСКОЙ ОСНОВЫ

Априорная информация:

1) список известных спектральных сигнатур LS размера NS $\{s_i\}_{i=0}^{NS-1}$ (не обязательно полный);

2) список известных типов областей/объектов на цифровой векторной карте LR размера NR ;

3) матрица соответствия δ сигнатур и областей с размером, соответственно, $NS \times NR$. Значение каждого элемента матрицы (i_{LS}, i_{LR}) задается следующим образом:

- $\delta(i_{LS}, i_{LR}) = 2$, если сигнатура i_{LS} может присутствовать в спектральной смеси для области i_{LR} ;
- $\delta(i_{LS}, i_{LR}) = -2$, если сигнатура i_{LS} не присутствует в спектральной смеси для области i_{LR} ;
- $\delta(i_{LS}, i_{LR}) \in (0, 1]$, если сигнатура i_{LS} используется с определённым коэффициентом.

Набор этих данных может храниться в виде базы данных и использоваться для множества задач.

Входные данные:

- 1) геопривязанное гиперспектральное изображение $v(n1, n2)$ некоторого разрешения R , размера $N1 \times N2$;
- 2) «маски» областей/объектов, каждой из которых соответствует индекс из LR . «Маски» областей могут быть, как векторными, так и растровыми, полученными с помощью средств ГИС. В последнем случае разрешение маски должно быть в несколько раз выше, чем R (разрешение входного изображения). Области не могут пересекаться, и размер каждой области должен быть больше, чем размер отсчёта изображения;
- 3) параметр Extend Spectrum. Если его значение "Истина", то список сигнатур при решении задачи может пополняться. Пополненный список будем называть LSE , размер списка – NSE будет $NSE \geq NS$. Если значение этого параметра – "Ложь", то списки LSE и LS совпадают;
- 4) необязательный параметр EPS, предназначенный для остановки процедуры пополнения списка сигнатур.

Считается, что геопривязка для входного изображения выполнена точно. Если данные о спектральных сигнатурах на изображении отсутствуют совсем, для инициализации списка сигнатур предлагается использовать любой из известных методов поиска "чистых" пикселей, например алгоритм N-FINDR [8].

Выходные данные:

- 1) пополненный список сигнатур LSE ;
- 2) NSE – канальное изображение размера $N1 \times N2$, содержащее в каждом отсчете коэффициенты представления соответствующего (по положению) гиперспектрального отсчета исходного изображения в виде спектральной смеси сигнатур LSE . Каждый канал выходного изображения $\lambda_i(n1, n2)$ соответствует набору коэффициентов для сигнатуры с номером i из списка LSE ;
- 3) изображение гиперспектральных остатков $x(n1, n2)$ представления гиперспектрального отсчета исходного изображения линейной смесью спектральных сигнатур.

Описание предлагаемого алгоритма спектрального разложения с использованием картографической основы.

1. Из множества V отсчетов всего изображения отбираются отсчёты, полностью лежащие в областях маски (не на границах). Обозначим эти множества отсчетов $\{V_j\}_{j=0}^{NR-1}$.
2. Для каждого отсчета множества V_j решается задача линейного спектрального разложения (4) при ограничениях

(2) и (3). В результате определяются доли каждой конкретной спектральной компоненты из множества LS , для которой $\delta(i_{LR}, i_{LS})$ допускает присутствие (если доля фиксирована – она в соответствующей системе также заранее фиксируется, а не является решением данной задачи). Результатом решения данной задачи являются значения $\lambda_i(n1, n2)$ для соответствующих индексов i из LS и отсчетов из V_j .

3. Формируется множество $X = \bigcup_{j=0}^{NR-1} V_j$ отсчетов, содержащих гиперспектральные остатки полученного на шаге 2 разложения:

$$x(n1, n2) \equiv v(n1, n2) - \sum_{i=0}^{NS-1} \lambda_i(n1, n2) s_i.$$

4. Если Extend Spectrum – "Истина" выполняются шаги 4.1 – Иначе выполняется переход на шаг 5.

4.1. Для множества X гиперспектральных остатков выполняется процедура поиска "чистых" пикселей с параметром EPS. Если исходный список сигнатур был не полным, то множество остатков будет содержать линейные комбинации недостающих сигнатур. Тогда "чистые" гиперспектральные остатки укажут на спектральные сигнатуры из множества V , которые необходимо внести в список LSE .

4.2. Для множества $\{V_j\}_{j=0}^{NR-1}$ решается задача (4) при ограничениях (2) – (3) для пополненного списка сигнатур. Полученные в результате коэффициенты разложения обозначим $\lambda_i(n1, n2)$, i – индекс из списка спектральных сигнатур.

4.3. Для отсчетов из $\{V_j\}_{j=0}^{NR-1}$ производится перерасчёт значений гиперспектральных остатков по пополненному списку сигнатур.

5. Для каждой пары область-сигнатура определяется закон распределения коэффициентов спектрального разложения $\{\lambda_i(n1, n2)\}_{(n1, n2) \in V_j}$. Обозначим соответствующие законы распределения (плотности вероятности): $\{p_{ij}(\lambda)\}_{i=0, NSE-1, j=0, NR-1}$.

В случае нормального распределения достаточно определить математическое ожидание коэффициентов и дисперсию.

6. Формируется множество пикселей $V^* = V \setminus \bigcup_{j=0}^{NR-1} V_j$, попадающих на границы областей/объектов.

7. Для каждого пикселя $(n1, n2)$ из списка V^* рассчитываются доли площади, которую в нём занимает конкретная область. Обозначаем площади: $\{S_j(n1, n2)\}_{j=0}^{NR-1}$.

Очевидно, должно выполняться ограничение $\sum_{j=0}^{NR-1} S_j(n1, n2) = 1$.

8. Для каждого отсчета $(n1, n2)$ из множества V^* расположенного на границе) с сигнатурой $v(n1, n2)$ определяются значения

$$\lambda_i(n1, n2) = \sum_{j=0}^{NR-1} S_j(n1, n2) \lambda_{ij}(n1, n2)$$

как решения следующей задачи:

$$\begin{cases} \alpha \left(v(n1, n2) - \sum_{i=0}^{NSE-1} \sum_{j=0}^{NR-1} \lambda_{ij}(n1, n2) S_j(n1, n2) s_i \right)^2 - \\ - (1 - \alpha) \sum_{i=0}^{NSE-1} \sum_{j=0}^{NR-1} \ln p_{ij}(\lambda_{ij}(n1, n2)) \rightarrow \min_{\lambda_{ij}(n1, n2) \Big|_{i=0, \dots, NSE-1, j=0, \dots, NR-1}} \\ \lambda_{ij}(n1, n2) \geq 0, \quad i = \overline{0, \dots, NSE-1}, j = \overline{0, \dots, NR-1}; \\ \sum_{i=0}^{NSE-1} \lambda_{ij}(n1, n2) = 1, \quad j = \overline{0, \dots, NR-1}. \end{cases}$$

Здесь $\alpha \in [0, 1]$ – некоторый параметр, характеризующий относительный вес каждого из слагаемых в целевой функции критерия. Для случая, когда распределения вероятностей $\{p_{ij}(\lambda)\}_{i=0, \dots, NSE-1, j=0, \dots, NR-1}$ являются нормальными, приведенный

выше критерий принимает вид:

$$\begin{cases} \alpha \left(v(n1, n2) - \sum_{i=0}^{NSE-1} \sum_{j=0}^{NR-1} \lambda_{ij}(n1, n2) S_j(n1, n2) s_i \right)^2 + \\ + (1 - \alpha) \sum_{i=0}^{NSE-1} \sum_{j=0}^{NR-1} \frac{(\lambda_{ij}(n1, n2) - m_{ij})^2}{\sigma_{ij}^2} \rightarrow \min_{\lambda_{ij}(n1, n2) \Big|_{i=0, \dots, NSE-1, j=0, \dots, NR-1}} \\ \lambda_{ij}(n1, n2) \geq 0, \quad i = \overline{0, \dots, NSE-1}, j = \overline{0, \dots, NR-1}; \\ \sum_{i=0}^{NSE-1} \lambda_{ij}(n1, n2) = 1, \quad j = \overline{0, \dots, NR-1}. \end{cases}$$

9. Для каждого отсчёта, расположенного на границе, производится пересчёт гиперспектральных значений остатков:

$$x(n1, n2) \equiv v(n1, n2) - \sum_{i=0}^{NSE-1} \lambda_i(n1, n2) s_i.$$

Примечание. В качестве процедуры пополнения списка на шаге 4.1 могут выступать любые методы извлечения спектрально чистых элементов, например алгоритм N-FINDR [8]. Величина и смысл параметра EPS определяются конкретным алгоритмом и связаны с остановкой процедуры пополнения списка. В случае использования алгоритма N-FINDR величина EPS характеризует порог, определяющий, в конечном итоге, число отбираемых сигнатур, в качестве которых выступают так называемые "чистые" пиксели. Связь между величиной порога EPS и числом отбираемых сигнатур устанавливается посредством ограничения суммы собственных значений корреляционной матрицы каналов изображения (используется разложение Карунена-Лоэва).

4. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТАЛЬНЫХ ИССЛЕДОВАНИЙ АЛГОРИТМА СПЕКТРАЛЬНОГО РАЗЛОЖЕНИЯ

Для исследования эффективности предложенного алгоритма спектрального разложения использовались синтезированные гиперспектральные изображения с 340 каналами с диапазоном длин волн от 0,8 до 2,5 мкм с шагом 0,005 мкм. Размеры исследуемых изображений составляли 64×64 пикселя. В качестве данных картографической основы использовалась

растровая маска областей размером 512×512. Для генерации тестовых изображений использовались сигнатуры из спектральной библиотеки IGCP-264 Library [5]. Коэффициенты спектральных сигнатур по областям задавались как стационарные случайные поля с биэкспоненциальной корреляционной функцией. При формировании изображения производилась предварительная коррекция значений коэффициентов с учетом ограничения (3) и нормировка коэффициентов в соответствии с ограничением (2). Для получения тестового (обрабатываемого) изображения сначала формировался его детальный прототип, т.е. крупномасштабное изображение с масками областей, из которого тестовое изображение получалось путём усреднения значений гиперспектральных отсчётов прототипа следующим образом:

$$v(n1, n2) = \frac{1}{T^2} \sum_{k_1, k_2=0}^{T-1} v_p(n1 \cdot T + k_1, n2 \cdot T + k_2) \quad (6)$$

где v_p – гиперспектральный отсчет детального изображения-прототипа; v – отсчет обрабатываемого изображения; $T \geq 2$ – отношение линейных размеров детального и обрабатываемого изображений.

Для пополнения списка сигнатур использовался алгоритм N-FINDR [8] с параметром $EPS=10^{-6}$.

Пример тестового изображения и маски областей – на рис. 1.

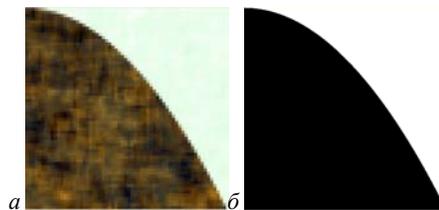


Рис. 1: а – тестовое изображение; б – маска областей

Для расчёта ошибки оценки коэффициентов использовалось следующее выражение:

$$\xi = \frac{1}{|M|} \sum_{(n1, n2) \in M} \frac{1}{NS} \sum_{i=0}^{NS-1} (\lambda_i^b(n1, n2) - \lambda_i(n1, n2))^2,$$

где M – множество отсчетов анализируемой области;

$|M|$ – мощность множества M ; $\lambda_i^b(n1, n2)$ – базовые коэффициенты для сигнатуры с номером i в списке LS ; $\lambda_i(n1, n2)$ – оценка коэффициентов для сигнатуры с номером i в списке LS . В качестве множества M выступало множество целых пикселей V или множество граничных пикселей V^* .

График значений среднеквадратичной ошибки (СКО) коэффициентов приведён на рис. 2. Видно, что коэффициенты разложения восстанавливаются достаточно точно в широком диапазоне значений отношения сигнал/шум (во всех экспериментах используется аддитивный некоррелированный шум с нормальным законом распределения с нулевым средним и заданной дисперсией).

При исследовании качества представления отсчётов исходного изображения при отсутствующей информации о сигнатурах в качестве показателя качества обработки применялась средняя абсолютная ошибка представления гиперспектрального пикселя с помощью полученных коэффициентов:

$$\varepsilon = \frac{1}{N} \sum_{i=0}^{N-1} \bar{\varepsilon}_i, \quad \bar{\varepsilon}_i = \frac{1}{|M|} \sum_{(n1,n2) \in M} \left| v_i(n1,n2) - \sum_{j=0}^{NSE-1} \lambda_j(n1,n2) s_{ji} \right|$$

где N – число спектральных компонент (каналов); $\bar{\varepsilon}_i$ – средняя абсолютная ошибка представления отсчета гиперспектрального изображения в i -м канале; $v_i(n1,n2)$ – i -я спектральная компонента пикселя $(n1,n2)$; s_{ji} – i -я спектральная компонента j -й сигнатуры из списка LSE .

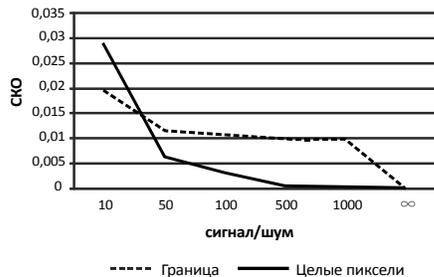


Рис. 2. SKO восстановления коэффициентов

На рис. 3 и 4 приведены, соответственно, значения ε для рассматриваемого тестового изображения в случаях известного списка сигнатур и при отсутствии информации о сигнатурах (используются сигнатуры, извлечённые N-FINDR). Можно заключить, что метод обладает хорошими свойствами помехоустойчивости.

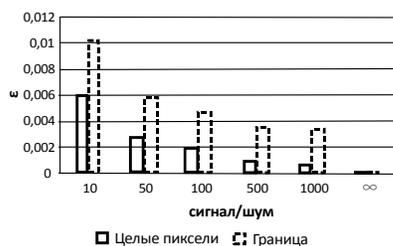


Рис. 3. Средняя абсолютная ошибка представления пикселей для случая с известным списком сигнатур

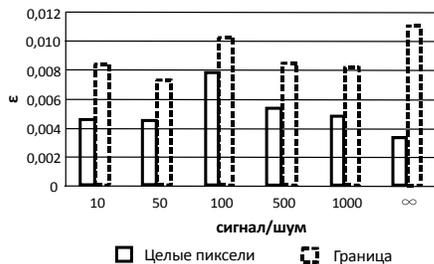


Рис. 4. Средняя абсолютная ошибка представления пикселей для случая с неизвестным списком сигнатур (сигнатуры извлечены NFINDR)

5. ЗАКЛЮЧЕНИЕ

Для представленного алгоритма картографическая основа используется для уточнения коэффициентов спектрального разложения на границах объектов вводом дополнительных ограничений в постановке задачи. Набор используемых спектральных сигнатур может быть, как задан заранее (с неопределёнными или предопределёнными коэффициентами), так и неизвестен, и извлечён в процессе

работы алгоритмов. Указанный факт позволяет относить разработанный алгоритм одновременно к двум классам методов гиперспектрального анализа – к методам управляемого (SLSMA) и неуправляемого (ULSMA) линейного спектрального анализа.

6. БЛАГОДАРНОСТИ

Работа выполнена при государственной поддержке Министерства образования и науки РФ в рамках реализации мероприятий Программы повышения конкурентоспособности СГАУ среди ведущих мировых научно-образовательных центров на 2013 – 2030 гг. А также при частичной финансовой поддержке: грантов РФФИ, проекты № 13-07-12103-офи-м, 13-01-12080-офи-м, 12-07-00021-а, 13-07-97006-р_поволжье_а; программы фундаментальных исследований Президиума РАН «Фундаментальные проблемы информатики и информационных технологий», проект 2.12; Министерства образования и науки Российской Федерации (в рамках постановления Правительства Российской Федерации от 09.04.2010 г. № 218: договор № 02.Г36.31.0001 от 12.02.2013).

7. ССЫЛКИ

- [1] Chang, C.I. Hyperspectral Data Processing: Algorithm Design and Analysis. / C.I. Chang. – John Wiley & Sons, 2013. – 1164 p.
- [2] Chang, C.I. Hyperspectral data exploitation: theory and applications / C.I. Chang. – Wiley-Interscience, 2007. – 456 p.
- [3] Chang, C.I. Hyperspectral imaging: techniques for spectral detection and classification / C.I. Chang. – Springer, 2003. – 370 p.
- [4] Chang, C.I. Constrained subpixel target detection for remotely sensed imagery/ Chang C. I., Heinz D. C. // Geoscience and Remote Sensing, IEEE Transactions on. – 2000. – V. 38 (3). – P. 1144 – 1159.
- [5] Clark, R.N. The U.S. Geological Survey, Digital Spectral Library: Version 1: 0.2 to 3.0 microns, U.S. Geological Survey Open File Report 93-592 / Clark, R.N., G.A. Swayze, A.J. Gallagher, T.V.V. King, W.M. Calvin – 1993. – 1340 p
- [6] Heinz, D.C. Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery/ Heinz D. C., Chang C. I. // Geoscience and Remote Sensing, IEEE Transactions on. – 2001. – V. 39 (3). – P. 529 – 545.
- [7] Мину, М. Математическое программирование: Теория и алгоритмы: Пер. с фр. и предисловие А.И. Штерна./ Мину М. – М.: Наука. Гл. ред. физ.-мат. лит. 1990. – 488 с
- [8] Plaza, A. Fast implementation of pixel purity index algorithm / A. Plaza, C.I. Chang // Proc. of the SPIE conference on Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XI. – 2005. – V. 5806. – P. 307 – 317.

Об авторах

Мясников В. – профессор СГАУ им. С.П. Королёва.

E-mail: vmyas@smr.ru.

Денисова А. – аспирант СГАУ им. С.П. Королёва.

E-mail: denisova_ay@geosamara.ru.

Representation of Linear Segment Voronoi Diagram by Bezier Curves*

L. Mestetskiy

Department of Mathematical Methods of Forecasting
Lomonosov Moscow State University
Moscow, Russia
l.mest@ru.net

Abstract – A new method to describe the Voronoi diagram of a set of line segments is presented. We introduce a new description of a segment Voronoi diagram by a straight control graph of a set of elementary and rational Bezier curves of the first and second degree. Also, the distance between Voronoi edges and line segments called radial function is represented in the same manner by Bezier curves description.

Keywords: Voronoi diagram, Voronoi edges, radial function, parabolic edges, Bezier curves, control graph

I. INTRODUCTION

Voronoi diagram (VD) for line segments is a well-studied and widely used geometric structure (Aurenhammer, 1991, Held, 2001, Karavelas, 2004). Particularly important is its application to the construction of skeletons of polygonal shapes, which is used in the image analysis and recognition (Drysdale, Lee, 1978, Kirkpatrick, 1979). There are known effective $O(n \log n)$ algorithms to construct VD for the general set of linear segments (Fortune, 1987, Yap, 1987) as well as for the sides of a simple polygon (Lee, 1982) or multiply-connected polygonal figures (Mestetskiy, Semenov, 2008).

Geometric construction of a segment VD is simple enough: it is a planar graph with straight-line and parabolic Voronoi edges (Fig. 1). Explicit expression of VD in the form of a geometric graph with coordinates of the vertices and curves of edges is required in many applications. In particular, it is used for the pruning of skeletons, or for the transformation of the object shape through transforms of skeleton and radial function. However, many authors point to the disadvantage of parabolic edges of VD. Parabolic edges are often replaced by piecewise linear polyline. Parabolic edges problem generates the tendency to handle structures having linear edges only. This idea is implemented in the concept of *straight skeleton* (Aichholzer, Aurenhammer, 1996). Sometimes reluctant to work with such edges indicated the primary motivation for the use of the straight skeleton (Tănase, 2004).

But the straight skeleton suffers from certain shortcomings, videlicet: complexity of mathematical definition, low computational efficiency, regularization complexity if noise effects are available.

In this paper, we propose a simple and robust method for segment VD describing in the form of a graph with straight edges.

1. The set of segment VD edges consists of the first and second order elementary and rational Bezier curves. This set we call *the compound Bezier curve* or *Bezier curve graph*.

2. A compound Bezier curve is described by its straight control graph, which is obtained from the dual graph of VD. The control graph consists of control polygons of Bezier curves.

Thus, to describe the segment VD, a straight-line control graph

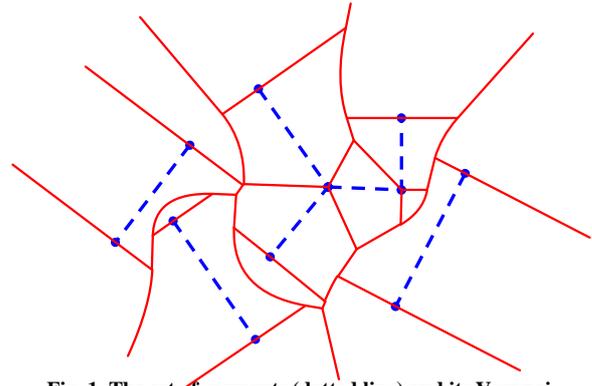


Fig. 1. The set of segments (dotted line) and its Voronoi diagram

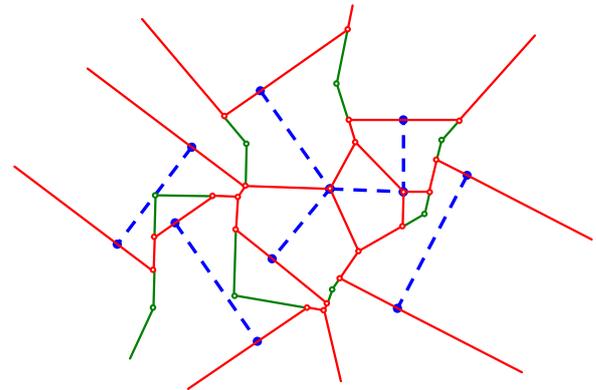


Fig. 2. The control graph of the Voronoi diagram of line segments (dotted line): line Voronoi edges (red) and control polygon edges (green)

is enough (Fig. 2). The set of control graph vertices consists of two subsets. The first subset is formed by Voronoi vertices of segment VD (they are depicted with red). And the second one consists of the certain control points called handles of Bezier curves (green). Straight edges of this graph are control polygons of Bezier curves. Some of them are described by the curve of the first degree (shown in red), and the other part - the curves of the second degree (green).

Usually algorithms for VD form the dual graph explicitly or implicitly. We show how based on this graph you can build a control graph that represents VD.

* The author is grateful to the Russian Foundation of Basic Researches, which has supported this work (grants 14-01-00716).

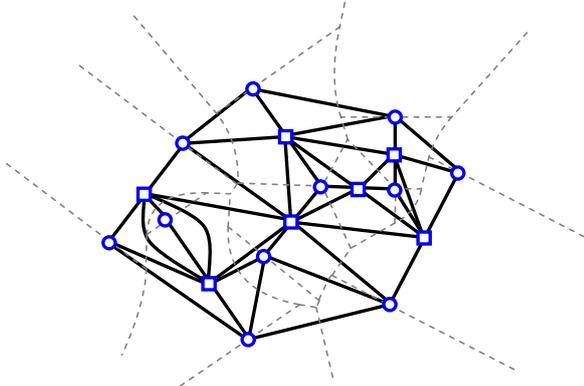


Fig. 3. Dual graph of Voronoi diagram from Fig.1. Vertex-sites depict circles, and segment-sites depict small squares

II. DEFINITIONS

Let's represent a set M of line segments as a finite set S of sites – points and open line segments (called *vertex-site* and *segment-site*). Endpoints of segments from M generate vertex-sites, and the segments without their endpoints generate segment-sites. Vertex-site and segment-site belonging to the same segment, called *neighboring* sites.

Thus, we distinguish three types of Voronoi edges. The first type (straight line) is defined by the pair “vertex-vertex”, the second one (straight line) is defined by the pair “segment - segment” and the third one (parabola) is defined by the pair “vertex - segment”. Every point of VD lies on these lines. Let us use the following terminology for Voronoi edges. There are *vv-bisectors*, *ss-bisectors* and *vs-bisectors* for the pairs of sites “vertex - vertex”, “segment - segment” and “vertex - segment”, respectively.

A *radial function* is determined at every point of VD. Radial function is equal to a euclidean distance between the point and its nearest site. The radial function at Voronoi edge assigns “the width of corridor” between two sites associated with this edge.

Adjacency graph or *Delaunay graph* (DG) of VD is an abstract graph whose set of vertices consists of the sites of VD, and the edge set contains all pairs of sites having adjacent Voronoi cells. DG is the dual graph of VD (Fig. 3).

DG defines the topological structure of VD. Each face of DG determines the Voronoi vertex. Each edge of DG determines the Voronoi edge. For visualization of DG, which is an abstract graph, we use the representation of DG as a geometric graph in the plane (Fig. 3).

We are going to build a representation of VD as a control graph (Fig. 2) based on the DG (Fig. 3), which is assumed to be known.

III. VORONOI VERTICES

The Voronoi vertices are equidistant to three or more sites. To find these vertices tangent circles can be constructed for the triplets of sites. Calculation of such circles involves a number of geometric tasks (Fig.4) related to the following combinations:

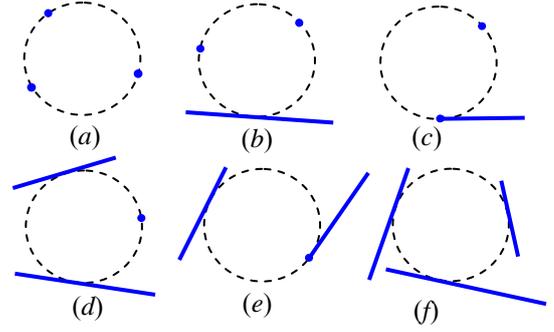


Fig. 4. Tangent circles for the triplets of sites

- three vertex-sites (Fig. 4a);
- two vertex-sites and one segment-site (Fig. 4 b,c);
- two segment-sites and one vertex-site (Fig. 4 d,e);
- three segment-sites (Fig. 4 f).

The second and third combinations involve different cases depending on whether triplet includes neighboring sites.

Assume that the tangent circle exists and the sequence of tangent points is defined. This condition holds for triples of vertices of each face of DG. Then the tangent circle is unique. To compute the center p of the circle tangent three sites

s_1, s_2, s_3 , the following system of equations is to be solved:

$$\begin{cases} d^2(p, s_1) = d^2(p, s_2) \\ d^2(p, s_1) = d^2(p, s_3) \end{cases}$$

In the cases in Fig. 4 a,c,e,f both equations are linear. But in the cases in Fig. 4 b,d one equation is linear, and the other is quadratic. After expressing the Y-coordinate of the point p through the X-coordinate in the linear equation it become possible to reduce the second equation to the usual quadratic equation, which is easily solved.

The obtained solution has to satisfy two auxiliary conditions, which are easily checked. The first condition requires the projections of p onto the segment-sites to lie on these segments themselves. The second condition requires the tangent circle to lie inside the figure. This means the center of tangent circle is required to lie to the left of the segment-site.

IV. VORONOI EDGES AS BEZIER CURVES

Explicit description of the parametric curve $V(t) = (x(t), y(t))$, $t \in [0,1]$ provides handy tools to deal with parabolic Voronoi edges. $V(t)$ determines the edge with Voronoi vertices $V(0)$ and $V(1)$. Also, a parametric representation is convenient to describe the radial function. Pair $\langle V(t), r(t) \rangle$ defines the edge of VD $V(t)$ and its “width” $r(t)$, which is a convenient form for medial representations of objects.

The basic idea is to represent the edges of VD and related radial function through conventional and rational Bezier curves.

First, we consider the representation of linear edges, which are *ss*-bisector. These edges are described by Bézier curves of the first degree

$$V(t) = V_0 B_0^1(t) + V_1 B_1^1(t), \quad t \in [0,1]. \quad (1)$$

Here points V_0, V_1 denote endpoints of bisector. $B_0^1(t) = 1-t$ and $B_1^1(t) = t$ are Bernstein polynomials.

Given the terminal points V_0 and V_1 of a linear *ss*-bisector together with radii r_0 and r_1 of the disks centered at V_0 and V_1 , respectively, we can find the radius of the empty disk centered at any inner point of the edge $V_0 V_1$. It is obvious that the radius of empty disk centered at the point $V(t)$ is $r(t) = r_0 B_0^1(t) + r_1 B_1^1(t)$.

As shown in (Kim, 1995), a parabolic line could be represented as a quadratic Bezier curve

$$V(t) = V_0 B_0^2(t) + V_1 B_1^2(t) + V_2 B_2^2(t), \quad t \in [0,1], \quad (2)$$

where $B_0^2(t) = (1-t)^2, B_1^2(t) = 2t(1-t), B_2^2(t) = t^2$ are Bernstein polynomials. This curve is determined by its control triangle $\{V_0, V_1, V_2\}$. The points V_0 and V_2 are called the endpoints, and the point V_1 is handle point of the Bezier curve.

Using this idea we represent a parabolic bisector of VD as a quadratic Bezier curve. Such a way of edge description is compact and easy-to-use since the only one handle point together with two endpoints defines every edge.

Consequently, in order to obtain *vs*-bisector as the Bezier curve it is necessary to calculate tangent lines at the endpoints of bisector and to find the intersection of these lines. Let us consider the solution of this problem.

Examine Voronoi edge $V_0 V_2$ that is *vs*-bisector for the vertex-site A and the segment-site B (Fig. 5). Let r_0 and r_2 denote the circle radii at the vertices V_0 and V_2 . For definiteness, we assume that when driving on the bisector from V_0 to V_2 the site A is to the left and the site B is to the right of the bisector. We choose a right-handed system, in which the axis Ox is parallel to the segment-site B and is equidistant from the sites A and B , while the axis Oy runs through the site

A perpendicular to the site B . In this system, the following points have the coordinates $V_0 = (x_0, y_0), V_2 = (x_2, y_2), A = (0, c)$. The equation of the line on which B lies is $y = -c$, and the equation of the parabola with focus $A = (0, c)$ and directrix $y = -c$ has the form $x^2 - 4cy = 0$.

Bisector is a segment of a parabola between the points (x_0, r_0) and (x_2, r_2) . It is described as quadratic Bezier curve (2), where V_0 and V_2 are Voronoi vertices, and a handle point of Bezier curve $V_1 = (x_1, r_1)$ is an intersection point of tangents lines of parabola in the end points $V_0 = (x_0, r_0)$ and $V_2 = (x_2, r_2)$. Therefore we can find V_1 through equations:

$$\begin{cases} 2x_0 \cdot x - 4c \cdot y = x_0^2 \\ 2x_2 \cdot x - 4c \cdot y = x_2^2 \end{cases}$$

The solution of the system is

$$x_1 = \frac{1}{2}(x_0 + x_2), \quad y_1 = \frac{1}{4c} x_0 \cdot x_2.$$

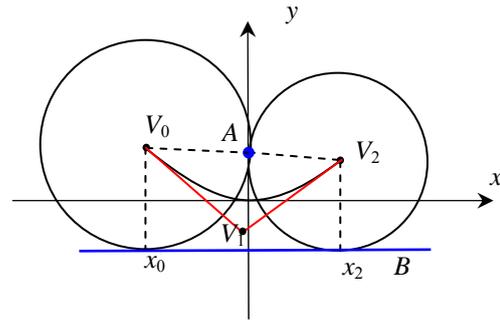


Fig. 5. Parabolic curve for *vs*-bisector

The radial function for *vs*-bisector can also be represented using parametric Bezier spline. In the local coordinate system (Fig. 5) we have simple relation between radii of disks and ordinates of the points of bisector $r(t) = y(t) + c$. From the property $B_0^2(t) + B_1^2(t) + B_2^2(t) = 1$ we obtain

$$\begin{aligned} r(t) &= y_0 B_0^2(t) + y_1 B_1^2(t) + y_2 B_2^2(t) + c = \\ &= (y_0 + c) \cdot B_0^2(t) + (y_1 + c) \cdot B_1^2(t) + (y_2 + c) \cdot B_2^2(t) = \\ &= r_0 \cdot B_0^2(t) + (y_1 + c) \cdot B_1^2(t) + r_2 \cdot B_2^2(t). \end{aligned}$$

If we set $r_1 = y_1 + c$ then

$$r(t) = r_0 B_0^2(t) + r_1 B_1^2(t) + r_2 B_2^2(t). \quad (3)$$

The disk centered at the handle point V_1 with radius r_1 is called a *handle disk*. As it follows from geometric analysis (Fig. 5), r_1 is equal to the distance from the point V_1 to the segment B .

Thus, equations (2) and (3) describe the shape and the radial function of νs -bisector.

Now let us consider the $\nu\nu$ -bisector. We see that the radial function can not be presented by Bezier spline of first degree. Therefore, we will use more complex splines called rational Bezier curves to describe $\nu\nu$ -bisector and its radial function.

Consider the edge V_0V_2 that is $\nu\nu$ -bisector for a couple of vertex-sites A and B . Let r_0 and r_2 the circle radii at the vertices V_0 and V_2 . We choose a right-handed system, which coincides V_0V_2 with the axis Ox and a positive direction is $\overrightarrow{V_0V_2}$ (Fig. 6a). The axis Oy passes through the sites A, B . In this system, the following points have the coordinates $V_0 = (x_0, 0)$, $V_2 = (x_2, 0)$, $A = (0, c)$ and $B = (0, -c)$. If r is the radius of the empty circle centered at the point x , then $r^2 - x^2 = c^2$. In the coordinate system Oxr this equation defines a hyperbola (Fig. 6b). We are interested only in the branch of the hyperbola, for which $r > 0$. As shown in (Kim, 1995) segment of the hyperbola between points (x_0, r_0) and (x_2, r_2) is described by the parametric equations of a rational Bezier curve

$$x(t) = \frac{x_0 \cdot B_0^2(t) + x_1 \cdot w_1 \cdot B_1^2(t) + x_2 \cdot B_2^2(t)}{B_0^2(t) + w_1 \cdot B_1^2(t) + B_2^2(t)}$$

$$r(t) = \frac{r_0 \cdot B_0^2(t) + r_1 \cdot w_1 \cdot B_1^2(t) + r_2 \cdot B_2^2(t)}{B_0^2(t) + w_1 \cdot B_1^2(t) + B_2^2(t)}$$

Here (x_1, r_1) is the control point of the Bezier curve, which is the intersection of the tangents to the hyperbola at the points (x_0, r_0) and (x_2, r_2) , i.e. is determined by solving the system of equations

$$\begin{cases} r_0 \cdot r - x_0 \cdot x = c^2 \\ r_2 \cdot r - x_2 \cdot x = c^2 \end{cases}$$

A weight coefficient w_1 is calculated as $w_1 = \frac{\lambda_1}{2\sqrt{\lambda_0\lambda_2}}$

where $(\lambda_0, \lambda_1, \lambda_2)$ - the barycentric coordinates of the point (x^*, r^*) of hyperbola in the triangle with vertices (x_0, r_0) , (x_1, r_1) , (x_2, r_2) . The point (x^*, r^*) can be selected

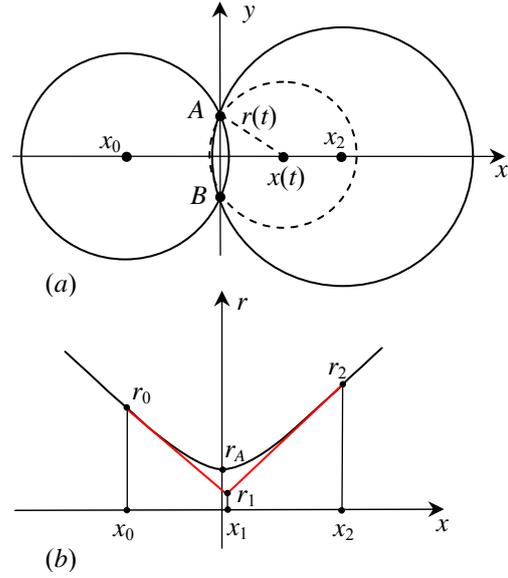


Fig. 6. Hiperbolic curve for νs -bisector

between (x_0, r_0) and (x_2, r_2) , for example, as $x^* = \frac{1}{2}(x_0 + x_2)$, $r^* = \sqrt{c^2 - (x^*)^2}$.

Coordinates $(\lambda_0, \lambda_1, \lambda_2)$ are calculated through the system of equations

$$\begin{cases} x_0 \cdot \lambda_0 + x_1 \cdot \lambda_1 + x_2 \cdot \lambda_2 = x^* \\ r_0 \cdot \lambda_0 + r_1 \cdot \lambda_1 + r_2 \cdot \lambda_2 = r^* \\ \lambda_0 + \lambda_1 + \lambda_2 = 1 \end{cases}$$

These arguments show that on the basis of the computed parameters x_1, r_1, w_1 $\nu\nu$ -bisector can be represented as a rational Bezier curve

$$V(t) = \frac{V_0 \cdot B_0^2(t) + V_1 \cdot w_1 \cdot B_1^2(t) + V_2 \cdot B_2^2(t)}{B_0^2(t) + w_1 \cdot B_1^2(t) + B_2^2(t)}, \quad (4)$$

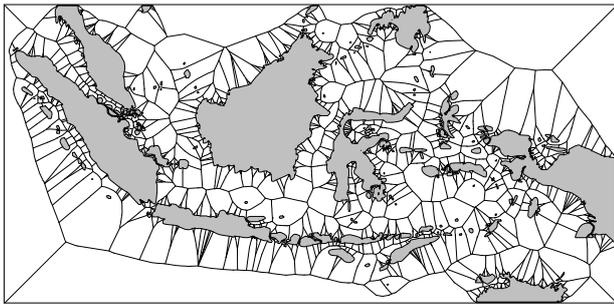
where $V_1 = V_0 \cdot (1 - \mu) + V_2 \cdot \mu$, $\mu = \frac{x_1 - x_0}{x_2 - x_0}$.

The corresponding radial function is defined as follows:

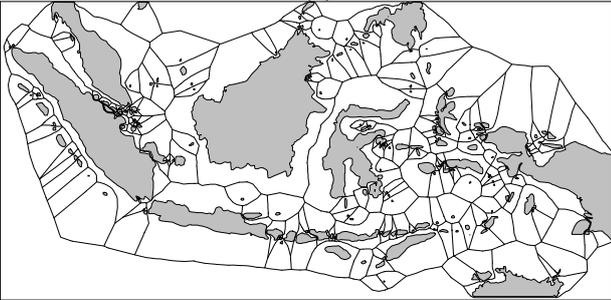
$$r(t) = \frac{r_0 \cdot B_0^2(t) + r_1 \cdot w_1 \cdot B_1^2(t) + r_2 \cdot B_2^2(t)}{B_0^2(t) + w_1 \cdot B_1^2(t) + B_2^2(t)}. \quad (5)$$

V. CONTROL GRAPH OF COMPOUND BEZIER CURVE

Thus, VD is a union of Bezier curves of first and second order. These curves describe a connected geometrical graph. We call this graph the *compound Bezier curve* or *Bezier curve graph*. The segment VD can be represented by a straight-line geometric control graph of compound Bezier curve.



(a)



(b)

Fig. 7. VD of the Malay Archipelago: (a) polygonal figure and its internal skeleton, (b) VD after pruning

The advantage of Bezier curve graph is the ability to describing Bezier curves by their control polygons. Thus compound Bezier curve can be described by a graph composed of polygons elementary Bezier curves. We call it the *control graph of the segment VD*. Fig.2 shows a representation of compound Bezier curve by the control graph. Control graph has straight line edges. The set of control graph vertices includes all Voronoi vertices and handle points of quadratic Bezier curves.

The data structure describing control graph of VD includes feature for each edge type (*ss*-, *vs*-, *vv*-bisector). The edge *vs*-bisector has kept additional parameters: coordinates and radius of control circle (V_1, r_1). The edge *vv*-bisector has kept parameters (x_1, r_1, w_1).

VI. IMPLEMENTATION AND EXPERIMENTS

The proposed method for representing of VD used in the program of constructing of continuous skeletons for binary images, developed by the author (Mestetskiy, 1999, Mestetskiy, Semenov, 2008). The example (Fig. 7) presents the application of the method for constructing VD for a set of polygon-sites.

The source binary image has the size 1271×620 pixels and presents the Malay Archipelago. We need to construct a Voronoi diagram for the islands depicted on the map. First, we approximate this binary image by a multiply connected polygonal figure. We get the bounding box with 190 holes (islands) and 3728 vertices. Second, we get the Voronoi diagram for this polygonal shape and highlight its internal skeleton by pruning (Fig. 7a). And third, we cut all terminal branches of the skeleton and obtain the subgraph, which is VD of islands (Fig.7b). Some characteristics of this example are

presented in the table. The total time for solving the problem, including the polygonal approximation, the construction of the skeleton and pruning is equal to 0.108 sec on the processor Intel 2.13 GHz.

	Skeleton	VD of islands
VD vertices	7822	4979
VD edges	8005	5162
ss-bisector	2822	1137
vs-bisector	3333	2479
vv-bisector	1850	1546

VII. CONCLUSIONS

In this paper, we have presented a new approach to describe the segment VD by stright line control graph of compound Bezier curves. One major advantage is the simplicity of this description. Another advantage is the independence from the algorithm of VD construction. Proposed form of segment VD representation gives the tool for storing and processing VD in geographical databases, computer graphics, and image processing systems.

REFERENCES

- [1] Aichholzer O., Aurenhammer F., 1996. Straight skeletons for general polygonal figures in the plane. *Lecture Notes in Computer Science*. Vol. 1090. Springer-Verlag, 1996.– P. 117 - 126.
- [2] Aurenhammer F., 1991. Voronoi diagrams – a survey of a fundamental geometric data structure. *ACM Computing Surveys*, vol.23, No.3, 1991.– P. 345 - 405.
- [3] Drysdale R., Lee D., 1978. Generalized Voronoi diagrams in the plane. *Proc. 16th Ann. Allerton Conf. Commun. Control Comput.*, 1978. – P. 833 - 842.
- [4] Fortune S., 1987. A sweepline algorithm for Voronoi diagrams. *Algorithmica*, No. 2, 1987. – P. 153 - 174.
- [5] Lee, D., 1982. Medial axis transformation of a planar shape. *IEEE Trans. Pat. Anal. Mach. Int.* PAMI-4(4): 1982. – P. 363 - 369.
- [6] Held M., 2001. VRONI: An engineering approach to reliable and efficient computing of Voronoi diagrams of points and line segments. *Computational Geometry*, 18 (2001). – P. 95 - 123.
- [7] Karavelas M., 2004. A robust and efficient implementation for the segment Voronoi diagram. In *Proceedings of the International Symposium on Voronoi Diagrams in Science and Engineering*, Japan, 2004. – P. 51 - 62.
- [8] Kim D-S., Hwang I-K., and Park B-J., 1995. Representing the Voronoi diagram of a simple polygon using rational quadratic Bezier curves *Computer-Aided Design* 27 (8). – P. 605 - 614.
- [9] Kirkpatrick D., 1979. Efficient computation of continuous skeletons. *Proc. 20th Ann. IEEE Symp. Foundations of Computer Science*, 1979. – P. 18 - 27.
- [10] Mestetskiy L., 1999. Skeletonization of polygonal figures based on the generalized Delaunay triangulation. *Programming and computer software*, vol.25, No.3, 1999. – P. 131 - 142.
- [11] Mestetskiy L., Semenov A., 2008. Binary image skeleton - continuous approach. In *VISAPP'2008, Int. conf. on computer vision theory and applications*, INSTICC Press, vol. 1, 2008. – P. 251 - 258.
- [12] Tănase M., Veltkamp R.C., 2004. A Straight Skeleton Approximating the Medial Axis. LNCS 3221, Springer-Verlag, Berlin Heidelberg, 2004. – P. 809 - 821.
- [13] Yap C., 1987. An $O(n \log n)$ algorithm for the Voronoi diagram of the set of simple curve segments. *Discrete Comput. Geom.*, No. 2, 1987. – P. 365 - 393.

Универсальный метод извлечения скрытых изображений из полиграфических документов

Вл. Сергеев, В. Федосеев
Институт систем обработки изображений РАН,
Самарский государственный аэрокосмический университет им. академика С.П. Королёва
(национальный исследовательский университет),
Самара, Россия
vsereg@geosamara.ru, vicanfed@gmail.com

Аннотация

В данной работе предложен метод извлечения из изображений полиграфических документов скрытых изображений (цифровых водяных знаков), встроенных в регулярные (двумерные квазипериодические) текстуры. Он заключается в спектральном анализе входного изображения, синтезе по результатам этого анализа банка фильтров Габора, многоканальной фильтрации изображения для формирования полей текстурных признаков, а также отборе, группировке и сегментации полей признаков. Приведены примеры моделирования работы предложенного метода на реальных и синтезированных образцах, подтверждающие его работоспособность и потенциальную эффективность.

Ключевые слова: полиграфический документ, регулярная текстура, текстурный анализ, цифровой водяной знак, фильтр Габора, сегментация.

1. ВВЕДЕНИЕ

Для защиты важных полиграфических документов (паспортов, виз, дипломов, билетов и т.д.) от подделок и/или несанкционированного копирования используются различные способы, применяемые на стадии их дизайна. Одним из наиболее распространенных способов является использование текстурных цифровых водяных знаков (ТВЗ), т.е. специальной защитной информации – графических маркеров, меток, изображений, «скрытых» в вариациях каких-либо параметров регулярной (двумерной квазипериодической) текстуры, покрывающей документ или его отдельные части.

Указанный тип текстур представляет собой множество примитивов (текстурных элементов), повторяющихся по полю документа в некотором регулярном или близком к регулярному порядке [6]. ТВЗ может быть встроен посредством изменения шага (периода) примитива, его формы, размера, пространственной ориентации или смещения относительно точной периодичности повторения (фазы). Локальная средняя яркость текстурированной области документа при этом не меняется, чем обеспечивается визуальная неразличимость водяного знака. В данной работе рассматривались следующие виды ТВЗ:

- 1) вариация направления линий;
- 2) вариация масштаба;
- 3) вариация фазы;
- 4) вариация размера точек;
- 5) вариация толщины линий;
- 6) вариация нескольких параметров.

Примеры водяных знаков первых двух типов показаны на рис. 1, а и б. Особенностью третьего типа является возможность его визуального выявления при помощи согласованной с ним оптической решетки (рис. 1 в и г). Более подробно все

упомянутые типы ТВЗ описаны в работах [9, 12]. Большинство из них предназначены для встраивания бинарных изображений, поэтому в данной работе будет рассматриваться именно этот случай.

Важной практической задачей является извлечение скрытых изображений, встроенных одним из рассмотренных методов, в условиях отсутствия достаточной априорной информации о конкретном способе их встраивания в текстурированный документ. Усложняется она тем, что в одной и той же области может быть встроено несколько изображений. Именно эта задача рассматривается нами.

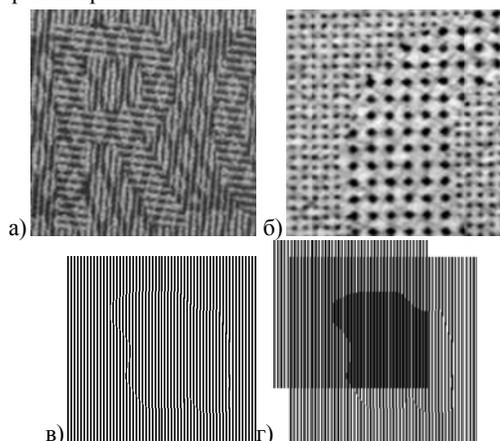


Рис. 1. Увеличенные фрагменты изображений с ТВЗ, встроенных за счёт вариации направления линий (а) масштаба (б) и фазы (в). Извлечение «фазового» ТВЗ с помощью решётки – линзового раstra (г).

2. ВЫБОР ПРИЗНАКОВ

Ввиду того, что в рассматриваемой группе методов встраиваемое изображение является бинарным, задача извлечения ТВЗ фактически – задача сегментации изображения на два класса по текстурным признакам. Если встроенных изображений несколько, то столько же будет и способов сегментации.

В литературе описано множество различных текстурных признаков, используемых в различных задачах текстурного анализа. К их числу среди многих других можно отнести признаки Харалика на основе матрицы вхождений, энергетические признаки Лоуса, локальные бинарные шаблоны (LBP), признаки на основе пирамиды Гаусса и пирамиды Лапласа, а также признаки на основе банков вейвлет-фильтров и габоровских фильтров [8]. Ряд известных работ также посвящён исследованиям разделяющих свойств различных текстурных признаков [2, 5, 7]. Разумеется, в каждой из них исследовались только несколько текстурных признаков и на

ограниченном классе изображений, однако в целом в большинстве случаев наилучшие результаты достигались при помощи признаков Харалика, Лоуса и Габора, причём последние наилучшим образом проявляли себя именно при анализе регулярных текстур.

С другой стороны, структурно-полиграфическое изображение, содержащее встроенный ТВЗ, состоит из двух областей, каждая из которых заполнена периодической или близкой к периодической текстурой, имеющей постоянные параметры (период, направление, форму шаблона). При этом один или несколько из этих параметров различны у двух областей. Таким образом, разумным способом выявления областей на изображении, заполненных каждой из текстур, является узкополосная фильтрация изображения, а подходящим фильтром – фильтр Габора, который помимо оптимальной локализации в пространственной и частотной областях обладает также и рядом других полезных свойств.

Импульсная характеристика фильтра Габора – произведение комплексной экспоненты на функцию Гаусса [4]:

$$g(n_1, n_2) = \frac{1}{2\pi\sigma_1\sigma_2} e^{-\frac{(n_1')^2}{2\sigma_1^2} - \frac{(n_2')^2}{2\sigma_2^2}} e^{2\pi i(n_1\Omega_1 + n_2\Omega_2)}.$$

Параметрами её являются угол θ поворота ядра функции Гаусса $((n_1')_\theta, (n_2')_\theta)$ – пространственные координаты (n_1, n_2) после поворота на угол θ , частота комплексной синусоиды (Ω_1, Ω_2) и пространственные масштабы σ_1 и σ_2 . Ввиду особенностей формирования анализируемых полиграфических документов для решения рассматриваемой задачи допустимо использовать параметры $\sigma_1 = \sigma_2 = \sigma$ и $\theta = 0$. Результатом обработки изображения $x(n_1, n_2)$ фильтром Габора $g(n_1, n_2)$ является комплексное поле.

В большинстве алгоритмов текстурного анализа, использующих фильтры Габора, рассматриваются только амплитудные составляющие откликов [4, 10], однако в предлагаемом методе принято решение использовать также и фазовые составляющие, особенно информативные при извлечении ТВЗ типа «Вариация фазы». Для её использования необходимо осуществить демодуляцию фазового поля с

последующей постобработкой, заключающейся в развёртке фазы. Расчёт выбранных признаков может быть быстро реализован при помощи алгоритмов БПФ или рекурсивного алгоритма [13].

3. ОБЩАЯ СХЕМА МЕТОДА. ФОРМИРОВАНИЕ БАНКА ФИЛЬТРОВ

Предлагаемая метод состоит из следующих последовательно выполняемых этапов:

1. Формирование банка фильтров Габора с импульсными характеристиками $g_k(n_1, n_2)$, $k = 0..N_y - 1$.
2. Линейная фильтрация изображения $x(n_1, n_2)$ каждым из фильтров $g_k(n_1, n_2)$:

$$y_k(n_1, n_2) = x(n_1, n_2) ** g_k(n_1, n_2).$$

3. Формирование $2N_y$ вещественных полей признаков на основе полученных результатов фильтрации.
4. Отбор информативных полей признаков и разделение их на N_z групп, отражающих различные встроенные изображения.
5. Многокомпонентная сегментация на два класса по каждой из групп признаков.

На рис. 2 показана структурная схема метода. Отличия его от известных методов текстурного анализа, использующих банки фильтров Габора [4, 10], заключаются в использовании фазы отклика, наличии стадий отбора и группировки признаков, возможности множественной сегментации, а также в ином методе формирования набора фильтров.

В отличие от традиционных методов, использующих predetermined набор фильтров, в предлагаемом методе применяется адаптивная процедура построения банка фильтров, заключающаяся в поиске периодически повторяющихся пиков на спектре изображения, обусловленных наличием несинусоидальных периодических структур. Частоты фильтров Габора формируемого набора определяются координатами найденных пиков. Подробнее данный метод описан в работе [3].

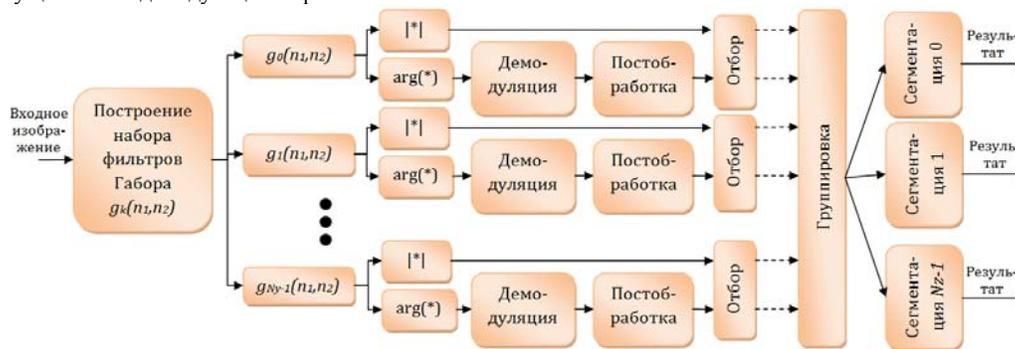


Рис. 2: Общая схема метода извлечения текстурного водяного знака

4. ОТБОР, ГРУППИРОВКА И СЕГМЕНТАЦИЯ ПОЛЕЙ ПРИЗНАКОВ

В результате фильтрации формируются $2N_y$ признаков, причём число N_y не является predetermined, а получено на этапе формирования банка фильтров. В дальнейшем необходимо отбросить неинформативные поля признаков, а оставшиеся разделить на группы, соответствующие разным скрытым изображениям. Обе задачи могут быть легко решены пользователем в интерактивном режиме, однако зачастую при

работе подобной системы желательна полная автоматизация всех процессов.

Информативные амплитудные отклики должны содержать высокие значения в областях текстуры, согласующихся с частотой фильтра Габора, и низкие значения – в остальных областях. Поэтому для проверки их информативности может быть использована сегментация на два класса (например, при помощи EM-алгоритма [1]) с последующим расчётом их характеристик (среднее, выборочная дисперсия, количество элементов) и проверкой ряда ограничений. Так, поле признака

отбрасывается, если максимум амплитуды по всему полю ниже определённого порога, если невелика разница между средними двух классов, если количество элементов одного из них слишком мало и т.д.

Критерием исключения фазовых полей является высокая плотность линий разрыва (линий одной фазы), а также их незамкнутость. В свою очередь, их малое количество и замкнутость свидетельствуют о точности определения частоты текстуры, а значит, об информативности признака. Примеры информативных и неинформативных амплитудных и фазовых признаков показаны на рис. 3 – 4.

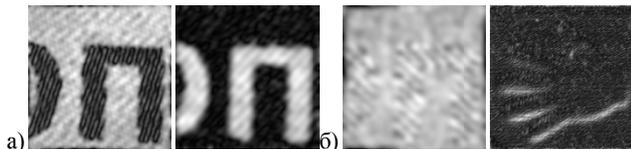


Рис. 3. Примеры информативных (а) и неинформативных (б) амплитудных откликов

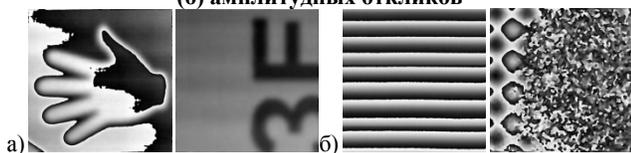


Рис. 4. Примеры информативных (а) и неинформативных (б) фазовых откликов



Рис. 5. Примеры информативных признаков, относящихся к одной группе

Как отмечалось выше, группировка осуществляется для разделения признаков, выявляющих разные скрытые изображения. Отнесение в одну группу нескольких полей признаков, дающих сходные результаты сегментации, может позволить улучшить итоговое качество сегментации по сравнению с однокомпонентной сегментацией. В то же время, наличие информативных признаков, чьи результаты сегментации разительно отличаются, свидетельствует о наличии двух скрытых ТВЗ и необходимости отнесения этих признаков в две разные группы. Таким образом, группировка признаков осуществляется путём сравнения результатов сегментации по каждому из отобранных полей признаков. На рис. 5 показаны примеры двух полей признаков, отнесённых к одной группе.

Отметим также, что итоговая многокомпонентная сегментация должна осуществляться тем же методом, что и однокомпонентная, использовавшаяся при отборе и группировке признаков.

5. ИССЛЕДОВАНИЕ РАБОТОСПОСОБНОСТИ ПРЕДЛОЖЕННОГО МЕТОДА

В данном разделе описаны результаты исследований, подтверждающие высокую разделяющую способность фильтров Габора при решении задачи извлечения ТВЗ и работоспособность процедуры формирования банка фильтров. При этом отбор и группировка признаков осуществлялись вручную. Анализ предложенных выше автоматических

процедур для данных этапов является предметом дальнейших исследований.

5.1. Исследование на реальных образцах

Предложенный метод, в первую очередь, был апробирован на девяти реальных отсканированных образцах изображений полиграфических документов, таких, как паспорт, виза, диплом, лотерейный билет, содержащих один или несколько встроенных водяных знаков. Некоторые примеры изображений и результатов извлечения ТВЗ представлены на рис. 6.



Рис. 6. Примеры реальных образцов (слева) и выделенные ТВЗ (справа)

5.2. Исследование на синтезированных образцах

Исследования на реальных образцах не позволяют получить количественные оценки точности извлечения скрытого изображения ввиду отсутствия оригинала. Поэтому нами было также проведено исследование работоспособности предложенного метода на синтезированных изображениях. Для этого три различных изображения размерами 256×256 пикселей встраивались в различные текстурированные контейнеры каждым из шести рассмотренных методов.

Вслед за этим происходило искажение каждого носителя скрытого изображения при помощи модели печати-сканирования, разработанной ранее в [14] на основе анализа реальных образцов и состоящей в гауссовском размыве и добавлении белого гауссовского шума. Далее осуществлялось извлечение скрытого изображения и сравнение его со встроенным изображением. Для этого использовалось побитовое сравнение двух изображений:

где

—

Полученные результаты сравнивались с результатами аналогичного эксперимента, проведённого с использованием разработанной в 2007 г. при участии авторов системы извлечения текстурных водяных знаков, в которой для извлечения каждого типа ТВЗ использовались отдельные алгоритмы [11] (ниже для краткости будем называть её «Система-2007»). Ввиду этой особенности данная система была плохо расширяемой, а при отсутствии информации о типе ТВЗ требовалось запускать все алгоритмы извлечения. Тем не менее, данная система показала высокие характеристики качества извлечения для каждого вида ТВЗ.

В таблице отражены результаты исследования – усреднённые значения величины TVZ при отсутствии искажений, а также при наибольших допустимых выбранной моделью

искажениях, достигаемых при дисперсии шума 100 и параметре размытия $\sigma = 2.5$. Абсолютные значения ρ , достигаемые с использованием предлагаемого метода, в среднем выше аналогичных значений для «Системы-2007» (на 3.3% при отсутствии искажений, на 2.2% – при наибольших искажениях). При этом, как видно из приведённых данных,

предложенный метод оказался чуть более чувствителен к искажениям носителя информации. Таким образом, полученные результаты позволяют сделать вывод о работоспособности указанного метода и отдать ему предпочтение в сравнении с «Системой-2007» не только по универсальности, но и по качеству извлечения.

Влияние искажений носителя информации на качество извлечения ТВЗ на синтезированных изображениях

Тип ТВЗ	Среднее значение $\rho(W, W^R)$					
	Разработанный метод			«Система-2007»		
	Неискажённые изображения	Наиболее искажённые изображения	Изменение	Неискажённые изображения	Наиболее искажённые изображения	Изменение
Вариация направления линий	0.995	0.992	0.003	0.987	0.973	0.014
Вариация нескольких параметров	0.946	0.961	-0.015	0.910	0.938	-0.028
Вариация толщины линий	0.955	0.900	0.055	0.949	0.958	-0.009
Вариация фазы	0.990	0.974	0.016	0.887	0.795	0.092
Вариация масштаба	0.948	0.916	0.032	0.917	0.881	0.036
Вариация размера точек	0.948	0.855	0.093	0.937	0.919	0.018
Среднее	0.964	0.933	0.031	0.931	0.911	0.021

6. ЗАКЛЮЧЕНИЕ

В работе предложен универсальный метод извлечения скрытых текстурных водяных знаков из полиграфических изображений, базирующийся на спектральном анализе и обработке изображения банком фильтров Габора. Проведены апробация метода на реальных изображениях и исследование качества извлечения информации на синтезированных образцах в сравнении с ранее разработанной системой, подтверждающие его работоспособность.

7. БЛАГОДАРНОСТИ

Работа выполнена при поддержке РФФИ (проекты 12-01-00822, 12-07-00021, 12-07-31056, 13-01-12080, 13-01-97007) и гранта Президента РФ МК-3863.2013.9.

8. ССЫЛКИ

- [1] Dempster A.P., Laird N.M., Rubin D.P. Maximum likelihood from incomplete data via the EM algorithm // J. of the Royal Statistical Society, Series B. – 1977. – Vol. 39, N. 1. – P. 1 - 38.
- [2] Du Buf J.M.H., Kardan M., Spann. M. Texture feature performance for image segmentation // Pattern Recognition. – Elsevier, 1990. – Vol. 23, N. 3 - 4. – P. 291 - 309.
- [3] Fedoseev V.A. Algorithm for estimating the spectral characteristics of regular textures // Proceedings of 11-th International Conference on Pattern Recognition and Image Analysis: New Information Technologies (PRIA-11-2013), September 23 - 28, 2013, Samara, Russia, Vol. 1. – P. 182 - 185.
- [4] Jain A.K., Farokhnia F. Unsupervised Texture Segmentation Using Gabor Filters // Pattern Recognition. – Elsevier, 1991. – Vol. 24, N. 12. – P. 1167 - 1186.
- [5] Koltsov P.P. Comparative Study of Texture Detection // Computational Mathematics and Mathematical Physics. – Springer, 2011. – Vol. 51, N 8. – P. 1460 - 1466.
- [6] Lin, H.-C. Regular-texture image retrieval based on texture-primitive extraction / H.-C. Lin, L.-L. Wang, Sh.-N. Yang // Image and Vision Computing, 1999. – Vol. 17. – P. 51 - 63.
- [7] Ojala T., Pietikäinen M., Harwood D. A comparative study of texture measures with classification based on feature distributions // Pattern Recognition. – Elsevier, 1996. – Vol. 29, N 1. – P. 51 - 59.

- [8] Tuceryan M., Jain A.K. Texture Analysis // Handbook of Pattern Recognition and Computer Vision / book auth. C. H. Chen L. F. Pau, P. S. P. Wang. – World Scientific Publishing Company, 1998. – 2nd ed.
- [9] van Renesse, R.L. Hidden and Scrambled Images - a Review // SPIE Conference on Optical Security and Counterfeit Deterrence Techniques IV, San Jose, CA, January 23 – 25, 2002. – 2002. – P. 333 - 348.
- [10] Weldon T.P. The Design of Multiple Gabor Filters for Segmenting Multiple Textures: Dissertation / Department of Electrical Engineering; The University of North Carolina at Charlotte. – Charlotte, NC, USA, 2007.
- [11] Глумов Н.И., Митекин В.А., Сергеев А.В., Федосеев В.А. Программное обеспечение стегоанализа отсканированных полиграфических изделий // Обозрение прикладной и промышленной математики, 2008. – Т. 6, № 15. – С. 1060 - 1061.
- [12] Митекин В.А., Сергеев А.В., Федосеев В.А. Богомолов Д.М. Построение модели стеганографической системы и обобщенного алгоритма встраивания ЦВЗ в полиграфические изделия // Компьютерная оптика, 2007. – Т 31, № 4. – С. 95 - 100.
- [13] Сергеев В.В., Мясников В.В. Алгоритм быстрой реализации фильтра Габора // Автометрия, 1999. – Т. 6. – С. 51 - 55.
- [14] Федосеев В.А., Митекин В.А. Метод извлечения водяных знаков, базирующихся на вариации размера растровой ячейки // Компьютерная оптика, 2009. – Т. 33, № 4. – С. 460 - 465.

Обавторах

Сергеев Владислав – д.т.н., профессор, зав. кафедрой геоинформатики и информационной безопасности СГАУ и лабораторией математических методов обработки изображений ИСОИРАН.

E-mail: vsereg@geosamara.ru.

Федосеев Виктор – к.ф.-м.н., ассистент кафедры геоинформатики и информационной безопасности СГАУ и научный сотрудник лаборатории математических методов обработки изображений ИСОИРАН.

E-mail: vicanfed@gmail.com.

Быстрый алгоритм совмещения контуров изображений, связанных изотропным аффинным преобразованием

Д. Сунгатуллина, А. Крылов

Лаборатория математических методов обработки изображений

Факультет вычислительной математики и кибернетики

Московский государственный университет им. М.В. Ломоносова, Россия

diana.sungatullina@gmail.com, kryl@cs.msu.ru

Аннотация

В данной работе предложен быстрый алгоритм совмещения изображений, основанный на поиске соответствий между точками контуров шаблона и наблюдения. Метод позволяет находить соответствия между точками шаблона и наблюдения за время $O(M \log M)$, где M – число совмещаемых точек на контуре, практически не ухудшая качество совмещения по сравнению с обычно используемым венгерским алгоритмом. Это достигается благодаря итеративному процессу исключения точек контура, совмещение которых прошло с максимальной ошибкой, и уточнению параметров преобразования для оставшихся точек.

Ключевые слова: совмещение изображений, совмещение контуров, аффинные преобразования, обработка гистологических изображений.

1. ВВЕДЕНИЕ

Совмещение изображений – фундаментальная задача в различных сферах обработки изображений, таких, как реконструирование 3-D сцены, распознавание объектов, построение карт глубины, обработка медицинских и аэрокосмических изображений, где данные, полученные с разных позиций, в различное время или с различных датчиков, должны быть выровнены или сравнены. При совмещении изображений происходит пространственное выравнивание двух изображений сцены так, что для каждой точки первого изображения (наблюдение) ищется преобразование, которое ставит в соответствие точку на втором изображении (шаблон). Большинство существующих методов направлены на поиск параметров линейного преобразования [8], [3] (подобия, аффинного), однако некоторые приложения содержат и нелинейные преобразования [18] (проективное, полиномиальное, эластичное). В нашей работе рассматриваются аффинные преобразования.

Методы совмещения изображений можно разделить на две большие категории: методы, основанные на анализе интенсивности, и методы, основанные на анализе ключевых точек. Методы, основанные на анализе интенсивности, либо оценивают параметры преобразования непосредственно, используя интенсивности в совмещаемых областях [11], [2], либо определяют метрику и находят решение с помощью вычислительно трудоемкой нелинейной оптимизации [5]. Недостатками данных методов являются предположение о малости преобразований, а также проблемы локальных минимумов при оптимизации. Во втором случае на каждом изображении выделяются ключевые точки (замкнутые границы, контуры, пересечения линий, углы и так далее), характеризующие данное изображение [11], [4]. Параметры

преобразования восстанавливаются на основе решения системы уравнений, которая была построена после извлечения ключевых точек и установления соответствий между ними. В качестве дескрипторов ключевых точек могут использоваться SIFT [10], SURF [1], функции Гаусса-Лагерра [12] и другие.

2. СОВМЕЩЕНИЕ ПО КОНТУРАМ

Данный метод относится к классу методов, основанных на анализе ключевых точек, и является модификацией метода, описанного в [15].

Процесс совмещения граничных точек включает в себя три этапа: вычисление дескриптора граничных точек, поиск соответствий между граничными точками и вычисление параметров аффинного преобразования.

2.1. Дескриптор граничных точек

В первую очередь, для каждой точки p замкнутого контура E в плоскости (O, X, Y) устанавливается ортогональная положительно определённая система координат (c, x_p, y_p) , где c – центр масс границы фигуры; x_p – ось, коллинеарная вектору cp ; y_p – ось, ортогональная оси x_p , как показано на рис. 1.

После определения локальной системы координат (c, x_p, y_p) все точки контура E проецируются на ось y_p . Положим v_{min} и v_{max} – минимальное и максимальное значения проекций контура E на ось y_p . Разделим отрезок $[v_{min}, v_{max}]$ на $2N + 1$ часть:

$$\left\{ \begin{array}{l} [v_{min}, v_{min} (1 - \frac{1}{2N})], \\ [v_{min} (1 - \frac{2k-1}{2N}), v_{min} (1 - \frac{2k+1}{2N})], k = \overline{\{1, N-1\}} \\ [v_{min} \frac{1}{2N}, v_{max} \frac{1}{2N}], \\ [v_{max} (1 - \frac{2k+1}{2N}), v_{min} (1 - \frac{2k-1}{2N})], k = \overline{\{1, N-1\}} \\ [v_{max} (1 - \frac{1}{2N}), v_{max}]. \end{array} \right. \quad (1)$$

На каждом из этих отрезков найдем проекции двух точек контура p_k^t и p_k^b , расстояние между которыми максимально. В том случае, если пар точек несколько, выбирается любая. Составим список точек, характеризующих точку p :

$$[x_1^t, x_1^b, x_2^t, x_2^b, \dots, x_{2N+1}^t, x_{2N+1}^b] \quad (2)$$

Поскольку двум разным точкам контура может соответствовать один и тот же список точек, в конец списка также добавим саму точку p . Для каждой точки p контура E по списку точек (2) вычисляется дескриптор DOPM [16]. Опишем его алгоритм.

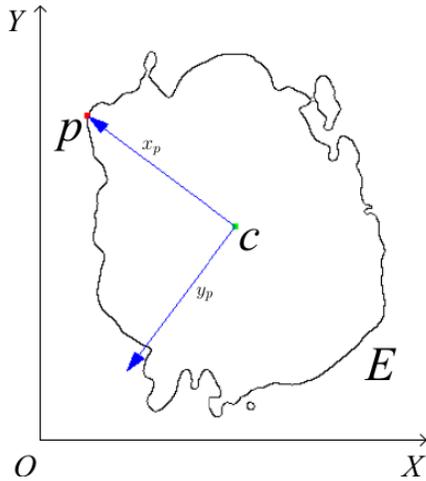


Рис. 1. Выбор локальной системы координат для каждой точки контура

Далее будем полагать, что x и y содержат центрированные координаты. Представим список точек (2) в виде матрицы:

$$(3)$$

где каждая строка (x_i, y_i) – координаты i -й точки в наборе точек (2). В предположении о том, что матрица полного ранга, запишем ее в однородных координатах:

$$(4)$$

Тогда матрица аффинного преобразования A в однородной системе координат имеет вид:

$$(5)$$

где (x_0, y_0) – вектор переноса; A – матрица линейного преобразования. Положим, что точки наблюдения получены из точек шаблона (x_s, y_s) в результате аффинного преобразования A и перестановки P :

$$(6)$$

где P – матрица перестановки, устанавливающая соответствия между точками шаблона и наблюдения. Было показано [17], что если A – матрица ортогональных проекций на пространства столбцов матриц X и Y соответственно, то верно

$$(7)$$

Данное равенство показывает, что матрицы A и P равны с точностью до перестановки. Известно [6], что для матриц, равных с точностью до перестановки, верно:

$$(8)$$

где d_i – диагональные элементы A и P , соответственно. Таким образом, диагонали содержат одни и те же элементы, но в разном порядке. Переупорядочив элементы, например, по убыванию, получим дескриптор, инвариантный к аффинным преобразованиям.

Подробное доказательство аффинной инвариантности полученного дескриптора приведено в работах [15] и [16].

2.2. Поиск соответствий между граничными точками

Пусть P – точка на контуре шаблона, C – точка на контуре наблюдения. Стоимость соответствия двух точек, вычисляется через J -статистику между дескрипторами:

где x_p и y_p – дескрипторы P и C точек, соответственно.

Для набора стоимостей J для всех пар точек контуров шаблона и наблюдения необходимо решить следующую задачу минимизации:

$$(9)$$

Задача (9) является частным случаем задачи о назначениях, классическим методом решения которой является венгерский алгоритм [9] с вычислительной сложностью $O(n^3)$, где n – ранг матрицы J .

В данной работе вместо венгерского алгоритма предлагается использовать более простой жадный алгоритм ближайшего соседа с вычислительной сложностью $O(n^2)$ [13] и итеративно анализировать правильность совмещенных точек контура. Предлагаемый ускоренный метод описан в следующем параграфе.

2.3. Вычисление параметров аффинного преобразования

Внесем координаты сопоставленных точек шаблона и наблюдения в столбцы матриц X и Y , соответственно. Для поиска аффинного преобразования с матрицей A и вектором переноса ограничимся классом изотропных аффинных преобразований, для которых матрицу A можно представить в виде:

$$(10)$$

где α – коэффициент масштабирования; R – матрица поворота.

Параметр α находится из соотношения площадей фигур, а вектор переноса (x_0, y_0) соединяет центры масс контуров шаблона и наблюдения. Таким образом, необходимо найти лишь ортогональную матрицу поворота R . Как и ранее, X и Y содержат центрированные координаты.

Задача поиска оптимальной матрицы поворота в смысле среднеквадратичного отклонения между двумя системами координат по набору точек известна как задача Вахба [14]:

$$(11)$$

Решение задачи (11) находится с помощью алгоритма Кабша [7], для чего вычисляется сингулярное разложение матрицы ковариации:

$$(12)$$

Тогда решение задачи (11) имеет вид:

$$(13)$$

где R – искомая матрица поворота.

Ошибки сопоставления точек шаблона и наблюдения приводят к решению R , сильно отличающемуся от точного решения задачи поиска матрицы поворота. Основная идея предлагаемого метода состоит в следующем: итеративно исключить половину точек, совмещение которых прошло с максимальной ошибкой, повторить алгоритм (10) – (13) для оставшихся координат. Итеративный процесс продолжается пока точность совмещения шаблона и наблюдения растет либо до тех пор, пока не останется одна пара точек.

Для оценки точности совмещения двух изображений используется коэффициент Жаккара:

$$J(x, A^{-1}y) = \frac{|x \cap A^{-1}y|}{|x \cup A^{-1}y|}, \quad (14)$$

где x – шаблон; y – наблюдение; A^{-1} – искомое преобразование. Как правило, для достижения точности совмещения более 0.9 достаточно трех итераций.

3. РЕЗУЛЬТАТЫ

Тестирование метода производилось на базе гистологических образцов, предоставленных РНЦХ имени академика Б.В. Петровского. Примеры работы предложенного метода совмещения при $N = 9$ представлены на рис. 2 – 3. Характерные результаты сравнительного анализа разработанного и существующих методов для гистологических изображений приведены в таблице. Тестирование показало практическую эффективность предложенного метода.

Метод	Время	Точность
ДОПМ + венгерский [15]	$O(M^3)$	0.9435
Метод моментов [2]	$O(M)$	0.8572
Предложенный	$O(M \log M)$	0.9429

Сравнение методов совмещения контуров изображений

4. ЗАКЛЮЧЕНИЕ

Разработан быстрый алгоритм совмещения медицинских изображений образцов тканей, основанный на поиске соответствий между точками контуров шаблона и наблюдения. Задача поиска соответствий известна как задача о назначениях, классическим методом решения которой является венгерский алгоритм, с вычислительной сложностью $O(M^3)$, где M – количество точек контура шаблона или наблюдения. Предложенный метод позволяет находить соответствия между точками шаблона и наблюдения за время $O(M \log M)$, практически не ухудшая качество совмещения, благодаря итеративному процессу исключения точек, совмещение которых прошло с максимальной ошибкой, и уточнения параметров преобразования для оставшихся точек.

5. БЛАГОДАРНОСТИ

Работа выполнена при поддержке гранта РНФ 14-11-00308.

6. ССЫЛКИ

- [1] Bay H., Ess A., Tuytelaars T., Gool L. SURF: Speeded up robust features, *Computer Vision and Image Understanding*, vol. 110, no. 3, 2008. – P.346 – 359.
- [2] Domokos C., Nemeth J., Kato Z. Nonlinear shape registration without correspondences, *IEEE Transactions on pattern analysis and machine intelligence*, vol. 34, no. 5, 2012. – P. 943 - 958.
- [3] Feldmar J., Ayache N. Rigid, affine and locally affine registration of free form surfaces, *International Journal of Computer Vision*, vol.18, no. 2, 1996. – P. 99 – 119.
- [4] Guo H., Rangarajan A., Joshi S., Younes L. Non-rigid registration of shapes via diffeomorphic point matching, *Proceedings of the IEEE International Symposium on*

Biomedical Imaging: From Nano to Macro, 2004. – P. 924 - 927.

- [5] Hansen M.S., Hansen M.F., Larsen R. Diffeomorphic statistical deformation models”, *Proceedings of the IEEE International Conference on Computer Vision*, 2007. – P. 1 - 8.
- [6] Horn R., Johnson C. Matrix analysis Cambridge, U.K.: Cambridge University Press, 1985.
- [7] Kabsch W. A solution for the best rotation to relate two sets of vectors, *Acta Crystallographica Section A*, vol. 32, no. 5, 1976. – P. 922 - 923.
- [8] Kadyrov A., Petrou M. Affine parameter estimation for the trace transform, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, 2006. – P. 1631 - 1645.
- [9] Kuhn H. The Hungarian Method for the assignment problem, *Naval Research Logistics Quarterly*, vol. 2, no. 1 - 2, 1955. – P. 83 - 97.
- [10] D.G. Low, “Distinctive image features from scale-invariant keypoints”, *International Journal of Computer Vision*, vol.60, pp.91-110, 2004.
- [11] Mann S., Picard R.W. Video orbits of the projective group a simple approach to featureless estimation of parameters, *IEEE Transactions on Image Processing*, vol. 6, no. 9, 1997. – P. 1281 - 1295.
- [12] Sorokin D., Krylov A. Gauss-Laguerre-Hermite methods of keypoint extraction, *Pattern Recognition and Image Analysis*, vol. 21, no. 2, 2011. – P. 332 - 334.
- [13] Vaidya P. An $O(n \log n)$ algorithm for the all-nearest-neighbors problem, *Discrete and Computational Geometry journal*, vol. 4, no. 1, 1989. – P. 101 - 115.
- [14] Wahba G. A Least Squares Estimate of Satellite Attitude, *SIAM Review*, vol. 7, no. 3, 1965. – P. 409 – 411.
- [15] Wang W., Jiang Y., Xiong B. and others. Contour matching using the-affine-invariant support point set, *IET Computer Vision*, vol. 8, no. 1, 2014. – P. 35 – 44.
- [16] Wang Z., Liang M. Locally affine invariant descriptors for shape matching and retrieval, *IEEE Signal Processing Letters*, vol. 17, no. 9, 2010.
- [17] Wang Z., Xia H. Dimension-free affine shape matching through subspace invariance, *IEEE Conf. Computer Vision and Pattern Recognition*, 2009. – P. 357 - 364.
- [18] Zagorchev L., Goshtasby A. A comparative study of transformation functions for nonrigid image registration, *IEEE Transactions on Image Processing*, vol. 15, no. 3, 2006. – P. 529 - 538,
- [19] Zitova B., Flusser J. Image registration methods: a survey, *Image and Vision Computing*, vol. 21, 2003. – P. 977 – 1000.

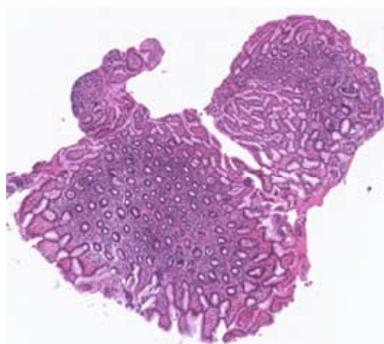
Об авторах

Сунгатуллина Диана – аспирантка кафедры математической физики факультета вычислительной математики и кибернетики Московского государственного университета имени М.В. Ломоносова.

E-mail: diana.sungatullina@gmail.com

Крылов Андрей – профессор, заведующий лабораторией математических методов обработки изображений факультета вычислительной математики и кибернетики Московского государственного университета имени М.В. Ломоносова.

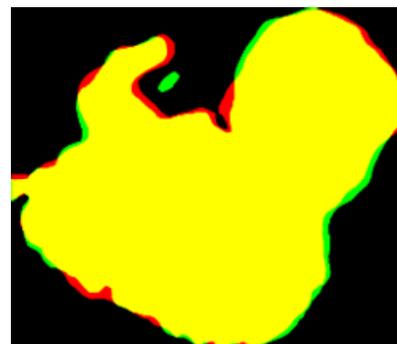
E-mail: kryl@cs.msu.ru



а) шаблон

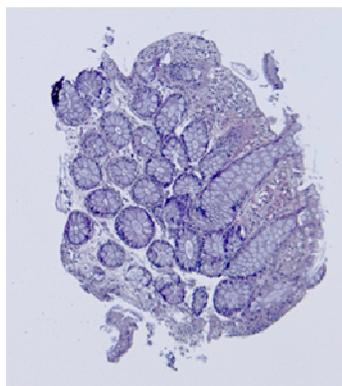


б) наблюдение

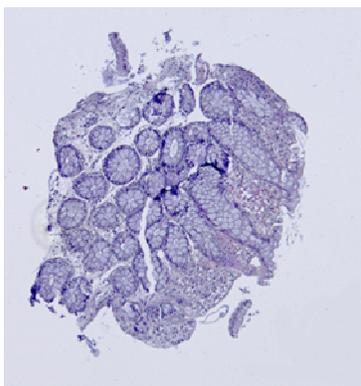


в) $J = 0,9363$

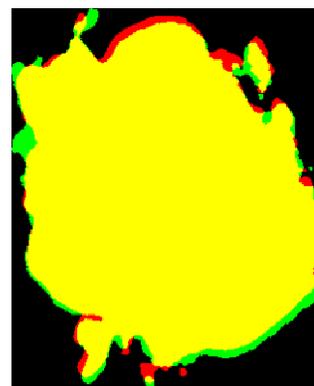
Рис. 2. Результаты работы программы по совмещению гистологических изображений по контурам: красная (темно-серая) фигура – шаблон ; зеленая (светло-серая) – наблюдение , к которому применили обратное преобразование ; желтая (белая) область – пересечение и .



а) шаблон



б) наблюдение



в) $J = 0,9214$

Рис. 3. Результаты алгоритма совмещения срезов тканей по контурам: красным (темно-серым) цветом обозначен шаблон ; зеленым (светло-серым) – наблюдение , к которому применили обратное преобразование ; желтая (белая) область – пересечение и .

Оценка состояния венул по MPT SWI изображениям с применением масок белого и серого вещества

И. Архипов¹, А. Ятченко², А. Гаврилов³, И. Куликов³, И. Кротенкова⁴, В. Брюхов⁴, Р. Коновалов⁴, А. Суслин⁴
¹РНЦХ РАМН, ²факультет вычислительной математики и кибернетики, ³НИИЯФ МГУ им. М.В. Ломоносова, ⁴Отделение лучевой диагностики ФГБУ «НЦН» РАМН, Москва, Россия

Аннотация

В данной статье предлагается метод, позволяющий определить объем венул на SWI изображениях. Метод включает в себя выделение вокселей венул, а также метод постобработки выделенных структур для избавления от шума. Приведен результат работы алгоритма в области интереса – в белом веществе головного мозга.

Ключевые слова: MPT, SWI, сегментация сосудов, скелетизация.

1. ВВЕДЕНИЕ

SWI (Susceptibility Weighted Imaging), или изображения, взвешенные по магнитной восприимчивости – это относительно новая импульсная последовательность магнитно-резонансной томографии (МРТ), впервые описанная Нааске и коллегами в 2004 г. [4]. В качестве контрастного агента используется магнитная восприимчивость таких веществ, как деоксигемоглобин, железо и кальций. Таким образом, МРТ в режиме SWI позволяет визуализировать следы мелких кровоизлияний по истечении длительного времени, отложения железосодержащих веществ в отдельных структурах головного мозга, а также хорошо контрастировать венозную кровь. Вышеописанные характеристики режима SWI позволяют применять его в диагностике различных неврологических заболеваний [1].

В последнее время широкое распространение приобрели исследования, направленные на оценку венозной системы головного мозга при рассеянном склерозе (РС) [2], [3]. В режиме SWI у пациентов с РС отмечается выраженное снижение контрастности между венозными сосудами и веществом головного мозга. Причина таких изменений до конца неясна, но вероятно, это говорит о снижении концентрации деоксигемоглобина в венозной крови. Таким образом, определение общего объема вен головного мозга может быть маркером в определении обширного гипометаболического статуса пациента.

Для лучшей визуализации венул врачи-диагносты используют технологию проекции минимальной интенсивности, при которой воксели изображений в достаточно толстом слое – 10 – 15 мм проецируются на плоскость по правилу минимального значения. Такие изображения дают хорошее представление о количестве венул в головном мозге, о перфузии тех областей, которые омываются данными сосудами. Пример такого изображения показан на рисунке 1 (а). Режим регистрации МРТ SWI позволяет наблюдать артериальные и венозные сосуды. Артериальные сосуды на таких изображениях регистрируются как светлые продолговатые структуры, а венозные сосуды – как темные продолговатые структуры (рис. 1 (b)).

Однако данный метод дает только качественное представление о состоянии венул. И врачу сложно оценить, попадает ли визуализированная венула в белое или серое вещество мозга. Для количественной оценки состояния пациента требуется оценить объем венозного русла в белом веществе мозга. В данной статье предлагается метод,



Рис. 1: (а) - SWI снимок; (b) - артерии и венулы на фрагменте изображения

позволяющий определить объем венул на SWI изображениях. Метод включает в себя выделение вокселей сосудов при помощи алгоритма, описанного в [8], а также метод постобработки выделенных вокселей для избавления от шума. Приведен результат работы алгоритма в области интереса – в белом веществе головного мозга.

2. МЕТОД

МРТ изображения на сегодняшний день редко имеют хорошую разрешающую способность для анализа мелких структур, поскольку врач-диагност всегда стоит перед выбором между качеством получаемых изображений и временем, которое занимает исследование. Исходные срезы, на которых описан метод, имеют разрешение 0.5мм по осям X, Y и 1.2 мм – по оси Z – такое разрешение является компромиссом для рутинного исследования головы на современном томографе.

Структуры, которые следует оценить, имеют размер, сравнимый с разрешением по оси Z, поскольку венулы имеют диаметр 0.5 – 3.5 мм. Для того чтобы такие мелкие структуры не сильно страдали от низкого разрешения по оси Z, исходные данные интерполируются с помощью метода Non-Local upsampling [5], приводят данные к анизотропному кубу с разрешением 0.5 мм по всем осям (рис. 2)

Для сегментации белого вещества головного мозга использовалась реализация в пакете SPM для неврологических исследований. Для этого исходные данные сегментируются по порогу для отсеивания областей, содержащих воздух и кости. Далее полученная область обрабатывается морфологическими операциями. Результат сегментации белого вещества загружается в виде NIFTY данных (пример результата сегментации белого вещества см. на рис. 3)

Для сегментации сосудов используется тот факт, что для областей в окрестности сосудов ярко выражено одно направление в матрице Гессе. Для определения таких вокселей был использован метод, описанный в [8].

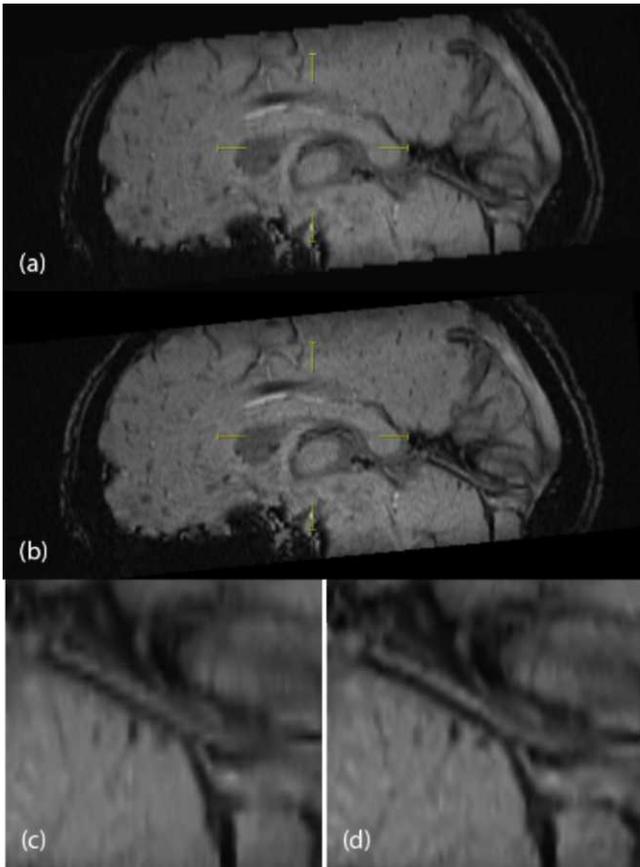


Рис. 2: (a), (c) – линейная интерполяция;
(b), (d) – NonLocal upsemping

На рис. 4 отображен результат выделения вокселей в зоне интереса. Для расчета использовались следующие параметры: $\sigma = 1.0\text{мм}$, $\alpha = 0.5$, $\alpha_2 = 2.0$.

Так как исходное разрешение изображений по оси Z сравнимо с размером структур, которые нам необходимо сегментировать, то при выделении трубчатых структур порог сегментации необходимо выбирать достаточно низким, чтобы не пропустить лишние воксели. В результате, помимо сосудов, в сегментацию попадает достаточно большое количество шума.

Сегментированная область содержит в себе большое количество односвязных областей. После сегментации внутри белого вещества мозга таких областей ~ 3000 . Объем таких структур сильно колеблется от структур в несколько вокселей до структур с объемом 10000 вокселей ($\sim 1\text{см}^3$).

Эксперименты показали, что чистка шума по порогу объема дает неприемлемый результат, поскольку сосудистые структуры могут иметь разрывы. В текущей работе предложен алгоритм, убирающий шум и при этом оставляющий небольшие структуры на концах венул.

3. АЛГОРИТМ СОЕДИНЕНИЯ СТРУКТУР

Первым этапом все найденные участки разбиваются по признаку связности. Для того чтобы формализовать дерево сосудов каждый участок скелетонируется. Для скелетизации используется связность по вершинам. Скелетизация в трехмерном пространстве довольно сложная задача. Существует множество алгоритмов скелетизации в двухмерном пространстве, однако для 3D их не так много [6], [7], [9]. В данной реализации для определения скелетов выделенных структур используется алгоритм [6]. Скелет каждого односвязного участка представляется в виде направленного простого графа-дерева. Скелет может

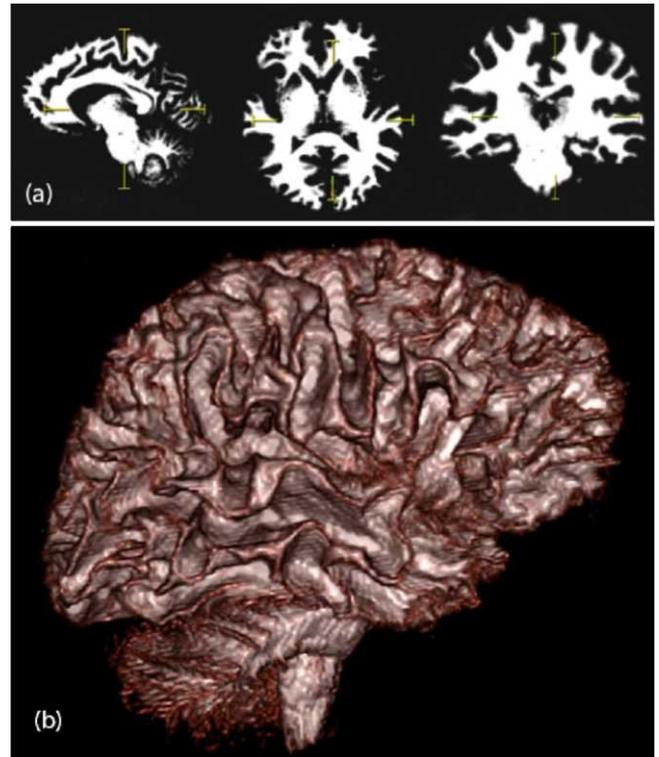


Рис. 3: (a) – сегментированное белое вещество головного мозга; (b) – 3D визуализация

содержать небольшие отростки, которые появляются из-за неровности поверхности сосуда. Поэтому после скелетизации производится стрижка графа, при которой удаляются все отростки, длина которых меньше определенного порога L_{cut} . Вершины графов, из которых выходит ровно одно ребро, будем называть крайними, или листьями. После чистки производится соединение отдельных графов в одну структуру. Для этого между некоторыми вершинами конечных ветвей графов добавляются ещё ребра, что позволяет исправить разрывы структуры венозного русла. Для каждого листа графа оценивается, возможно ли определить касательную к участку скелета, соответствующему ребру, выходящему из этой вершины. Для этого введем величину L_0 – это величина длины скелета, при которой мы считаем, что на концах участка мы достоверно можем определить направление участка сосуда. Если длина скелета меньше L_0 , то лист мы добавляем в группу сомнительных листов.

Для определения касательных в начале и конце скелета определяется прямая, соединяющая лист и узел графа, который отстоит на L_0 вокселей по скелету. Далее вводится величина L_w , которая характеризует, на каком расстоянии от конца скелета мы можем соединять сомнительные участки. Вводится величина U , которая характеризует, насколько может отклониться касательная к скелету участка при экстраполяции к концу сомнительного участка. Для каждого участка производим поиск ближайшего участка из группы сомнительных. Соединяем концы найденных скелетов, удовлетворяющие условиям выше и помечаем все воксели, которые попадают на прямую как воксели венозного русла. Сомнительные участки, скелеты которых ни с чем не соединены, помечаются как шум и исключаются из сегментации.

Для того чтобы из анализа исключить ложные структуры, участки, объем которых превышает значение V_{max} , исключаются из анализа. Также исключаются те структуры, касательные к скелету которых направлены вдоль границы исходной области интереса

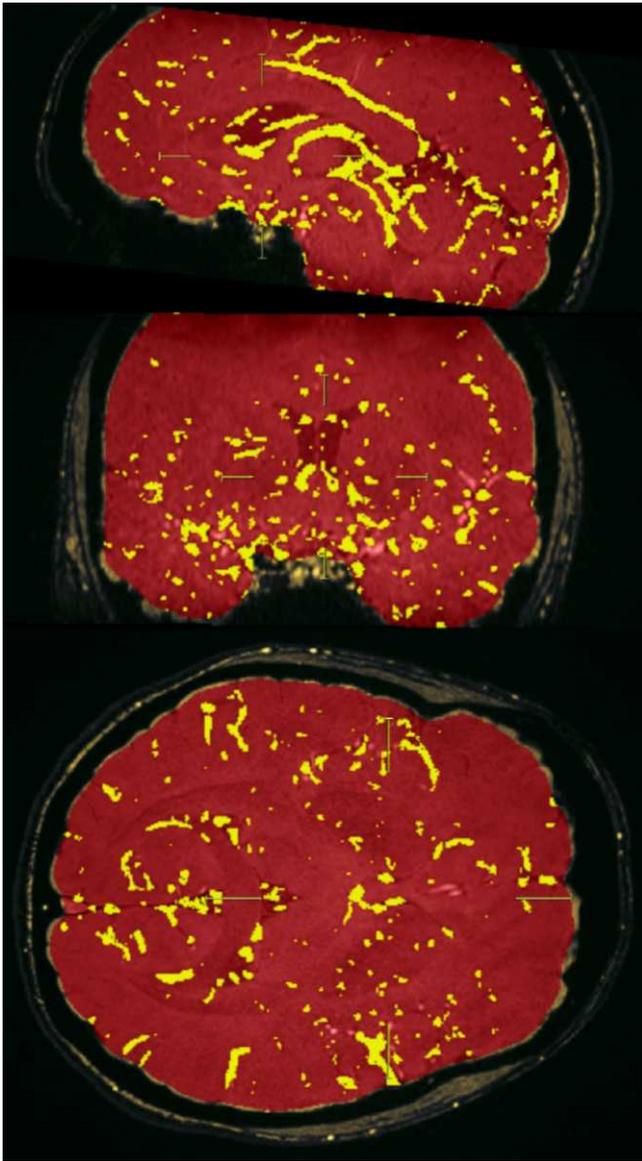


Рис. 4. Результат сегментации венул

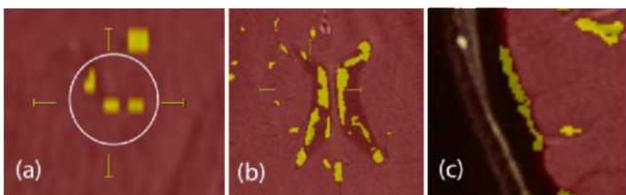


Рис. 5: (а) – разрыв структуры сосуда; (b), (с) – ошибочно сегментированные участки

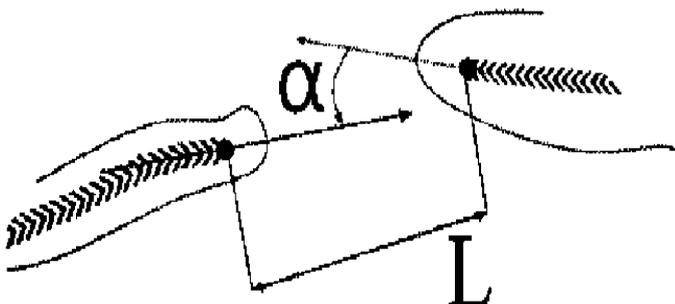


Рис. 6. Разрыв структуры сосуда

Заметим, что использование сегментированных объектов белого и серого вещества как области интереса значительно уменьшает количество ложных структур, которые попадают в результирующий объем венозного русла.

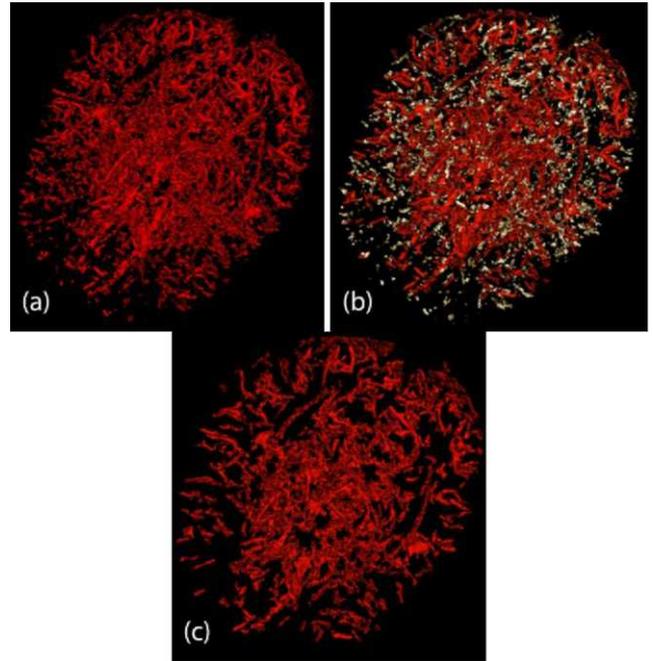


Рис. 7: (а) – сегментированное сосудистое русло; (b) – ярким цветом выделены области, детектированные как шум; (с) – сегментированное сосудистое русло с убранными шумами

4. РЕЗУЛЬТАТЫ

Эксперимент был проведен на МРТ серии, полученной из НИИ неврологии. Серия содержит 72 среза. При помощи SPM пакета для обработки неврологических данных сотрудниками НИИ неврологии была построена маска белого и серого вещества мозга. Маска из SPM экспортируется как файл с данными сегментаций NIFTI. В белом веществе были определены трубчатые структуры. При сегментировании сосудов использовались параметры $\sigma = 1.0\text{мм}$, $\alpha_1 = 0.5$, $\alpha_2 = 2.0$. Затем для каждой односвязной области был построен простой граф. Во время предварительной чистки были удалены те области, у которых длина скелета не превосходит пороговой ($L_{cut} = 5\text{ мм}$) и рядом с листьями графа которого нет листьев графов скелетов из соседних областей ($\sim 15\text{ мм}$).

Для чистки сегментированного сосудистого русла от шумов использовались следующие параметры: $L_0 = 3\text{ мм}$, $L_w = 2.5\text{ мм}$, $U = 35^\circ$ и $V_{max} = 1.5\text{ см}^3$.

Результат сегментации венозного русла в белом веществе до и после чистки приведен на рис. 7.

Вклад в объем отсеянных областей составляет величину около $\sim 25\%$ (с отсеиванием 14.2 см^3 , без отсеивания 20.5 см^3); это говорит о том, что пост-обработка сегментированных областей для оценки объема является необходимым шагом. В результате был разработан метод количественной оценки объема вен на SWI МРТ изображениях. Метод был реализован как модуль к рабочей станции MultiVox, применяемой для просмотра, 3D визуализации сегментированных областей.

5. ССЫЛКИ

- [1] Gasparotti R., Pinelli L., Liserre R. New MR sequences in daily practice: susceptibility weighted imaging. A pictorial essay // *Insights Imaging* Vol. 2(3), 2011. – P. 335 – 347.
- [2] Ge Y., Zohrabian VM., Robert I. Diminished visibility of cerebral venous vasculature in multiple sclerosis by susceptibility-weighted imaging at 3.0 T. // *Journal of magnetic resonance imaging* Vol. 29(5), 2009. – P. 1190 - 1194,
- [3] Haacke EM., Makki M., Ge Y., Maheshwari M., Sehgal V., Hu J., M Selvan, Z Wu, Z Latif, Y Xuan, O Khan, J Garbern, Robert I. Characterizing iron deposition in multiple sclerosis lesions using susceptibility weighted imaging // *Journal of magnetic resonance imaging*, Vol. 29(3), 2009. – P. 537 - 544,.
- [4] Haacke EM., Xu Y., Cheng YN., JR Reichenbach. Susceptibility weighted imaging (SWI) // *Magnetic Resonance in Medicine*, Vol.52(3), 2004. – P. 612 - 618.
- [5] Manjon JV., Coupe P., Buades A., Fonov V., Collins DL., Robles M. Non-Local MRI Upsampling // *Medical Image Analysis*, Vol. 14(6), 2010. – P. 784 - 92.
- [6] Manzanera A., Bernard TM. N-dimensional skeletonization: a unified mathematical framework // *J. of Electronic Imaging*, Vol. 11, 2002. – P. 25 - 37.
- [7] Mian P., Gisela K. A revision of a 3D Skeletonization algorithm // *CITR, The University of Auckland, New Zealand, Research Tech. Rep.* 143, 2004.
- [8] Sato Y., Nakajima S., Atsumi H., Koller T., Gerig G., Yoshida S., Kikinis R. 3D multi-scale line filter for segmentation and visualization of curvilinear structures in medical images // *Lecture Notes in Computer Science* Vol. 1205, 1997. – P. 213 – 222.
- [9] Ятченко А.М., Крылов А.С., Гаврилов А.В., Архипов И.В. Построение 3D модели кровеносных сосудов по серии КТ изображений печени // *ГрафиКон'2009*. – М., 2009. – С. 344 - 347.

Об авторах

Архипов Иван – ведущий программист РНЦХ РАМН.
E-mail: arkhivania@gmail.com

Ятченко Артем – к.ф.-м.н. н.с. лаб. мат. методов обработки изображений ф-та ВМиК МГУ.
E-mail: artyom@yatchenko.com.ua

Гаврилов Андрей – к.т.н. зав. лаб. мед. комп. систем отдела микроэлектроники НИИЯФ МГУ.
email: agavrilov49@gmail.com

Куликов Игорь – программист лаб. мед. комп. систем отдела микроэлектроники НИИЯФ МГУ.
E-mail: igor@multivox.ru

Кротенкова Ирина – аспирант отделения лучевой диагностики ФГБУ «НЦН» РАМН.
E-mail: irina.krotenkova@mail.ru

Брюхов В. – к.м.н., н.с. отделения лучевой диагностики ФГБУ «НЦН» РАМН.
E-mail: in-ray@yandex.ru

Коновалов Родион – с.н.с. отделения лучевой диагностики ФГБУ «НЦН» РАМН.
E-mail: krn74@mail.ru

Суслин Александр – н.с. отделения лучевой диагностики ФГБУ «НЦН» РАМН.
E-mail: in-ray@yandex.ru

ABSTRACT

The method for venule volume determination in the human brain have been proposed. This method uses SWI MRI images. Метод включает в себя выделение вокселей венул, а так же метод постобработки выделенных структур для избавления от шума. Также в статье приведен результат работы алгоритма в области интереса - в белом веществе головного мозга.

SAR Image Processing for Sea Surface Monitoring

B. Alpatov, O. Balashov, M. Ershov, V. Muraviev, A. Feldman
 Department of Automation and Information Technologies in Control
 Ryazan State Radio Engineering University, Ryazan, Russia
 aitu@rsreu.ru

Abstract

Processing of SAR images are extensively used for control and monitoring of sea surface. Image data can be acquired from Earth observation satellites, such as TerraSAR-X, ERS, COSMO-SkyMed. Key areas of research include following the most relevant: detection of oil spills, detection of surface objects of artificial origin (ships and other vessels) and mapping of sea currents speed.

The paper describes several complex algorithms that solve these problems. The results of computer modeling on real SAR images are presented. Based on these results it is concluded that the proposed approaches can be used in practice.

Keywords: SAR image, ship detection, oil spill detection, sea currents mapping.

1. INTRODUCTION

It is known that about two-thirds of the Earth's surface is covered by water. Humanity is actively using ocean for freight shipping. Therefore, the tasks related to the monitoring of water surface are relevant. These tasks include ship detection, oil pollution control and sea currents research. Due to the fact that water surface covers wide areas, remote sensing is the most successful way of getting information about the ocean environment. Currently, the satellites equipped with synthetic aperture radar (SAR) antennas have received wide prevalence in solution of problems of water surface monitoring (TerraSAR-X, ERS-2, COSMO-SkyMed). The radar remote sensing satellites can obtain images of the ocean surface in any weather and any time.

2. OIL SPILL DETECTION

Mass use of petroleum products around the world creates preconditions for increasing volumes of production of crude oil and transportation of petroleum products. This increases the probability of water surface pollution. Problem of ecological monitoring of marine environment gets more acuity; thereby the development of oil spill detection algorithm is an important task [7].

2.1. Radar imaging and oil slicks

Radio waves are sensitive to the state of the sea surface because wind waves are present on it. Reflection mechanism of radio waves from such surface is described by the Bragg scattering [8]. Oil pollution inhibits the interaction of wind with the water surface. Due to this effect, the scattering of radio waves decreases in the antenna direction (Figure 1).

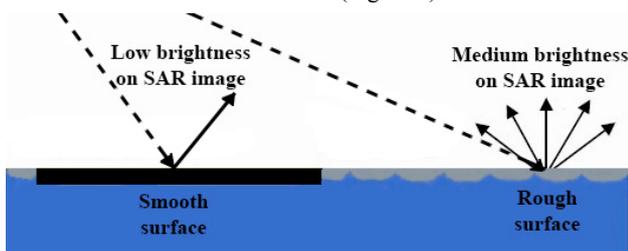


Fig. 1. Scattering of radio waves from the sea surface

2.2. Oil spill detection algorithm

A SAR image $I(m,n)$ presented by grayscale image is input data for the oil spills detection algorithm. The algorithm consists of the following main steps: image preprocessing, detection of dark segments, parameter extraction and classification of detected dark segments.

2.2.1. Image preprocessing

Preprocessing may consist of two parts: filtering of a speckle noise and masking of land areas.

The speckle noise is an integral component of the SAR image and can significantly affect the results of processing. So we can use mean, median, adaptive Lee, Frost filters or other filters to reduce the noise of images.

As the area of interest is water surface, the second step of the preprocessing is masking of land areas (Figure 2). This preprocessing step leads to the avoidance of false detections which are associated with relief roughness and other features of the land surface. For this purpose, the vector database of coastlines is used.

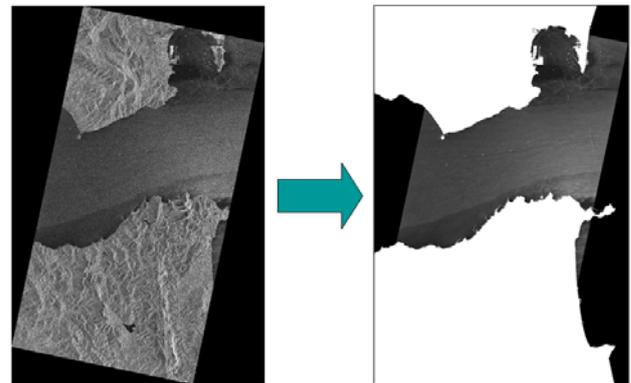


Fig. 2. Example of land masking

2.2.2. Detection of dark segments

An important step of the algorithm is detection of dark spots in the processing image. Segmentation is performed in order to pick out the dark areas which generally correspond to oil pollution. An adaptive threshold filtering is performed at this stage of the oil spill detection algorithm. This filtering is based on an estimate of roughness of the surrounding sea surface and consists of the following steps:

1. The mean value $\mu(m,n)$ and the standard deviation of brightness $\sigma(m,n)$ are computed in local window with size $w \times w$ for each pixel of the image $I(m,n)$, according to (1) and (2):

$$\mu(m,n) = \frac{1}{(w+1)^2} \sum_{x=m-w/2}^{m+w/2} \sum_{y=n-w/2}^{n+w/2} I(x,y), \quad (1)$$

$$\sigma(m,n) = \sqrt{\frac{1}{(w+1)^2 - 1} \sum_{x=m-w/2}^{m+w/2} \sum_{y=n-w/2}^{n+w/2} (I(x,y) - \mu(m,n))^2}. \quad (2)$$

2. Calculation expression $\lambda(m,n) = \frac{\sigma(m,n)}{\mu(m,n)}$ for each pixel

of the image $l(m,n)$. Homogeneity category K is determined in accordance with the calculated value $\lambda(m,n)$ in a neighborhood of the point (m,n) .

3. Calculation of the threshold value $T(m,n) = \frac{\mu(m,n)}{d(K)}$.

The coefficient $d(K)$ is selected from a premade table according to a certain category of homogeneity.

4. Obtaining of the binary image $b(m,n)$ by thresholding according to expression

$$b(m,n) = \begin{cases} 1, & \text{if } l(m,n) \geq T(m,n) \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

2.2.3. Parameter extraction from detected dark segments

Marking and parameterization of the binary image is performed after the detection of dark spots. As a result of this operation, all connected regions in the image are combined into individual segments. A set of parameters is determined for each segment: area S , perimeter P , coordinates of the segment's center, circularity c , eccentricity ε and others.

The marking and parameterization algorithm of the binary image $b(m,n)$ is performed by adding a label matrix Q and line by line image processing. We assume that the white points are part of objects – the oil spills and the black points are part of background – the sea surface.

2.2.4. Classification

Dark segments classification is an important step of the oil spill detection algorithm and requires a complex estimation of each segment. The set of slicks that are likely oil spills are formed at this stage.

One of the main features that characterize the oil pollution is its geometric shape which is determined by the circularity and the eccentricity. Other important parameter of the oil slick is its size. Segments having too small area are likely formed due to the strong sea surface disturbance and the interference effects. Too large segments are usually slicks of natural origin (colonies of algae) or calm areas. Edge detection of pollution areas plays an important role because oil spills typically have a well-defined smooth contour. Moreover, distinctive feature of oil slicks from natural pollution is the homogeneity of their texture.

3. SHIP DETECTION

One of the important problems arising in the field of SAR images interpretation is the detection of ships and vessels of various classes. Ship detection is a crucial application for global environmental monitoring and security. This allows to monitor traffic, fisheries and to associate ships with oil discharge.

Although the objects in the image are clearly visible, the presence of intense noise and clutter makes the problem of detection much more difficult. To improve the obtained results a reliable and fail-safe approach should be developed. Ship detection algorithm proposed in this paper consists of the following main stages: prescreening, land masking, image segmentation combined with parameter measurement, ship orientation estimation and size correction, object discrimination.

Prescreening involves the processing of the original SAR image using a multi-window averaging filter followed by adaptive local thresholding. The small window of size $w_s \times w_s$ is

constructed in the neighborhood of the current pixel (m,n) and the local average $\bar{\mu}_s(m,n)$ is calculated. The window is used to smooth the noise and to find a more accurate estimate of the object brightness. For the pixel under test the background statistics is estimated in the region bounded by window of size $w_s \times w_s$ on the one side and the filtering window of $w_b \times w_b, w_b \gg w_s$ size on the other side (Figure 3). It is assumed that the background in local region can be approximated by the normal distribution with $\mu_b(m,n)$ mean and variance $\sigma_b^2(m,n)$. All windows are moved one pixel at a time across the whole image.

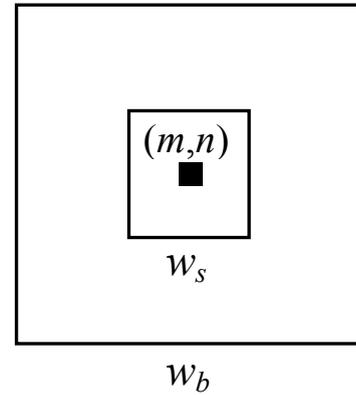


Fig. 3. Window setup for filtering

A binary image is obtained according to the rule (4):

$$b(m,n) = \begin{cases} 1, & \text{if } \bar{\mu}_s(m,n) > \bar{\mu}_b(m,n) + \bar{\sigma}_b(m,n) t / w_s, \\ 0, & \text{otherwise,} \end{cases} \quad (4)$$

where $\bar{\mu}_b(m,n)$, $\bar{\sigma}_b(m,n)$ – background average and standard deviation, t – threshold. The exact value t is chosen based on a given probability of false alarms $P_{fa} = 0,5 - 0,5 \operatorname{erf}(t / \sqrt{2})$, where $\operatorname{erf}(\dots)$ is the error function. In a binary image nonzero pixels belong to objects and others to background. At this stage the land mask can be implemented to allocate areas that don't contain sea surface. The described approach is similar to the CFAR detector, which is widely used in practice [2, 3].

However, the extended objects may be fragmented in the binary image, which leads to incorrect ship parameters estimation. Therefore, to increase the reliability of the algorithm an image segmentation stage is suggested for object mask restoration purposes.

Region growth algorithm is used as a basis for image segmentation. In the process of line by line scanning of image $b(m,n)$ the first pixel with nonzero value is treated as the start position for subsequent region growth. Then the neighboring pixels are viewed in the original SAR image.

It is considered that the analyzed point belongs to the segment, if it satisfies the homogeneity criterion $|l(m,n) - \mu_l| < T_l$, where $l(m,n)$ is the brightness of the analyzed pixel in the original image, μ_l – the average brightness of the segment, T_l – the predetermined threshold. To improve binding for all pixels that belongs to original mask no checks are performed. If a new pixel is added to the segment, its area S and average brightness μ_l are recalculated.

This process continues as long as a segment increases in its size also taking into account the connectivity conditions. Thus the form of the segment can be recovered and such parameters as

area, average brightness and bounding box can be simultaneously estimated.

With a restored object binary mask the corresponding pixels values in image $b(m, n)$ are set to zero. After that the searching of remaining nonzero pixels in $b(m, n)$ continues. If such pixel is found then the region growth algorithm is started in the new start position. Segmentation stage terminates at the end of the search procedure.

For correct ship size measurement it is important to accurately estimate its orientation. For a connected region R , which describes the binary segment mask, central moments of the second degree by row μ_{nn} , column μ_{mm} and a mixed central moment μ_{mn} are evaluated (5):

$$\begin{aligned} \mu_{mm} &= \sum_{(m,n) \in R} (m - m_c)^2 / S, \quad \mu_{nn} = \sum_{(m,n) \in R} (n - n_c)^2 / S, \\ \mu_{mn} &= \sum_{(m,n) \in R} (m - m_c)(n - n_c) / S, \end{aligned} \quad (5)$$

where (m_c, n_c) is the region centroid. Usually the object corresponding to the ship has a well distinguishable principal axis. It is understandable that the orientation of a ship is actually the direction of its principal axis with angle $\alpha = \frac{1}{2} \arctg(2\mu_{mn} / (\mu_{nn} - \mu_{mm}))$.

Object discrimination is the rejection of false alarms using object parameters analysis. Typically the width-to-length ratio of a ship lies in the range from 1/10 to 1/7 [8].

As the satellite image resolution is known the maximum and minimum area of a ship in meters can be simply converted to pixels. Objects with inappropriate area and aspect ratio are excluded from consideration. When all the required checks are made the list of final object parameters includes center, orientation, size and average brightness.

4. MEASUREMENT OF THE SEA CURRENTS SPEED

Sea currents allow track processes in the oceans. The described algorithm is designed to measure speed of sea currents. The result of the algorithm is a contour map of the speed of sea currents.

To measure sea currents information about the amplitude and phase of the reflected signal is required [1]. To find the solution of a considered task it is necessary to use SAR images such as images of SSC level processing, obtained by the spacecraft TerraSAR-X.

Suppose that we have SAR image of the sea surface, containing information about the amplitude and phase of the reflected signal. The process of mapping of sea surface speed can be described as follows. It is known that according to Doppler's law the frequency of the reflected signal changes in proportion to the object speed. Therefore, movement of the water surface (horizontal movement) as a part of the sea current affects the formation of SAR images [4, 5].

Sea currents have an influence on the satellite speed relative to the ocean surface. Due to the Doppler effect, the frequency f_{DC} of the reflected signal changes to the value

$$f_{DCA} = f_{DC} - f_{DCM}, \quad (6)$$

where f_{DCM} is the frequency of the reflected signal for the case of no sea currents [1]. For the TerraSAR-X image the frequency f_{DCM} can be computed from the data stored in the image passport file [6] using the expression $f_{DCM} = k_0 + k_1 t$, where t is

the period of a signal passing from the beginning to the current point of the image, k_0, k_1 – coefficients stored in the image passport file.

The SAR image data allow us to measure the Doppler centroid f_{DC} . The Doppler shift is calculated using the expression (6). Sea current radial velocity is evaluated from [4]

$$V_D = -\frac{\pi f_{DCA}}{k_R \sin \theta}, \quad (7)$$

where V_D – projection of an object's speed to the observation direction; θ – the incidence angle; $k_R = 2\pi/\lambda$; λ – the wavelength of the SAR-radar. The final expression for the sea current speed is obtained from (7), thus

$$V = V_D \sin \theta \cos \psi, \quad (8)$$

where ψ is the angle between the observation plane and the water flow direction.

5. EXPERIMENTAL RESULTS

The SAR images obtained from remote sensing satellites TerraSAR-X, ERS-2 were used as the input data for experiments.

The result of the oil spill detection algorithm is shown in Figure 4. Spots classified as oil spills are outlined by rectangles. Size of these spills is displayed in square kilometers.

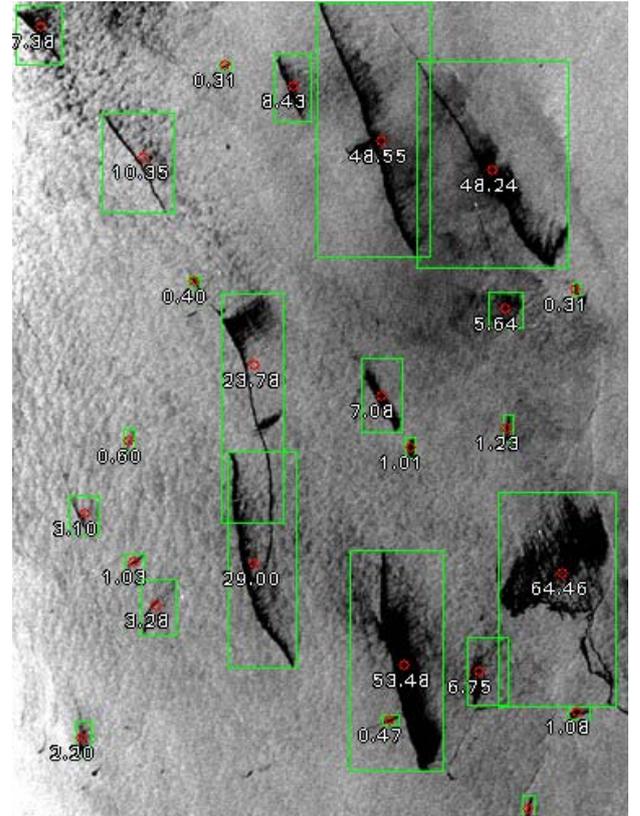


Fig. 4. Result of oil spills detection in SAR image

The result of the ship detection algorithm is shown in Figure 5. Objects of interest are marked by ellipses.

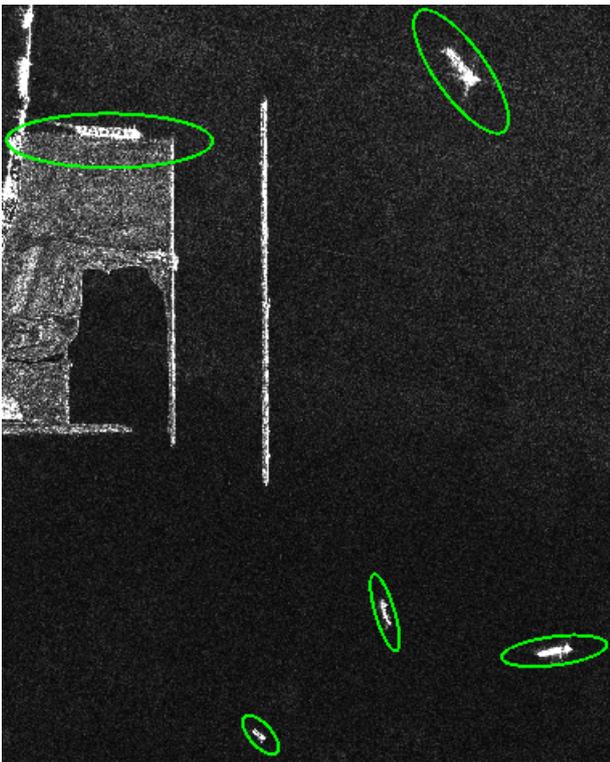


Fig. 5. The result of ship detection

6. CONCLUSION

This paper describes the developed algorithms of SAR image processing. The proposed approaches are modeled on the real TerraSAR and ERS-2 images. Presented algorithms are focused to solve a number of important scientific and practical tasks, including control of illegal fishing, ocean pollution, discovery of new oil deposits, studies of phytoplankton life, mapping of sea currents.

In practice the algorithms can be used for:

- high-precision measurement of ship characteristics and parameters;
- measurement of spills characteristics on the sea surface;
- distinguishing spills, formed by both artificial and natural pollutions;
- studies of surface masses movement.

Thus, the developed algorithms can be used as a very efficient tool for sea surface monitoring.

7. ACKNOWLEDGMENTS

This work was supported by grant RFMEFI57414X0056 from the Ministry of Education and Science of the Russian Federation.

8. REFERENCES

- [1] Chapron B., Fabrice C., Fabrice A. Direct Measurements of Ocean Surface Velocity from Space: Interpretation and Validation. *Journal of Geophysical Research*, 2005, Vol. 110. – P. 1 – 17.
- [2] Hinz S., Meyer F. Automatic ship detection in space-borne SAR imagery // *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2009, Vol. 38. – 6 p.

- [3] Jiayuan L., Mingzhu L., Shuhong J. An adaptive ship detection method in SAR image based on CFAR // *2nd International Conference on Information Communication and Management*, Singapore, 2012, Vol. 55. – P. 51 - 55.
- [4] Johannessen J.A., Kudryavtsev V., Chapron B., Collard F., Akimov D., Dagestad K.-F. Backscatter and Doppler signals of surface current in SAR images: a step towards inverse modeling. *Proceedings of SEASAR 2006*, 23 - 26 January 2006, Frascati, Italy.
- [5] Kang M., Lee H. Estimation of Ocean Current Velocity near Incheon using Radarsat-1 SAR and HF-radar Data. *Korean Journal of Remote Sensing*, 2007, Vol. 23, No. 5. – P. 421 - 430.
- [6] TerraSAR-X Ground Segment Level 1b Product Format Specification. 2007. – P. 257.
- [7] Topouzelis K.N. Oil Spill Detection by SAR Images: Dark Formation Detection, Feature Extraction and Classification Algorithms. *Journal Sensors*, 2008, Vol. 8. –P. 6642 - 6659.
- [8] Verba V.S., Neronskiy L.B., Osipov I.G., Turuk V.E. *Spaceborne Earth Surveillance Radar Systems*. – M.: Radiotekhnika, 2010. – 681 p., in Russian.

About the authors

Alpatov Boris (D.Sc.Tech.) is a professor, head of Department of Automation and Information Technologies in Control (DoAITC), Ryazan State Radio Engineering University (RSREU).

Balashov Oleg (Ph.D.) is an associate professor at DoAITC, RSREU.

Ershov Maksim is an engineer at DoAITC, RSREU.

Muraviev Vadim (Ph.D.) is an associate professor at DoAITC, RSREU.

Feldman Alexander (Ph.D.) is an assistant researcher at DoAITC, RSREU.

Автоматическое построение комбинаций алгоритмов обработки изображений в задаче обработки УЗ изображений сонных артерий

Т. Махно

Институт прикладной математики и механики НАН Украины
belikova.taisija@gmail.com

Аннотация

Одной из задач систем, призванных облегчать оценку степени атеросклеротического поражения артерий на основании УЗИ, является разделение плотных тканей и просвета артерий на УЗ снимке. В большинстве работ эта задача решается либо при помощи автоматической бинаризации изображения, либо вручную экспертом. При выборе методов подавления шумов различной природы и алгоритмов бинаризации необходимо рассматривать огромное множество комбинаций. В данной статье предложен алгоритм автоматического построения и тестирования таких комбинаций на различных изображениях. А также алгоритм, позволяющий проводить предварительный анализ изображения и выбор подходящих методов обработки.

Ключевые слова: обработка ультразвуковых изображений, генетический алгоритм, матрицы смежности яркостей.

1. ВВЕДЕНИЕ

Анализ атеросклеротического поражения артерий на основании УЗИ остаётся субъективным процессом, эффективность которого во многом зависит от эксперта. Вспомогательные программные приложения, призванные улучшить точность установленного диагноза, позволяют извлекать информацию из УЗ изображений. Одной из задач подобных систем является разделение плотных тканей и просвета артерий на УЗ снимке. Она зачастую решается с помощью разнообразных алгоритмов автоматической бинаризации изображения [2 - 3, 5 - 6], или подобную работу выполняет эксперт вручную [8].

Приведение УЗ изображения к бинарному виду можно условно разбить на два этапа: фильтрацию от шумов, а затем пороговое преобразование. И при удалении шумов, и при разделении изображения по порогу могут применяться различные алгоритмы, их комбинации и модификации. Выбрав некоторую последовательность обработки изображения и затем, тестируя её на различных примерах, можно сделать вывод об эффективности некоторого подхода. За последние более 30 лет было предложено огромное количество методов обработки изображений. Поэтому, при выборе методов подавления шумов различной природы и алгоритмов бинаризации, необходимо перебирать огромное множество комбинаций существующих алгоритмов, а также разрабатывать новые. Построение таких комбинаций и тестирование их на различных изображениях занимает немало времени и зачастую делает сложным сравнение различных вариантов обработки изображений между собой. Поэтому для решения задачи автоматизации этого процесса в данной нами предлагается использовать генетический алгоритм (ГА).

2. СУЩЕСТВУЮЩИЕ ПОДХОДЫ

В работах, посвященных обработке УЗ изображений, фильтрации шумов уделяется немало внимания [9]. Наиболее распространены фильтры Гаусса, например в [6]. Иногда, для уменьшения влияния шумов на результат работы алгоритмов, модификации фильтра Гаусса применяют не на первых стадиях обработки изображения [7]. Также популярными стали алгоритмы фильтрации шумов, предложенные в [2], и подробно описанные автором в [3]. Эти алгоритмы были предложены С. Loizou на основе анализа природы шумов на медицинских УЗ изображениях. Из большого количества разработанных и протестированных фильтров автором были отмечены, в частности, медианные фильтры и группа фильтров, объединённых С. Loizou под названием *lsmv* [2]. В основу *lsmv* фильтров были положены фильтры Lee [4]. Предложенный впервые ещё в 1990 г. С. Loizou фильтр *lsmv* вошел в его диссертацию [2] в 2005 г. До настоящего времени *lsmv* описывается в многочисленных сборниках, посвященных обработке медицинских изображений. В 4 пункте приведены результаты экспериментального сравнения последовательности обработки С. Loizou и последовательностей обработки, предложенных в данной работе.

Подходы к бинаризации изображений не менее разнообразны, чем к очищению от шумов. Существует огромное множество алгоритмов преобразования гистограммы, адаптивных локальных и глобальных пороговых фильтров, основанных на различных статистиках и параметрах. В примерах С. Loizou [2] использовался пороговый фильтр Отсу. Также автор использовал морфологический фильтр дилатации, с целью устранения «узких впадин» значений яркости на изображении.

3. ПРЕДЛОЖЕННЫЕ АЛГОРИТМЫ

Генетический алгоритм, осуществляющий автоматическое построение последовательностей обработки изображений из некоторых наперед заданных фильтров и алгоритмов, был предложен для ускорения процесса конструирования и проверки новой последовательности, а также в некоторой степени устранения субъективности подхода автора [13].

3.1. Применяемые алгоритмы обработки изображений

Для построения последовательностей фильтрации ГА использует различные линейные сглаживающие фильтры и фильтры низких частот, фильтры Собела и Превитта, Магг-Хилдрет (лапласиан-гауссианы); медианные фильтры, морфологические (эрозия, дилатация, открытие, закрытие). Алгоритмы преобразования гистограммы: линейное растяжение, эквализация, приведение гистограммы. Простые и адаптивные пороговые локальные и глобальные фильтры.

Фильтры Гаусса, фильтры высоких частот, основанные на преобразовании Фурье. Всего в рассмотрение было взято более 50 популярных фильтров и алгоритмов обработки изображений.

Отдельно можно остановиться на алгоритме приведения гистограммы. В данном случае гистограмма изображения приводится к виду, задаваемому кривой Джонсона класса Sb. Расчёты коэффициентов при построении кривой Джонсона производились по гистограммам изображений, для которых ГА быстро находил множество различных последовательностей фильтрации, дающих очень хороший результат обработки. Рассчитанные коэффициенты для вышеупомянутых изображений имели очень близкие значения, и для результирующей кривой было взято их усреднённое значение.

3.2. Алгоритм поиска последовательностей обработки изображений

Особь популяции ГА – это упорядоченный набор алгоритмов обработки изображения. Алгоритмы выбираются из наперёд заданного множества, описанного в пункте 3.1, они могут занимать любое место в последовательности, а также повторяться любое количество раз. Результат обработки изображения попиксельно сравнивается с шаблоном, и на основании сравнения рассчитывается значение фитнеса решения. Различные модификации ГА были протестированы и выполнен сравнительный анализ [12]. В результате работы ГА были найдены различные комбинации алгоритмов обработки изображений, позволяющие достичь требуемого эффекта обработки с достаточной точностью.

ГА использует классическую схему работы. Может использовать одно и двухточечный кроссинговер. Тип мутации меняется в ходе эволюции популяции. На ранних стадиях выше вероятность случайной мутации. Под случайной мутацией понимается замена случайно выбранного фильтра или алгоритма из особи популяции на другой, случайно выбранный из начального множества фильтров или алгоритмов. В более «поздних» популяциях выше вероятность замены «неудачного» фильтра или алгоритма, на некоторую подпоследовательность фильтров и/или алгоритмов, выбираемую из динамически создаваемой библиотеки. Такой оператор мутации призван не только уменьшить вероятность схождения алгоритма к локальному экстремуму, но и эмитировать работу человека при подборе последовательностей обработки изображений.

Найденные при помощи ГА последовательности, к сожалению, не являются универсальными. Для некоторых изображений они дают результат, отличный от желаемого. Однако практически для всех изображений можно построить такую последовательность, которая обеспечит необходимую близость результата фильтрации и шаблона. Таким образом, для некоторого нового изображения, не задавая шаблон, нельзя выбрать последовательность из ранее построенных ГА, и ожидать требуемого результата обработки.

Для выбора способа обработки нового изображения необходимо провести его предварительную оценку. Проведение такой оценки можно обеспечить, установив связь между значениями параметров изображений и теми методами обработки, которые дают хороший результат бинаризации.

3.3. Алгоритм поиска соответствий между методами обработки изображений и значениями текстурных параметров изображения

Для параметризации УЗ изображений было решено использовать текстурные характеристики изображений [10]. При выборе текстурных параметров сложно предположить, какова будет зависимость эффективности различных методов фильтрации (обработки) от значений того или иного параметра. Также не определён выбор из самого множества параметров [10 - 11] и их комбинаций. Кроме того, при построении двумерных экспериментальных графиков зависимостей результатов фильтрации изображений от их параметров визуально не определяются ни кластеры, ни функциональные зависимости. Следовательно, при параметризации изображений с целью быстрого выбора методов его фильтрации необходимо свыше двух параметров; и на момент реализации, и начала тестирования алгоритма невозможно определить, какие это параметры. Среди текстурных параметров, рассчитанных на основании матриц смежности яркостей, существует большое количество таких, которые теоретически могут влиять на результат обработки изображения по некоторым используемым алгоритмам. Мощность множества всевозможных таких параметров превышает 500.

Для поиска зависимости параметров изображений и методов их обработки применялся параллельный генетический алгоритм (ПГА). В данном случае необходимо разделить изображения на группы по некоторому критерию, так, чтобы после обработки любого изображения из группы некоторой заданной последовательностью фильтрации мы получали желаемый результат. Разделение на группы выполнялось на основании диапазонов значений текстурных параметров изображения. Особью популяции был вектор, состоящий из диапазонов значений параметров, и соответствующий ему список методов фильтрации. Вектор диапазонов параметров имеет произвольную длину, так как заранее неизвестно, какими параметрами и каким количеством этих параметров можно задать изображения в группах.

Тип ПГА, используемого в работе – «островной». Обмен особями параллельно развивающихся популяций происходит синхронно, с удалением из исходной популяции мигрирующей особи. Миграция однонаправленная, по кольцу. Мигрирующая особь выбирается случайным образом, не зависимо от значения фитнес функции. Частота обмена особями между подпопуляциями ПГА, по результатам экспериментов, считается оптимальной в пределах вероятности обмена 0,2 – 0,4 для каждого поколения. Уменьшение этого значения ведёт к излишним затратам вычислительных ресурсов, а увеличение – к сходимости различных подпопуляций к одним и тем же решениям. В алгоритме используется турнирный отбор родительских особей, оптимальный размер группы по результатам экспериментов – 4 особи. Сокращение промежуточной популяции осуществляется методом селекции. Кроссинговер выполнялся аналогично правилам промежуточной и линейной рекомбинации для нижней и верхней границей диапазона, в зависимости от начальных настроек алгоритма.

Использовалась случайная мутация, представляющая собой изменение верхней и нижней границ диапазона любого тектурного параметра особи на случайные допустимые вещественные числа. Изменение мутации в пределах 0,05 – 0,15 в среднем влияет на значение целевой функции в пределах 0,5%. Такое изменение значения целевой функции соответствует ошибке эксперимента. Таким образом, считаем, что большая часть модификаций данного алгоритма нечувствительна к изменениям вероятности мутации в вышеуказанных пределах. В алгоритме предложено несколько вариантов расчёта фитнес функции. В зависимости от требований к размеру групп изображений более удачными могут быть различные виды расчёта фитнеса.

Текстурные параметры перед использованием были проверены экспериментально на наличие зависимостей. Параметры, на множестве которых осуществляется поиск, попарно не коррелировали. Для построения регрессионных моделей и проверки зависимостей использовалась программная среда R.

4. ТЕСТИРОВАНИЕ

Теоретически, предложенные алгоритмы должны находить более универсальные и/или точные решения задачи построения последовательностей обработки изображений благодаря перебору большого количества вариантов. Поэтому в дальнейшем имеет смысл проводить исключительно экспериментальное численное сравнение построенных алгоритмом решений с существующими «готовыми» решениями данной прикладной задачи.

4.1. Сравнение с существующими результатами

Наиболее привлекательным для сравнения является последовательность обработки, на которую делается акцент в [2]. Это последовательность, состоящая из авторского метода lsmv и порогового фильтра Огсу. Кроме того, в одном из таких сборников [3] есть фрагмент кода реализации lsmv в среде matlab, приведенный автором. То есть в данном случае есть возможность достаточно точно повторить авторский алгоритм для сравнения.

Сравнение проводилось по 50 тестовым изображениям, для каждого из которых был задан шаблон-результат. Приведенные на рисунке примеры исходных изображений – это фрагменты УЗ изображений сонных артерий небольшого размера. В основном они содержат фрагмент стенки и просвета артерии на различных стадиях заболевания.

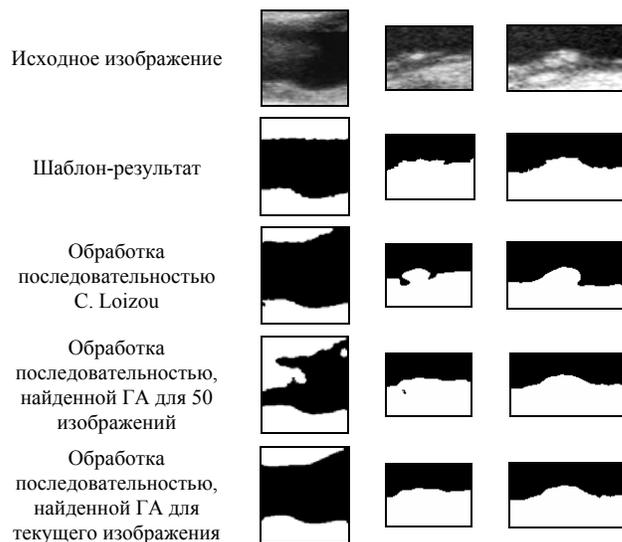
После применения к 50 фрагментам изображений последовательности, состоящей из алгоритмов lsmv и Огсу, было рассчитано среднее значение фитнеса обработанных фрагментов изображений. Аналогично было рассчитано значение для последовательностей, построенных ГА. Производилось несколько запусков ГА, осуществляющих поиск одной последовательности, в среднем дающей лучшее значение фитнеса для 50 изображений одновременно. Результаты различных запусков ГА могут отличаться, так как начальная популяция генерируется случайным образом. Полученные по обработанным фрагментам изображений значения фитнес функции приведены в таблице. Значения по всем запускам ГА усреднены.

	1	2	3
Среднее значение фитнес функции	0,76066	0,81625	0,91307
Минимальное значение фитнес функции	0,23880	0,38929	0,62427
Максимальное значение фитнес функции	0,97678	0,96395	0,99003

Среднее значение фитнес функции по 50 изображениям для: 1 – обработки последовательностью С. Loizou; 2 – обработки последовательностью, найденной ГА для 50 изображений; 3 – обработки последовательностью, найденной ГА для каждого изображения

При поиске одной «универсальной» последовательности (таблица, столбец «2»), после обработки некоторых из 50 изображений был получен результат, более чем на 90% совпадающий с шаблоном (что является достаточно хорошим результатом). Для некоторых изображений это значение не достигало порога в 80%, что является неудовлетворительным результатом. При поиске последовательностей обработки отдельно для каждого из 50 изображений среднее значение фитнеса было значительно выше, чем при поиске одной «универсальной» последовательности.

Примеры результатов обработки изображений, шаблонов результатов и исходных фрагментов приведены на рисунке.



Примеры результатов обработки

ГА поиска последовательностей обработки находит сравнительно более удачные и универсальные решения, по значению фитнес функции. Последующая параметризация при помощи ПГА призвана улучшить результат, сделав его приемлемым для большего числа изображений. А также для улучшения качества обработки всех изображений за счёт подбора подходящих последовательностей.

4.2. Тестирование ПГА поиска соответствий между значениями текстурных параметров и последовательностями фильтрации

Найденные ПГА на тренировочном множестве значения параметров, определяющие выбор последовательностей фильтрации, были проверены на тестовом множестве фрагментов изображений. Тренировочное множество состояло из 250 изображений и соответствующих им шаблонов-результатов, а тестовое – из 125. Изображения были взяты из УЗ снимков и кадров различных УЗ видеорядов на различных стадиях заболевания. Преимущественное большинство пациентов нуждалось в оперативном вмешательстве. В общей совокупности всех 375 изображений присутствуют такие, попадание которых в тестовое множество резко увеличивает ошибку тестирования, ввиду отдалённости данных элементов выборки от прочих в пространстве поиска. Так, при наличии в тестовой выборке некоторых таких удалённых точек произвольное увеличение тренировочного множества в 4 раза не даёт ожидаемого улучшения качества решений. На данный момент дать оценку отдалённости таких элементов выборки от прочих представляется возможным на основании значения ошибки тестирования. Для устранения столь сильной зависимости необходимо расширить множество изображений обучающей выборки. На настоящий момент по результатам экспериментов для произвольного тестового изображения значение вероятности получения удовлетворительного качества обработки была от 0,55 до 0,9 для различных вариантов тренировочной и тестовой выборок (совпадение с шаблоном более 80%).

5. ЗАКЛЮЧЕНИЕ

Результатом работы двух вышеупомянутых ГА является набор последовательностей обработки изображений и соответствующих им векторов диапазонов текстурных параметров. Система обработки УЗ изображений, использующая результаты работы алгоритмов, должна позволить с большей точностью разделять плотные ткани и просветы артерий на УЗ изображениях сонных артерий человека. При дополнении тренировочного множества изображений новыми экземплярами алгоритмы могут быть перезапущены и результаты их работы скорректированы.

6. БЛАГОДАРНОСТИ

Работа проводится в сотрудничестве с Институтом неотложной и восстановительной хирургии им. В.К. Гусака АМН Украины.

7. ССЫЛКИ

[1] Mitrea D., Nedeveschi S., Lupsor M., Badea R. Exploring Texture-Based Parameters for Noninvasive Detection of Diffuse Liver Diseases and Liver Cancer from Ultrasound Images //Proc. of the 8th WSEAS Int. Conf. on Mathematical Methods and Computational Techniques in Electrical Engineering, Bucharest, October 16 - 17, 2006.

[2] Christos P. Loizou Ultrasound Image Analysis of the Carotid Artery //PhD Thesis, Kingston University London, UK, 2005.

[3] Christos P. Loizou, Constantinos S. Pattichis, Costas Pattichis Despeckle Filtering Algorithms and Software for Ultrasound Imaging //A Publication in the Morgan & Claypool Publishers series, 2008.

[4] Lee J.S. Digital image enhancement and noise filtering by use of local statistics. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-2:165 - 168, 1980.

[5] Lin Y. Feature synthesis and analysis by evolutionary computation for object detection and recognition. //PhD Thesis, University of California, Riverside, 2003.

[6] Miguel A., Santos F., Maria R. dos Santos, Miguel P., Castro A.C., Azevedo E., Sousa L., João Manuel R.S. Tavares. (2013) A novel automatic algorithm for the segmentation of the lumen of the carotid artery in ultrasound B-mode images. Expert Systems with Applications 40:16, 6570-6579. Online publication date: 1-Nov-2013.

[7] Molinari F., Rajendra Acharya U., Zeng G., Kristen M. Meiburger, Jasjit S. Suri. (2011) completely automated robust edge snapper for carotid ultrasound IMT measurement on a multi-institutional database of 300 images. Medical & Biological Engineering & Computing 49:8, 935 - 945. Online publication date: 1-Aug-2011.

[8] Stoitsis J., Golemati S., Konstantina S., Nikita A Modular Software System to Assist Interpretation of Medical Images— Application to Vascular Ultrasound Images //IEEE Transactions On Instrumentation And Measurement, Vol. 55, No. 6, December 2006.

[9] Suri J.S., Yuan C., Wilson D.L., Laxminarayan S. Plaque Imaging: Pixel to Molecular Level //Volume 113 of Studies in Health Technology and Informatics, May 2005. – P. 488.

[10] Joan S. Weszka, Charles R. Dyer, And Azriel Rosenfeld A Comparative Study of Texture Measures for Terrain Classification, IEEE Transactions On Systems, Man, And Cybernetics, Vol. SMC-6, No. 4, April 1976.

[11] Yu-Hsiang Wu, Jhu-Yun Huang, Shyi-Chyi Cheng, Chen-Kuei Yang, Chih-Lang Lin Evolutionary Feature Construction for Ultrasound Image Processing and Its Application to Automatic Liver Disease Diagnosis //International Conference on Complex, Intelligent, and Software Intensive Systems, 2011.

[12] Беликова Т.А., Скобцов В.Ю. Генетический алгоритм в задаче фильтрации УЗ изображений и анализ эффективности его модификаций // Вестник Херсонского Национального технического университета № 1 (44), 2012. – С. 331 - 338.

[13] Беликова Т.А., Скобцов В.Ю. Эволюционный поиск эффективных последовательностей фильтров в задаче бинаризации УЗ изображений // Труды ИПММ НАН Украины. – Т. 23, 2011. – С. 21 - 34.

Об авторах

Махно Таисия – инженер 1 категории Институт прикладной математики и механики НАН Украины.

E-mail: belikova.taisija@gmail.com.

Визуализация и анализ тонкой тепловой структуры волны горения СВС в программе ImageJ

М. Бороненко, А. Серегин
Политехнический институт, кафедры физико-химии процессов и материалов
Югорский государственный университет, Ханты-Мансийск, Россия
MarinaB7@yandex.ru

Аннотация

Для определения наиболее эффективных путей управления структурой и свойствами получаемых материалов необходимо исследование влияния различных теплофизических параметров, при которых протекает СВС, на получаемый при этом продукт. Детальное исследование механизма взаимодействия реагентов требует применения комплексных методик эксперимента, дополняющих друг друга. В качестве экспериментальной методики исследования процесса СВ-синтеза может рассматриваться метод и аппаратура яркостной пирометрии высокого разрешения, основанной на анализе и визуализации экспериментальных данных в программе ImageJ.

Ключевые слова: Визуализация, межкадровый вычит, пиксель, изолинии, пороговая сегментация, оконтуривание, самораспространяющийся высокотемпературный синтез, ImageJ.

1. ВВЕДЕНИЕ

Успехи в области экспериментальной физики, техники, производства тесно связаны с совершенствованием методов получения и обработки измерительной информации. Наиболее востребованы установки, позволяющие осуществлять бесконтактное получение сведений об изучаемом объекте. Неотъемлемым элементом таких систем является узел, отвечающий за быстрый ввод зарегистрированных изображений в компьютер, в котором происходит их обработка, извлечение из изображений количественных характеристик изучаемого процесса, анализ полученных данных и их визуализация. При этом обычно возникает вопрос, в какой программе осуществлять обработку экспериментальных данных. Для тех, кто не в совершенстве владеет навыками программирования, подходит свободно распространяемая программа ImageJ. Но подходит ли эта программа для изучения таких быстротекущих процессов, как плазменное напыление или самораспространяющийся высокотемпературный синтез (СВС)? Целью данной статьи является представление результатов обработки и визуализации экспериментальных данных СВС, полученных с помощью высокоскоростного микропирометрического комплекса. Для достижения цели решались следующие задачи:

- регистрация тонкой тепловой структуры волны горения СВС, позволяющая наблюдать динамику первичных очагов СВ-синтеза;
- обработка и анализ изображений тепловой микроструктуры волны горения в программе ImageJ.

2. ПРИБОРЫ И ТЕХНИКА ЭКСПЕРИМЕНТА

Техническим средством системы обработки информации является высокоскоростной микропирометрический комплекс измерения температуры и скорости распространения пламени [2].

Программное обеспечение включает в себя пакет анализа экспериментальных данных Origin7 и свободно распространяемую программу обработки графических изображений ImageJ. Для использования видеокамеры в качестве высокоскоростного пирометра была проведена калибровка по эталонной температурной лампе ТРУ1100-2350 [2, 3]. Поставив в соответствие яркости пикселей ток, подаваемый на лампу, можно ввести новую калибровочную шкалу, связывающую яркость пикселей изображения и температуру регистрируемого видеокамерой на данной экспозиции светящегося объекта. Калибровка пространственного масштаба проводится по числу пикселей, укладывающихся на изображении деления 1 мм измерительной линейки. В качестве исходных материалов использовались порошки никеля ПНК-УТ1 и алюминия АСД-1 дисперсностью 5 – 15 мкм, приготовленная шихта Ni-18 мас. % Al, Ni-31.5 мас. % Al формовалась с насыпной плотностью 2,7 г/см³, начальная температура синтеза – 293 К. Синтез в реагирующей смеси инициировался путем локального нагревания верхней части поверхности насыпки электрической спирали. Съемке подвергалась произвольно выбранная область шихты. Регистрация процесса горения производилась с боковой поверхности образца при прохождении волны СВС в поле зрения 2 x 2 мм видеокамеры через оптический канал бинокулярного микроскопа МБС-10 на частоте 400 кадров в секунду и экспозиции 200 мкс. Полученные видеофайлы анализировали в программе ImageJ. Введенная калибровочная шкала позволяет провести статистический анализ локальной температуры каждого элемента всей выбранной области. Результатом исследования является набор характеристических параметров реакции: времена тепловыделения и теплоотвода, скорость фронта горения и др. Также существует возможность оценить размеры и времена существования фаз.

3. МЕТОДИКА ОБРАБОТКИ ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ

Покадровые изображения распространения волны горения СВС, полученные видеокамерой, сохраняются в формате .avi. Этот файл содержит информацию о пространственном распределении энергетической освещенности на плоскости, в виде двумерных массивов точек. Для выявления особенностей течения СВ-реакции был произведен межкадровый вычит. Разность двух изображений $f(x,y)$, $h(x,y)$ получается вычислением разностей между парами значений всех соответствующих пикселей изображений $f(x,y)$ и $h(x,y)$. При анализе хаотических волн твердотопливного горения [6 – 9] образуется множество очагов горения, которые движутся в разных направлениях с разной скоростью. Вычитание позволяет удалить все неподвижные, присутствующие на обоих кадрах очаги, становиться возможным наблюдение движущихся или выявление вновь возникших очагов горения на последовательности кадров. Таким образом, была обнаружена «обратная волна», визуализация прохождения фронта горения которой представлено на рис.

1 – 3. Появление и исчезновение активных областей, а также прохождение «обратной волны» можно видеть на последовательности маскированных кадров, в которой отсутствуют кадры, не содержащие новых активных областей.

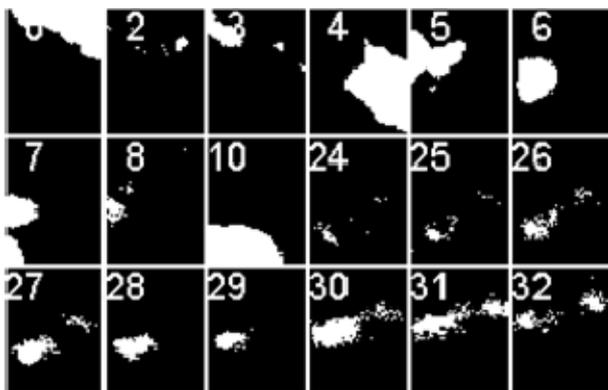


Рис. 1. Визуализация изменения площади, занимаемой «обратной волной»

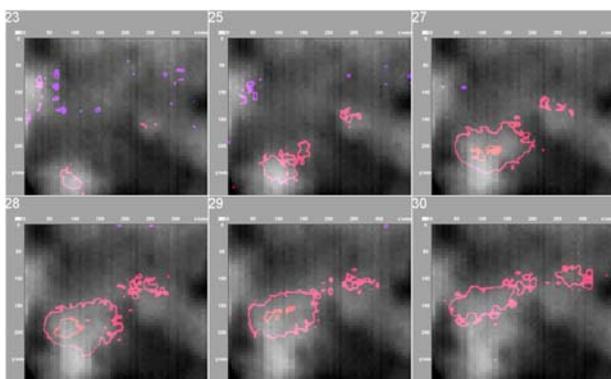


Рис. 2. Визуализация соответствия изолиний температурного поля «обратной волны» области на оригинальном кадре

Также можно исследовать температурный профиль волны горения. Для этого, выбрав интересующую область, выделяем линию, вдоль которой хотим выяснить температурный профиль – температуру вдоль линии в один и тот же момент времени. Линию можно выбирать, как стандартную – прямую, так и произвольной формы. Инструмент Plot Profile отображает в двумерном графике яркость пикселей вдоль выбранного направления. Изменение температурного профиля «обратной» волны горения представлено на рис. 3.

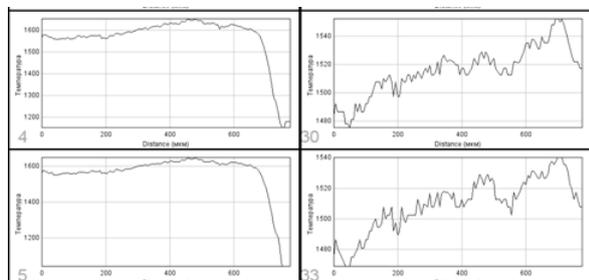


Рис. 3. Визуализация прохождения прямой и обратной волны вдоль выбранного направления

Динамику изменения площади обратной волны можно осуществить, преобразовав изображение в черно/белое (Make Binary) и выбрав в Set Measurements параметры, которые необходимо измерить: Centroid – центральная точка активного выделения и др. Команда Analyze Particles (Анализ частей) измеряет выбранные параметры по принципу сканирования всего изображения или выделения, пока не находит края объекта. Используя данный инструмент, можно автоматически или интерактивно настроить верхние и нижние значения порога для сегментирования области интереса. Результаты измерения представляются в текстовом файле Results. Наглядно разделить области с определенной температурой можно инструментами Threshold или Color Threshold. Hue позволяет устанавливать пороги на основе оттенков изображения, Saturation (Насыщенность) – на основе цветовой насыщенности участков изображения, Brightness (Яркость) регулирует порог по значению яркости пикселей изображения. Установив необходимый порог яркости пикселей на изображении, соответствующий после калибровки определенной температуре, мы визуализируем области, в которых реакция СВС протекает в одинаковой фазе (рис. 4). Здесь представлены 4 кадра, на которых методом пороговой сегментации выделена граничная область, в которую включены температуры, соответствующие $\epsilon+\delta$ (1668 K), ϵ (1653 K), $\epsilon+\chi$ (1658 K) областям на фазовой диаграмме. Внутри этих областей расположена реакционная зона с температурой, превышающей 1653 K, т.е. зона, в которой формируется δ -фаза.

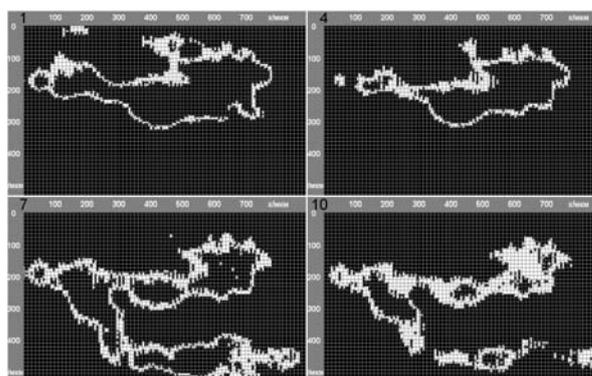


Рис. 4. Визуализация изменений граничной области ϵ -фазы

Выделение объектов на изображении – это одна из характерных задач обработки изображений [1, 4 – 5]. Мерой, определяющей границы контура, может служить отношение сигнал/шум. Выбор метода, улучшающего изображение, всегда индивидуален. Наиболее качественное выделение границ дает применение локального увеличения контраста. При этом использовалась Unsharp Mask (Контурная резкость). Эта команда позволяет управлять процессом усиления резкости: программа находит в изображении области со значительными изменениями цвета (резкий переход, сильный контраст) и заостряет их края, что приводит к усилению резкости контуров (контраста «на границах») и не влияет на остальные («ровные») участки. Здесь же существует возможность управлять пороговым значением, определяющим минимальное изменение яркости. Пороговое значение выбирают, исходя из соотношения сигнал/шум. Считается, что пиксель, имеющий соотношение сигнал/шум ниже порогового, не принадлежит контуру.

Еще одним важным параметром, характеризующим СВС, является скорость распространения волны горения. Оценка скорости возможна двумя способами: по изменению яркости пикселей в выбранных областях (сравнению термограмм горения) и по изменению размеров выбранной области реакции. При визуализации волны горения в виде изолиний температуры или 3D-поверхности хорошо видно, как распространяется тепловая волна: происходит отток тепла из одной области в другую, при котором происходит прогрев вещества в холодной области. При этом возможен расчёт скорости по изменению положения температурных изолиний.

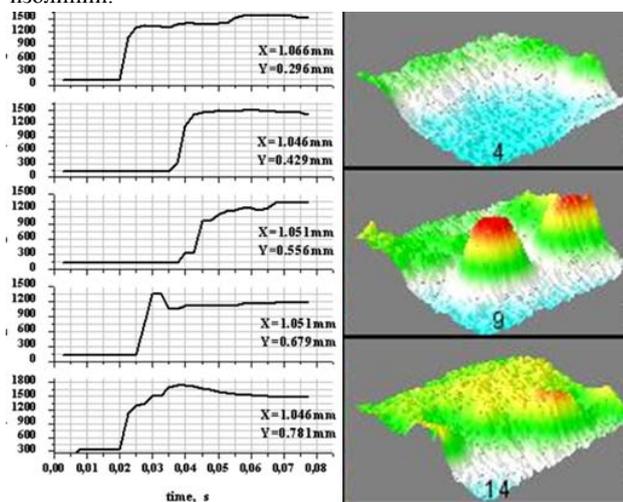


Рис. 5. Термограммы горения (а) и визуализация активной области (б)

Прослеживание изменения температуры, основываясь на статистических данных о яркости пикселей, позволяет более точно о процессах фазообразования в ходе реакции СВС. Изображение выбранной активной области 500x800 мкм² образовано пикселями, размер которых 6,25x6,25 мкм². В каждом пикселе определяется локальная температура. Статистику можно осуществлять, как в одном кадре, так и по всему видеофрагменту.

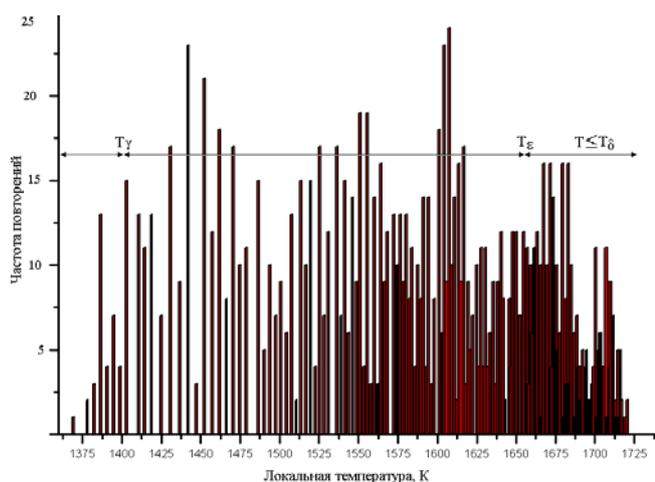


Рис. 6. Результаты статистического анализа яркости пикселей изображения

4. ЗАКЛЮЧЕНИЕ

Микровидеосъемка тепловых полей в волне горения СВС и применение комплексных методик анализа видеокadres в программе ImageJ позволяет получать количественные значения основных теплофизических параметров СВ-синтеза, а также проводить детальное исследование фазообразования. Таким образом, использование программы ImageJ дает возможность существенно облегчить визуализацию получаемых данных, а также качественный и количественный анализ быстропротекающих процессов СВ-синтеза.

5. ССЫЛКИ

1. Белоусов А.А., Спицын В.Г. Применение пространственных методов улучшения изображений на видеопоследовательностях // Известия Томского политехнического университета. – 2013. – Т. 322. – № 5.
2. Бороненко М.П., Гуляев П.Ю. Телевизионная измерительная система наносекундного разрешения // Доклады Томского государственного университета систем управления и радиоэлектроники, 2014. – № 1 (31). – С. 60 – 64.
3. Гуляев И.П., Гуляев П.Ю., Иордан В.И. Разрешающая способность виртуальных приборов контроля температуры частиц в плазменных потоках по суммарному спектру // Ползуновский альманах, 2008. – № 2. – С. 13 – 14.
4. Гуляев П.Ю. Байесовское восстановление цвета цифровых изображений / П.Ю. Гуляев, Ю.П. Гуляев, А.В. Долматов // Вестник Сибирской государственной геодезической академии, 1997. – № 2. – С. 114 – 115.
5. Потапов А.А. и др. Новейшие методы обработки изображений / под ред. А.А. Потапова. – М.: ФИЗМАТЛИТ, 2008. – Т. 496.
6. Development prospects of SHS technologies in Altai state technical university / V.V. Evstigneev, P.J. Guljaev, I.V. Miljukova, et al // International Journal of Self-Propagating High-Temperature Synthesis, 2006. – V. 15. – № 1. – P. 99 – 104.
7. Gulyaev I.P., Ermakov K.A., Gulyaev P.Yu. New High-Speed Combination of Spectroscopic And Brightness Pyrometry For Studying Particles Temperature Distribution In Plasma Jets // European Researcher, 2014. – № 3 – 2 (71). – P. 564 – 570.
8. Gulyaev P.Yu. Plasma spraying of protective coatings from ferromagnetic SHS-materials // Research Journal of International Studies, 2013. – № 12 – 1 (19). – P. 74 – 77.
9. In-situ selfpropagating-high-temperature-synthesis controlled by plasma / P.Yu. Gulyaev, I.P. Gulyaev, I.V. Milyukova, et al. // Вестник Югорского государственного университета, 2012. – № 2 (25). – С. 28 – 33

Об авторах

Бороненко Марина – ведущий инженер кафедры ФХПМ Политехнического института Югорского государственного университета.

E-mail: m_boronenko@ugrasu.ru,

Серегин Александр – инженер кафедры ФХПМ Политехнического института Югорского государственного университета.

E-mail: alex_seregin@mail.ru

Face Quality Assessment for Face Verification in Video

M. Nikitin¹, V. Konushin², A. Konushin¹

¹Lomonosov Moscow State University, ²Video Analysis Technologies, LLC
mnikitin@graphics.cs.msu.ru, vadim@tevia.ru, ktosh@graphics.cs.msu.ru

Abstract

Performance of biometric systems depends on quality of acquired biometric samples. Low sample quality is the main reason for matching errors in biometric systems and may be the principal weakness of some implementations. Therefore, when a biometric system obtains a sequence of person images from a surveillance camera, the quality of the different face images has to be evaluated before performing any analysis on the face of a person. In this paper, we propose an approach for face image quality assessment, which is based on four facial features including facial symmetry, sharpness, quality of illumination and the image resolution. To produce overall face quality score we perform weighted fusion of facial features with automatically tuned weights. Experimental evaluation of the proposed method has demonstrated its high accuracy and efficiency.

Keywords: *face quality assessment, facial symmetry, video surveillance.*

1. INTRODUCTION

When a person is observed by a surveillance camera, a sequence of images of that person is captured. Most of these images are useless due to problems like not facing the camera, motion blur, poor illumination and too small size of the region of interest in that image. For most biometric applications considering some (one or two) of the best images is sufficient to obtain accurate results. Therefore, there is a need for a mechanism, which can choose the best image from a sequence in terms of quality. This is called Quality Assessment. Automatic face quality assessment (FQA) can be used to monitor image quality for different applications such as face logging, video-based face classification [6] and identification [7].

Fig. 1 shows a framework of a face identification system using face quality assessment component. Face images are preprocessed and their quality is evaluated. Low quality images are discarded and only images with acceptable qualities are received for recognition. This allows to significantly accelerate matching speed in cases of large gallery. Moreover, in [13] it was shown that using quality assessment component in video-based face identification system can greatly improve its performance. Also, the quality score may be useful in image-based face recognition. For example, images of different qualities can be processed in different ways or high quality image may be necessarily required for reliable matching.

In different works related to FQA, different quality metrics were analyzed. X. Gao et al.[4] tried to standardize the quality of face images by facial symmetry based methods. Wong et al.[13] developed a method for simultaneous handling issues such as pose variations, cast shadows and blurriness, which quantifies the similarity of a given face to a probabilistic face model, representing an ‘ideal’ face, via patch-base local analysis. Fournery et al [3] proposed a quality fusion approach to combine head pose, sharpness, human skin presence, resolution, and two illumination measurements. Nasrollahi and Moeslund[8] proposed a similar quality assessment method, by using out-of-plane rotation, sharpness, brightness, and image resolution qualities.

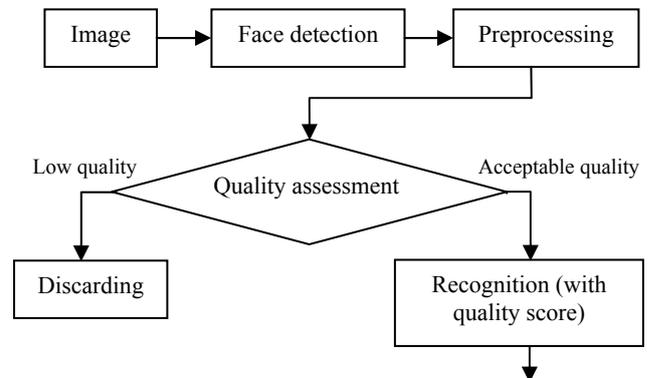


Fig. A framework of face recognition system with quality assessment component

In order to obtain an overall image quality score, it is necessary to combine all the scores of the used quality metrics, measuring different quality parameters. There are various methods of quality metrics fusion. Some works do it by thresholding each quality metric and counting the number of satisfied metrics. Others perform weighted averaging. The significant drawback of many existing works is that all thresholds and weights are obtained experimentally [3, 8].

In this paper, we propose a face quality assessment method, which is based on four facial features including facial symmetry, sharpness, illumination quality and face size. In order to combine our facial quality features we perform weighted averaging. For automatic weights tuning we adapted Ozay's et al [9] face recognition match score based technique.

The majority of existing pose estimation based FQA methods use an analysis of gradient image in order to locate left and right sides and face's axis of symmetry. Such method is not stable when subjects are wearing glasses, or when faces are not upright. To avoid this problem we perform facial symmetry evaluation based on feature points from facial features detector. In addition, using facial points allowed us to conduct sharpness analysis more accurately. In practice modern facial features detectors[5,10] only take about one millisecond per image and that is why they can be used as a part of FQA algorithm without notable performance loss.

There are two main contributions in this paper. The first contribution is that we proposed a new facial symmetry estimation method, which is based on the analysis of face local features. The second contribution is that we adapted weights tuning technique of low-level features for our high-level facial features.

2. PROPOSED ALGORITHM

When determining the quality of a face image I , we consider four facial features, weighted to varying degrees of importance. These features are facial symmetry, sharpness, illumination and resolution. The following sections describe how each of these features can be measured, and how they contribute to the overall quality score of an image.

2.1. Facial symmetry

The pose variations and illumination unevenness are two main issues that cause serious performance degradation for the most existing biometric systems, because wide variations in pose and illumination direction can hide most of the useful face image features. We propose to use facial symmetry to assess quality degradations caused by improper facial pose and non-frontal lighting.

The symmetry may be analyzed using some local image features, e.g., the raw image pixel values, or locally-filtered pixel values. When a local filter is chosen properly, it provides a better basis for computing facial symmetry. The degree of similarity between image features at the corresponding left-right pixel locations provides local measures of symmetry. If the face image is strictly left-right symmetric, the similarity scores should all take the maximum value.

We propose (see Fig. 2) to use similarity score between histograms H^L and H^R of facial local features as local measures of symmetry, because local feature histograms are more stable relative to slight misalignments than the local features itself. We use FaceSDK library [2] to find the locations of facial features. One of the compared patches (left or right) is horizontally mirrored before histogram computation. To find similarity score of two normalized histograms we use histogram intersection distance, which can be calculated as follows:

$$d_{intc}(l) = \sum_t \min(H_t^L(l), H_t^R(l)), \quad (1)$$

where l denotes l -th pair of symmetric facial points. The larger the intersection distance value is, the larger the left-right symmetric of the face image is, and the larger the image quality is in some aspects.

Facial symmetry should be measured based on pose-sensitive image local features. In this paper, the Histograms of Oriented Gradients (HoG)[1] are used for this purpose.

The facial symmetry score S_1 is calculated as the mean value of all the histogram distances:

$$S_1 = \text{Symmetry}(f) = \frac{1}{N} \sum_{l=1}^N d_{intc}(l), \quad (2)$$

where N is the number of pairs of symmetric facial points. The larger the $\text{Symmetry}(f)$ value, the more the face is symmetric.

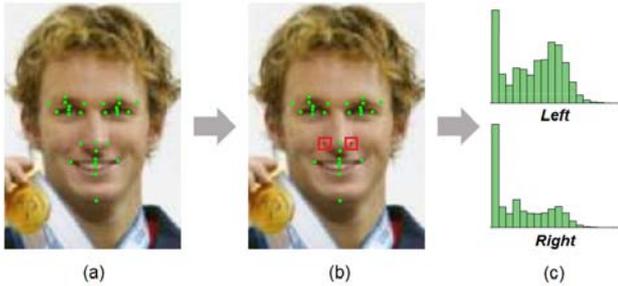


Fig. 2. (a) Face image with detected local facial features. (b) Used areas of symmetric points for local feature histogram computing. (c) Comparing left-right histograms of oriented gradients H^L and H^R

2.2. Sharpness

Since in real world applications the objects are moving in front of the camera, it is possible that the captured image is affected by motion blur, so defining a sharpness feature can be useful for FQA. The sharpness of a face image refers to the degree of clarity in both coarse and fine details in the face region. Well-focused images, which have a better sharpness compared to blurring images, should get a higher score for this feature.

We use a modified discrete Laplace operator to estimate image sharpness:

$$L(f) = \left| \frac{\partial^2 f}{\partial x^2} \right| + \left| \frac{\partial^2 f}{\partial y^2} \right|. \quad (3)$$

The Laplace operator is an example of a second order method of image spatial filtering. It is particular good at finding the fine details in an image. Any feature with a sharp discontinuity will be enhanced by a Laplace operator. The discrete second derivatives can be computed as convolution with the following kernels: $(1, -2, 1)$ and $(1, -2, 1)^T$.

We perform sharpness estimation strictly inside the facial area (see Fig. 3). For this purpose, we construct a mask using facial points found by face features detector. The mask is constructed in such a way that there would not be background pixels on masked image for any allowable face pose. The image sharpness score S_2 is calculated as the averaged Laplace operator response by masked image. Such scheme allows to achieve independence of sharpness score from image background.

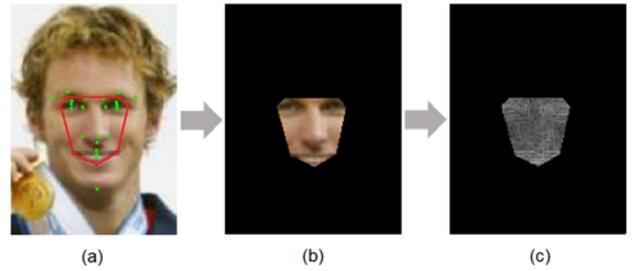


Fig. 3. (a) Face image with detected local facial features and mask contour. (b) Masked image. (c) Response of modified Laplace operator on masked image

2.3. Illumination quality

Variations caused by changes in illumination constitute yet another significant challenge encountered by automated biometric systems. In order to compensate for different lighting conditions some implementations may perform histogram equalization or similar histogram dependent techniques in order to normalize an image before its analysis. For this reason, it is highly important to begin with images which make the best (maximum) use of the available dynamic range.

We estimate quality of illumination by determining the length R of available range of gray intensities excluding 5% of the darkest and brightest pixels (see Fig. 4). The illumination quality score S_3 is simply the percentage of the total dynamic range represented in R :

$$S_3 = \text{Illumination}(f) = \frac{R}{256} \quad (4)$$

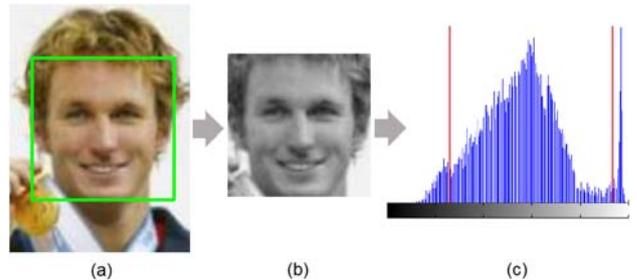


Fig. 4. (a) Face image with marked bounding box. (b) Cropped grayscale image. (c) Pixel intensity histogram; used range R marked with red vertical lines

2.4. Face size

The face image resolution score is perhaps the easiest of the aforementioned quality features to measure. The face size quality score S_4 is defined to be the linear function of the size of its bounding box. In general, high resolution images are preferred over low resolution images. We define lower threshold for face size as 50 pixels and upper threshold as 150 pixels. If bounding box size is below the lower threshold, face size score takes the minimum value, and in the case of bounding box size is beyond the upper threshold, face size score takes the maximum value, because it is no longer useful to achieve higher resolution:

$$S_4 = \text{Size}(x) = \begin{cases} 0, & x < 50 \\ 0.01x - 0.5, & 50 \leq x \leq 150, \\ 1, & x \geq 150 \end{cases} \quad (5)$$

where x is the bounding box size.

2.5. Overall quality score

Each of the four facial quality features discussed in the previous sections can score in the range $[0,1]$, but these features should not contribute equally to an overall quality score $q(I)$ of image I . For this reason, they are combined according to the weighted sum

$$q(I) = w_0 + \sum_{i=1}^4 w_i S_i, \quad (6)$$

where w_0 is the bias term and the coefficients $w_i, i = 1,2,3,4$ determine the impact, which the quality features have on the overall score. For automatic weights tuning Ozay's et al.[9] technique has been adapted.

We consider the problem of weights tuning as the problem of linear regression learning, where facial quality features S_i act as predictor variables and overall image quality score $q(I)$ acts as dependent variable. The main problem that arises with such approach is the difficulty of training set preparation. Indeed, it is difficult for human to quantify the image quality, especially if it is necessary to consider several factors. For this reason, we decided to obtain overall quality scores for objects of training set using the information about their mutual similarity (in terms of face recognition match scores). Let us define it more formally.

A matching algorithm \mathcal{A} produces a score for a given pair of images:

$$s_{i_k i_l} = \mathcal{A}(i_k, i_l), \quad (7)$$

where i_k denotes the k^{th} image of the i^{th} individual in training set. Considering the match score as a similarity measure, a quality measurement algorithm should satisfy the following property: face image of a subject should be assigned a high quality score if it is similar to other images of the same subject while it is different from the image of other subjects. This rule makes it necessary to define a measure of the match score quality. In [11], the normalized match score (NMS) was proposed for this purpose. The NMS between the k^{th} and l^{th} images of i^{th} individual is defined as:

$$NMS(i_k, i_l) = \frac{s_{i_k i_l} - \mu_{i_k}(s_{\text{non-match}})}{\sigma_{i_k}(s_{\text{non-match}})}, \quad (8)$$

where $\mu_{i_k}(s_{\text{non-match}})$ and $\sigma_{i_k}(s_{\text{non-match}})$ are respectively the mean and standard deviation of the match scores between the image i_k and the images from other individuals $j \neq i$.

The NMS provides some information about the quality of the images i_k and i_l , but it is not symmetric in its arguments as the

non-match score distribution will vary when NMS arguments are interchanged. This variation could be especially strong when the images are of different quality. Hence, it is not a good measure of the quality of the match score, which is symmetric by its definition.

In order to avoid mentioned problem, in [9] symmetric normalized match score ($SNMS$) was defined:

$$SNMS(i_k, i_l) = \frac{1}{2}(NMS(i_k, i_l) + NMS(i_l, i_k)). \quad (9)$$

$SNMS$ satisfies desired property: it will be high for high-quality image pairs and will be low for low-quality image pairs. Therefore, $SNMS$ can be used as a measure of the quality of a match.

Once we have a way to measure the quality of the match, the next step is to separate it into the qualities $q(i_k)$ and $q(i_l)$ of matched images. We model the quality of the match as the average quality of compared images:

$$\frac{q(i_k) + q(i_l)}{2} \approx SNMS(i_k, i_l). \quad (10)$$

If we have at least three images for each individual in the training dataset, the separation problem can be solved. In particular, by combining all the equations for an individual i , we obtain the following least squares problem:

$$\mathbf{A}\mathbf{q}_i + \mathbf{e} = \mathbf{y}_i, \quad (11)$$

where

$$\mathbf{A} = \begin{bmatrix} 0.5 & 0.5 & 0 & 0 & \dots & 0 \\ 0.5 & 0 & 0.5 & 0 & \dots & 0 \\ \vdots & & \ddots & & & \vdots \\ 0 & \dots & 0 & 0 & 0.5 & 0.5 \end{bmatrix}, \quad \mathbf{e} = \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_T \end{bmatrix},$$

$$\mathbf{q}_i = \begin{bmatrix} q(i_1) \\ \vdots \\ q(i_N) \end{bmatrix}, \quad \mathbf{y}_i = \begin{bmatrix} SNMS(i_1, i_2) \\ SNMS(i_1, i_3) \\ \vdots \\ SNMS(i_{N-1}, i_N) \end{bmatrix}.$$

Here, \mathbf{A} is a $T \times N$ matrix with two non-zero elements in each row where N is the number of samples from individual i , and T is the number of possible pairs of i^{th} individual images. When $N \geq 3$, \mathbf{A} is a full column rank; hence the solution with minimum squared error is given by $\mathbf{q}_i = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{y}_i$.

The sample qualities for all the images in the training set can be obtained using this separation scheme. Once we have a quality value assigned to each image in the training dataset, we can find coefficients $\mathbf{W} = (w_0, w_1, w_2, w_3, w_4)^T$ of the linear regression that would predict face image quality $q(I)$ based on values of facial quality scores $\mathbf{f}_I = (1, S_1, S_2, S_3, S_4)^T$:

$$q(I) = \mathbf{W}^T \mathbf{f}_I. \quad (12)$$

3. EXPERIMENTS

First of all we have performed visual analysis of the proposed FQA method by estimating quality of video frames and sorting them in descending order. Fig. 5 illustrates the example of such ranking. In most cases, the relative position of sorted frames corresponds to the intuitive idea.

Further, we evaluated our face quality assessment method for the video-based face verification task on real-world YouTube Faces [12] dataset.

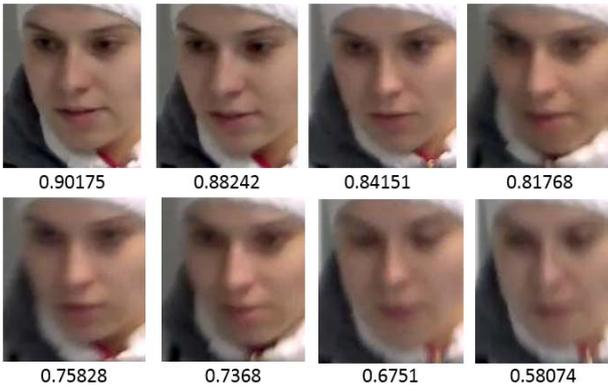


Fig. 5. Example of image ranking based on the proposed face quality assessment method. Numbers below images indicate corresponding quality score

3.1. Experimental setup

YouTube Faces (YTF) is a video dataset, which contains 3,425 videos of 1,595 different subjects downloaded from YouTube. It is collected in unconstrained conditions and contains large variations in pose, expressions, illumination, etc. All YTF video clips are provided with labels indicating the identity of a person appearing in each video. It also contains meta-data defining benchmark protocols for video-based face verification task.

YouTube Faces dataset follows a ten-fold, cross validation, pair-matching ('same'/'not-same') test. Specifically, 5,000 video pairs were randomly selected and divided into 10 subject-mutually exclusive splits. Each split contains 250 'same' and 250 'not-same' pairs. The goal is to determine for each split, which are the same and which are the not-same pairs, by training on nine remaining splits.

We perform video-based face recognition with FQA component in the following way:

1. Find the highest quality frame of a video sequence.
2. Use it as input for the image-based face recognition algorithm.

3.2. Experimental comparisons

The proposed FQA method was compared against two other methods: Fourney's et al.[3] and Wong's et al.[13] patch-based method. Our and patch-based methods were trained on PubFig[7] dataset. For selected frames matching, face recognition module from the FaceSDK library[2] was used. All experiments were conducted on the machine equipped with a 2.3GHz Intel Core i7-3610QM processor and 8GB of RAM.

Results for MATLAB implementations are presented in Table.

Method	Accuracy	FQA speed
Fourney et al.[3]	69.82%	142.5 fps
Wong et al.[13]	79.92%	1.6 fps
Proposed	74.46%	28.5 fps

The comparison on the YouTube Faces dataset

The results indicate that patch-based FQA outperforms our method when considering verification accuracy. However, it works much slower. So to handle a pair of videos with a frame rate of 24 fps and a duration of 3 seconds each, patch-based method will take approximately 90 seconds for frames quality estimation, while the proposed method can do it in 5 seconds. In this way, our method provides a trade-off between speed and accuracy.

4. CONCLUSION

In this paper, we presented a new face image quality assessment algorithm, which is based on evaluating a set of facial features. The proposed approach is capable of handling issues such as pose and illumination variations, motion blurring and insufficient face image resolution. We developed a new face symmetry estimation method, which is based on analysis of symmetrically located face features. To get the overall face image quality weighted averaging of facial features is performed. For automated weights learning we have adapted Ozay's et al [9] technique.

Our FQA method has been applied as a quality assessment component in video-based face verification system. Comparison with other methods has demonstrated that our method provides a trade-off between accuracy and efficiency of the verification system.

5. ACKNOWLEDGMENTS

This paper was supported by RFBR, research project No. 14-01-00849 a.

6. REFERENCES

- [1] Dalal N. and Triggs B. Histograms of Oriented Gradients for Human Detection. In Proc. IEEE Conf. Computer Vision and Pattern Recognition, vol. 2, 2005. – P. 886 – 893.
- [2] FaceSDK – face analysis library. <http://www.tevian.ru/ru/products/facesdk>
- [3] Fourney A. and Laganier R. Constructing Face Image Logs that are Both Complete and Concise. In 4th Canadian Conference on Computer Vision and Robot Vision, 2007. – P. 488 - 494
- [4] Gao X., Li S.Z., Liu R. and Zhang P. Standardization of face image sample quality. In Proc. Int. Conf. Biometrics, 2007. – P. 242 - 251.
- [5] Kazemi V. and Josephine S. One Millisecond Face Alignment with an Ensemble of Regression Trees. In Computer Vision and Pattern Recognition (CVPR), 2014.
- [6] Konushin V., Lukina T., Kuharenko A., Konushin A. Simile classifiers for face classification. In Proc. GraphiCon, 2012. – P. 108 - 112.
- [7] Kumar N., Berg A., Belhumeur P. and Nayar S. Attribute and Simile classifiers for face verification. In Proc. ICCV, 2009. – P. 365 - 372.
- [8] Nasrollahi K. and Moeslund T.B. Face quality assessment system in video sequences. In BIOD, Lecture Notes in Computer Science (LNCS), vol. 5372, 2008. – P. 10 - 18.
- [9] Ozay N., Tong Y., Frederick W. and Liu X. Improving face recognition with a quality-based probabilistic framework. In Computer Vision and Pattern Recognition (CVPR) Biometrics Workshop, 2009. – P. 134 - 141.
- [10] Ren S., Cao X., Wei Y. and Sun J. Face Alignment at 3000 FPS via Regressing Local Binary Features. In Computer Vision and Pattern Recognition (CVPR), 2014.
- [11] Tabassi E., Wilson C.L. and Watson C.I. Fingerprint image quality. Technical report, NIST, 2004.
- [12] Wolf L., Hassner T. and Maoz I. Face Recognition in Unconstrained Videos with Matched Background Similarity. In IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2011. – P. 529 - 534.
- [13] Wong Y., Chen S., Mau S., Sanderson C. and Lovell B.C. Patch-based probabilistic image quality assessment for face selection and improved video-based face recognition. In Computer Vision and Pattern Recognition Workshops (CVPRW), 2011. – P. 74 – 81.

Выделение зрачка на изображении глаза преобразованием Хафа границ компонент связности

И. Матвеев¹, Н. Чинаев²

¹Вычислительный центр им. А.А. Дородницына Российской академии наук, Москва, Россия

²Московский физико-технический институт, Москва, Россия
matveev@ccas.ru

Аннотация

Предложен метод поиска зрачка на изображении глаза. Граница зрачка определяется как граница или часть границы одной из компонент связности, получаемых бинаризацией исходного изображения. Для выделения части границы, соответствующей зрачку, используется модификация преобразования Хафа для окружностей. Метод испытан на 70000 изображениях радужки из баз, находящихся в открытом доступе.

Ключевые слова: сегментация радужки, преобразование Хафа, компоненты связности.

1. ВВЕДЕНИЕ

Выделение области радужки на изображении глаза – интересная и актуальная задача в системах биометрической идентификации и медицинских приложениях. Важнейшей подзадачей является определение границы зрачка, по сути, поиск округлого тёмного объекта. Проблема поиска круга/окружности – одна из наиболее известных в обработке изображения. Для её решения создано большое количество различных методов, часть из которых применима в данном частном случае. Можно назвать морфологическое выделение круга [1], проекции яркости или её градиента [7], построение оптимального контура [10]. В морфологическом подходе и методах, основанных на проекциях яркости, применяется бинаризация изображения с некоторым порогом, после чего рассматривается наибольшее связное множество точек с яркостями ниже пороговой, исходя из предположения, что зрачок – наиболее тёмная область на изображении глаза. Преимуществом этих методов является скорость исполнения. Следует отметить, что границы, выделяемые при бинаризации, состоят из сравнительно небольшого количества точек. Однако при бинаризации часто выделяется несколько компонент связности, кроме того, на изображениях радужки ресницы и веки могут сливаться со зрачком, образуя одну компоненту неадекватной формы. Поэтому недостатком этого класса методов является значительное число ошибок, связанных с искажением формы объектов, удалением частей фигур. Большая группа методов [2, 5, 8] основана на идеологии преобразования Хафа. Преимуществом преобразования Хафа является устойчивость к удалению части фигуры, например, с его помощью можно найти окружность по одной или нескольким дугам. Недостаток – большая вычислительная сложность.

В работе сделана попытка совместить преимущества метода бинаризации, извлекающего малое количество представляющих интерес пикселей границ областей связности, и преобразования Хафа, которое выделяет форму заданного типа из сильно зашумлённых данных. На бинаризованном изображении выделяются компоненты связности, граница каждой из которых подвергается преобразованию Хафа с целью поиска

округлых контуров. Использование такого сочетания методов из литературы авторам неизвестно.

2. ПОСТАНОВКА ЗАДАЧИ И МЕТОД РЕШЕНИЯ

На изображении глаза с прилегающими частями лица (брови, нос, часть щеки) необходимо найти окружность, являющуюся наилучшей аппроксимацией границы зрачок-радужка. Граница может быть частично закрыта веками, ресницами, бликами. Зрачок является тёмным, но не обязательно самым тёмным объектом на изображении, см. рис. 1 (а).

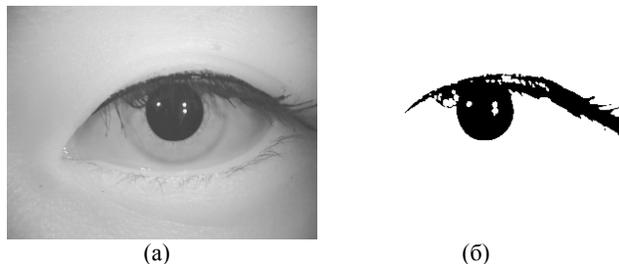


Рис. 1. Исходное изображение (а) и результат его бинаризации (б)

Обозначим исходное растровое изображение $I(x, y) = I(\vec{p})$. Необходимо найти координаты и радиус окружности, аппроксимирующей зрачок. На изображении методом бинаризации выделяются компоненты связности C , соответствующие тёмным областям. Для каждой компоненты определяется её граница $L(C) = \{l_i\}$ – циклическая последовательность пикселей. Следует отметить, что количество пикселей в такой последовательности значительно меньше числа пикселей с высоким градиентом яркости, которые можно было бы считать граничными. Зрачок не всегда граничит только с более светлой радужкой, но часто бывает прикрыт веками и/или ресницами, что создаёт помехи двух видов: граница зрачка на изображении присутствует лишь частично; яркость век и ресниц может быть такой же или даже ниже яркости зрачка, и при выделении компоненты связности зрачок сливается с ними; см. рис. 1 (б). В этом случае лишь небольшая часть границы компоненты связности является границей зрачка, но метод должен определить ту часть границы, которая представлена на изображении. Проводится преобразование Хафа, состоящее в построении в аккумуляторном пространстве луча из каждого пикселя границы перпендикулярно её направлению. В результате, если граница содержала дуги окружности, в аккумуляторе возникают значимые локальные максимумы на месте центров гипотетических окружностей, содержащих эти дуги. Глобальный максимум аккумулятора соответствует наиболее правдоподобному положению окружности. Для определения её радиуса строится гистограмма расстояний от найденного положения

центра до граничных пикселей. Максимум гистограммы соответствует радиусу. Из нескольких гипотетических окружностей, построенных таким образом для разных компонент связности, выбирается наилучшая, согласно критериям качества. Блок-схема метода представлена на рис. 2.

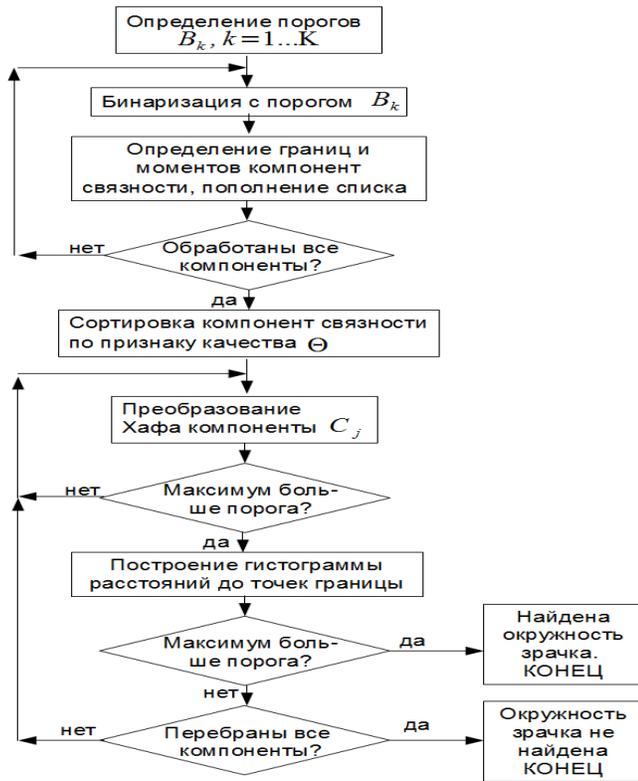


Рис. 2. Блок-схема работы метода

Более подробно рассмотрим его работу.

2.1. Определение порогов бинаризации.

Поскольку заранее неизвестны уровни яркости зрачка и радужки, а, следовательно, порог бинаризации, который разделит их, производится несколько проходов (обозначим их количество K) алгоритма при разных порогах. Величины используемых порогов определяются так, чтобы количество пикселей с яркостью ниже порога составляло определённую долю их общего числа на изображении. Для этого используется гистограмма яркости:

$$H(b) = \left| \left\{ \bar{p} : I(\bar{p}) \leq b \right\} \right|. \quad (1)$$

Анализируется часть гистограммы, соответствующая малым яркостям. Доли точек α_k , определяющих пороги бинаризации, задаются равномерно на участке от 0 до 1/4 общего числа точек:

$$\alpha_m = \frac{1}{4} \frac{k}{K}, \quad k = 1, \dots, K. \quad (2)$$

Пороги бинаризации B_k вычисляются, согласно (1) и (2):

$$B_k : H(B_k) = \alpha_k WH. \quad (3)$$

2.2. Бинаризация

Операция бинаризации проводится так, чтобы обнулить яркости, превышающие порог, и выделить пиксели с меньшими яркостями, с тем, чтобы далее рассматривать объекты, полученные из тёмных областей. Пример бинаризованного изображения приведён на рис. 1 (б).

2.3. Определение границ и моментов компонент связности

На бинаризованном изображении тёмные (с яркостью меньше порога) области формируют несколько компонент связности. Для каждой компоненты определяются упорядоченный список граничных точек и моменты нулевого, первого и второго порядка, которые используются далее для вычисления качества компоненты. Моменты определяются посредством перебора всех точек компоненты методом заливки. Каждая компонента, удовлетворяющая условиям (4), вносится в список, общий для всех порогов бинаризации.

2.4. Отбраковка и сортировка компонент

Выделенные на этапах бинаризации компоненты необходимо обработать преобразованием Хафа. Однако уже по статистическим признакам этих компонент до преобразования можно отбросить их часть, заведомо не содержащую зрачка, а оставшиеся отсортировать так, чтобы обрабатывать вначале компоненты, содержащие зрачок с наибольшей вероятностью. Для удаления заведомо не содержащих зрачок (или имеющих неадекватную форму) компонент связности используются параметры *эквивалентного эллипса*, имеющего те же моменты, что и компонента. Компонента связности, содержащая зрачок, должна иметь достаточно большую площадь, а отношение длины большой полуоси к малой не должно быть слишком большим. Кроме того, в случае, когда на изображении глаза ресницы/веки имеют яркость такую же или меньшую, чем зрачок, искажение формы компоненты связности, вносимое этим, вытягивает её по горизонтали, но не в вертикальном направлении. Соответственно, объекты с эквивалентным эллипсом, имеющим значительный эксцентриситет и большой осью, расположенной ближе к вертикали, должны быть отброшены. Эти условия записываются как:

$$\begin{aligned} ab &> \rho_{\min}^2, \\ \frac{a}{b} &< T_1, \\ \neg \left(\left(\frac{a}{b} > T_2 \right) \cap \left(\varphi \in \left[\frac{\pi}{4}; \frac{3\pi}{4} \right] \right) \right), \end{aligned} \quad (4)$$

где ρ_{\min} – минимальный возможный радиус зрачка в пикселях. Оптимальные значения порогов $T_1 = 5$, $T_2 = 2.5$ установлены экспериментально. Для сортировки оставшихся компонент вычисляется критерий качества:

$$\Theta(C) = \frac{M(C) - s(C)}{s^2(C)}, \quad (5)$$

где $s(C) = |L(C)|$ – длина границы компоненты связности, $M(C)$ – её масса. Данный признак увеличивается по мере приближения формы области к круговой, также он возрастает при увеличении размера области. Таким образом, при обработке компонент связности, отсортированных по убыванию

признака (5), предпочтение отдаётся округлым компонентам значительного размера.

2.5. Преобразование Хафа

Проводится процедура голосования: строятся внутренние нормали в каждом пикселе $\vec{l}_i \in L$, на луче нормали выбирается отрезок, ограниченный некоторыми значениями ρ_{\min} и ρ_{\max} , после чего значения элементов аккумуляторного пространства в точках отрезка увеличиваются на единицу:

$$A(\vec{p}) := A(\vec{p}) + u(\vec{p}, \vec{l}_i),$$

$$u(\vec{p}, \vec{l}_i) = \begin{cases} 1, & \text{if } \vec{p} = r\vec{\eta} + \vec{l}_i, r \in [\rho_{\min}; \rho_{\max}] \\ 0, & \text{otherwise} \end{cases}. \quad (6)$$

Вектора нормали $\vec{\eta}$ строится следующим образом: выбираются ближайшие соседи пикселя \vec{l}_i в списке L , обозначим их \vec{l}_j , где отрицательный индекс соответствует перемещению вверх по циклическому списку, положительный – перемещению вниз. По координатам выбранных пикселей аппроксимируется касательное направление $\vec{\tau}$:

$$\tau = \sum_{j=-n}^{-1} \frac{\vec{l}_i - \vec{l}_j}{|\vec{l}_i - \vec{l}_j|} + \sum_{j=1}^n \frac{\vec{l}_j - \vec{l}_i}{|\vec{l}_j - \vec{l}_i|}, \quad (7)$$

где n – величина окрестности пикселя, выбирается равной $\rho_{\min} / 2$. Из двух возможных нормальных направлений выбирается внутренняя нормаль, к направлению обхода. Отрезок голосования строится в растре аккумулятора алгоритмом Брезенхема. По окончании процедуры голосования значения в аккумуляторном пространстве сглаживаются фильтром низкой частоты. Центр гипотетической окружности находится как глобальный максимум в сглаженном аккумуляторе.

На рис. 3 (а) представлен результат описанного преобразования Хафа для наибольшей из компонент связности изображения рис. 2 (б), которая содержит зрачок. Это единственная обработанная компонента, остальные были отбракованы на предыдущем шаге.

2.6. Построение гистограммы расстояний до пикселей границы

Строится гистограмма $H(r)$ расстояний от найденного центра \vec{c} области до всех точек границы:

$$H(r) = \left| \left\{ \vec{l}_i : \vec{l}_i \in L, r - 0,5 \leq \|\vec{c} - \vec{l}_i\| < r + 0,5 \right\} \right|. \quad (8)$$

Гистограмма в случае принадлежности зрачка компоненте имеет специфический вид. Например, на рис. 3 (б), дана гистограмма для компоненты рис. 2 (б). Виден пик, соответствующий радиусу зрачка, присутствующего на изображении $r_p = \arg \max H(r)$. Окончательный вывод относительно того, содержится ли зрачок в компоненте, делается на основе анализа гистограммы. Вычисляется её масса в окне шириной

W с центром в точке r_p , где наблюдается максимальное среди всех значение гистограммы. Полученное значение представляет собой оценку длины видимого контура зрачка и сравнивается с r_p . Считается, что зрачок найден, если

$$\sum_{r=r_p-W}^{r_p+W} H(r) > \pi r_p, \quad (9)$$

то есть достаточным условием для обнаружения зрачка является видимость половины его окружности.

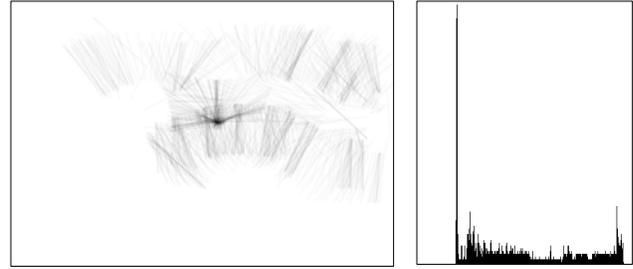


Рис. 3. Обработка компонент связности: (а) - результат голосования по компоненте изображения 1 (б) с наибольшим значением Θ и удовлетворяющей условиям (4); (б) - гистограмма расстояний (8) от гипотетического центра окружности до пикселей границы

Таким образом, метод состоит из следующих шагов:

Шаг 1. Построение гистограммы яркостей (1). Вычисление порогов бинаризации B_k , согласно (3).

Далее для каждого k выполняются шаги 2 - 3.

Шаг 2. Бинаризация с порогом B_k .

Шаг 3. Определение моментов и границ каждой из компонент связности, отбраковка несоответствующих критериям (4) и сохранение оставшихся в списке.

Шаг 4. Сортировка накопленных компонент связности согласно критерию качества (5).

Дальнейшие шаги 5 - 7 выполняются поочередно над элементами отсортированного списка, пока не будет встречен элемент, удовлетворяющий условиям (как правило, это первый элемент отсортированного списка) или не произойдет исчерпание списка.

Шаг 5. Преобразование Хафа (6) над границами, находится максимум аккумулятора.

Шаг 6. Если максимум достаточно велик, строится гистограмма расстояний (8) от его положения до точек границ.

Шаг 7. Если на гистограмме существует пик, то это значит, что существует округлый участок границы с центром, близким к положению максимума и радиусом, равным координате пика на гистограмме расстояний.

2.7. Оценка алгоритмической сложности

Высота и ширина изображения – величины одного порядка, поэтому для простоты обозначим линейный размер изображения W , а число точек W^2 . Вычисление порогов бинаризации состоит из построения гистограммы яркости и её обработки, сложность этого равна $\max\{W^2, Z\}$, где Z - число

градаций яркости изображения. Используются 8-битовые монохромные изображения, а значит, $Z = 256 \ll W^2$. Для одного шага бинаризации вычисление самой бинаризации и выделение компонент связности имеют сложность W^2 . Число компонент связности можно оценить сверху как W^2 / ρ_{\min}^2 , где ρ_{\min} - минимальный ожидаемый радиус зрачка. Длина границы компоненты связности оценивается как W . Сложность K проходов бинаризации составляет KW^2 , в результате выделяется KW^2 / ρ_{\min}^2 компонент связности. Голосование в процедуре Хафа для одной точки имеет сложность W , для всей границы одной компоненты W^2 . Поиск максимума в аккумуляторе также имеет сложность W^2 . Итак, с учётом числа компонент общая сложность работы всего метода составляет KW^4 / ρ_{\min}^2 .

3. ВЫЧИСЛИТЕЛЬНЫЕ ЭКСПЕРИМЕНТЫ

Для тестирования были использованы изображения открытых баз [3, 9]. Эти изображения были размечены экспертами: на каждом была построена окружность радиуса r_0 с центром в точке (x_0, y_0) , приближающая границу зрачка. Найденные алгоритмом центр (x, y) и радиус r окружности зрачка на каждом изображении сравнивались с экспертными, и вычислялись отклонения положения центра $|x - x_0| + |y - y_0|$ и величины радиуса $|r - r_0|$. В таблице приведено сравнение метода с некоторыми известными аналогами по средним отклонениям центра и радиуса.

Сравнение с некоторыми известными методами

База	Ошибка	Методы			
		Masek [4]	Ma et al. [3]	Daugman [10]	Данный
CASIA	центр	3,67	4,79	1,19	2,82
	радиус	5,15	5,39	3,02	1,26
NDARIS	центр	5,59	5,92	1,79	3,94
	радиус	7,23	7,39	3,11	1,89

4. ЗАКЛЮЧЕНИЕ

Предложен метод выделения контура зрачка на изображении глаза. Контур строится из точек, лежащих на границе одной из компонент связности множества тёмных точек, полученных при бинаризации изображения. Для выбора правильной (наиболее близкой к окружности) компоненты (или её части) используется преобразование Хафа для граничных точек, приводящее к выделению центра её округлой части. Радиус окружности определяется как максимум гистограммы расстояний граничных точек до выделенного центра. Сами точки контура зрачка затем выбираются из всего множества точек границы как имеющие близкое к этому радиусу расстояние до центра. Проведена апробация работы алгоритма на наборе изображений из открытых баз данных.

Использование преобразования Хафа позволяет корректно выделять истинные центр, радиус и контур зрачка даже при наличии больших помех, при условии лишь частичной видимости контура зрачка на изображении. Недостатком метода является потенциально большое время работы, возникающее при переборе многих компонент связности и нескольких порогов бинаризации. Этот недостаток частично скомпенсирован введением признака качества компоненты связности.

5. AKNOLEDGMENTS

Работа выполнена при финансовой поддержке РФФИ (проект №12-01-00893).

6. REFERENCES

- [1] Bakshi S., Mehrotra H., Majhi B. Real-time Iris Segmentation based on Image Morphology // Proc. Int. Conf. Communication, Computing and Security. Odisha, India, February 12-14, 2011.
- [2] Chen D., Bai J., and Qu Z. Research on pupil center location based on improved Hough transform and edge gradient algorithm. Proc. National Conf. Information Technology and Computer Science, 2012. – P. 47 – 51,
- [3] Chinese Academy of Sciences Institute of Automation. Iris image database, version 3, 2005.
- [4] J. Daugman. New methods in iris recognition. IEEE Trans. Systems, Man and Cybernetics-Part B: Cybernetics, 37:1167 – 1175, 2007.
- [5] Ma L., Wang Y. and Tan T. Iris recognition using circular symmetric filters. Proc. 16th Int. Conf. Pattern Recognition, 1(2):414 – 417, 2002.
- [6] L. Masek. Recognition of human iris patterns for biometric identification. <http://www.csse.uwa.edu.au/pk/studentprojects/libor>, 2003.
- [7] Mohammed G.J., Hong B.R. and Jarjes A.A. Accurate pupil features extraction based on new projection function. Computing and Informatics, 29:663 – 680, 2010.
- [8] L. Pan, W.S. Chu, J.M. Saragih, and F. de la Torre. Fast and robust circular object detection with probabilistic pairwise voting. Signal Processing Letters, 18(11):639–642, 2011.
- [9] Phillips P., Scruggs W. and O’Toole A. et al. Frvt 2006 and ice2006 large-scale experimental results. IEEE PAMI, 5(32):831–846, 2010.
- [10] Ritter N.J. and Cooper J.R. Locating the iris: A first step to registration and identification. Proc. 9th IASTED Int. Conf. Signal and Image Processing, 2003. – P. 507 – 512.

Информация об авторах

Матвеев Иван – к.ф.-м.н., зав. сектором ВЦ РАН.

Чинаев Николай – студент МФТИ.

Алгоритмы построения и хранения навигационной когнитивной карты для взаимодействия с человеком

М. Пестун

Институт прикладной математики им. М.В. Келдыша Российской академии наук

max.pestun@gmail.com

Аннотация

В данной статье рассматриваются алгоритмы построения и хранения навигационных когнитивных карт, предоставляющих удобное для человека описание маршрута из точки А в точку В (в городском окружении). Система учитывает персональные знания пользователя об объектах недвижимости и организациях. В тех случаях, когда местность незнакома или малознакома пользователю, в описании маршрута используются популярные, с точки зрения общественного мнения, объекты.

Ключевые слова: когнитивная навигационная карта, алгоритмы построения и хранения, взаимодействие с человеком.

1. ВВЕДЕНИЕ

В данной работе представлены алгоритмы построения и хранения навигационных когнитивных карт. Задача когнитивной карты: предоставление удобного для человека описания маршрута из одной точки в другую (преимущественно в городском окружении, однако алгоритмы не накладывают жестких ограничений) в текстовом виде, обладающем лаконичностью, легкостью запоминания и максимальной однозначностью. Алгоритмы построения и хранения карты нацелены на использование на мобильных устройствах, учитывают известные пользователю места на основе его передвижений. Иначе, для неизвестных мест учитывается общая популярность окружающих пользователя объектов на основе данных из открытых источников.

Целью статьи является разработка методов построения когнитивных навигационных карт (также создание методов навигации по ним) для упрощения взаимодействия человека с компьютером в области описания маршрута, а также реализация алгоритмов построения и хранения навигационной когнитивной карты, используемой в основном исследовании.

В наши дни карты не подстраиваются под конкретного пользователя и не выделяют важную только для него информацию. Ввиду этого усложняется их использование.

Кроме того, следует отметить, что существующие методы хранения карт обладают избыточной информацией, поэтому с для уменьшения их объема необходимо сконцентрироваться только на важных для конкретного человека деталях, расставив им приоритеты, и, при необходимости, отбрасывать маловажные [2].

Проблема описания маршрута актуальна и в настоящее время недостаточно исследована. Подробнее об актуальности с приведением примеров использования в реальной жизни можно ознакомиться в обзорной статье исследования [1].

2. СТРУКТУРА ДАННЫХ КОГНИТИВНОЙ КАРТЫ В ХРАНЕНИЯ ПАМЯТИ

Структура данных для представления когнитивной карты представляет собой связный граф с циклами [3, 8]:

- ребра графа представляют собой дороги и другие пути, по которым может передвигаться человек;
- вершины графа символизируют объекты недвижимости (здания, сады, офисы компаний и т.д., обычно это всякие вершины), а также пересечения дорог.

Ребра имеют набор параметров, характеризующих элемент пути:

- ширину (популярность) дороги;
- название улицы, если оно применимо;
- направление (актуально для одностороннего движения);
- статистические данные, основанные на перемещении пользователя карты и другие персональные параметры.

Вершины, представляющие собой перекрестки дорог, обладают следующими параметрами:

- географическими координатами;
- углами, под которыми выходят ребра.

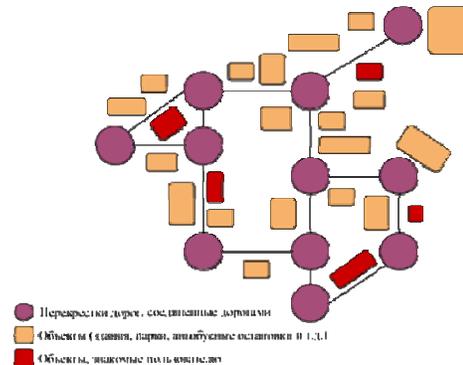


Рис. 1. Пример графа когнитивной карты (ребра, соединяющие вершины, описывающие объекты, не отображены для облегчения изображения)

Вершины, представляющие собой объекты, обладают параметрами:

- географическими координатами;
- названием (или несколькими), если оно применимо (адрес);
- словесным описанием;

- находящимися внутри организациями;
- другими индивидуальными для каждого пользователя параметрами (частотой посещения, предпочтениями, известностью и т.д.).

2.1. Программная реализация

С программной точки зрения структура данных для хранения когнитивной карты в памяти компьютера реализована следующим образом:

- база данных (может быть использована любая, но в нашем случае для быстроты реализации и простоты использования была выбрана реляционная СУБД SQLite; кроме того, ее реализация присутствует на мобильных платформах, что позволяет делать хранилище кроссплатформенным);
- Программная оболочка для доступа к данным API.

Структура таблиц выглядит следующим образом:

- Таблица «edges-nodes» для хранения ребер и вершин графа;
 - Столбцы:
 - идентификатор ребра или вершины;
 - булево значение (ответ на вопрос «вершина ли это?»).
 - Индекс: по столбцу с идентификатором.
- Таблица «graph» для хранения информации о связях ребер и вершин:
 - Столбцы:
 - идентификатор связи,
 - идентификатор вершины,
 - идентификатор ребра.
 - Индекс: по всем столбцам.
- Таблица «data» для хранения различной дополнительной информации (обычно такие таблицы называют «ресурсной системой»):
 - Столбцы:
 - идентификатор записи;
 - идентификатор вершины или ребра (из таблицы «edges-nodes»);
 - идентификатор поля (из таблицы «data-fields»);
 - значение поля в текстовом виде.
 - Таблица «data-fields» для хранения имен допустимых полей в таблице «data»:
 - Столбцы:
 - идентификатор поля;
 - строковое представление имени поля.
 - Индекс: по столбцу с идентификатором.

В таблице «data» хранится вся информация о ребрах и вершинах, перечисленная в разделе выше: названия, параметры и т.д. Текстовое представление значений полей вполне допустимо с учетом современных вычислительных мощностей. Кроме того, такой подход делает систему хранения информации очень гибкой и легко расширяемой: нет четких ограничений на хранимую информацию, нет ограничений на количество полей для каждого ребра/вершины.

В случае коллизионных ситуаций, когда на одно и то же ребро/вершину одновременно назначено несколько одинаковых полей (например, имя или длина), то выбирается последнее. Такой подход хорошо зарекомендовал себя на высоконагруженных системах, так как позволяет быстро переписать данные, не удаляя старые.

3. АЛГОРИТМ ПОСТРОЕНИЯ НАВИГАЦИОННОЙ КОГНИТИВНОЙ КАРТЫ В ПАМЯТИ КОМПЬЮТЕРА

В общем виде построение навигационных когнитивных карт, хранимых в памяти компьютера, подразделяется на этапы:

- а) формирование базовой карты: преобразование информации из обычной карты с видом сверху в структуру данных когнитивной карты, получение персонализированной информации о перемещениях пользователя;
- б) сбор данных о пользователе когнитивной карты;
- с) объединение результатов пунктов а) и б) для персонализации карты и уменьшения объема хранимых данных.

3.1. Формирование базовой карты

Частично процесс формирования карты происходит автоматически на основе существующих карт (например, «Google Maps» или «Яндекс.Карты»), содержащих избыточную информацию, как о самой схеме дорог и адресах зданий, так и об организациях, расположенных внутри. Кроме того, из них можно получить информацию о популярности организации, типе предлагаемых услуг, а также оценку ее деятельности (обычно в баллах – от 1 до 5).

3.2. Сбор данных о пользователе

Для создания персонализированной когнитивной карты для конкретного пользователя требуется сбор дополнительной подробной информации о нем:

- пол;
- возраст;
- вкусовые предпочтения;
- основные места пребывания;
- другие.

Получение таких данных возможно из социальных сетей либо непосредственно при помощи их ввода самим пользователем в интерфейс программы.

Дополняются карты информацией, полученной при непосредственном перемещении пользователя по миру. Происходит автоматическая запись координат положения мобильного телефона, их последующий анализ и запись в структуру данных карты следующим образом:

- отслеживаются промежутки времени, при которых телефон был относительно неподвижен (остановки);
- отслеживаются быстрые (движение на автомобиле или в общественном транспорте) и медленные (перемещение пешком) изменения координат;
- при остановках дополнительно уточняется у пользователя, какое конкретно место (заведение, фирму, магазин) он посещает (аналогично сделано в социальной сети «foursquare»).

Список мест для посещения доступен благодаря тем же картографическим сервисам, предлагающим доступ к накопленной информации.

Дополнительно в структуру данных когнитивной карты заносится следующая информация:

- частота перемещений по конкретному маршруту или его участку;
- частота посещения различных мест;
- эмоциональное отношение пользователя к конкретному участку дороги и/или месту.

Все три перечисленные выше пункта позволяют более точно персонализировать карту, сделав ее максимально удобной для использования. Благодаря им:

- при прокладывании маршрута будут использоваться статистические частотные данные, позволяющие проложить маршрут, наиболее знакомый пользователю, несмотря на возможную меньшую оптимальность траектории пути (из предположения, что чем чаще человек пользуется одним и тем же маршрутом, тем более хорошо он ему знаком и может быть описан менее детально, нежели новый неизвестный) [7];
- вводимое пользователем в ручном режиме эмоциональное отношение к участкам пути и/или конкретным объектам на нем дает возможность создать такой вариант маршрута, который будет казаться ему более коротким и оптимальным [2, 4, 5] (основано на подтвержденных исследованиях в системе виртуальной CAVE, проводимых факультетом психологии МГУ им. М.В. Ломоносова [6]).

3.3. Программная реализация

С программной точки зрения структура данных для хранения накопленной о пользователе информации реализована следующим образом:

- база данных (аналогично с пунктом 2.1 данной статьи);
- программная оболочка для доступа к данным API.

Структура таблиц выглядит следующим образом:

- Таблица «user» для хранения редко изменяемой информации о пользователе; таблица организована по типу «ресурсной системы», описанной выше в пункте 2.1.
- Таблица «moves-gaw» содержит исходные данные по всем перемещениям пользователя:
 - Столбцы:
 - идентификатор записи;
 - долгота;
 - широта;
 - временная метка.
 - Индекс: по столбцу с идентификатором.
- Таблица «moves» содержит обработанную информацию из таблицы «moves-gaw»:
 - Столбцы:
 - идентификатор записи;
 - идентификатор ребра или вершины, к которому относится данная запись;
 - тип записи (быстрое перемещение, медленное перемещение, остановка);
 - Индекс: по столбцам с идентификаторами.
- Таблица «prefeg» содержит информацию о предпочтительных местах или маршрутах:
 - Столбцы:

- идентификатор записи;
- индекс предпочтения (целое число без знака, где 0 самая низкая оценка популярности/предпочтения);
- индекс эмоционального отношения (целое число со знаком, 0 в качестве нейтрально отношения);
- идентификатор ребра или вершины, к которому относится данная запись;
- идентификатор группы записей (для объединения записей в маршрут).
- Индексы: по столбцам с идентификаторами.
- Таблица «custom» для хранения дополнительной информации о местах и маршрутах; таблица организована по типу «ресурсной системы», описанной выше в пункте 2.1; в основном необходима для расширения системы и быстрого прототипирования нового функционала.

Программная составляющая для управления данными реализована на языке программирования Java и не является предметом рассмотрения данной статьи.

4. ПРАКТИЧЕСКОЕ ПРИМЕНЕНИЕ

Описанные выше алгоритм построения карты и структура данных для ее хранения были частично применены на практике для навигации внутри высотного офисного здания «SkyLight» в городе Москве.

Система когнитивной навигации была интегрирована в корпоративный портал. Любой сотрудник может найти другого сотрудника в адресной книге, просмотреть его личную карточку, узнать место его посадки, а также получить краткую и легкую для запоминания инструкцию, «как пройти» к нему.

Система на данный момент не использует данных о пользователе для представления результатов: все пользователи увидят один и тот же маршрут. Несмотря на это она показала свою востребованность у сотрудников, преимущественно у новичков и персонала из отдела кадров, которым приходится часто перемещаться по зданию.

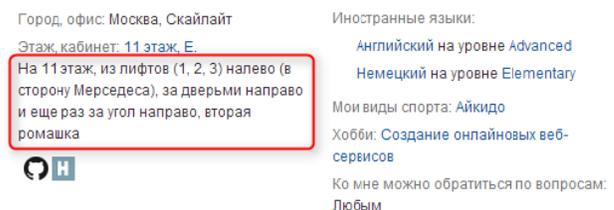


Рис. 2. Пример предложенного пользователю текстового описания маршрута прохода к рабочему месту коллеги

В текущий момент идет разработка мобильного приложения для платформы Android, реализующего описанные выше алгоритмы и структуры данных для хранения когнитивной навигационной карты. По завершению работ над приложением, оно будет опубликовано в Google Play в свободном бесплатном доступе.

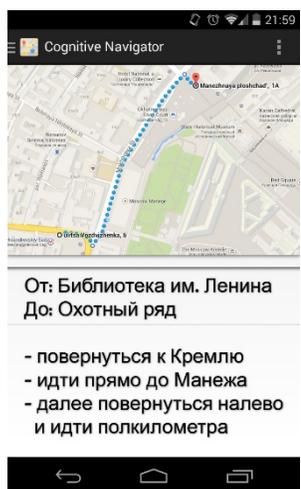


Рис. 3. Прототип интерфейса приложения для платформы Android для предоставления описания маршрута в текстовом виде, удобном для быстрого запоминания

5. ЗАКЛЮЧЕНИЕ

Развитие технологий голосового взаимодействия человека и машин (компьютеров, роботов) приведет к ситуациям, когда станет необходима передача знания о маршруте от машины к человеку и, наоборот, в форме, удобной именно человеку. Вычислительная система всегда может быть приспособлена к особенностям восприятия и передачи информации человеком, но не наоборот. Именно для этих проблемных ситуаций и разрабатывается комплекс систем по описанию маршрута в удобном для человека формате, часть которого была рассмотрена в данной статье. Подробнее про само исследование можно прочитать в обзорной статье исследования [1].

6. ССЫЛКИ

- [1] Пестун М.В. Компьютерная система описания маршрута в удобном для человека формате. Научно-практический семинар "Новые информационные технологии в автоматизированных системах". – Москва, Россия, 2014. – С. 125 - 134.
- [2] Allen G.L., Kirasic K.C., Siegel A.W. and Herman J.F. Development Issues in Cognitive Mapping: The Selection and Utilization of Environmental Landmarks, *Child Development* 50, 1979. – P. 1062 - 1070.
- [3] Cousins J.H., Siegel A.W. and Maxwell S.E. Way-Finding And Cognitive Mapping in Large-Scale Environments: A Test of a Developmental Model, *Journal of Experimental Child Psychology* 35, 1983. – P. 1 - 20.
- [4] Klatzky R.L., Freksa C., Habel C., Wender K.F. Allocentric and Egocentric Spatial Representations: Definitions, Distinctions, and Interconnections, *Artificial Intelligence* (Eds.), Springer, 1997. – V. 1404. – P. 1 - 17.
- [5] Klatzky R.L., Loomis J.M., Golledge R.G., Cicinelli J.G., Doherty S. & Pellegrino J.W. Acquisition of route and survey knowledge in the absence of vision. *Journal of Motor Behavior*, 22, 1990. – P. 19 - 43.

- [6] 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.
- [7] Loomis J.M., Klatzky R.L. & Golledge R.G. (in press). Human navigation by path integration. In R. Golledge (Ed.), *Wayfinding: Cognitive mapping and spatial behavior*. Baltimore, MD: Johns Hopkins University, 1999.
- [8] Yeap W.K., Jefferies M.E. On early cognitive mapping. *Spatial Cognition and Computation*, The Netherlands: Kluwer Academic Publishers, 2000. – V. 2(2). – P. 85 - 116.

Об авторах

Пестун Максим – аспирант Института прикладной математики им. М.В. Келдыша Российской академии наук.
E-mail: max.pestun@gmail.com.

Кластеризация пикселей для сегментации цветового изображения

М. Харинов

Федеральное государственное бюджетное учреждение науки Санкт-Петербургский институт информатики
И автоматизации Российской академии наук (СПИИРАН)
khar@iias.spb

Аннотация

В докладе рассматривается проблема сегментации изображения посредством иерархической последовательности кусочно-постоянных приближений, минимально отличающихся от изображения по суммарной квадратичной ошибке. Предлагается получать приближения комбинированным методом кластеризации/сегментации на основе метода Уорда кластеризации пикселей изображения, в котором предусматривается снижение числа сегментов в бинарной иерархии кластеров за счет реклассификации пикселей из одного кластера в другой. В случае цветowych изображений, предлагаемый метод применяется сначала для выравнивания и снижения числа кластеров пикселей в каждой из RGB-компонент, а затем для кластеризации, в частности, сегментации, цветового изображения в целом. В результате достигается заметное улучшение качества приближений изображения, а их кодирование в инвариантном виде обеспечивает разметку изображения для обнаружения объектов.

Ключевые слова: изображение, приближение, бинарная иерархия, минимизация, суммарная квадратичная ошибка, метод Уорда, метод Оцу, модель Мамфорда–Шаха.

1. ВВЕДЕНИЕ

Доклад относится к области распознавания цифровых изображений на первоначальной стадии «приведения данных к удобному для распознавания виду» [6], «локализации объектов» [5], извлечения и упорядочения «глобально-локальной информации» [1]. Более конкретно, речь идет о *проблеме сегментации*, трактуемой нами как стадия автоматического выделения на изображении, доступных для анализа и распознавания вложенных изображений, именуемых *объектами*, которые обнаруживает на исходном изображении компьютер. Допускается, что в общем случае, пиксели объекта составляют на изображении *кластер* из того или иного числа несмежных связанных сегментов. Таким образом, в нашем понимании, термины «изображение», «объект», «кластер» употребляются в сходном смысле.

Если изображение состоит из изображений, то последние, в свою очередь, также разделяются на изображения, и сегментация выполняется посредством итеративного разделения изображения до тех пор, пока оно не распадется на некоторое множество *тривиальных* кластеров из одинаковых пикселей, которые полагаются неделимыми. Таким образом, мы вычисляем не единственное разбиение изображения, как при традиционной сегментации, а иерархию разбиений изображения на объекты, которая отражает неоднозначность изображения. Другим отличием нашей обобщенной интерпретации сегментации является то, что для каждого объекта автоматически генерируется целочисленная *инвариантная* метка, обозначение объекта, которое не зависит от предусмотренных преобразований или искажений исходного изображения. При этом числовые значения инвариантных меток кодируют иерархию сопоставляемых им объектов.

2. ПОСТАНОВКА ЗАДАЧИ

Для нахождения иерархии объектов решается задача оптимизации качества требуемых разбиений [2], которое при каждом значении числа кластеров k оценивается по величине суммарной квадратичной ошибки $E^{(k)}$ изображения относительно своего *кусочно-постоянного приближения* с усредненными по кластерам значениями пикселей. Так как последовательность оптимальных приближений изображения, описываемая минимально возможными значениями суммарной квадратичной ошибки $E_{\min}^{(k)}$, в общем случае, не является иерархической, возникает задача аппроксимации оптимальных приближений иерархией *квазиоптимальных* приближений, описываемых значениями $E^{(k)}$:

$$E^{(k)} \approx E_{\min}^{(k)}, \quad k = 1, 2, 3, \dots, g, \quad (1)$$

где g – общее число *тривиальных* кластеров.

При этом полагается, что для последовательности $E^{(k)}$ сохраняется свойство выпуклости:

$$E^{(k)} \leq \frac{E^{(k-1)} + E^{(k+1)}}{2}, \quad k = 2, 3, \dots, g-1, \quad (2)$$

которое характерно для мажорируемой последовательности оптимальных значений $E_{\min}^{(k)}$.

Если для кластеров пикселей устанавливается условие, что они не должны состоять из нескольких несмежных сегментов, то задача кластеризации сводится к задаче иерархической сегментации изображения, т.е. получению последовательности приближений изображений разбиениями с $k = 1, 2, 3, \dots, g$ связными сегментами, и соответствующей последовательности $\tilde{E}^{(1)}, \tilde{E}^{(2)}, \tilde{E}^{(3)}, \dots, \tilde{E}^{(g)}$ значений суммарной квадратичной ошибки. Если условие не минимизирует, а только ограничивает число связанных сегментов в составе кластера, то ставится задача получения приближений с промежуточными значениями $\tilde{E}^{(1)}, \tilde{E}^{(2)}, \tilde{E}^{(3)}, \dots, \tilde{E}^{(g)}$ суммарной квадратичной ошибки:

$$\tilde{E}^{(k)} \geq \tilde{E}^{(k)} \geq E^{(k)} \geq E_{\min}^{(k)}, \quad k = 1, 2, 3, \dots, g. \quad (3)$$

По существу, требуется алгоритм комбинированной кластеризации/сегментации, который в одном предельном случае обеспечивает иерархическую квазиоптимальную кластеризацию пикселей, а в другом – иерархическую *сегментацию* изображения, при которой число кластеров совпадает с числом связанных сегментов.

3. АНАЛИТИЧЕСКОЕ ОБОСНОВАНИЕ

Суммарная квадратичная ошибка $E^{(k)}$ для приближения из k кластеров вычисляется суммированием суммарных квадратичных отклонений E_i значений пикселей каждого кластера от их среднего значения:

$$E^{(k)} = \sum_{i=1}^k E_i, \quad (4)$$

Если i -й кластер рассматривать как результат слияния j вложенных кластеров $i = 1 \cup 2 \cup \dots \cup j$, то суммарное квадратичное отклонение E_i для i -го кластера получается суммированием квадратичных отклонений E_p для вложенных кластеров и некоторой неотрицательной добавки, характеризующей средний взвешенный разброс квадратов евклидовых попарных расстояний $\|I_p - I_q\|^2$ между центрами I_p, I_q кластеров p и q из n_p и n_q пикселей в цветовом RGB пространстве [12]:

$$E_i = \sum_{p=1}^j E_p + \frac{1}{2} \cdot \frac{\sum_{p=1}^j \sum_{q=1}^j n_p n_q \|I_p - I_q\|^2}{n_i}, \quad (5)$$

где $n_i = \sum_{p=1}^j n_p$ – число пикселей в i -м кластере.

Если в качестве j вложенных кластеров иметь в виду тривиальные кластеры, на которые распадается обсуждаемый i -й кластер, то в (5) сумма E_p , очевидно, обращается в 0.

Если в качестве тривиальных кластеров рассматриваются отдельные пиксели, то в (5) следует также положить $n_p = n_q = 1$ и $j = n_i$. Тогда (5), с точностью до множителя, совпадает с выражением для дисперсии значений пикселей i -го кластера, которое активно применяется в [8 – 10] для теоретической и практической разработки методов кластеризации. Таким образом, (5) обобщает аналогичную формулу из [8 – 10] на случай тривиальных кластеров из нескольких или многих пикселей.

Для случая $j = 2$ пары вложенных кластеров, из (4), (5) несложно получить [12] выражения для приращений суммарной квадратичной ошибки ΔE_{merge} и $\Delta E_{correct}$ при слиянии пары кластеров 1 и 2, а также их коррекции посредством реклассификации пикселей кластера $3 \subset 1$ из кластера 1 в кластер 2:

$$\Delta E_{merge} = \frac{\|I_1 - I_2\|^2}{\frac{1}{n_1} + \frac{1}{n_2}}, \quad (6)$$

$$\Delta E_{correct} = \frac{\|I_3 - I_2\|^2}{\frac{1}{n_3} + \frac{1}{n_2}} - \frac{\|I_3 - I_1\|^2}{\frac{1}{n_3} - \frac{1}{n_1}}, \quad (7)$$

где n_1, n_2, n_3 – число пикселей, в кластерах 1, 2, 3, а I_1, I_2, I_3 – трехмерные центры кластеров 1, 2, 3.

Именно выражение (6) минимизируется при слиянии сегментов изображения в версиях [4, 12] модели сегментации Мамфорда-Шаха [14], и либо с дополнительным множителем, либо с аддитивной добавкой используется в ее версиях [7, 13]. Формула (7) интересна тем, что из нее выводится классический метод K -means [2] и обобщенный метод K -meanless [10], который в нашей версии [12] отличается тем, что предусматривает операции с наборами пикселей.

4. ИНВАРИАНТНОЕ ПРЕДСТАВЛЕНИЕ

Для автоматической разметки поля изображения целочисленными метками, обозначениями объектов используется *инвариантное представление* изображения, разработанное и запатентованное в задачах стеганографии [11]. Представление $H(u, v, w)$ цветового изображения (u, v, w) из трех RGB -компонент называется *инвариантным*, так как не зависит от умножения значений пикселей изображения на положительное число $\alpha > 0$ и сдвига значений пикселей любой компоненты на одинаковую величину:

$$H(u, v, w) = H(\alpha \cdot u, \alpha \cdot v, \alpha \cdot w) = H(u + c_R, v + c_G, w + c_B), \quad (8)$$

где c_R, c_G, c_B – *тривиальные* компоненты изображения, состоящие из одинаковых пикселей.

Преобразование H коммутирует с преобразованием P :

$$HP = PH, \quad (9)$$

где под преобразованием P можно понимать: а) произвольную перестановку цветовых компонент; б) преобразование цветовой компоненты в негатив; в) масштабирование изображения посредством дублирования пикселей. Перестановочность аргументов в $H(u, v, w)$ означает, что функция преобразования цветовой компоненты зависит только от двух других компонент и не зависит от номера компоненты:

$$H_{u,v} : H(u, v, w) \equiv (H_{v,w}u, H_{u,w}v, H_{u,v}w), \quad (10)$$

причем, не меняется при перестановке индексов: $H_{u,v} = H_{v,u}$. При этом преобразование цветовой компоненты серого изображения из одинаковых компонент совпадает с преобразованием данной цветовой компоненты при двух других тривиальных:

$$H_{c,c}u = H_{u,u}u, \quad (11)$$

где c – *тривиальная* матрица из одинаковых пикселей, например, из одних 0. Таким образом, программная реализация преобразования H цветового изображения поддерживает одно и то же преобразование одномерной компоненты в составе серого изображения из трех одинаковых цветовых компонент или монохромного изображения при обнулении остальных.

При заданном алгоритме разделения нетривиального кластера на пару вложенных инвариантное представление $(H_{v,w}u, H_{u,w}v, H_{u,v}w)$, по существу, является результатом целочисленной «оцифровки» средних яркостей вложенных кластеров. Оно строится посредством кодирования знаков разностей средних яркостей вложенных кластеров в итеративном алгоритме, подобном алгоритму получения двоичного кода числа, в котором для учета при кодировании неделимых тривиальных кластеров используется не двоичное, а троичное значение коэффициентов разложения по степеням 2. При этом по результирующему инвариантному представлению вычисляются предыдущие – посредством арифметических операций [11].

5. КЛАСТЕРИЗАЦИЯ И СЕГМЕНТАЦИЯ

Комбинированный метод сегментации изображения посредством кластеризации пикселей поясняется на примере стандартного изображения «Лена» из 256×256 пикселей и трех одинаковых RGB -компонент (рис. 1).

На рис. 1 показаны кусочно-постоянные приближения изображения, в которых усредненные по кластерам значения

пикселей на первом шаге кодирования [11] просто заменены значениями 0 и 2 числовых кодов инвариантного представления яркости и нормализованы на рабочий диапазон.



Рис. 1. Кластеризация пикселей и сегментация изображения в инвариантном представлении

Суть обсуждаемого метода состоит в том, что сначала для изображения методом Уорда [16] строится квазиоптимальная бинарная иерархия кластеров, которая получается итеративным слиянием кластеров начального разбиения изображения на пиксели из условия $\Delta E_{merge} = \min$, где ΔE_{merge} раскрывается в (6). Затем берется разбиение множества пикселей изображения на пару вложенных кластеров (рис. 1, слева), и число сегментов итеративно редуцируется за счет реклассификации пикселей неизолированных сегментов из одного вложенного кластера в другой. Очередность реклассификации устанавливается из условия $\Delta E_{correct} = \min$, где $\Delta E_{correct}$ вычисляется согласно (7). Коррекция пары вложенных кластеров по числу сегментов продолжается до выполнения установленного правила останова, например, условия неповторяющихся площадей сегментов (рис. 1, в центре), или, при безусловной коррекции, продолжается до тех пор, пока во вложенных кластерах не останется по одному сегменту (рис. 1, справа).

По завершению формирования вложенных кластеров, каждый из них рассматривается как самостоятельное изображение, для которого обновляется иерархия вложенных кластеров, и вся процедура повторяется до тех пор, пока множество пикселей изображения не распадется на *тривиальные* кластеры из одинаковых пикселей. По мере формирования иерархии кластеров пикселей продолжает формироваться и инвариантное представление, которое визуально выглядит как приближение изображения на очередной итерации, так как связано с ним *изотонным* преобразованием, сохраняющим яркостный порядок [11].

В отличие от модели Мамфорда-Шаха [4, 7, 13, 14], в методе Уорда [16] при минимизации суммарной квадратичной ошибки $E^{(k)}$ принимаются во внимание все пары кластеров. Однако в частном случае серого изображения эквивалентный результат получается при анализе только пар кластеров с последовательными значениями средней яркости. При этом квазиоптимальные приближения (рис. 1, слева) можно получать в простейшем гистограммном алгоритме [3], если вместо эвристического критерия слияния [3] воспользоваться критерием $\Delta E_{merge} = \min$.

С другой стороны, в случае серого изображения для разделения очередного кластера надвое можно воспользоваться классическим методом Оцу [15], что представляет интерес для сравнения (рис. 2).

Рис. 2 иллюстрирует поведение суммарного числа сегментов s в кластерах и среднее квадратичное отклонение

$$\sigma^{(k)} = \sqrt{\frac{E^{(k)}}{3 \cdot N}}$$

приближения от изображения в зависимости от числа k кластеров в изображении из N пикселей и трех одинаковых *RGB* компонент. Сплошные серые кривые описывают наш метод при выполнении разделения кластера

надвое посредством обращения последнего слияния в методе Уорда, а черные пунктирные кривые описывают разделение кластера надвое методом Оцу.

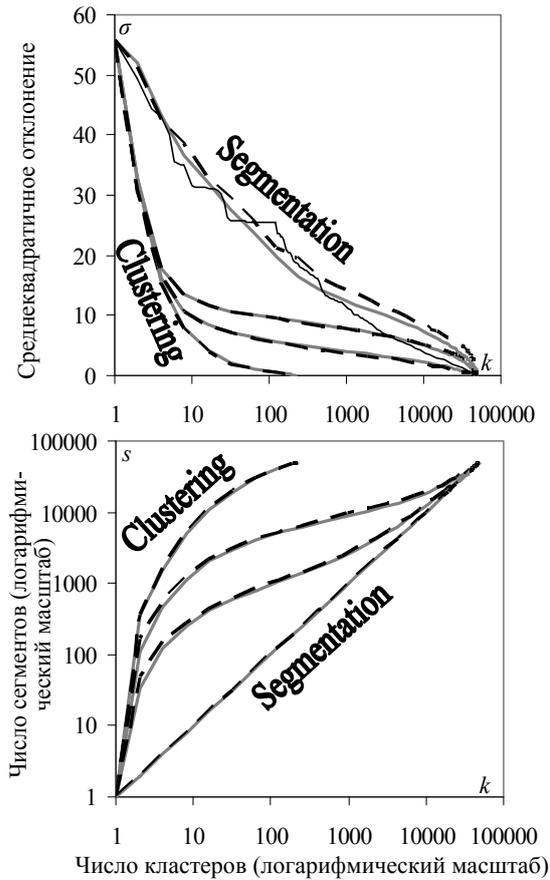


Рис. 2. Сопоставление методов по качеству приближений σ (вверху) и числу сегментов s (внизу) в зависимости от числа кластеров k

На верхних графиках рис. 2 тонкая сплошная черная кривая описывает результаты сегментации в модели Мамфорда-Шаха по версии FLSA [7]. Переплетающиеся с ней серая и пунктирные кривые описывают сегментацию изображения посредством кластеризации с безусловной коррекцией кластеров. Нижняя пара кривых описывает квазиоптимальную кластеризацию пикселей методами Уорда и иерархическим методом Оцу без коррекции кластеров по числу сегментов. Промежуточные пары кривых описывают кластеризацию пикселей по очереди с итеративной коррекцией, до останова при отсутствии повторений сегментов по площади и средней яркости (третья и вторая снизу пара кривых).

На нижних графиках рис. 2 соответствующие кривые следуют в обратном порядке. Верхняя пара кривых отражает лавинообразное возрастание числа сегментов с ростом числа кластеров при кластеризации без коррекции. Совпадение числа сегментов с числом кластеров при сегментации изображения описывается нижней прямой. Промежуточные кривые описывают подавление возрастания числа сегментов за счет коррекции разделения кластеров.

В целом, наш метод позволяет получать приближения изображения с заранее установленным качеством, так как область между кривыми на рис. 2 оказывается доступной при варьировании правил останова коррекции. Для серого изображения варианты комбинированной иерархической кластеризации/сегментации на основе метода Уорда и иерархического метода Оцу являются взаимно-

дополнительными, причем коррекция кластеров не нарушает согласованности результатов, как по качеству разбиений, так и по количеству сегментов (рис. 2).

Так как метод Оцу [15] не имеет очевидного обобщения для цветового изображения, то в этом случае из двух вариантов нашего метода остается применимым только метод кластеризации/сегментации на основе метода Уорда, который поясняется на примере обработки цветового изображения «Лена» из 512×512 пикселей (рис. 3).

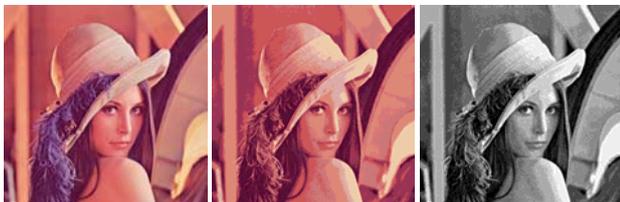


Рис. 3. Цветовое изображение (слева) и его квазиоптимальное приближение в исходном (в центре) и в инвариантном (справа) представлениях

Демонстрируемые на рис. 3 представления получены методом Уорда сначала снижением числа цветов в изображении с 148279 до 3434 за счет снижения числа тонов в каждой цветовой компоненте исходного изображения до 25. Затем, повторным применением метода Уорда для кластеризации трехмерных пикселей рассчитаны 3434 квазиоптимальных приближения изображения и столько же инвариантных представлений. Из них на рис. 3 показано приближение изображения в 8 цветах вместе с соответствующим инвариантным представлением, которое является серым:

$$H_{v,w}^u = H_{u,v}^w = H_{u,v}^w, \quad k = 1, 2, \dots, 8, \quad (12)$$

что не является тривиальным свойством, так как оно нарушается, например, при преобразовании одной из исходных цветовых компонент в негатив. Тем не менее, как показывает эксперимент, совпадение компонент инвариантного представления характерно для начального числа кластеров и согласуется с тем, что образы одних и тех же объектов в различных цветовых компонентах помечаются одинаковыми метками. Обнаруженное свойство планируется проиллюстрировать в докладе на примере выделения объектов на дистанционном изображении из 15 млн. пикселей.

6. ЗАКЛЮЧЕНИЕ

Скоростные вычисления с миллионами приближений изображения выполняются в ограниченном объеме оперативной памяти благодаря интенсивному применению структуры данных динамических деревьев Слэйтора-Тарьяна, которые не имеют альтернатив при программировании операций с произвольными множествами пикселей [11, 12], но, оптимизируя вычисления, никак не влияют на прозрачный смысл алгоритмов. Для содержательной разработки, развития и обоснования программно-алгоритмического обеспечения гораздо более важна интерпретация инвариантного представления как массива кодов информации, содержащихся в пикселях изображения [11]. Упомянутая интерпретация позволяет предложить целочисленную оценку количества информации в пикселе изображения и сопоставить ее с классическими оценками, что планируется обсудить в дальнейших работах.

7. ССЫЛКИ

[1] Абламейко С.В., Недзьведь А.М., Белоцерковский А.М., Руцкая Е.А. Сегментация трехмерных изображений компьютерной томографии на основе гло-

бально-локальной информации // Вестник Белорусского государственного университета. Сер. 1. Физика. Математика. Информатика, 2009. – № 1. – С. 58 - 64.

- [2] Айвазян С.А., Бухштабер В.М., Енюков И.С., Мешалкин Л.Д. Прикладная статистика: классификация и снижение размерности. – М.: Финансы и статистика, 1989. – 607 с.
- [3] Arifin A.Z., Asano A. Image segmentation by histogram thresholding using hierarchical cluster analysis // Pattern Recogn. Letters 2006. Vol. 27. №. 13. – P. 1515 - 1521.
- [4] Бугаев А.С., Хельвас А.В. Поисковые исследования и разработка методов и средств анализа и автоматического распознавания потоковой информации в глобальных информационных системах. Шифр «Лацкан» // Отчет по НИР – М.: МФТИ, 2001. – Т 1. – 140 с.
- [5] Визильтер Ю.В., Желтов С.Ю. Проблемы технического зрения в современных авиационных системах // Механика, управление и информатика, 2011. – № 6. – С. 11 - 44.
- [6] Gurevich I.B. and Yashina V.V. Descriptive approach to image analysis: image formalization space // Pattern Recogn. Image Anal.: Adv. Math. Theory Appl. 2012. Vol. 22, №4. – P. 495 - 518.
- [7] Crisp D.J., Tao T.C. Fast Region Merging Algorithms for Image Segmentation // The 5th Asian Conf. on Computer Vision (ACCV2002), Melbourne, Australia, 23 – 25 January 2002. – P. 1 - 6.
- [8] Dvoenko S.D. Clustering of a set of objects// Pattern Recognition and Information Processing (PRIP-2007) / Proc. of the 9th Int. Conf., Minsk, 2007. Vol. 1. – P. 93 - 97.
- [9] Dvoenko S.D. Meanless k-means as k-meanless clustering with the bi-partial approach // Pattern Recognition and Information Processing (PRIP'2014) / Proc. of the 12th Int. Conf., Minsk, 2014. – P. 50 - 54.
- [10] Двоенко С.Д. Неиерархический дивизимный алгоритм кластеризации // Автоматика и телемеханика, 1999. №4, — С. 117–124.
- [11] Харинов М.В. Запоминание и адаптивная обработка информации цифровых изображений; под ред. Р.М. Юсупова. – СПб: Изд-во С.-Петерб. ун-та, 2006. – 138 с.
- [12] Харинов М.В. Обобщение трех подходов к оптимальной сегментации цифрового изображения // Труды СПИИРАН, 2013. – Вып. 2 (25). – С. 294 - 316.
- [13] Koepller G., Lopez C., Morel J. A Multiscale Algorithm for Image Segmentation by Variational Method // SIAM J. on Numerical Analysis, Vol. 31, 1994. – № 1. – P. 282 - 299.
- [14] Mumford D., Shah J. Boundary detection by minimizing functionals, I // Proc. IEEE Comput. Vision Patt. Recogn. Conf., San Francisco, 1985. – P. 22 - 26.
- [15] Otsu N.A Threshold Selection Method from Gray-Level Histograms // IEEE Trans. on systems, MAN, and CYBER-NETICS, January 1979. Vol. SMC-9, № 1. – P. 62 - 66.
- [16] Ward J.H., Jr. Hierarchical grouping to optimize an objective function. // J. Am. Stat. Assoc. 1963. Vol. 58, Issue 301. – P. 236 - 244.

Об авторах

Харинов Михаил – старший научный сотрудник СПИИРАН. E-mail: khar@iias.spb.su.

Детектирование стадии фиброза печени при помощи текстурного анализа ультразвуковых изображений

А. Хвостиков¹, А. Крылов², Ю. Камалов³,

Лаборатория математических методов обработки изображений⁴,

Факультет вычислительной математики и кибернетики

Московский государственный университет им. М. В. Ломоносова, Россия

¹xubiker@gmail.com, ²kryl@cs.msu.ru, ³kamalov53@yandex.ru, ⁴http://imaging.cs.msu.ru

Аннотация

Предложен комплексный текстурный метод анализа ультразвуковых изображений печени для определения стадии фиброза. Разработан алгоритм, основанный на использовании классификаторов Rotation Forest и KNN применительно к текстурным характеристикам. 720 текстурных характеристик получены методами, основанными на применении матрицы смежности, энергетических масок Laws', матрицы равномерности и статистических характеристик изображений. С помощью корреляционного метода произведён выбор 22 существенных признаков. Тестирование метода на изображениях печени 57 пациентов при разделении на 5 стадий фиброза показало 72.7% точность классификации отдельных областей интереса. При классификации целых изображений алгоритм правильно определил стадию фиброза в подавляющем большинстве случаев.

Ключевые слова: текстурный анализ, ультразвуковые изображения, фиброз печени, алгоритмы классификации

1. ВВЕДЕНИЕ

Фиброз печени – заболевание, при котором вследствие хронического воспаления печени происходит постепенное разрастание или уплотнение соединительной ткани. Развитие фиброза приводит к серьёзным последствиям (нарушению процессов обмена между клетками печени и кровью, шунтированию крови, прогрессированию хронического заболевания печени в сторону цирроза, сужению спектра и снижению эффективности лечебных мероприятий). Поэтому диагностика и оценка стадии фиброза представляет собой важную и актуальную задачу.

В силу того, что динамическое наблюдение за фиброзом печени с помощью биопсии затруднено ввиду инвазивности самого метода, наиболее востребованными сейчас являются неинвазивные методы диагностики, например, ультразвуковое исследование. Для определения стадии фиброза печени обычно используются текстурные методы, основанные на энергетических масках Laws'[1], матрицах смежности[2, 3], фрактальных размерностях [3], матрицах равномерности [4].

В данной статье предлагается комплексный подход к определению стадии фиброза печени, сочетающий в себе сразу несколько текстурных методов.

В качестве материала для исследования использовались ультразвуковые снимки печени 57 пациентов с различной подтверждённой стадией фиброза. Изображения имеют разрешение $\sim 770 \times 770$ пикселей и глубину 8 бит. На используемых снимках печень представлена в трёх сечениях: вдоль брюшной средней линии, вдоль правой парастеральной линии и

вдоль правой межключичной линии. В дальнейшем будем именовать эти сечения как $C1$, $C2$ и $C3$.

Для классификации стадии фиброза печени используется 5-уровневая ($F0-F4$) шкала *METAVIR*.

Обучение и классификация проводятся следующим образом:

1. *Обучение.* На каждом изображении из обучающей базы выделяется несколько областей для анализа текстуры. Для областей составляются векторы текстурных признаков, на которых тренируется ансамбль классификаторов.
2. *Классификация.* Для каждого вновь поступившего изображения выделяются области интереса, вычисляются такие же векторы признаков и классифицируются при помощи обученного ансамбля.

2. ВЫБОР ОБЛАСТЕЙ ИНТЕРЕСА

На исходном изображении выбирают области интереса (ROI), представляющие собой квадратные участки фиксированного радиуса. Выбираемые области должны содержать «чистую» текстуру без сосудов, затемнений и других артефактов. Обычно для изображений такого разрешения радиус областей интересов выбирают от 16 до 32 пикселей [1, 3].

Так как информативная часть изображения на ультразвуковом снимке представлена в виде конуса, текстура печени имеет разный угол направленности, зависящий от удаления от оси конуса, что оказывает сильное влияние на значения неустойчивых к повороту текстурных признаков. Для устранения этого эффекта предлагается выбирать ROI с учётом угла поворота текстуры. Выбор областей интереса проводится вручную, поиск оси конуса на изображении и поворот ROI выполняется автоматически.



Рис. 1. Пример выделения ROI с учетом поворота

3. ПРЕДОБРАБОТКА ИЗОБРАЖЕНИЯ

Перед вычислением текстурных характеристик анализируемое изображение нормализуется по яркости. Для этого рас-

смагивается множество пикселей, образующее области интереса:

$$\Omega = \bigcup_{k=1}^n \{i \in ROI_k\}, \text{ где } n - \text{ число выделенных областей.}$$

Для множества Ω вычисляются среднее значение и среднеквадратичное отклонение. После чего линейным преобразованием исходное изображение приводится к изображению со средним значением яркости 128 и среднеквадратичным отклонением 30 (рис. 2).

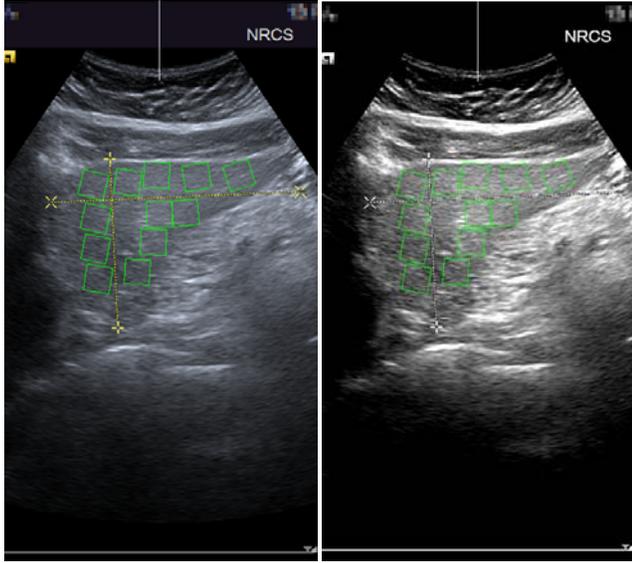


Рис. 2. Применение нормализации:
а – исходное изображение; б – результат

4. ВЫЧИСЛЕНИЕ ПРИЗНАКОВ

Для каждой ROI нормализованного изображения вычисляются текстурные признаки, позволяющие ёмко охарактеризовать основные параметры текстуры.

4.1. Статистические характеристики

Ряд признаков (среднее значение, среднеквадратичное отклонение, вариация, энергия, асимметрия, эксцесс, энтропия) вычисляется непосредственно для самого изображения.

4.2. Энергетические характеристики Laws'

Из одномерных ядер L5, E5, S5, W5, R5 путём попарного произведения векторов и удаления повторяющихся элементов составляются 15 двумерных масок. Для каждой ROI вычисляются свёртки со всеми полученными масками, причём, в случае несимметричных масок результат усредняется:

$$Laws(ROI, E5L5) = \frac{ROI * E5L5 + ROI * L5E5}{2}.$$

Для каждого отфильтрованного изображения вычисляются коэффициенты среднего значения, среднеквадратичного отклонения, асимметрии, эксцесса и энергии [1].

4.3. Характеристики матрицы смежности

Также используется матрица смежности (GLCM) [5]. Она строится для полутоновых изображений по фиксированному

вектору $d = (dr, dc)$, задающему смещение между пикселем яркостью i и пикселем яркостью j :

$$GLCM(i, j) = |\{(r, c) | I[r, c] = i, I[r + dr, c + dc] = j\}|.$$

Каждая ROI нормализованного изображения постеризуется до 64 градаций яркости, после чего для неё вычисляются матрицы смежности по всем возможным смещениям $dr = 0,9$, $dc = 0,9$.

Для каждой GLCM вычисляются коэффициенты энергии, энтропии, контрастности, однородности, различия и корреляции.

4.4. Характеристики матрицы равномерности

Матрица равномерности (GLRLM) [6] имеет размер $n \times m$, где n - число градаций яркости, m - число, не превышающее длины диагонали изображения. GLRLM строится по фиксированному углу $\theta = \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$, задающему направление построения матрицы:

$$P[i, j | \theta] = |\omega|,$$

где ω - множество групп из j подряд идущих пикселей яркости i в направлении θ .

Каждая ROI постеризуется до 64 градаций яркости, после чего для неё вычисляются GLRLM по всем четырём направлениям ($0^\circ, 45^\circ, 90^\circ, 135^\circ$).

Для каждой матрицы вычисляются признаки SRE, LRE, LGRE, HGRE, SRLGE, SRHGE, LRLGE, LRHGE, GLNU, RLNU, RPC.

В итоге, для каждой ROI вычисляется вектор текстурных характеристик, состоящий из 720 признаков.

5. ПОСТОБРАБОТКА ПРИЗНАКОВ

На этапе выделения областей интереса иногда возникает ситуация, при которой выделяемая область частично «задевает» какой-либо артефакт, в этом случае значения вычисленных текстурных характеристик этой ROI сильно отличаются от аналогичных значений остальных ROI на изображении. Причём, если такое отклонение имеет место, то отличия заметны сразу на большом количестве признаков.

Для предотвращения этой ситуации в алгоритме предусмотрена возможность исключения выбросов в объёме до 40%.

6. КЛАССИФИКАЦИЯ И РЕЗУЛЬТАТЫ

Для сравнения возможностей различных классификаторов применительно к данной задаче используется ПО с открытым исходным кодом *Weka* [7, 8].

На момент написания статьи база данных содержит снимки 57 пациентов (7 пациентов со стадией F0, 34 пациента со стадией F1, 8 пациентов со стадией F2, 2 пациента со стадией F3, 6 пациентов со стадией F4).

Для балансирования выборки обучающая база данных умеренно прореживается так, чтобы число пациентов с разными стадиями фиброза было примерно одинаковым.

Далее изображения разбиваются на три группы по сечению. Внутри каждой группы вычисляются векторы текстурных признаков для всех ROI на всех изображениях, полученные векторы перемешиваются случайным образом. После чего составленный набор векторов длины 720 передаётся класси-

фикатору для обучения. Для каждого типа сечения ($C1$, $C2$, $C3$) используется независимый классификатор.

В ходе поиска наиболее подходящего классификатора были протестированы алгоритмы SVM, MultilayerPerceptron, KNN, K^* [9], RandomForest [10], Rotation Forest [11], CHIRP, Decorate.

Для проверки качества обучения использовался скользящий контроль с разбиением на 10 подвыборок.

Подавляющее большинство классификаторов показали лучший результат на сечении $C1$, что объясняется большей однородностью текстуры на этом сечении.

Лучший результат показал классификатор Random Forest со следующими параметрами: глубина деревьев = 15, количество признаков = 100, количество деревьев = 2000. Точность классификации отдельно взятой области интереса составляет 66.8% для $C1$, 63.6% для $C2$, 64.4% для $C3$.

Далее с помощью корреляционного метода CFS [12] из 720 признаков были выбраны существенные 22 признака для $C1$, 11 признаков для $C2$ и 12 признаков для $C3$. Например, для $C1$ такими признаками являются: среднее квадратичное отклонение и эксцесс для самого изображения, среднее квадратичное отклонение для свертков с масками Laws' LW, EE, SS, RR, RW, WW и энергия для SS, контраст для GLCM(9,1), энтропия для GLCM(1,0) и GLCM(9,6), энергия для GLCM(1,1), однородность для GLCM(2,6) и GLCM(7,9), корреляция для GLCM(5,2), различие для GLCM(7,1) и GLCM(7,2), SRLGE для GLRLM(0), LRE для GKRLM(45), LRLGE для GLRLM(90) и GLNU для GLRLM(135).

На сжатых таким образом данных были еще раз протестированы различные классификаторы. В итоге, для изображений из группы $C1$ была достигнута точность:

- KNN – 72.7%.
- Random Forest – 68.0%.
- Rotation Forest – 69.2%.
- K^* - 70.4%.

Увеличение точности классификации подтверждает правильность выбора существенных признаков.

predicted \ actual	$F0$	$F1$	$F2$	$F3$	$F4$
$F0$	38	4	5	0	1
$F1$	3	25	7	0	3
$F2$	5	5	29	0	4
$F3$	0	0	0	0	0
$F4$	4	4	1	0	31

Матрица ошибок на сечении $C1$ для классификатора KNN

Для окончательной проверки предложенного алгоритма было проведено тестирование в условиях, приближенных к реальным. Именуемая база данных пациентов была разделена на непересекающиеся обучающую и тестовую выборки. Тестовая выборка состояла из 1 пациента с $F0$, 4 пациентов с $F1$, 1 – с $F2$, 1 – с $F4$. Для классификации использовался выбранный ранее Random Forest. Из-за небольшого размера базы

данных подобное разделение проводилось несколько раз с выбором разных пациентов для тестирования.

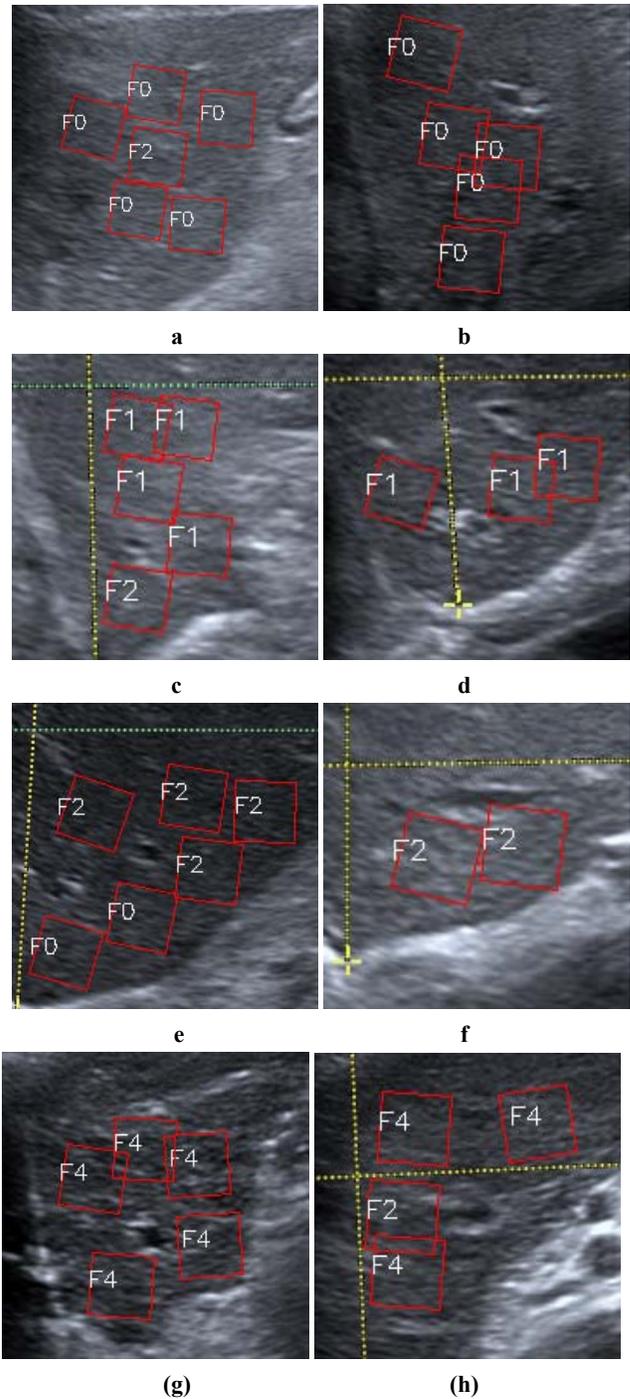


Рис. 3. Примеры тестирования алгоритма. Реальные стадии фиброза: a, b – $F0$, c, d – $F1$, e, f – $F2$, g, h – $F4$

В качестве итогового результата классификации каждого изображения выбирался класс с наибольшим количеством ROI, за него проголосовавших.

Из результатов тестирования (рис.3) можно сделать вывод, что в подавляющем большинстве случаев классификация проводится верно. Предположительно, причиной встречаю-

шихся ошибок классификации является относительно малый объём обучающей выборки и недостаточно эффективная настройка методов обучения.

7. ЗАКЛЮЧЕНИЕ

В статье предложен алгоритм, определяющий стадию фиброза по ультразвуковому изображению печени. Указанный алгоритм был проверен на базе данных пациентов с подтверждённой стадией фиброза. Точность классификации отдельных фрагментов текстуры достигает 72.7%. При обучении и тестировании на данных разных пациентов алгоритм в большинстве случаев правильно определил стадию фиброза.

8. БЛАГОДАРНОСТИ

Работа выполнена при поддержке гранта РФФИ 14-11-00308.

9. ССЫЛКИ

- [1] Virmani J. et al. Prediction of cirrhosis from liver ultrasound B-mode images based on Laws' masks analysis //Image Information Processing (ICIP), 2011 International Conference on. – IEEE, 2011. – С. 1 - 5.
- [2] Гаврилов А.В., Камалов Ю.Р., Парусников А.В., Сандриков В.А., Лопаткина Т.Н., Калайдзидис Я.Л. Первый опыт компьютерного анализа ультразвуковой текстуры паренхимы печени при распознавании ее некоторых диффузных заболеваний // Клиническая физиология, 1998. – С. 128 – 136.
- [3] Cao G., Shi P., Hu B. Liver fibrosis identification based on ultrasound images captured under varied imaging protocols //Journal of Zhejiang University. Science. B. – 2005. – Т. 6. – №. 11. – С. 1107.
- [4] Selvarajah S., Kodituwakku S. R. Analysis and comparison of texture features for content based image retrieval //International Journal of Latest Trends in Computing. – 2011. – Т. 2. – №. 1.
- [5] Хвостиков А.В. Обработка текстовых изображений с использованием матрицы смежности // Труды 15-й международной конференции «Цифровая обработка сигналов и её применение» (DSPA'2013). Т. 2. – С. 41 - 43.
- [6] Xu D. H. et al. Run-length encoding for volumetric texture //Heart. – 2004. – Т. 27. – С. 25.
- [7] Bouckaert R. R. et al. WEKA Manual for Version 3-7-11, 2014.
- [8] Witten I. H., Frank E. Data Mining: Practical machine learning tools and techniques. – Morgan Kaufmann, 2005.
- [9] Cleary J. G. et al. K*: An Instance-based Learner Using an Entropic Distance Measure //ICML. – 1995. – С. 108 - 114.
- [10] Breiman L. Random forests //Machine learning. – 2001. – Т. 45. – №. 1. – С. 5 - 32.
- [11] Rodriguez J. J., Kuncheva L. I., Alonso C. J. Rotation forest: A new classifier ensemble method //Pattern Analysis and Machine Intelligence, IEEE Transactions on. – 2006. – Т. 28. – №. 10. – С. 1619 - 1630.
- [12] Hall M. A. *Correlation-based feature selection for machine learning*: дис. – The University of Waikato, 1999.

Об авторах

Хвостиков Александр – студент факультета ВМК МГУ.

Email: xubiker@gmail.com

Крылов Андрей – проф., д. ф.-м. н., заведующий лабораторией математических методов обработки изображений факультета ВМК МГУ.

Email: kryl@cs.msu.ru

Камалов Юлий – проф., д.м.н., руководитель лаборатории ультразвуковой диагностики РНЦХ им. Петровского РАМН.

Email: kamalov53@yandex.ru

Abstract

In this work, a comprehensive method of texture analysis of B-mode ultrasound images for the determination of the liver fibrosis stage is reported. The algorithm is based on the use of Rotation Forest and KNN classifiers for the texture characteristics. 720 textural characteristics were extracted using methods based on Laws' masks analysis, Co-occurrence Matrix, Gray Level Run Length Matrix and statistical characteristics of the images. An optimal subset of 22 essential features was selected using correlation-based method. Testing the algorithm with liver images of 57 patients divided into 5 stages of fibrosis showed 72.7% classification accuracy for separate regions of interest. Using the entire image classification method the algorithm correctly identified fibrosis stage in the vast majority of cases.

Keywords: texture analysis, ultrasound images, liver fibrosis, classification algorithms

Аппроксимация фазы проекционным методом Эрмита при восстановлении изображения по фазе

Е. Павельева, А. Крылов

Факультет вычислительной математики и кибернетики

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

E-mail: paveljeva@yandex.ru, kryl@cs.msu.ru

Аннотация

В работе предложен алгоритм восстановления изображения по фазе аппроксимации преобразования Фурье, полученной в результате проекционного метода Эрмита. Тестирование метода на изображениях базы данных SIPI показало хорошее качество восстановления аппроксимации изображений по фазе. Показано, что итерационный метод восстановления достаточно устойчив к области расширения фазы-оригинала нулями.

Ключевые слова: функции Эрмита, преобразование Фурье, фаза, проекционный метод Эрмита, восстановление изображения.

1. ВВЕДЕНИЕ

Фаза преобразования Фурье содержит больше информации, чем амплитуда преобразования Фурье [1, 2]. Использование информации о фазе преобразования Фурье позволяет решать различные задачи обработки изображений: поиск контуров на изображениях [3 - 4], определение уровня размытости [5] и повышение резкости изображений [6], шумоподавление [7], биометрические задачи [8] и др. [9 - 11].

В работе [1] приведены условия, при которых сигнал однозначно (с точностью до постоянного множителя) определяется фазой своего преобразования Фурье. Среди алгоритмов восстановления сигналов по фазе [1, 12] наиболее применимым на практике является итерационный алгоритм, основывающийся на методе проекций на выпуклые множества [13].

В данной статье рассматривается задача восстановления аппроксимации изображения по фазе аппроксимации преобразования Фурье на основе проекционного метода Эрмита. В проекционном методе Эрмита [14] используется аппроксимация разложения функции в ряд Фурье по функциям Эрмита [15], являющимся собственными функциями преобразования Фурье. На основе проекционного метода Эрмита решается ряд задач обработки сигналов и машинного зрения [16 - 17].

2. АППРОКСИМАЦИЯ ИЗОБРАЖЕНИЙ С ПОМОЩЬЮ ФУНКЦИЙ ЭРМИТА

Функции Эрмита

$$\psi_n(x) = \frac{(-1)^n e^{-\frac{x^2}{2}}}{\sqrt{2^n n! \sqrt{\pi}}} \cdot \frac{d^n (e^{-x^2})}{dx^n}$$

являются собственными функциями преобразования Фурье

$$F[f] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) e^{-ix} dx \quad \text{с собственными значениями}$$

$\pm 1, \pm i$: $F[\psi_n] = (-i)^n \psi_n$, образуют полную ортонормированную систему функций в пространстве $L_2(-\infty, \infty)$. Функции Эрмита $\psi_n(x)$ хорошо локализованы на отрезке $[-\sqrt{2n+1}, \sqrt{2n+1}]$, на котором сосредоточены все точки перегиба функции Эрмита.

Двумерные функции Эрмита (рис. 1) определяются через одномерные функции Эрмита:

$$\psi_{m,n}(x, y) = \psi_m(x) \cdot \psi_n(y),$$

поэтому двумерные функции Эрмита обладают аналогичными свойствами в пространстве $L_2(R^2)$.

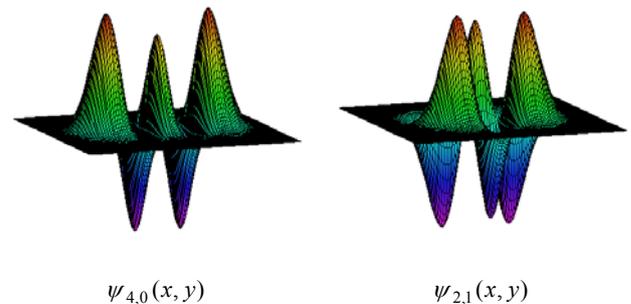


Рис 1. Двумерные функции Эрмита

Опр. 1. Пусть $f_{m,n}(x, y)$ - аппроксимация разложения функции $f(x, y) \in L_2(R^2)$ в ряд Фурье по функциям Эрмита (проекционный метод Эрмита [14]):

$$f(x, y) \approx f_{m,n}(x, y) = \sum_{k=0}^m \sum_{l=0}^n c_{k,l} \psi_{k,l}(x, y). \quad (1)$$

Выражение

$$HF_{f,m,n}(x, y) = F[f_{m,n}(x, y)] = \sum_{k=0}^m \sum_{l=0}^n c_{k,l} (-i)^{k+l} \psi_{k,l}(x, y)$$

назовем аппроксимацией преобразования Фурье двумерной функции $f(x, y)$ с использованием функций Эрмита (АПФЭ).

Функцию $HF_{f,m,n}(x, y)$ можно записать в показательной форме записи: $HF_{f,m,n}(x, y) = A_{f,m,n}(x, y) \cdot e^{i\varphi_{f,m,n}(x, y)}$, где $A_{f,m,n}(x, y)$ - амплитуда; $\varphi_{f,m,n}(x, y) = \arg HF_{f,m,n}(x, y)$ - фаза $HF_{f,m,n}(x, y)$.

Опр. 2. Пусть $F_{m,n}(x, y)$ - аппроксимация разложения комплексной функции $F(x, y) \in L_2(R^2)$ в ряд Фурье по функциям Эрмита:

$$F(x, y) \approx F_{m,n}(x, y) = \sum_{k=0}^m \sum_{l=0}^n C_{k,l} \psi_{k,l}(x, y),$$

где $C_{k,l}$ – комплексные коэффициенты. Выражение

$$HF^{-1} F_{m,n}(x, y) = \sum_{k=0}^m \sum_{l=0}^n C_{k,l} (-i)^{k+l} \psi_{k,l}(-x, -y)$$

назовем аппроксимацией обратного преобразования Фурье функции $F(x, y)$ с использованием функций Эрмита (АОПФЭ). Можно показать, что АОПФЭ, примененное к АПФЭ дает аппроксимацию $f_{m,n}(x, y)$.

Пусть задано изображение с интенсивностью $I(x, y)$. Можно интерпретировать это изображение как двумерную функцию $f(x, y) \in L_2(R^2)$, которая совпадает с $I(x, y)$ внутри некоторого прямоугольника $D = [-A, A] \times [-B, B]$ и равна нулю вне D . Для применения алгоритма восстановления изображения по фазе АПФЭ растянем функции Эрмита $\psi_{k,l}(x, y)$, $k = 0, 1, \dots, m$, $l = 0, 1, \dots, n$ в $A/\sqrt{2m+1}$ раз вдоль оси OX и в $B/\sqrt{2n+1}$ раз вдоль оси OY , чтобы функция $\psi_{m,n}(x, y)$ в разложении (1) была локализована с вычислительной точки зрения в прямоугольнике D . Для подавления эффекта ложного оконтуривания в аппроксимации из исходного изображения перед разложением вычитается базовая функция [18].

3. АЛГОРИТМ ВОССТАНОВЛЕНИЯ ИЗОБРАЖЕНИЙ

В работе [1] предложен итерационный алгоритм восстановления дискретного сигнала $f(x)$ длины N по фазе преобразования Фурье сигнала $\hat{f}(x)$, совпадающего с $f(x)$ в N точках и продолженного по длине нулями не менее чем в два раза (до длины $M \geq 2N$):

$$\hat{f}(x) = \begin{cases} f(x), & x = 0, \dots, N-1 \\ 0, & x = N, \dots, M-1 \end{cases}$$

Этот алгоритм представляет собой двухшаговый метод проектирования на выпуклые множества [13], в котором на первом шаге каждой итерации используется информация о сигнале из пространственной области, а на втором шаге – из частотной.

Алгоритм восстановления изображения по фазе АПФЭ использует информацию только о фазе АПФЭ исходного изображения $f(x, y) \in L_2(R^2)$. Пусть $f(x, y) = 0$ вне прямоугольника $D = [-A, A] \times [-B, B]$, и число функций Эрмита $\psi_{k,l}(x, y)$, $k = 0, 1, \dots, m$, $l = 0, 1, \dots, n$ зафиксировано. Вычислим аппроксимацию $f_{m,n}(x, y)$ и фазу АПФЭ $\varphi_{f,m,n}(x, y)$.

Рассмотрим два выпуклых множества G_1 и G_2 :

$$G_1 = \{f(x, y) : f(x, y) = \sum_{k=0}^m \sum_{l=0}^n c_{k,l} \psi_{k,l}(x, y), c_{k,l} \in \mathbb{R}\},$$

$$G_2 = \{F(x, y) : \arg(F(x, y)) = \varphi_{f,m,n}(x, y)\}.$$

Пусть $h_{p,q}(x, y) = \sum_{k=0}^p \sum_{l=0}^q d_{k,l} \psi_{k,l}(x, y)$, где $p > m, q > n$, является аппроксимацией (1) некоторой функции

$h(x, y) \in L_2(R^2)$. Тогда АПФЭ функции $h(x, y)$ имеет вид:

$$HF_{h,p,q}(x, y) = |H_{h,p,q}(x, y)| \cdot e^{i\varphi_{h,p,q}(x, y)}. \quad \text{Определим}$$

проекторные операторы P_1 и P_2 на множества G_1 и G_2 следующим образом:

$$P_1 h = h_{m,n}(x, y) = \sum_{k=0}^m \sum_{l=0}^n d_{k,l} \psi_{k,l}(x, y),$$

$$P_2 h = |H_{h,p,q}(x, y)| \cdot e^{i\varphi_{f,m,n}(x, y)}.$$

Таким образом, под действием оператора P_1 в аппроксимации $h_{p,q}(x, y)$ остаются только слагаемые

$$\sum_{k=0}^m \sum_{l=0}^n d_{k,l} \psi_{k,l}(x, y), \quad \text{и происходит обнуление}$$

$(p+1) \cdot (q+1) - (m+1) \cdot (n+1)$ слагаемых, соответствующих высокочастотным функциям Эрмита, а под действием оператора P_2 фаза АПФЭ заменяется на заданную фазу $\varphi_{f,m,n}(x, y)$.

Рассмотрим прямоугольник $D' = [-A_1, A_1] \times [-B_1, B_1]$, $A_1 > A$, $B_1 > B$ и определим количество $(p+1) \cdot (q+1)$ функций Эрмита $\psi_{k,l}(x, y)$, $k = 0, 1, \dots, p$, $l = 0, 1, \dots, q$, исходя из соотношений:

$$\frac{A_1}{A} = \frac{\sqrt{2p+1}}{\sqrt{2m+1}}, \quad \frac{B_1}{B} = \frac{\sqrt{2q+1}}{\sqrt{2n+1}}. \quad (2)$$

Таким образом, все функции Эрмита $\psi_{k,l}(x, y)$, $k = 0, 1, \dots, p$, $l = 0, 1, \dots, q$ хорошо локализованы в прямоугольнике D' , поскольку функции Эрмита $\psi_{k,l}(x, y)$, $k = 0, 1, \dots, m$, $l = 0, 1, \dots, n$ хорошо локализованы в D . Итерационный алгоритм восстановления аппроксимации $f_{m,n}(x, y)$ по известной фазе АПФЭ $\varphi_{f,m,n}(x, y)$ выглядит следующим образом.

Начальная итерация. Рассмотрим начальное приближение неизвестной амплитуды АПФЭ $A^0(x, y)$ и получим начальное приближение АПФЭ: $F^0(x, y) = A^0(x, y) e^{i\varphi_{f,m,n}(x, y)}$. Вычислим АОПФЭ для $F^0(x, y)$ и получим локализованное в D' начальное приближение $g^1(x, y)$ аппроксимации изображения: $g^1(x, y) = HF^{-1} F^0(x, y)$.

$(k+1)$ -я итерация состоит из следующих действий:

1. Применение ограничения в пространственной области: $f^k(x, y) = P_1 g^k$.

$$2. G^k(x, y) = HF_{f^k,p,q}(x, y) = |A^k(x, y)| e^{i\varphi_{f^k,p,q}(x, y)}.$$

3. Применение ограничения в частотной области: $F^k(x, y) = P_2 f^k(x, y) = |A^k(x, y)| e^{i\varphi_{f,m,n}(x, y)}$.

$$4. g^{k+1}(x, y) = HF^{-1} F^k(x, y).$$

Алгоритм сходится, поскольку множества G_1 и G_2 – выпуклые [13]. Условие останова алгоритма после K итераций является условием:

$$\|f^K(x, y) - f^{K-1}(x, y)\|_{L_2(R^2)} < \varepsilon.$$

Важно отметить, что, в отличие от классического метода восстановления по фазе, в предложенном алгоритме восстановление аппроксимации изображения происходит по фазе АПФЭ того же изображения, а не другого изображения большего размера.

4. РЕЗУЛЬТАТЫ РАБОТЫ АЛГОРИТМА

На рис. 2 приведен пример работы алгоритма для тестового изображения “Lena” размера 128×128 пикселей для $K=10, 50, 100, 300, 500, 1000$ итераций. Для аппроксимации (1) изображения в предложенном алгоритме взяты значения $m=n=40$. В алгоритме восстановления аппроксимации изображения по фазе АПФЭ размеры прямоугольника $D' = [-A_1, A_1] \times [-B_1, B_1]$ являются параметрами алгоритма.

Возьмем $d = \frac{A_1}{A} = \frac{B_1}{B} = 1.5$ и начальное приближение

амплитуды $A^0(x, y) \equiv 1$. На рис. 2 на изображениях в левом верхнем углу показан номер итерации K , а снизу приведены значения метрик PSNR и MS-SSIM [19] между полученным изображением после K -й итерации и аппроксимацией $f_{m,n}(x, y)$. Видно, что уже после небольшого числа итераций изображение визуально малоотлично от аппроксимации $f_{m,n}(x, y)$.

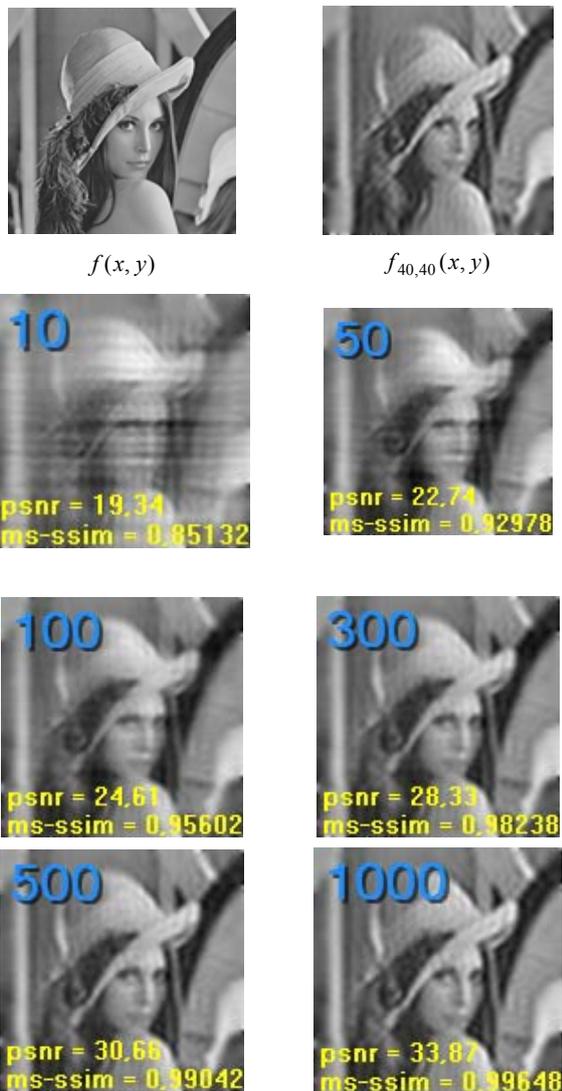


Рис. 2. Пример работы алгоритма восстановления изображений по фазе АПФЭ

На рис. 3 приведен пример работы алгоритма при различных вариантах выбора параметра d . Для аппроксимации (1) брались значения $m=n=40$.

$d=1.2$	$d=1.5$ (рис. 2)	$d=2$
 10 psnr = 20.14 ms-ssim = 0.88408	 10 psnr = 19.34 ms-ssim = 0.85132	 10 psnr = 18.35 ms-ssim = 0.82501
 100 psnr = 23.13 ms-ssim = 0.96950	 100 psnr = 24.61 ms-ssim = 0.95602	 100 psnr = 24.38 ms-ssim = 0.95523
 500 psnr = 23.82 ms-ssim = 0.97173	 500 psnr = 30.66 ms-ssim = 0.99042	 500 psnr = 29.72 ms-ssim = 0.98604

Рис. 3. Влияние параметра d на работу алгоритма

Для тестирования работы алгоритма рассмотрим изображения из базы данных SIPI Miscellaneous [20] размера 128×128 пикселей. На рис. 4 приведены аппроксимации изображений из базы данных и изображения после 500 итераций, а также значения метрик PSNR и MS-SSIM между изображениями.

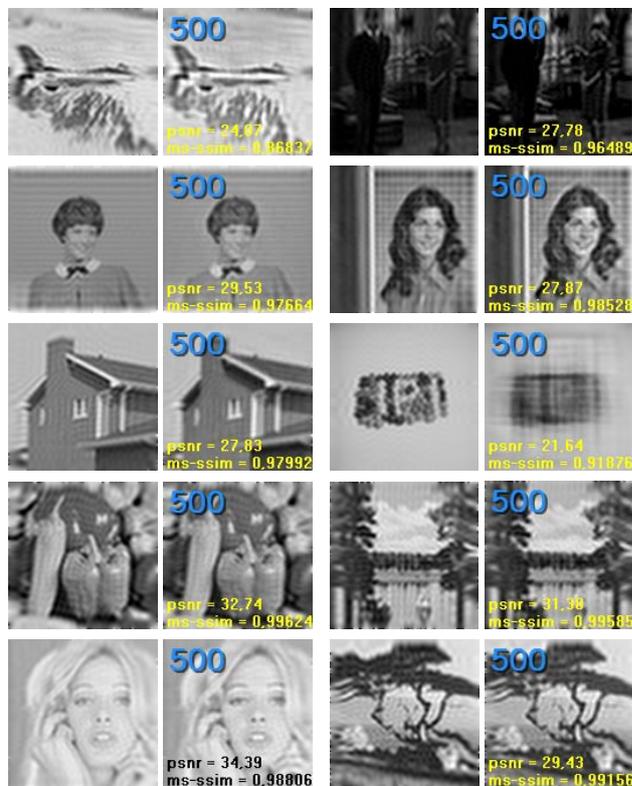


Рис. 4. Результаты работы алгоритма

5. ЗАКЛЮЧЕНИЕ

Достаточно реалистичным является сценарий, когда у преобразования Фурье изображения фаза известна точно, а амплитуда дана с ошибкой. В этой ситуации достаточно интересно восстановить правильную амплитуду по фазе и, тем самым, восстановить изображение. Классический метод восстановления требует знания информации о фазе не исходного сигнала, а сигнала, расширенного не менее чем в два раза нулями. Однако такой информации для данной задачи получить нельзя. Отличием предложенного метода является возможность восстановления аппроксимации сигнала по фазе аппроксимации преобразования Фурье рядом из $(m+1) \cdot (n+1)$ функций Эрмита. Это тоже требует дополнительной информации о количестве слагаемых в аппроксимации (1), однако объем этой дополнительной информации существенно меньше.

Цель дальнейшей работы – оценка устойчивости восстановления изображения при ошибке в задании параметров m и n .

Работа выполнена при поддержке гранта РФФИ 13-07-00438.

6. ССЫЛКИ

- [1] Oppenheim A.V. and Lim J.S. The importance of phase in signals. *Proc. of the IEEE*, 69(5). – P. 529 - 541, 1981.
- [2] Ni X.S., Huo X. Statistical interpretation of the importance of phase information in signal and image reconstruction, *Statistics & Probability Letters*, vol. 77, no. 4. – P. 447 - 454, Feb. 2007.
- [3] Kovese P. Phase congruency detects corners and edges, the Australian pattern recognition society conference: DICTA 2003. – P. 309 - 318, 2003.
- [4] Skarbnik, Sagiv C., Zeevi Y.Y. “Edge detection and skeletonization using quantized localized phase,” in *Proc. EUSIPCO*, 2009, p. 1542 - 1546.
- [5] Hassen R., Wang Z., Salama M. No-reference image sharpness assessment based on local phase coherence measurement, in *ICASSP'10*, 2010. – P. 2434 - 2437
- [6] Blanchet G., Moisan L. and Rougé B. Measuring the global phase coherence of an image, in *ICIP*, Oct. 2008. – P. 1176 - 1179.
- [7] Kovese P. Phase preserving denoising of images, in *DICTA'99*, Perth WA, Dec. 1999. – P. 212 - 217.
- [8] Miyazawa K., Ito K., Aoki T., Kobayashi K., Nakajima H. A Phase-Based Iris Recognition Algorithm, *Advances in Biometrics, LNCS*, vol. 3832. – P. 356 - 365, 2006.
- [9] Jing X., Hong Y.X., Xin S.X., Ying M.X. Medical Image Mosaic Technology Based on Image Phase Correlation, in *ICDH*, 2012. – P. 274 - 277.
- [10] Narwaria M., Lin, W.I. McLoughlin V., Emmanuel S., Chia L.T. Fourier transform-based scalable image quality measure, *IEEE Trans. on Image Processing*, vol. 21, no. 8. – P. 3364 - 3377, Aug. 2012.
- [11] Reddy B.S., Chatterji B.N. An FFT-based technique for translation, rotation, and scale-invariant image registration, *IEEE Trans. on Image Processing*, vol.: 5, no. 8. – P. 1266 - 1271, Aug. 1996.
- [12] Ma C. “Novel criteria of uniqueness for signal reconstruction from phase”, *IEEE Trans. on Sig.Proc.*, vol. 39, no. 4.– P. 989 - 992, Apr. 1991.
- [13] Гурин Л.Г., Поляк Б.Т., Райк Э.В. Методы проекций для отыскания общей точки выпуклых множеств //

Вычисл. матем. и матем. физ. – Т. 7, 1967 – № 6. – С. 1211 - 1228.

- [14] Krylov A.S. and Korchagin D.N. Fast Hermite Projection Method, *LNCS*, vol. 4141, 2006. – P. 329 – 338.
- [15] Titchmarsh E.C.. Introduction to the theory of Fourier integrals, Oxford: Clarendon Press, vol. 337, 1937.
- [16] Pavelyeva E.A., Krylov A.S. An Adaptive Algorithm of Iris Image Key Points Detection, in: 20-th International Conference on Computer Graphics GraphiCon'2010. St. Petersburg, Russia, 2010. – P. 320 - 323.
- [17] Stankovic S., Orovic I., Krylov A. Video Frames Reconstruction based on Time-Frequency Analysis and Hermite projection method, *EURASIP Journal on Advances in Signal Processing*, Vol. 2010, 2010. – P. 6:1 - 6:11.
- [18] Pavelyeva E.A., Krylov A.S. Image reconstruction from phase using Hermite projection method, in 11-th Int. Conf. “PRIA: New Information Technologies”, vol. 1. Samara, Russia, 2013. – P. 296 – 299.
- [19] Wang Z., Simoncelli E.P., Bovik A.C. “Multiscale structural similarity for image quality assessment”, in *Signals, Systems and Computers*, 2003. – P. 1398 - 1402.
- [20] <http://sipi.usc.edu/database/database.php?volume=misc> SIFI Miscellaneous images database.

Об авторах:

Павельева Елена – ассистент кафедры математической физики факультета ВМК МГУ.
E-mail: paveljeva@yandex.ru

Крылов Андрей – д. ф.-м. н., профессор, зав. лабораторией математических методов обработки изображений факультета ВМК МГУ.
E-mail: kryl@cs.msu.ru

Встраивание данных в плоскости Грея цифрового полутонового изображения

В. Горбачев, Л. Денисов, Е. Кайнарова, Е. Яковлева *

Северо-Западный институт печати Санкт-Петербургского государственного университета технологии и дизайна, Санкт-Петербург, Россия

*Национальный минерально-сырьевой университет «Горный», Санкт-Петербург, Россия
{Елена Кайнарова}helenkainarova@gmail.com

Аннотация

Рассмотрена стеганографическая система, использующая плоскости Грея полутонового цифрового изображения, которые построены с помощью кодов Грея. Предложены алгоритмы встраивания и детектирования. Найдено, что данные, скрытые в плоскостях Грея, испытывают при JPEG-сжатии меньшее разрушение, чем данные, встроенные в битовые плоскости.

Ключевые слова: стеганография, коды Грея, цифровое изображение, JPEG-сжатие.

1. ВВЕДЕНИЕ

Коды Грея хорошо известны в теории информации [1]. Построенные с их помощью плоскости для случая цифрового изображения позволяют выделить одно из его главных свойств - избыточность, что представляет интерес для современной стеганографии, которая занимается встраиванием цифровых данных.

Плоскости Грея получают из битовых плоскостей и определяют представление изображения в пространственной области. Для пространственной области описано большое число методов встраивания цифровых данных или цифровых водяных знаков (ЦВЗ) [2], среди которых варианты LSB (Least Significant Bit), блочное и аддитивное встраивание, где используется битовое представление.

После того как данные встроены полученное стегоизображение часто сохраняют в каком-либо графическом формате для дальнейшего использования. Преобразование в графический формат, например, JPEG, который осуществляет сжатие с потерей, может разрушать ЦВЗ. Причина простая: встраивание основано на избыточности изображения, а избыточность удаляется сжатием с потерей. Простым решением служит выбор формата без потерь типа tiff или png. Однако JPEG очень широко используется на практике, поэтому в литературе уделяется большое внимание стеганографии устойчивой к JPEG-сжатию. Техника устойчивая к сжатию с потерями позволяет решать разные задачи, в число которых входит распознавание образов, повышение качества изображения (см., например, [3, 4]) и другие. Встраивание ЦВЗ в jpg-файл с последующим JPEG сжатием известно как J2J (JPEG-to-JPEG) преобразование [6]. Оно изучалось во многих работах, где представлены методы, предлагающие разнообразные компромиссы между уровнем сжатия и деградацией ЦВЗ.

Цель нашей работы – рассмотреть встраивание бинарного изображения в плоскости Грея с последующим JPEG-сжатием. Методы встраивания в плоскости Грея, предложенные в [7] и развитые в [7, 8, 9], продемонстрировали их устойчивость к RS (Regular-Singular) анализу, атакам типа χ^2 и SPA (Sample Pair Analysis). Детекторы для обнаружения ЦВЗ, которые

построены на основе SPA, могут иметь высокую эффективность [10], однако их проектирование сильно усложняется при использовании битовых плоскостей, начиная со второй [11]. В нашей работе для встраивания выбрана четвертая плоскость Грея, детектор для которой авторам не известен. В отличие от приведенных работ, где рассматривались разнообразные атаки, мы интересовались устойчивостью к JPEG-сжатию. Как показали результаты численного эксперимента встраивание в плоскости Грея является более устойчивым к JPEG-сжатию, чем использование битовых плоскостей.

2. ПРЕДСТАВЛЕНИЕ ПОЛУТОНОВОГО ИЗОБРАЖЕНИЯ С ПОМОЩЬЮ ПЛОСКОСТЕЙ ГРЕЯ

Коды Грея для пиксела c получаются из двоичного представления его яркости $c = b_W b_{W-1} \dots b_1$ с помощью соотношения $g_W = g_W$ и $g_V = b_V + b_{V+1}$, $V < W$, где все $b_V, g_V \in 0, 1$. Здесь и далее все биты складываются по модулю два. Двоичные разряды b_V имеют вес 2^{V-1} , $V = 1, \dots, W$, где старший разряд W , для кодов Грея это не так, поскольку они относятся к системам счисления с неестественными весами. В полутоновом изображении совокупность одноименных двоичных разрядов b_V образует битовую плоскость, это бинарное изображение B_V , где пиксел имеет два значения яркости. Аналогично совокупность кодов Грея g_V будем называть плоскостью Грея G_V , которая также является бинарным изображением. Любое полутоновое изображение можно представить с помощью плоскостей Грея.

Пусть C – это 8-битовое полутоновое изображение. Его представление через битовые плоскости имеет вид

$$C = \sum_{V=1}^8 2^{V-1} B_V. \text{ Из определения кода Грея следует, что}$$

$$B_8 = G_8,$$

$$B_T = G_8 + \dots + G_V + \dots + G_T, \quad (1)$$

где $T < 8$. Отсюда вытекает важное свойство, которое используется дальше. Так, согласно (1) модификация одной плоскости Грея, например, G_V приводит к модификации сразу нескольких битовых плоскостей $B_1, \dots, B_T, \dots, B_V$. В итоге представление 8-битового изображения имеет вид

$$C = \sum_{V=1}^8 2^{V-1} (G_8 + \dots + G_V) \quad (2)$$

3. АЛГОРИТМЫ ВСТРАИВАНИЯ И ДЕТЕКТИРОВАНИЯ

Будем рассматривать задачу встраивания бинарного изображения M в полутоновое, используя его плоскости Грея. Для этого возьмем простой алгоритм побитового сложения

$$G_V \rightarrow G_{VM} = G_V + M,$$

который представляет собой вариант метода LSB в пространственной области. Заметим, если использовать младшие плоскости V , где бинарное изображение выглядит как шум, то G_V играет роль секретного ключа, который выбирается случайным.

В результате получается стегоизображение

$$S = 2^7 B_8 + \dots + 2^{V-1} B_{VM} + \dots + 2^0 B_{1M}, \quad (3)$$

где $B_{TM}, T=1, \dots, V$ битовые плоскости со встроенным бинарным изображением M

$$B_{VM} = B_V + M.$$

Отсюда следует, что встраивание в одну плоскость Грея G_V эквивалентно встраиванию в набор битовых плоскостей B_1, \dots, B_V . Из этого свойства возникает две возможности детектирования, когда скрытое изображение извлекается либо из плоскости Грея, либо из битовой плоскости.

Будем рассматривать не слепое детектирование, где требуется исходное изображение C . Для этого случая скрытое изображение можно извлечь из плоскости Грея G_V путем побитового сложения $M = G_{VM} + G_V$. Пусть функция $bitget(A, V)$, обозначение взято из MATLAB, вычисляет битовую плоскость V изображения A . Тогда, с учетом определения плоскости Грея, формула детектирования принимает вид

$$D_1 : M = bitget(C, V) + bitget(C, V+1) + bitget(S, V) + bitget(S, V+1) \quad (4)$$

где побитовая сумма $bitget(C, V+1) + bitget(S, V+1) = 0$, поскольку в плоскость $V+1$ ничего не встраивалось. Кроме этого есть вторая возможность детектирования. Так, можно взять одну из битовых плоскостей, поскольку $M = B_T + B_{TM}$ тогда

$$D_2 : M = bitget(C, T) + bitget(S, T) \quad (5)$$

где $T = 1, 2, \dots, V$.

Избыточность изображения позволяет модифицировать не одну, а несколько плоскостей. Это обстоятельство служит основой для слепого детектирования, когда исходное изображение не требуется. Схема следующая. В плоскость с номером V встраивается содержимое плоскости K вместе с M , а детектирование осуществляется из плоскостей K и V стего-изображения.

Для плоскостей Грея данная схема имеет следующий вид. Встраивание

$$G_V \rightarrow G_{VM} = G_K + M,$$

детектирование

$$M = G_{VM} + G_K = bitget(S, V) + bitget(S, V+1) + bitget(S, K) + bitget(S, K+1) \quad (6)$$

Если выбрать $K = V+1$, то $M = bitget(S, V) + bitget(S, V+2)$. Приведенный вариант со слепым детектированием в отличие от не слепого, нельзя свести к эквивалентному встраиванию в битовые плоскости (3). Поэтому никаких двух путей детектирования типа (4) и (5) не возникает. Поясним на примере. Пусть $V = 2$, $K = 3$ и $G_2 \rightarrow G_{2M} = G_3 + M$. Запишем (2) в виде

$$C = Z + 2^2(Y + G_3) + 2^1(Y + G_3 + G_2) + 2^0(Y + G_3 + G_2 + G_1)$$

где $Y = G_8 + \dots + G_4$. Используя выражение для G_{2M} , вместо (2) найдем

$$S = Z + 2^2(Y + G_3) + 2^1(Y + M) + 2^0(Y + G_1 + M).$$

Согласно (6), детектирование определяется выражением $M = bitget(S, 2) + bitget(S, 4)$.

4. СЖАТИЕ С ПОТЕРЕЙ

Формат JPEG осуществляет сжатие с потерями, это сложный процесс, который будем описывать преобразованием

$$A \rightarrow A_q,$$

где $q = 1, 2, \dots, 100$ – параметр качества. Он представляет собой вес, с которым используются матрицы квантования DCT (Discrete Cosine Transform) коэффициентов в формате JPEG. Большие значения q соответствуют высокому качеству и, соответственно, низкой степени сжатия.

Чтобы определить возможности плоскостей Грея для встраивания ЦВЗ, рассмотрим следующую схему, которая включает сжатие. Бинарное изображение M встраивается в битовую плоскость B_V и плоскость Грея G_V полутонового контейнера C . В результате получается два стегоизображения S_B и S_G , которые сжимаются вместе с контейнером

$$M, C \rightarrow S_B, S_G, \\ C, S_B, S_G \rightarrow C_q, S_{Bq}, S_{Gq}.$$

Затем осуществляются разные варианты детектирования: слепое и не слепое, D_1 и D_2 . Сообразно встраиванию и детектированию будем рассматривать следующие четыре извлеченных ЦВЗ

M_b, M_{gb}, M_g, M_{gc} , которые определены соотношениями

$$M_b = bitget(C_q, V) + bitget(S_{Bq}, V),$$

$$M_{gb} = bitget(C_q, T) + bitget(S_{Gq}, T),$$

$$M_g = bitget(C_q, V) + bitget(S_{Gq}, V),$$

$$M_{gc} = bitget(S, V) + bitget(S, V+2),$$

где $T = 1, \dots, V$. Здесь M_b получается при встраивании и извлечении из битовой плоскости при не слепом детектировании, M_{gb}, M_g получаются при встраивании в плоскость Грея и извлечении из битовой плоскости и плоскости Грея при не слепом детектировании, наконец, M_{gc} получается при встраивании в плоскость Грея и слепом детектировании.

5. ЭКСПЕРИМЕНТ

Основная цель эксперимента сравнить встраивание в битовые плоскости и плоскости Грея и определить какой из путей детектирования изображения, скрытого в плоскости Грея, будет лучше.

Для этого мы рассчитывали меры искажения $d(M, X)$ между исходным изображением M и извлеченными после сжатия $X = M_b, M_{gb}, M_g, M_{gc}$. В качестве мер были выбраны евклидово расстояние $e(M, X)$, пиковое отношение сигнал шум $PSNR(M, X)$ и относительная энтропия $Q(M \| X)$. Для двух матриц M и X эти величины определяются следующим образом

$$e(M, X) = \sqrt{(1/Z) \sum_{m,n} (M_{mn} - X_{mn})^2},$$

$$PSNR(M, X) = 20 \log_{10} \max |M_{mn}| / e(M, X),$$

$$Q(M \| X) = \sum_j p_M[j] (\log_2 p_M[j] - \log_2 p_X[j]), \quad (7)$$

где Z число элементов, $p_M[j]$, $p_X[j]$ гистограммы яркости M и X . Смысл этих величин хорошо известен. Чем меньше евклидово расстояние и соответственно больше PSNR, тем изображения «ближе» друг к другу. Считается, что значение PSNR от 20 до 40 децибел соответствует приемлемому визуальному качеству [11]. Относительная энтропия характеризует различие гистограмм, чем она меньше, тем меньше различаются гистограммы.

Встраивание в битовую плоскость и плоскость Грея. Бинарное изображение M и полутоновый контейнер C , представлены на рис. 1 а и б. Использовалась битовая плоскость B_4 и плоскость Грея G_4 . Выбор $V = 4$, а не младших плоскостей $V = 1, 2$ обусловлен следующим компромиссом. При JPEG-сжатии сильнее разрушаются младшие битовые плоскости, поэтому лучше встраивать в более старшие, что, однако, может приводить к «просвечиванию» скрываемого изображения. В нашем случае при $V = 4$ контейнер и стегонизображение визуально неразличимы. Заметим, для этого случая простые атаки типа χ^2 могут быть неэффективны. Для сжатия с параметром качества $q = 90$ извлеченные ЦВЗ из битовой плоскости и плоскости Грея M_b и M_g показаны на рис. 1 в и г. Здесь использовано не слепое детектирование. Визуальное качество у M_g лучше. Однако изображение, которое извлечено из плоскости Грея после сжатия, имеет не только более высокое визуальное качество, но и лучшие меры искажения, определенные согласно (7). Для примера на рис. 2 приведена зависимость PSNR, усредненная по 80 изображениям. Мы выбрали изображения из коллекции Caprichos де-Гой. Они имеют сложную текстуру, что позволяет встраивать ЦВЗ в старшие битовые плоскости. Для параметра качества $q > 50$, эта область представляет наибольший интерес, для M_g получаются значения PSNR от 15 до 30 дБ, что свидетельствует в пользу приемлемого визуального качества.

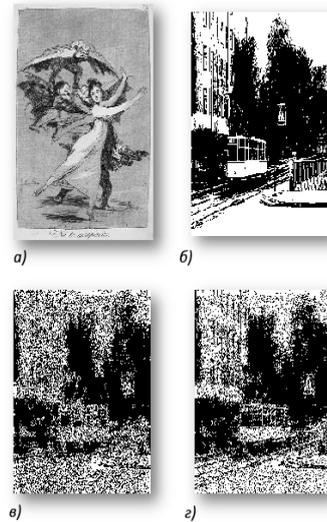


Рис. 1. Встраивание в битовую плоскость и плоскость Грея:

- а) – полутоновый контейнер; б) Бинарный ЦВЗ;
- в) – ЦВЗ, извлеченные из битовой плоскости, $q=90$;
- г) ЦВЗ, извлеченные из плоскости Грея, $q=90$

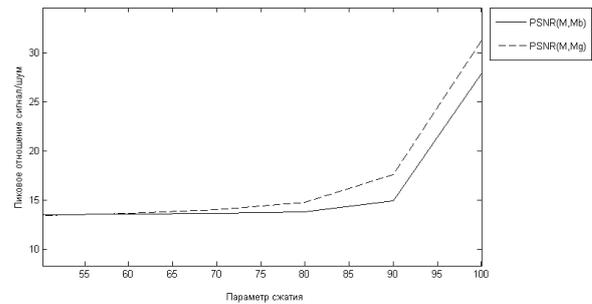


Рис. 2. Пиковое отношение сигнал шум при встраивании в битовую плоскость и плоскость Грея

Из полученных результатов следует, что выбор между встраиванием в битовые плоскости и плоскости Грея решается в пользу последних.

Встраивание в плоскость Грея с детектированием из битовой плоскости и плоскости Грея. Здесь мы рассмотрели варианты слепого и не слепого детектирования для массива из 200 полутоновых изображений. ЦВЗ встраивалось в плоскости $V = 2, 3, 4$ с характерными значениями $q = 70, 80, 90$. Были рассчитаны меры искажения $d(M, X)$, определенные согласно (7) между исходным изображением M и извлеченными после сжатия $X = M_b, M_{gb}, M_g, M_{gc}$. На рис. 3 а и б приведены значения евклидова расстояния и относительной энтропии, усредненные по 200 изображениям, в зависимости от V . Все функции убывают, это связано со следующим обстоятельством. При сжатии большее искажение испытывают младшие битовые плоскости, поскольку они представляют избыточность изображения. Меры приведены для параметра качества $q = 80$, что является не сильно щадящим режимом сжатия. Видно, что наилучшие показатели, это наименьшее евклидово расстояние и относительная энтропия между M и M_g , для не слепого детектирования. Аналогичный результат для PSNR. На рис. 3 в и г приведены две

соответствующие гистограммы евклидова расстояния и относительной энтропии. Это распределение значений этих мер, полученное из 200 изображений. Для евклидова расстояния и относительной энтропии возникают три хорошо различающиеся распределения. Их средние значения соответствуют точкам $V = 4$ кривых в верхнем ряду. Хорошо различимые распределения характеризуют различие выбранных случаев встраивания.

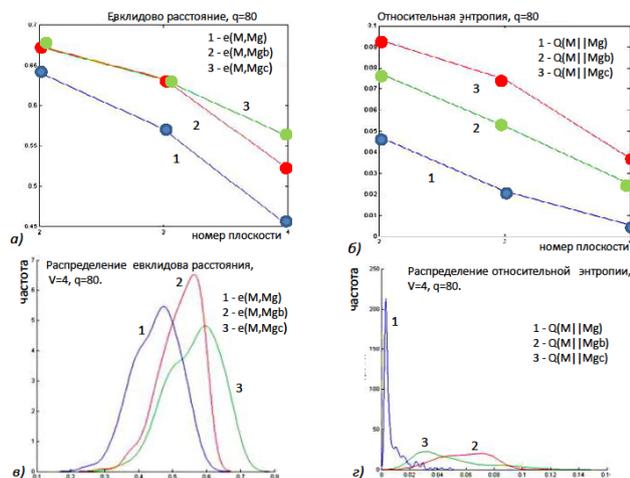


Рис. 3: Меры искажения и гистограммы для встраивания в плоскости Грея. а) Евклидово расстояние; б) Относительная энтропия; в) Распределение значений евклидова расстояния для $V=4$; г) Распределение значений относительной энтропии для $V=4$.

Из найденных зависимостей следует, что при сжатии наиболее благоприятным вариантом детектирования является слепое.

6. ВЫВОДЫ

Плоскости Грея цифрового изображения наряду с битовыми плоскостями могут быть использованы для сокрытия бинарного изображения. Для алгоритмов типа LSB встраивание в плоскость Грея эквивалентно использованию нескольких битовых плоскостей. Особенности проявляются при JPEG сжатии стегоизображение. Это преобразование возникает естественным образом, когда изображение с ЦВЗ сохраняется на диске в jpg-файле. Наличие канала сжатия приводит к различным вариантам детектирования, среди которых возникает слепое детектирование. Чтобы оценить возможности разных вариантов, мы вычислили меры искажения между исходными и извлеченными после сжатия ЦВЗ. Использовались такие меры как евклидово расстояние, PSNR и относительная энтропия. Из расчетов, проведенных для массива изображений, следует, что лучше всего ЦВЗ встраивать и извлекать в плоскости Грея.

7. ССЫЛКИ

- [1] Хэмминг Р.В. Теория кодирования и теория информации. М., Радио и связь, 1983.
- [2] I.J. Cox, M.L. Miller, J.F. Bloom, J. Fridrich, T. Kaler. Digital Watermarking and Steganography. ELSEVIER, Morgan Kaufmann Publishers, 2008.
- [3] Тимбай Е.И. Применение корректирующего фильтра для повышения качества изображений, сжатых методом JPEG. Компьютерная оптика, том 35, №4, 2011. – С. 513 - 518.

[4] Лапшенков Е.М. Модель оценки потерь качества графического изображения при сжатии с потерями, ориентированная на системы распознавания образов. Компьютерная оптика, том 35, № 3, 2011. – С. 408 - 415.

[5] P. H.W. Wong, O.C. Au. Capacity Estimation Technique for JPEG-to-JPEG Image Watermarking.

IEEE Transaction on circuit and system for video technology, Volume 13, No 8, 2003. – P. 74 - 78.

[6] Nguyen B.C., Yoon S.M., Lee H-K. Proceedings of the 5th international conference on Digital Watermarking, Y.Q. Shi and B. Jeon (Eds.): IWDW 2006, LNCS 4283. – P. 61 - 70. Springer-Verlag Berlin Heidelberg 2006.

[7] Aгаian, S., Rodriguez, B., Perez, J.: Stego sensitivity measure and multibit plane based steganography using different color models. In: Delp, E., Wong, P. (eds.) Proc.Security, Steganography, and Watermarking of Multimedia Contents VIII, February 2006, vol. 6072. – P. 279 - 290.

[8] Barbier J., Mayour K. Steganalysis of Multi Bit Plane Image Steganography. Proceeding IWDW '07 Proceedings of the 6th International Workshop on Digital Watermarking Pages 99 - 111, Springer-Verlag Berlin, Heidelberg 2008, ISBN: 978-3-540-92237-7 (2008).

[9] Xiangyang Luo, Fenlin Liu, Chunfang Yang, Shiguo Lian, Ying Zeng Multimedia Tools and Applications (Impact Factor: 0.62). 08/2013; DOI:10.1007/s11042-010-0663-3, 2013.

[10] Ker A.D. Improved detection of LSB steganography in grayscale images. In: Proc. 6th Information Hiding Workshop. Volume 3200 of Springer LNCS, 2004. – P. 97 - 115.

[11] Ker A.D. Steganalysis of Embedding in Two Least-Significant Bits. IEEE Transactions on Information Forensics and Security, vol. 2, no. 1, March 2007 – P. 46 - 54.

[12] Сэлмон Д. Сжатие изображений и звука. – М.: Техносфера, 2004.

Об авторах

Горбачев Валерий – преподаватель Северо-Западного института печати Санкт-Петербургского государственного университета технологии и дизайна (СЗИП СПб ГУТД).

Денисов Лев – выпускник СЗИП СПб ГУТД.

E-mail: denisovlev@yandex.ru

Кайнарова Елена – преподаватель СЗИП СПб ГУТД.

E-mail: helenkainarova@gmail.com

Яковлева Елена – преподаватель НМСУ «Горный».

E-mail: 2305lena@mail.ru

Modification of the Multi-target Tracking Algorithm Based on Energy Minimization

A. Gringauz¹, E. Shalnov², A. Konushin³
^{1,2,3}Lomonosov Moscow State University

Department of Computational Mathematics and Cybernetics

¹grin3s@mail.ru, ²shalnov.eugen@gmail.com, ³ktosh@graphics.cs.msu.ru

Abstract

In the current work we consider a multi-target tracking problem. The proposed algorithm is based on energy minimization within a temporal sliding window and is a modification of the approach from [10]. We propose a method to impose physical constraints on the trajectory such as curvature or maximum velocity using the energy. Experimental evaluation of the algorithm shows that we were able to achieve a performance improvement compared to the base method [10].

Keywords: tracking, MCMC DA, energy minimization.

1. 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. Despite a significant progress in recent years, humans are still far ahead of existing automatic algorithms in terms of solving this task.

In the current work we consider tracking by detection as one of the most promising approaches to this task.

All methods of multi-target tracking can be divided into three groups.

Methods from the first group determine the position of the object in the current frame based its position in the previous frame. Some examples of this approach are algorithms based on Kalman filtering [5] or particle filtering [6].

Methods from the second group use information from the following frames to estimate the person's position in the current frame [10, 3]. These algorithms use energy minimization within a temporal sliding window. The current frame is somewhere in the sliding window, and people's estimated positions might change when the window moves.

The third group is similar to the second. These methods use energy minimization on the whole video fragment, that is the size of the sliding window is equal to the duration of the video. The possibility of defining a continuous energy function that depends on every person's position in all frames was researched in [1, 7].

Besides this approach the same authors developed a discrete-continuous method [2]. In this approach the energy function is divided into two parts: continuous and discrete. Every trajectory is modeled by a cubic B-spline. The continuous part contains terms that impose constraints on how close the trajectory is to detections, velocity, inter-object occlusions. The discrete part is responsible for associating detections from adjacent frames.

The proposed algorithm is a modification of [10]. Due to the improvements the modified algorithm shows better performance than the base method.

2. PROPOSED METHOD

The base algorithm and its modification are described below. It consists of the following steps: applying object detector to

every frame of the video, building tracklets, building trajectories, estimating people's positions in intermediate frames.

2.1. Searching For Objects

The objects we want to track in the video are people. So we use a HOG based detector [9] to find all people's heads. Using a head detector instead of a full body detector allows us sometimes to find a person even in case of an occlusion.

2.2. Building Tracklets

Then for every found detection a tracklet is built. A tracklet is an object containing information about a detection and a set of its motion estimates. A tracklet is built based on the information obtained using the "Flock of features" tracking algorithm. It uses only one detection and tracks it for several frames forwards and backwards. The position of the head of the person in the frame where it has been detected and its position found by the visual tracking algorithm are used to build the estimate. It is important that as the time of tracking with this algorithm increases the probability of finding a wrong position for a person also increases. That is the reason why it is used only to determine a position of a person in a small temporal neighborhood of the detection.

2.3. Building Trajectories

Let D denote the set of all detections in the temporal sliding window. H is the hypothesis that shows how D is divided into trajectories, that is: $H = \{T_1, \dots, T_J\}$, $T_i = \{d_n^i\}$, d_n^i is the n^{th} detection in the j^{th} trajectory. Using Bayes' theorem:

$$(1) \quad p(H | D) = \frac{p(D | H)p(H)}{p(D)} \propto p(D | H)p(H)$$

We must find:

$$(2) \quad H^* = \arg \max_H p(H | D)$$

Following [10] let's assume that:

$$(3) \quad p(H) = J! \prod_{T_j \in H} \left(\frac{|T_j|}{|D|} \right)^{|T_j|} p(c_j)$$

Here $|X|$ is the number of elements in the set X , c_j is the type of the j^{th} trajectory. Like in the base algorithm we propose two types of a trajectory: c_{fp} (a trajectory of false positives) and c_{ped} (a person's trajectory).

Let's consider the factor $p(D | H)$. In the base algorithm the trajectory was modeled by a Markov chain (see figure 1).

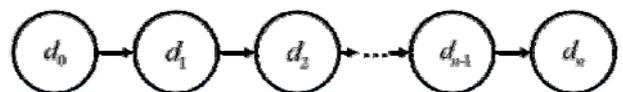


Fig. 1. Trajectory Model

The likelihood of this Markov chain is:

$$(4) \quad p(D|H) = \prod_{T_j \in H} \left[p(d_1^j | c_j) \prod_{d_i^j \in T_j, d_i^j} p(d_n^j | d_{n-1}^j, c_j) \right]$$

This representation has its drawbacks:

- It imposes constraints on the behavior of the trajectory in the neighborhood of every point but it doesn't impose global constraints on the whole trajectory.
- The likelihood of the trajectory doesn't depend on other trajectories' behavior.

To overcome some limitations of the base algorithm we propose to use the idea from [2] to use B-splines to model the trajectory. The new trajectory model is shown in figure 2.

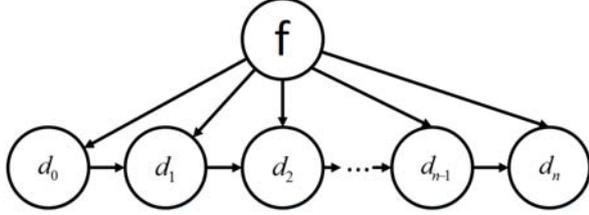


Fig. 2. Modified Trajectory Model

The likelihood of this Bayesian network is:

$$(5) \quad p(D|H) = \prod_{T_j \in H} \left[p(f_j | c_j) p(d_0^j | f_j, c_j) \prod_{d_i^j \in T_j, d_0} p(d_n^j | d_{n-1}^j, f_j, c_j) \right]$$

Here f_j is the spline, constructed for the trajectory T_j .

Following the work [10], every detection d_n^j is described by its size s_n , position x_n and motion estimate m_n . The likelihood of a single detection is:

$$(6) \quad p(d_n^j | c_j, f_j) = p(s_n) p(x_n) p(m_n | c_j)$$

$$(7) \quad p(d_n^j | d_{n-1}^j, c_j, f_j) = p(s_n | s_{n-1}) p(x_n | x_{n-1}, c_j) p(m_n | c_j)$$

2.3.1. Detection Size

The size of the first detection of a trajectory can't be estimated based on previous frames, so let's define an a priori distribution:

$$(8) \quad \ln(s_1) \sim N(\mu_p, \sigma_p^2)$$

The sizes of the following detections depend on the previous ones:

$$(9) \quad \ln\left(\frac{s_i}{s_{i-1}} | c_{ped}\right) \sim N(0, \delta_i \sigma_{s,p}^2)$$

$$(10) \quad \ln\left(\frac{s_i}{s_{i-1}} | c_{fp}\right) \sim N(0, \delta_i \sigma_{s,f}^2)$$

Here δ_i is the time difference between frames which the corresponding detections were found on.

2.3.2. Detection Position

We assume that the position of false positives may change only due to the noise.

$$(11) \quad x_i | x_{i-1}, c_{fp} \sim N(x_{i-1}, 2\Sigma_d)$$

Let v_p be a motion estimate, obtained from the tracking results from the previous frame and the next frame. This estimate is considered the most reliable. Consider the distribution $x_i | x_{i-1}, c_{ped}$. First let's derive the estimate for the person's position at time i using the motion estimate v_p :

$$(12) \quad x_p^{i-1} = x_{i-1} + \delta_i v_p$$

$$(13) \quad \Sigma_p = \delta_i \Sigma_v$$

Here Σ_v is a parameter modeling an error in the speed v_p . In

order to improve the estimate x_p^{i-1} we propose to use the remaining motion estimates. Let $Y_i = \{y_i\}$ be the set of motion estimates for a detection at time i . The more accurate estimate x_y^{i-1} is given by:

$$(14) \quad x_y^{i-1} = x_p^{i-1} + \Delta x_y$$

$$(15) \quad x_p^{y_{i-1}} = x_{i-1} + \delta_i y_{i-1}$$

$$(16) \quad \Delta x_y = \Sigma_p (\Sigma_p + \delta_i \Sigma_{local})^{-1} (x_p^{y_{i-1}} - x_p^{i-1})$$

$$(17) \quad \Sigma_y = (I - \Sigma_p (\Sigma_p + \delta_i \Sigma_{local})^{-1}) \Sigma_p$$

The parameter Σ_{local} describes the error of the algorithm used for building the tracklets (see section 2.2), δ_i is the time difference between detections d_i and d_{i-1} .

Although this estimate is more accurate, it is less reliable, because the error of the tracking algorithm used for building tracklets increases. Therefore, we propose to define the distribution for a person's position at time i as a mixture of normal distributions:

$$(18) \quad x_i | x_{i-1}, Y_{i-1}, c_{ped} \sim \frac{\alpha^{\delta_i}}{|Y_{i-1}|} \sum_{y \in Y_{i-1}} N(x_y^{i-1}, \Sigma_y + 2\Sigma_d) + (1 - \alpha^{\delta_i}) N(x_p^{i-1}, \Sigma_p + 2\Sigma_d)$$

Here α is the probability of the visual tracking algorithm to lose a person in a given frame.

In the work [10] it was shown that using motion estimates Y_i of the tracklet d_i , besides the motion estimates Y_{i-1} of the tracklet d_{i-1} , improves tracking performance. Therefore it was proposed to use motion estimates from both associated tracklets.

$$(19) \quad p(x_i | x_{i-1}, Y_i, Y_{i-1}, c_{ped}) = \beta p(x_i | x_{i-1}, Y_i, c_{ped}) + (1 - \beta) p(x_i | x_{i-1}, Y_{i-1}, c_{ped})$$

$$(20) \quad \beta = \frac{|Y_{i-1}|}{|Y_{i-1}| + |Y_i| + 2}$$

2.3.3. Motion Magnitude

The motion estimate m is needed to distinguish a person's trajectory from a trajectory of false positives. Following the work [10], it is modeled by a histogram of 4 bins. The boundaries of the bins represent the movement of $\frac{1}{8}, \frac{1}{4}, \frac{1}{2}$ of

the detection's size. We assume that the histogram corresponds to one of the two multinomial distributions, depending on the type of the track:

$$(21) \quad m_i | c_{ped} \sim \text{Mult}(m_{ped})$$

$$(22) \quad m_i | c_{fp} \sim \text{Mult}(m_{fp})$$

2.3.4. Spline Distribution

Let's define an a priori distribution for a spline $p(f)$. Let K be a number of parts in the spline. Then it can be defined by a coefficient matrix $C \in \mathbb{R}^{2K \times 4}$. Number 2 represents the fact that splines are constructed for two dimensions x and y . Supposing a spline consists of polynomials:

$$(23) \quad f_i(t) = a_i t^3 + b_i t^2 + c_i t + d_i$$

$$C = [a_i, b_i, c_i, d_i], i = \overline{1, 2K}$$

Let's denote:

$$(24) \quad A = \max_i |a_i|$$

And assume:

$$(25) \quad f \sim N(A|0, \sigma^2)$$

In the work [2] it was mentioned that the coefficient of the highest degree of the polynomial has the greatest influence on the person's speed. The parameters of the described distribution were estimated using the maximum likelihood method. The training set consisted of PETS-S2-L1¹, PETS-S2-L2, PETS-S2-L3 and TownCentre² datasets, containing ground truth. Modeling parameter a_i with a discrete distribution (see figure 3) showed that a normal distribution is a reasonable approximation of the distribution of this parameter obtained from the real data.

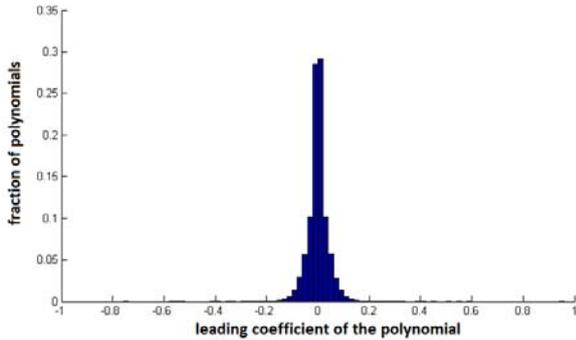


Fig. 3. Statistic for A

2.3.5. Associating Algorithm

As it was described in the beginning of section 2.3, at every moment the algorithm searches for an optimal splitting of tracklets into trajectories within a temporal sliding window. There is no efficient algorithm to find a trajectory hypothesis H^* (see eq. (2)), which is a global maximum point of the likelihood function. Therefore we use an approximate inference method MCMC DA [8]. The Metropolis–Hastings algorithm is used to obtain a sequence of samples from the distribution.

We use an initial hypothesis with high likelihood as an initialization for the algorithm in order to accelerate convergence. This hypothesis can be obtained from the tracking results from the sliding window corresponding to the previous time step. New tracklets are added using a greedy algorithm, tracklets that went beyond the sliding window are removed from the hypothesis.

In order to generate new hypotheses, three types of trajectory transforms are used: "swap", "switch", "change type".

When a "swap" move occurs, two randomly chosen trajectories exchange tracklets having equal time stamps, or if no such tracklets exist, a random tracklet from the first trajectory is moved into the second. The "switch" move makes two random trajectories exchange tracklets from the beginning of the sliding window to a random moment of time. The "change type" move changes the type of a randomly chosen trajectory.

While generating new hypotheses, each of the three types of transforms can be chose with equal probability. The trajectories and moments of time are chosen from a uniform distribution.

The probability of accepting a new hypothesis in the Metropolis-Hastings algorithm is given by:

$$(26) \quad p(H_{i+1} \leftarrow \bar{H}) = \min \left(\frac{p(\bar{H} | D)q(H_i | \bar{H})}{p(H_i | D)q(\bar{H} | H_i)}, 1 \right)$$

¹ <http://www.cvg.rdg.ac.uk/PETS2009/>

² http://www.robots.ox.ac.uk/ActiveVision/Research/Projects/2009bбенfold_headpose/project.html#datasets

Here $q(H_i | \bar{H})$ and $q(\bar{H} | H_i)$ is the probability of switching from the hypothesis \bar{H} to H_i and from H_i to \bar{H} respectively.

2.4. Restoring People's Positions

After finding an approximation of an optimal hypothesis, the people's positions at a certain time within a sliding window are estimated.

It is not necessary for the detections to be obtained in every frame for the algorithm to work. They may be obtained once in several frames. The linear interpolation is used to estimate a person's position in an intermediate frame.

3. EXPERIMENTS

The aim of the experimental evaluation was to compare the base method [10] with its modification. The modified part consists of modeling trajectories using splines, and thus taking into account some physical features of a person's movement (see sec. 2.3 and 2.3.4).

The base algorithm and its modification were evaluated on the TownCentre dataset. It contains a high resolution video sequence (1920×1080 / 25 fps), filmed from a static camera. The calibration matrix and the ground truth are also provided.

To evaluate the methods standard precision and recall metrics were used along with the widely known CLEAR MOT [4] metrics. Here is a brief description of some of them. 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 table.

The results of comparing the base method [10] with its modification

Algorithm	Baseline	Modification
Precision	74.73	76.03
Recall	47.92	50.02
FP	976	950
FN	3137	3011
ID	76	71
MOTA	30.46	33.08
MOTP	44.24	43.95

As it was mentioned in [10], the proposed approach can be used to process a video in real time. But in order to simplify the development process we chose MATLAB to implement the algorithm. Thus the time to process one frame significantly increased. Therefore we had to resort to a very rare usage of the head detector (only once in 20 frames). Our experiments showed that the proposed algorithm was able to perform tracking even in such poor conditions, although the quality of the result significantly dropped.



Fig. 4. An example of tracking results

4. CONCLUSION

In this work we have proposed a modification of the base algorithm [10]. The experimental results showed that the modification improved the base method. It proves that imposing global constraints on the trajectory influences tracking performance. In future work we plan to add new factors such as occlusion reasoning. It is also possible to integrate an appearance model into the likelihood function.

5. ACKNOWLEDGMENTS

This paper was supported by RFBR, research project No. 14-01-00849 a.

6. REFERENCES

- [1] Andriyenko A., Schindler K. Multi-target tracking by continuous energy minimization. CVPR, 2011.
- [2] Andriyenko A., Schindler K., Roth S. Discrete-continuous optimization for multi-target tracking. CVPR, 2012.
- [3] Benfold B. Reid I. Stable multi-target tracking in real-time surveillance video, CVPR, 2011.
- [4] Bernardin K. Stiefelhagen R. Evaluating multiple object tracking performance: The CLEAR MOT metrics. Image and Video Processing, 2008(1):1 – 10, 2008.
- [5] Black J., Ellis T., Rosin P. Multi view image surveillance and tracking. In *Motion & Video Computing Workshop*, Dec. 2002.
- [6] Breitenstein M.D., Reichlin F., Leibe B., Koller-Meier E., Van Gool L. Robust tracking-by-detection using a detector confidence particle filter. In ICCV, 2009.
- [7] Milan A., Schindler K., Roth S. Continuous Energy Minimization for Multi-Target tracking, PAMI, 2013.
- [8] Oh S., Russell S., Sastry S. Markov chain monte carlo data association for general multiple-target tracking problems // Decision and Control. 2004. 1. No 43. – P. 735 - 742.
- [9] Prisacariu V.A., Reid I.D. FastHOG – a real-time GPU implementation of HOG. Technical Report 2310/09.
- [10] Shalnov E., Konushin A. Improvement of MCMC-based video tracking algorithm. Pattern recognition and image analysis (PRIA-11-2013), 2013.

Модификация нейронной сети неокогнитрон для увеличения качества распознавания текста

А. Кугаевских

Институт математики и компьютерных наук

Тюменский государственный университет, Тюмень, Россия

a-kugaevskikh@yandex.ru

Аннотация

В работе представлен новый подход к распознаванию иероглифических текстов со слабо изученной лингвистикой. Классическая нейронная сеть была модифицирована для улучшения качества распознавания. Применение фильтра Габора позволило улучшить выявление границ символов, а совмещение сегментации и распознавания в одной нейросетевой архитектуре улучшить восстановление структуры текста.

Ключевые слова: распознавание текстов, нейронная сеть, фильтр Габора.

1. ВВЕДЕНИЕ

Изучение письменности исчезнувших культур порождает дополнительные трудности, связанные со слабой изученностью лингвистики языка и существованием в древних языках синтаксических элементов, связанных с группировкой иероглифов, а также наличием поврежденных иероглифов и других синтаксических элементов.

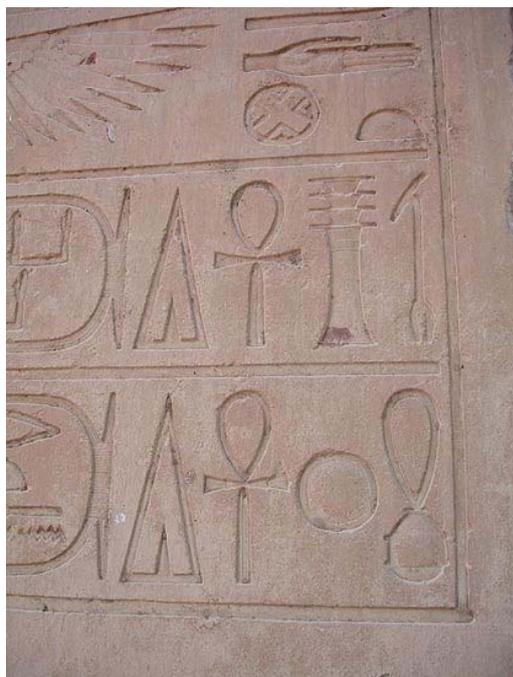


Рис 1. Исходное изображение

Так как лингвистика слабо изучена, мы не можем полагаться на лингвистический анализ при исправлении ошибок распознавания для увеличения качества работы.

В [1] приведены результаты поиска оптимального метода сегментации иероглифического текста и показано, что наиболее подходящий принцип – выявление границ на перепадах яркости. Там же приведена архитектура нейронной сети распознавания выявленных изображений символов.

К настоящему времени наиболее качественным методом для этого является фильтр Габора [3].

Существующие методы сегментации плохо приспособлены для подобного типа текстов ввиду высокой степени повреждения изображения, также возможно неправильное разделение на строки из-за разной высоты символов.

Нет гарантии верной сегментации из-за невозможности достоверно классифицировать компоненты изображения на предмет «шум / элемент изображения символа». А это возможно только после распознавания символа. Поэтому необходимо совместить процесс сегментации и распознавания в одной системе.

2. ОБЩИЙ АЛГОРИТМ

Распознавание иероглифического текста осуществляется в два этапа:

- 1) идентификация структурных элементов текста;
- 2) определение синтаксических параметров текста и его элементов (группы, высота текста и т.д.).

За первый этап отвечает модифицированная нейронная сеть неокогнитрон (рис. 2). Второй этап основывается на результатах первого, поэтому нейронная сеть не только должна указать класс, к которому относится изображение символа, но и определить некоторые параметры (угол поворота, степень повреждения, направление письма, высоту символа).

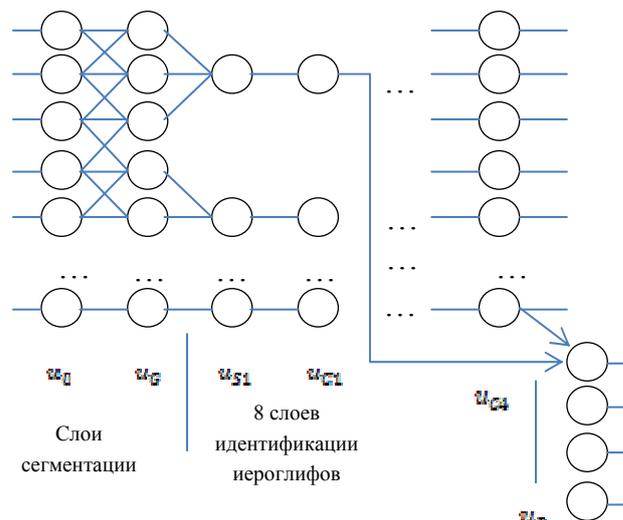


Рис. 2. Модифицированная структура нейросети

Для решения задачи сегментации на вход нейронной сети должно поступать изображение всего текста, а не отдельного символа.

Для улучшения качества выделения границ компонентов изображения в слое U_G стал применяться фильтр Габора, адаптированный под нейросетевую парадигму.

3. МОДИФИКАЦИЯ ИЗВЛЕЧЕНИЯ ГРАНИЦ И СЕГМЕНТАЦИЯ

Ядро фильтра Габора является произведением гауссиана и гармонической функции, фильтрует сигнал, основываясь на параметре предпочтительной пространственной частоты λ^{-1} :

$$g[k, l] = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cos\left(\frac{2\pi x'}{\lambda} + \psi\right), \quad (1)$$

Непосредственно же выявление границ сегментов осуществляется при проведении математической операции свертки ядра фильтра Габора и изображения в градациях серого:

$$r = \iint_{\Omega} f(u - k, v - l) g(k, l) dk dl, \quad (2)$$

Размер слоя U_G , отвечающего за выделение контуров сегментов, определяется размером исходного изображения. Значения свертки больше нуля означают наличие границы, поэтому в качестве передаточной функции нейрона используется функция Хевисайда. Преобразуя (2) к взвешенной сумме и подставив формулу (1), получим передаточную функцию нейрона:

$$u_G(m, n) = \varphi \left[\sum_{|k| < A_G} u_G(m + k, n + l) \exp\left(-\frac{x'^2 + 0.25y'^2}{2A_G^2}\right) \cos\left(\frac{2\pi x'}{\lambda} + \frac{\pi}{2}\right) \right], \quad (3)$$

где $\varphi[x] = \max(x, 0)$ – пороговая функция;

A_G – размер рецептивного поля;

m, n – позиция центра рецептивного поля;

k, l – позиция светового импульса в рецептивной области;

λ – количество параллельных возбуждающих и тормозных зон в рецептивном поле

Согласно исследованиям, опубликованным в [2], оптимальное значение предпочтительной пространственной частоты может быть вычислено, исходя из соотношения:

$$A_G = 0.56\lambda$$

$$x' = k \cos \theta + l \sin \theta$$

$$y' = -k \sin \theta + l \cos \theta$$

При этом для текущего пикселя рассматриваются яркости соседних пикселей в некоторой области A_G , которые влияют на степень граничности текущего пикселя. Таким образом, можно оценить степень граничности всех пикселей на изображении в зависимости от их соседей, и такая локальная обработка будет инвариантна к смене освещенности или цвета изображения.

Выход уровня U_G подается на слой извлечения признаков (U_{S1}). Он содержит ровно столько плоскостей, сколько неповторяющихся признаков встречается на всей обучающей выборке.

На рис. 3 представлен результат работы слоя извлечения границ

Сегментация осуществляется следующим образом. Каждый нейрон слоя извлечения признаков при активации запоминает координаты нейронов слоя извлечения границ, по активированным связям, соединенным с ним. При передаче от слоя к слою координаты отдельных признаков преобразуются в координаты и размер прямоугольника,

описанного около распознанного изображения символа. Таким образом, после распознавания символа, пройдя обратный путь по нейронной сети по активированным нейронам, можно выявить позицию сегмента на изображении. После распознавания всего текста, на основании этой информации, можно формировать строки текста.

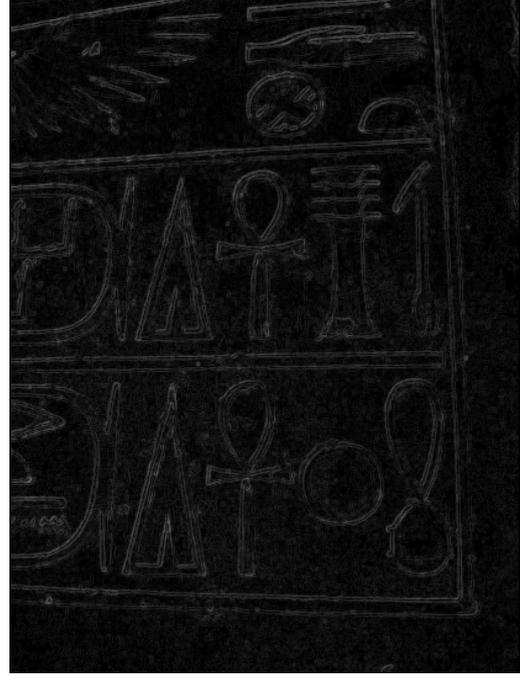


Рис. 3. Применение фильтра Габора

4. МОДИФИКАЦИЯ ДЛЯ ОПРЕДЕЛЕНИЯ ПАРАМЕТРОВ СИМВОЛА

Для придания сети инвариантности к поворотам символов был добавлен параметр α ($\alpha \in [0, \pi]$ с шагом 15°). В дальнейшем параметр угла наклона признака символа понадобится только при определении общего угла наклона символа при восстановлении структуры текста.

Шаг изменения параметра α выбран не случайно. Оптимально настроенный и обученный неоконитрон перестает отвечать на признаки, отклоненные более чем на $15 - 20$ градусов.

Подход, предложенный Фукушимой, позволяет классифицировать изображение по признакам, но не дает ответа на вопрос о степени повреждения изображения элемента текста. Для выявления степени повреждения сегмента нейронная сеть дополняется слоем U_D , состоящим из 4 нейронов, отвечающих за соответствующую четверть сегмента.

Каждый нейрон слоя U_D получает связи от нейронов слоя U_{C1} . Ответ такого нейрона должен обладать обратной пропорциональной зависимостью от степени активации признаков символа (т.е. слоя U_{C1}). Чем меньше активация соответствующих нейронов слоя U_{C1} , тем выше вероятность, что символ поврежден в соответствующей четверти. Поэтому передаточной функцией является функция Хевисайда, при которой нейрон будет активироваться только при положительном сигнале. Ответом каждого нейрона U_D будет:

$$u_D(m) = \varphi \left[1 - \frac{\sum_{k=1}^{K_{C1}} \sum_{|l| < A_D} a_D(v) u_{C1}(n + v, k, \alpha)}{\theta_D} \right] \quad (4)$$

где A_D – размер области активации нейрона;

$\alpha_D(v)$ – сила постоянных возбуждающих связей;

θ_D – порог чувствительности.

Модификация нейросетевой модели К. Фукушимы в части дополнения слоев выявления степени повреждения требует обучения нейронов этого слоя.

Обучение нейрона этого слоя производится с применением правила Хебба, ставящего изменение весовых коэффициентов в зависимость от входных и выходных значений нейрона. Алгоритм обучения слоя выявления повреждения заключается в выполнении следующих шагов:

Шаг 1. Обучение слоев распознавания.

Шаг 2. Начальным значениям $\alpha_D(v)$ присваиваются соответствующие значения $\alpha_{S2}(v)$.

Шаг 3. На вход нейросети подается поврежденный образ элемента

Шаг 4. Расчет изменения веса

$$\Delta \alpha_D(v) = q_D u_D(m) u_{C1}(n + v, k, \alpha) \quad (5)$$

Шаг 5. Переход к следующему нейрону и возврат на шаг 4.

Шаг 6. Если не конец обучающей выборки – возврат на шаг 3, иначе – выход.

В результате обучения нейроны этого слоя становятся отзывчивыми к отсутствию наиболее важных признаков, характеризующих степень повреждения элемента текста. При этом признаки обладают разной степенью важности.

5. РЕЗУЛЬТАТЫ

На неповрежденных образцах изображения нейросеть показала среднее качество распознавания на уровне 82%. Этот показатель зависит от материала-носителя текста, степени повреждения.

Тестовая выборка для проведения экспериментов по качеству распознавания текстов составила 358 изображений текстов.

Тестовую выборку текстов можно условно разделить на четыре группы по материалу:

1. папирус, холст, кожа;
2. камень;
3. керамика;
4. дерево.

Материал из каждой группы обладает разной фактурой и степенью повреждения, что по-разному влияет на качество распознавания. Результаты проверки на текстах из каждой группы и сравнение показателей качества распознавания без применения и с применением фильтра Габора приведены в таблице.

Результаты проверки качества распознавания

Материал	Кол-во текстов	Качество распознавания, без применения фильтра Габора	Качество распознавания, с применением фильтра Габора
Папирус, холст, кожа	32	79,9%	82,1%

Камень	294	79,1%	84,6%
Керамика	7	78,4%	80,3%
Дерево	25	78%	81%
Итого	358	78,85%	82%

На рис. 4 приведена зависимость качества распознавания от степени повреждения образа символа. При увеличении степени повреждения до 50% и более наблюдается резкое снижение качества распознавания. Это объясняется большим количеством символов алфавита и их похожестью, так как при повреждении изображений символов может быть потеряна информация о важных признаках, отличающих символ от других.

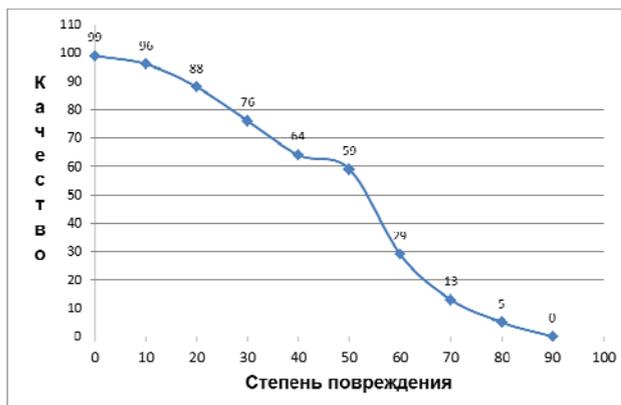


Рис. 4. Зависимость качества распознавания от степени повреждения

6. ЗАКЛЮЧЕНИЕ

Результаты проведенных экспериментов подтверждают, что применение фильтра Габора улучшает распознавание символов. А совмещение сегментации и распознавания в одной системе позволяет восстановить структуру текста. Среднее качество распознавания синтаксических элементов составило 82%, что подтверждает адекватность разработанных методов и моделей распознавания иероглифических текстов.

7. ССЫЛКИ

- [1] Кугаевских А.В., Ивашко А.Г. Возможность применения искусственных нейронных сетей "неокогнитрон" для распознавания древнеегипетских иероглифов // Вестник Тюменского государственного университета. – Тюмень: ТюмГУ, 2009 – № 6. – С. 209 - 214.
- [2] DeValois R.LAlbrecht., D.G. and Thorell L.G. Spatial frequency selectivity of cells in macaque visual cortex, Vis. Res., vol. 22, 1982. – P. 545 - 559.
- [3] Grigorescu S.E., Petkov N. and Kruizing P. Comparison of Texture Features Based on Gabor Filters, IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 11, NO. 10, OCTOBER 2002. – 1160 - 1167

Об авторах

Кугаевских Александр – к.т.н., доцент кафедры ИСТюмГУ.
E-mail: a-kugaevskikh@yandex.ru

Анализ и визуализация многомерных данных в задачах вычислительной газовой динамики

А. Бондарев, В. Галактионов, Л. Шапиро
Институт прикладной математики им. М.В. Келдыша РАН
Москва, Россия
bond@keldysh.ru; vlgal@gin.keldysh.ru; pls@gin.keldysh.ru

Аннотация

В докладе рассматриваются вопросы анализа и визуализации многомерных объемов данных в задачах вычислительной газовой динамики. Многомерные данные появляются в вычислительной газовой динамике как результаты параметрических исследований и решения задач оптимизационного анализа. Предлагается приближенный подход, предполагающий отображение многомерного объема в пространстве главных компонент и аппроксимацию данных в объеме с помощью плоскостей. Приведен пример практического применения подхода.

Ключевые слова: визуализация многомерных данных, метод главных компонент, вычислительная газовая динамика

1. ВВЕДЕНИЕ

Задачи обработки, анализа и визуализации многомерных данных являются на сегодняшний день важным и актуальным направлением. Проблемы изучения многомерных объемов данных, задачи определения взаимного расположения точек в многомерном облаке данных, задачи выявления определяющих факторов и скрытых взаимосвязей между ними возникают практически во всех областях знания. Анализ многомерных данных (Data Analysis) интенсивно развивается как научная дисциплина, которая включает в себя: метод главных компонент (PCA-Principal Component Analysis) и его обобщения на нелинейные случаи, факторный анализ, кластерный анализ, дискриминантный анализ, построение самоорганизующихся карт (SOM – Self-Organized Maps) и упругих карт (Elastic Maps) [1,2]. Комбинированное применение методов, алгоритмов и подходов, разработанных в этих разделах, позволяет провести всестороннее исследование многомерного объема данных вне зависимости от их происхождения.

В задачах вычислительной газовой динамики проблемы анализа многомерных данных ранее практически не встречались. Для обработки и визуального представления результатов даже самых сложных расчетов вполне хватало наработанных методов и приемов научной визуализации [3]. Однако в настоящее время интенсивное развитие высокопроизводительных и параллельных вычислений позволяет решать задачи параметрического исследования и задачи оптимизационного анализа [4, 5].

Параметрические численные исследования позволяют получать решение не для одной конкретной задачи математического моделирования, а для класса задач, заданного в многомерном пространстве определяющих параметров. Также применение параллельных алгоритмов на высокопроизводительной вычислительной технике позволяет

численное исследование задач оптимизационного анализа, когда обратная задача решается в каждой точке сеточного разбиения многомерного пространства определяющих параметров. Основная особенность, с точки зрения задач анализа и визуализации решений, в подобных вычислениях заключается в том, что их результаты представляют собой многомерные массивы, размерность которых соответствует количеству определяющих параметров. Эти массивы нуждаются в обработке и визуальном представлении с целью их анализа и выявления внутренних взаимосвязей между определяющими параметрами. Подобные задачи начинают встречаться на практике все чаще, хотя следует отметить, что размерность подобных массивов сегодня ограничивается вычислительными мощностями и обычно составляет 4 – 5, в исключительных случаях – 6.

В этой ситуации естественно хотелось бы применить уже наработанный аппарат методов и алгоритмов Data Analysis к подобным задачам. Однако здесь возникают некоторые проблемы, обусловленные спецификой целей исследования и происхождения самих данных. В задачах Data Analysis многомерные данные рассматриваются как набор точек $A_i(x_1, \dots, x_n), i = 1, \dots, m$, и основной интерес для исследователя представляет их взаиморасположение с целью выделения кластеров, решения задачи классификации новых объектов. Когда мы рассматриваем многомерные данные в задачах вычислительной газовой динамики (CFD), полученные как результаты решения задач оптимизационного анализа или параметрических исследований, нас в гораздо меньшей степени интересует взаиморасположение точек, так как разбиения по определяющим параметрам x_1, \dots, x_{n-1} задаются нами. Основная цель здесь – изучение зависимости $x_n = F(x_1, \dots, x_{n-1})$, представленной по результатам вычислений в виде многомерного объема данных, визуализация этой зависимости и, по возможности, представление ее в квазианалитическом виде с помощью приближений.

Таким образом, задача адаптации методов Data Analysis для целей исследования многомерных результатов расчетов газодинамических задач является актуальной.

2. АНАЛИЗ И ВИЗУАЛИЗАЦИЯ МНОГОМЕРНЫХ ДАННЫХ

Наиболее эффективным путем анализа многомерных данных, получаемых в результате решения задач вычислительной газовой динамики, является визуальное представление зависимости $x_n = F(x_1, \dots, x_{n-1})$ и получение информации о характере этой зависимости. Далее следует аппроксимация

зависимости с помощью поверхностей достаточно простого вида и получение, как следствие, искомого квазианалитического выражения.

В работе [4] рассматривались современные попытки построения визуальной концепции для представления многомерных данных, а также отмечалось отсутствие в настоящее время адекватного и надежного способа подобного визуального представления для объемов, имеющих размерность, превышающую 3. Следовательно, для анализа информации, содержащейся в полученном многомерном массиве, необходимо понизить размерность массива. Рассмотрим наиболее распространенные практические способы понижения размерности.

Рассматриваемые способы основаны на анализе дисперсий данных массива по координатным направлениям или нахождении в изучаемом многомерном пространстве, по направлению которого дисперсия максимальна.

Первый способ представляет собой поиск координатного направления с наименьшей дисперсией. Вычисляются дисперсии D_i по всем координатным направлениям, выбирается наименьшая из них, и в том случае, когда минимальная дисперсия существенно (на порядок) меньше остальных, значения исследуемой функции по координатному направлению с наименьшей дисперсией заменяются на константу, равную среднему значению по направлению. Таким образом, размерность исходного многомерного пространства понижается на единицу.

Более радикальный вариант данного способа выглядит следующим образом. Вычисляются дисперсии по всем координатным направлениям и ранжируются в порядке убывания. Выбираются три направления, соответствующих максимальным дисперсиям D'_1, D'_2, D'_3 . Далее проверяется условие $D'_j \gg \varepsilon * D'_i, i \neq 1, 2, 3$, где ε – малая величина, $j=1, 2, 3$

задаваемая пользователем. Если это условие выполнено, то полагаем значения искомой функции по всем направлениям, кроме трех, соответствующих максимальным дисперсиям, константами, равными соответствующим средним значениям по направлениям. Таким образом, мы радикально понижаем размерность исходного пространства и оказываемся в рамках стандартного трехмерного пространства.

Изложенный подход обладает рядом недостатков:

- он далеко не всегда осуществим, например, если данные в многомерном пространстве близки к гиперсфере;
- в выборе малой величины ε заложен произвол.

Однако, несмотря на эти недостатки, для пространств небольшой размерности $n = 4, 5$ во многих практических случаях данных подход работает успешно.

Второй распространенный способ понижения размерности заключается в построении графических проекций на стандартное число измерений $n \leq 3$ с фиксацией переменных, не участвующих в построении проекции. В тех случаях, когда из набора дисперсий по направлениям нельзя выделить существенно наименьшую, часто используется метод разделения переменных.

Если из вида проекций в стандартных измерениях удастся сделать вывод о том, что для двух переменных при

фиксированных остальных переменных исследуемая функция может быть выражена с помощью аналитической зависимости Φ_1 , а для остальных переменных при фиксированных первых двух – с помощью зависимости Φ_2 , то выдвигается гипотеза о том, что итоговая аналитическая зависимость для искомой функции может быть представлена в виде комбинации этих функций со сшивкой при фиксированных значениях.

Оба вышеизложенных подхода не являются строго обоснованными. Скорее, это алгоритмы выдвигения гипотез, нуждающихся в проверках. Однако эти методы позволяют получать реальные практические результаты.

Не менее эффективным является применение метода главных компонент (РСА). Суть метода состоит в переходе от исходной системы координат к новому ортогональному базису в рассматриваемом многомерном пространстве, оси которого ориентированы по направлениям максимальной дисперсии массива данных. Реализации метода главных компонент и алгоритмам его применения в различных областях посвящено большое количество литературы. Различные варианты реализации метода главных компонент и его обобщений для нелинейных случаев подробно представлены в работах [1, 2]. Геометрическая постановка задачи нахождения главных компонент формулируется согласно [1, 2] следующим образом. В многомерном пространстве ищется вектор направления, задающий прямую, вдоль которой дисперсия максимальна (или сумма квадратов расстояний от точек данных до прямой минимальна). Так определяется первая главная компонента. Далее рассчитывается множество векторов первых остатков, которое лежит в пространстве, ортогональном первой главной компоненте и имеющем размерность на единицу меньше исходной размерности. Для нового пространства, образованного этим множеством векторов, снова ищется направление с максимальной дисперсией. Так рассчитывается вторая главная компонента. Снова рассчитывается множество векторов вторых остатков и т.д.

Применение главных компонент дает нам возможность отобразить исследуемый многомерный массив на плоскость или в трехмерное пространство, образованное первыми тремя главными компонентами. В этом случае схема обработки, анализа и визуализации исходного многомерного объема данных будет выглядеть следующим образом.

- Для исходного объема вычисляются три первые главные компоненты Y_1, Y_2, Y_3 , где каждая главная компонента является линейной комбинацией исходных переменных $Y(x_1, \dots, x_n) = \sum B_i x_i$.

- Далее координаты исходных точек исследуемого объема выражаются в координатах главных компонент:

$$A_i(x_1, \dots, x_n) = A_i(Y_1(x_1, \dots, x_n), Y_2(x_1, \dots, x_n), Y_3(x_1, \dots, x_n)) -$$

Реализуется визуальное представление массива в двумерном виде $A_i(Y_1, Y_2)$ или в трехмерном $A_i(Y_1, Y_2, Y_3)$.

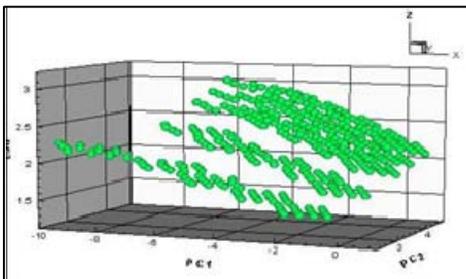
- Далее изучается полученное визуальное представление многомерного массива в главных компонентах и предпринимается попытка аппроксимации данных массива с помощью примитивных функций, имеющих аналитическое выражение. В простейшем случае применяется грубая

линейная аппроксимация с помощью плоскости вида $E_1Y_1 + E_2Y_2 + E_3Y_3 = C_y$. Так как плоскость при переходе от главных компонент к исходным переменным сохраняет свои свойства, с помощью обратного преобразования получаем $E'_1x_1 + E'_2x_2 + \dots + E'_nx_n = C_x$, которое уже можно рассматривать как искомую квазианалитическую зависимость $x_n = F(x_1, \dots, x_{n-1})$. В том случае, когда $A_i(Y_1, Y_2, Y_3)$ нельзя аппроксимировать одной плоскостью, можно использовать кусочно-линейный подход, применив несколько плоскостей. Следует также заметить, что применение квадратичных поверхностей может также оказаться весьма полезным, однако этот вопрос заслуживает отдельного рассмотрения.

3. ПРАКТИЧЕСКИЕ РЕЗУЛЬТАТЫ

Данный приближенный подход был применен к многомерному объему данных, полученному как результат решения задачи оптимизационного анализа нестационарного взаимодействия сверхзвукового потока вязкого сжимаемого теплопроводного газа со струйной преградой [5]. Преграда возникает благодаря недорасширенной спутной струе, истекающей из сопла, помещенного во внешний сверхзвуковой поток. При повышении скорости изменения степени нерасчетности струи возникает специфический режим течения, когда вещество струи распространяется вверх по потоку, по внешней стенке сопла. Скорость изменения степени нерасчетности струи рассматривается как управляющий параметр задачи оптимизационного анализа. В качестве определяющих параметров задачи рассматривались характерные числа Маха, Рейнольдса, Прандтля и Струхала. Эти четыре параметра варьировались в определенных диапазонах. Целью решения задачи было нахождение скорости изменения степени нерасчетности струи, при которой реализуется специфический режим течения во всех диапазонах изменения характерных чисел задачи.

В качестве результата решения задачи был получен 5-мерный объем данных, где в качестве переменных были 4 характерных числа задачи $M_\infty, \lg Re_\infty, Pr, Sh_\infty$ и искомая скорость V^* . Для полученного многомерного объема были определены три первые главные компоненты. После перехода к главным компонентам строилось визуальное представление точек массива в главных компонентах (см. рисунок). Полученное визуальное представление многомерного массива в главных компонентах позволило предположить, что точки массива могут быть грубо аппроксимированы параметрически заданной плоскостью.



Представление многомерного объема в пространстве трех первых главных компонент

После определения конкретного вида плоскости и ее коэффициентов было проведено обратное преобразование к исходным переменным и определение конкретного вида аппроксимирующей плоскости в исходных координатах. Это дало возможность получить искомую зависимость $V^* = F(M_\infty, \lg Re_\infty, Pr, Sh_\infty)$ в аналитическом виде. Полученные результаты представляют собой решение для класса задач, заданного в многомерном объеме определяющих параметров.

4. ЗАКЛЮЧЕНИЕ

Предложенный приближенный подход, предполагающий отображение многомерного объема в пространстве главных компонент и аппроксимацию данных в объеме с помощью плоскостей, позволяет проводить оценку скрытых взаимосвязей в многомерных объемах данных, получаемых в задачах вычислительной газовой динамики, как результаты решения задач параметрического поиска и оптимизационного анализа.

5. БЛАГОДАРНОСТИ

Данная работа выполнена при поддержке гранта Российского фонда фундаментальных исследований (проект N 14-01-00769a)

6. ССЫЛКИ

- [1] Gorban A., Kegl B., Wunsch D., Zinovyev A. (Eds.), Principal Manifolds for Data Visualisation and Dimension Reduction, LNCSE 58, Springer, Berlin – Heidelberg – New York, 2007.
- [2] Зиновьев А. Ю. Визуализация многомерных данных, Красноярск, Изд. КГТУ, 2000. – 180 с.
- [3] Бондарев А.Е., Галактионов В.А., Четкин В.М. Анализ развития концепций и методов визуального представления данных в задачах вычислительной физики // Вычислительная математика и математическая физика, 2011. – Т. 51, N 4. – С. 669 – 683.
- [4] Бондарев А.Е., Галактионов В.А. Анализ многомерных данных в задачах многопараметрической оптимизации с применением методов визуализации // Научная визуализация, 2012. – Т. 4, № 2. – С. 1 - 13,
- [5] Bondarev A.E, Galaktionov V.A. Parametric Optimizing Analysis of Unsteady Structures and Visualization of Multidimensional Data // International Journal of Modeling, Simulation and Scientific Computing, Vol. 4, suppl. issue 1, 2013, DOI: 10.1142/S1793962313410043 <http://www.worldscientific.com/doi/abs/10.1142/S1793962313410043>

Abstract

This article is devoted to the questions of multidimensional data analysis and visualization for CFD problems. Multidimensional data are considered as results of parametrical search and optimizing analysis. Rough approximate approach is proposed for data processing. The approach includes data visualization in principal components and data approximation by planes. An example of approach application to practical problem is considered.

Keywords: *multidimensional data visualization, PCA, CFD problems*

Об авторах

Бондарев Александр – к.ф.-м. н., старший научный сотрудник ИПМ им. М.В. Келдыша РАН.

E-mail: bond@keldysh.ru

Галактионов Владимир – д.ф.-м. н., профессор, заведующий отделом компьютерной графики ИПМ им. М.В. Келдыша РАН.

E-mail: vlgal@gin.keldysh.ru

Шапиро Лев З – к.т.н., старший научный сотрудник ИПМ им. М.В. Келдыша РАН.

E-mail: pls@gin.keldysh.ru

СРАВНЕНИЕ ФОРМЫ ИЗОБРАЖЕНИЙ НА ОСНОВЕ СПЕКТРА ШИРИНЫ ОБЪЕКТОВ

Е. Зак, Н. Ломов, Л. Местецкий

Факультет вычислительной математики и кибернетики

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

e_zak@mail.ru, nikita-lomov@mail.ru, mestlm@mail.ru

Аннотация

Работа посвящена разработке и сравнительному анализу методов вычисления так называемых морфологических спектров для решения задач классификации изображений гибких объектов. В качестве базового примера гибкого объекта, у которого форма может изменяться в широких пределах, рассматривается изображение человеческой ладони. Необходимость в сравнении ладоней по форме возникает при решении задач биометрической идентификации личности. Морфологические спектры, как показано в работе, позволяют выявить информативные признаки формы таких объектов и построить метрики для определения их сходства и различия.

Приводится сравнительное исследование трех типов спектров: морфологический спектр (pattern spectrum), предложенный в работе П. Марагоса, медиальный спектр и спектр дистанционной функции формы, предложенные авторами статьи.

Ключевые слова: морфологический спектр, медиальный спектр, спектр дистанционной ширины, анализ формы.

1. ВВЕДЕНИЕ

Ширина объектов на изображениях – важная характеристика их формы. Поэтому естественным является стремление использовать информацию о ширине объектов для распознавания формы в задачах компьютерного зрения. Поскольку ширина объекта различна для разных его частей, она не может быть описана какой-нибудь скалярной величиной, например, «средней» шириной. В качестве дескриптора ширины желательно построить такое описание, которое характеризовало бы весь спектр ее значений. В этой роли может выступить описание диапазона вариации значений ширины в различных частях объекта с указанием «частоты присутствия» каждого значения в данном объекте. Такое интегральное описание ширины удобно представить в виде функции, определяющей меру, например площадь, той части объекта, в которой ширина не превосходит заданного порога. При этом аргументом функции будет величина этого порога. Таким образом, дескриптор ширины приобретает вид, аналогичный функции распределения вероятности, и его вполне уместно назвать функцией распределения ширины объекта.

Для того чтобы полностью формально определить такой дескриптор, нужно сделать следующее:

- ввести понятие ширины объекта в каждой его точке;

- построить описание области «заданной ширины», представляющей собой множество точек объекта, в которых ширина не превосходит заданного порога;

- построить меру для подмножества заданной ширины, как функцию от величины порога.

В данной статье мы рассматриваем три возможных варианта построения функции распределения ширины, различающиеся способами формализации понятия ширины объекта в точке. Все три способа основываются на понятии пустого круга.

В качестве объектов изображения, для которых нужно строить дескриптор ширины, будем использовать понятие фигуры замкнутой ограниченной области на евклидовой плоскости. Пустым кругом называется круг, целиком принадлежащий фигуре. Максимальным пустым кругом или вписанным кругом будем называть пустой круг, который не содержится ни в одном другом пустом круге. Вписанный круг всегда имеет общие точки с границей фигуры. Радиус-вектор, соединяющий центр вписанного круга с точкой, в которой он касается границы, называется спицей.

Мы рассматриваем три варианта определения ширины фигуры в точке: морфологическая, медиальная и дистанционная ширина.

Морфологической шириной фигуры в точке называется радиус наибольшего пустого круга, покрывающего эту точку.

Дистанционной шириной фигуры в точке называется радиус наибольшего пустого круга с центром в этой точке.

Медиальной шириной фигуры в точке называется максимальное значение длины спицы, проходящей через эту точку.

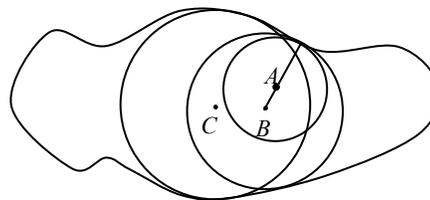


Рис. 1. Пример морфологической, медиальной и дистанционной ширины для точки A фигуры

В примере на рис. 1 для точки A морфологическая ширина равна радиусу круга C, дистанционная ширина – радиусу круга A, медиальная ширина – радиусу круга B.

Для этой же фигуры на рис. 4 серым цветом выделены области заданной ширины, соответствующие этим определениям. Области представляют собой подмножества точек фигуры, в которых значение ширины не превосходит радиуса изображенного на рисунке круга.

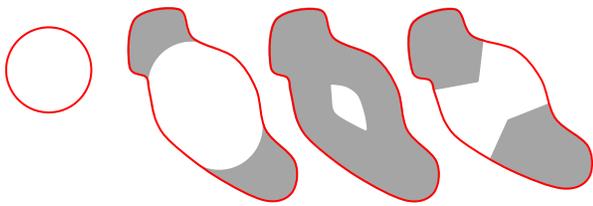


Рис. 2. Области заданной ширины (морфологической, дистанционной, медиальной), соответствующие размеру круга (слева)

2. ОПРЕДЕЛЕНИЯ И ПОДХОДЫ К ВЫЧИСЛЕНИЮ ШИРИНЫ И СПЕКТРА

Понятие морфологического спектра было введено Марагосом в работе [3]. Приведем определение спектра для дискретного случая, то есть когда мы работаем с объектом, заданным на растровой решетке.

Определение 1. Морфологическим спектром PS_X фигуры X относительно примитива (структурного элемента) B называется функция

$$PS_X(r_i, B) = - \frac{S(X \circ B(r_i)) - S(X \circ B(r_{i+1}))}{r_{i+1} - r_i}$$

где $(r_{i+1} - r_i)$ – шаг растровой решетки; $r_i \geq 0$, $X \square B(r)$ – операция морфологического открытия фигуры X с примитивом B ; $S(X)$ – площадь X .

Если в качестве примитива $B(r)$ взять диск радиуса r , то результатом операции открытия $X \square B(r)$ будет подмножество X , покрываемое множеством дисков радиуса r , которые целиком лежат в X , то есть подмножество равно объединению всех дисков, вписанных в X . Следовательно, подмножество остальных точек из X – это те точки, которые покрываются вписанными кругами меньшего радиуса. Таким образом, функция $F(r) = S(X \setminus X \square B(r))$ есть функция распределения морфологической ширины фигуры X в терминологии, введенной в предыдущем разделе. В определении 1 фигурирует разностная производная этой функции.

Другой подход к описанию ширины объекта основывается на понятии срединной оси, или скелета, которые вводятся с помощью непрерывной морфологии [7]. Скелетом фигуры называется множество точек-центров всех вписанных кругов фигуры. Скелет имеет вид плоского геометрического графа. Введем необходимые для понятия медиальной ширины определения.

Определение 2. Скелетом фигуры называется множество точек-центров всех вписанных кругов фигуры.

Определение 3. Радиальная функция определена в точках скелета и ставит в соответствие точке скелета радиус вписанного круга с центром в этой точке.

Определение 4. Спицей называется отрезок прямой, соединяющий точку скелета с ближайшей точкой границы фигуры.

Отметим некоторые важные свойства, связывающие точки фигуры с множеством спиц этой фигуры.

Утверждение 1. Через каждую точку фигуры проходит хотя бы одна спица. Следовательно, спицы покрывают всю фигуру.

Утверждение 2. Через каждую внутреннюю точку фигуры, не являющуюся точкой скелета, проходит только одна спица.

Теперь определим понятие медиальной ширины [8] фигуры в точке.

Определение 5. Медиальная ширина фигуры в точке равна длине минимальной спицы, инцидентной этой точке.

Из утверждения 1 следует, что медиальная ширина определена для всех точек фигуры. Все спицы, инцидентные одной точке скелета, имеют одинаковую длину, равную радиусу пустого круга с центром в этой точке. Поэтому для точек скелета медиальная ширина просто равна радиальной функции. Для каждой внутренней точки, не являющейся точкой скелета, существует, согласно утверждению 2, единственная инцидентная спица, поэтому медиальная ширина в такой точке определена однозначно.

Точки фигуры, имеющие несколько инцидентных спиц разной длины, могут находиться лишь на границе фигуры. Именно для таких случаев в определении 5 медиальная ширина определена через понятие минимальной спицы.

Введем обозначения:

- $\phi(g)$ – медиальная ширина в точке g ;
- $G_r = \{g \in G, \phi(g) \leq r\}$ – подмножество точек фигуры, в которых медиальная ширина не превосходит заданное значение r ;
- $F(r)$ – функция медиальной ширины фигуры, площадь множества точек G_r .

Тогда, под медиальным спектром будем понимать

$$f(r) = dF(r) / dr.$$

Еще одним способом описать фигуру с помощью дескрипторов, основанных на ширине фигуры, является вычисление дистанционной ширины с помощью карты расстояний (Distance transform map, далее по тексту DT) [1, 2]. Карта расстояний (Distance map) – это такое представление изображения фигуры, в котором каждой точке объекта ставится в соответствие минимальное расстояние до точек фона.

Определение 6. Функция построения карты расстояний DT является отображением фигуры X , при котором каждой точке фигуры x ставится в соответствие минимальное расстояние до точек фона, то есть:

$$DT(x) = \min_p (d(x, p) | x \in X, p \notin X).$$

Функция $DT(x)$ описывает дистанционную ширину в точке x .

Таким образом, функция распределения дистанционной ширины строится следующим образом:

$$F(r) = \sum_{x \in X} [DT(x) \leq r].$$

3. РЕАЛИЗАЦИЯ И ЭКСПЕРИМЕНТЫ

Для проверки применимости спектра для решения задач анализа формы объектов и классификации была собрана выборка из 160 фотографий ладоней 36 различных людей.

Выборка состоит из изображений с разрешением 640x480 px. Примеры некоторых объектов из выборки приведены на рис. 3. Как видно из примеров, положение ладони на изображении и другие ограничения не фиксированы.



Рис. 3. Элементы выборки

Все три способа вычисления функции распределения ширины были реализованы в соответствии с данными выше определениями. Сравнение этих методов выполнялось по критериям скорости вычислений и по качеству построенных на их основе алгоритмов классификации.

Скоростная характеристика алгоритма является одним из самых главных критериев, с точки зрения использования его в системах машинного зрения. Для сравнения скоростей вычисления функций распределения ширины были проведены эксперименты для 100 случайно выбранных изображений (табл. 1). Вычисление проводилось на процессоре Intel Core i5 2.67 GHz.

Эксперимент показал, что вычисление морфологической ширины существенно отстает от других видов спектров по времени расчета. Разница между временем вычисления медиальной ширины и дистанционной ширины в 4 раза связана с относительной сложностью операции построения карты глубины, которая сравнима по времени с алгоритмом обхода в глубину графа. Результаты показывают, что время построения функции распределения медиальной ширины позволяет использовать ее в анализе видео последовательностей, так как алгоритм работает достаточно быстро для видео с 24 кадрами в секунду.

Морфологическая ширина (сек.)	Медиальная ширина (сек.)	Дистанционная ширина (сек.)
2.548	0.025	0.093

Таблица 1. Усредненное по 100 вычислениям на CPU время вычисления ширины на изображениях выборки (640x480 px)

Для сравнения классификационной способности в экспериментах был использован метрический алгоритм kNN . Оценка качества классификации (Q) выполнена с помощью

скользящего контроля по схеме leave-one-out как отношение числа верно классифицированных объектов (N_{true}) к общему числу объектов (N):

$$Q = N_{true} / N.$$

Алгоритм k -ближайших соседей имеет 3 параметра:

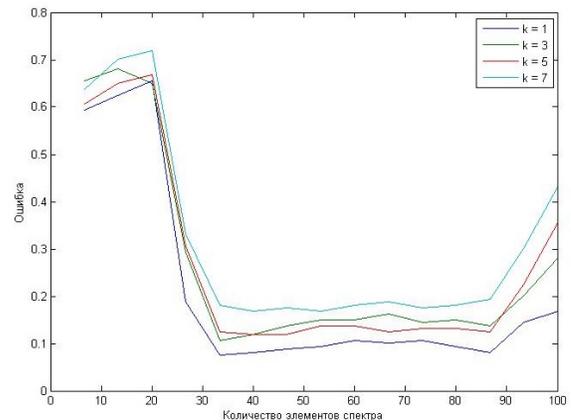
- метрика – $d(H_1, H_2)$;
- количество «соседей» – k ;
- набор элементов спектра – $I = i_{p1}, \dots, i_{pm}$.

Метрика сравнения является одним из основных параметров модели классификации. В исследовании были рассмотрены 4 метрики (табл. 2). На данной выборке лучшее качество показала так называемая транспортная метрика (EMD – метрика).

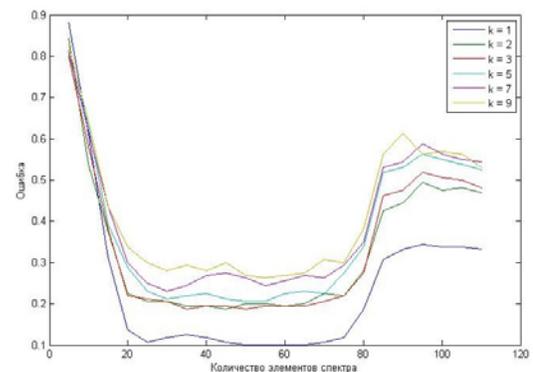
$d(H_1, H_2)$	Q
Simple	0.8313
χ^2	0.8663
Бхаттачария	0.7375
EMD	0.9012

Таблица 2. Сравнение метрик сходства морфологических спектров на выборке по качеству классификации

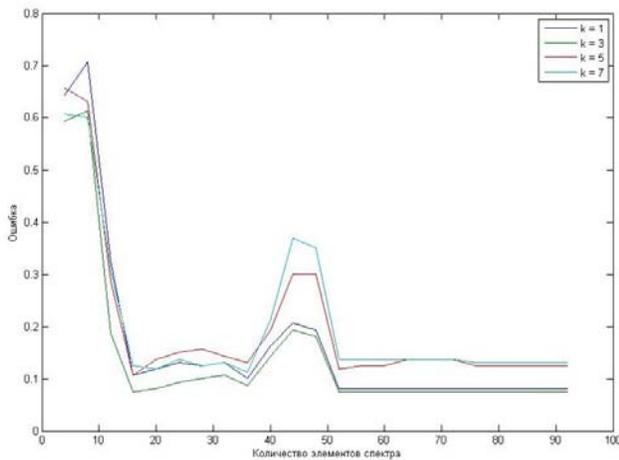
Для подбора оптимального количества соседей и количества первых элементов спектра, участвующих в классификации, была проведена серия экспериментов, в рамках которых для каждого типа спектров оценивалось качество классификации при различных значениях параметров. Результаты приведены на рис. 4.



(а)



(б)



(в)

Рис. 4. Значение ошибки классификации алгоритмом kNN в зависимости от выбора параметров: количество «соседей», количество первых элементов спектра, участвующих в рассмотрении алгоритма для: морфологической ширины (а); медиальной ширины (б); дистанционной ширины (в)

Из графиков на рис. 4 можно определить оптимальные значения параметров. Важно отметить, для всех трех методов эксперименты показали, что на имеющихся данных наиболее эффективно работает алгоритм, определяющий класс по одному ближайшему объекту обучающей выборки. Количество первых элементов спектра, используемых в классификации, для каждого из видов спектра разное, что объясняется их различиями при построении.

С учетом подобранных параметров были проведены эксперименты по вычислению качества идентификации с помощью каждого вида спектра. На рис. 5 представлены графики зависимости ошибки классификации с ростом объема выборки для моделей классификации на основе каждого из 3 спектров. Графики построены на основании проведения 100 экспериментов, в ходе каждого из которых исходная выборка случайным образом делилась на 40 блоков. После чего, в каждом из экспериментов рассматривались результаты классификации для моделей, обученных на 1, 2, 3 и т.д. блоках.

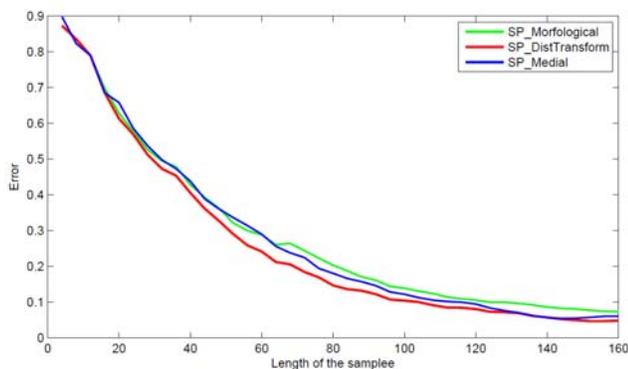


Рис. 5. Сравнение качества классификации в зависимости от длины выборки с использованием морфологического спектра (синий цвет), медиального спектра (красный цвет) и дистанционного спектра (зеленый цвет)

Итоговое значение качества классификации для моделей на основе морфологического спектра Марагоса, медиального спектра и спектра на основе карты расстояний составило **93.7 %, 94.5 %, 95.2 %**, соответственно.

4. ЗАКЛЮЧЕНИЕ

В данной работе рассмотрен один из подходов к решению задачи классификации объектов по их форме. Метод, основанный на построении спектров ширины, то есть дескрипторов ширины фигуры, не подразумевает существенных ограничений к объектам выборки и вычислительным ресурсам; при этом он обладает набором полезных свойств, например регулярность, инвариантность относительно положения объекта на изображении и т.д.

Приведены скоростные оценки спектров, показано, что вычисление морфологического спектра работает слишком медленно, и поэтому не может использоваться в прикладных системах реального времени, в отличие от медиального и дистанционного спектров. Показано также, что по критериям качества классификации все три спектра показывают качественно близкие результаты.

Эти результаты позволяют сделать следующие выводы: во-первых, спектры ширины являются хорошим инструментом, который применим как признаковое описание формы объекта. Это позволяет рассчитывать на его использование не только в задаче биометрической идентификации личности на основе формы ладони, но и в других задачах классификации изображений. Использование медиального спектра позволяет решать задачи анализа формы в режиме реального времени, так как время построения спектра составляет $\approx 1/40$ секунды, что пригодно для обработки видеопоследовательностей с высокой частотой кадров.

Спектр на основе дистанционной ширины также вычисляется за время, существенно меньшее, чем морфологический спектр, но уступает медиальному – примерно в 4 раза. При этом он обладает рядом полезных свойств: построение карты расстояний является чисто растровой операцией, эффективная реализация которой присутствует во многих современных прикладных системах обработки изображений, например в Matlab. Также экспериментально показано, что спектр на основе карты расстояний показывает относительно лучший результат классификации среди трёх рассматриваемых спектров на имеющейся выборке.

Вместе с тем, продемонстрированные результаты исследования (качество классификации 93 - 95%) не являются высокими по сравнению с существующими методами биометрической идентификации. Это говорит о том, что спектр целесообразно использовать в мультимодальных системах как один из нескольких признаков для описания объекта, отвечающий за свойства ширины объекта.

5. ССЫЛКИ

[1] Fabbri R., Costa L.Da F., Torelli J. C., Bruno O. M., 2D Euclidean Distance Transform Algorithms: A Comparative Survey // ACM Computing Surveys, Vol. 40, No. 1, Article 2, 2008.

[2] Meijster A., Roerdink J. B. T. M., Hesselink W. H., A General Algorithm for Computing Distance Transforms in Linear Time // *Mathematical Morphology and its Applications to Image and Signal Processing Computational Imaging and Vision* Volume 18, 2000.

[3] Petros Maragos. Pattern Spectrum and Multiscale shape representation // *IEEE transaction on pattern analysis and machine intelligence*, VOL II, № 7 – 1989.

[4] Ramirez-Cortes J.M., Gomez-Gil P., Sanchez-Perez G., Baez-Lopez D., A Feature Extraction Method Based on the Pattern Spectrum for Hand Shape Biometry // *Proceedings of the World Congress on Engineering and Computer Science, WCECS 2008, October 22 - 24, San Francisco, USA – 2008.*

[5] Serra J. *Image Analysis and Mathematical Morphology* // Academic Press, London, 1982.

[6] Визильтер Ю.В., Сидякин С.В. Робастное сравнение морфологических спектров двумерных фигур и изображений, семинар «Морфологический анализ данных» под руководством проф. Ю.П. Пытьева, 2012.

[7] Местецкий Л.М. Непрерывная морфология бинарных изображений: фигуры, скелеты, циркуляры. – М.: Физматлит, 2009.

[8] Местецкий Л.М., Чернышев В.Г. Медиальная ширина объектов – дескриптор формы изображений // *Сб. докл. всероссийской конференции «Математические методы распознавания образов» (ММРО-16)*, 2013.

[9] Сидякин С.В. Разработка алгоритмов построения морфологических спектров для анализа цифровых изображений и видеопоследовательностей // *Диссертация на соискание ученой степени кандидата технических наук.* – М., 2013.

ОБ АВТОРАХ

Местецкий Леонид – д.т.н., профессор ВМК
МГУ им. М.В. Ломоносова.
E-mail: mestlm@mail.ru

Зак Евгений – студент ВМК МГУ им. М.В. Ломоносова.
E-mail: e_zak@mail.ru

Ломов Никита – студент ВМК МГУ им. М.В. Ломоносова.
E-mail: nikita-lomov@mail.ru

Многоклассовая модель формы со скрытыми переменными

А.Н. Кириллов¹, М.И. Гавриков¹, Е.М. Лобачева¹, А.А. Осокин¹ и Д.П. Ветров^{1, 2}

akirillov@cs.msu.ru, gavrmike@gmail.com, lobacheva.tjulja@gmail.com, osokin@bayesgroup.ru, vetrovd@yandex.ru

¹Московский Государственный Университет им. М.В. Ломоносова

²Национальный исследовательский университет Высшая школа экономики

27 июля 2014 г.

Аннотация

Модель формы Больцмана (SBM, [1]) и ее многоклассовый вариант (MSBM, [2]) являются генеративными моделями, позволяющими задавать форму объекта. Многоклассовая модель обладает большей выразительной способностью, чем бинарная, но для ее обучения требуются данные с полной (многоклассовой) разметкой. В данной работе предложен новый алгоритм обучения MSBM, требующий только бинарные маски объектов и семена, задающие приближенное положение их частей. Семена при этом могут быть получены автоматически на основе использования детектора частей объекта. На этапе экспериментов мы показали, что MSBM, обученная нашим алгоритмом, превосходит по качеству SBM в задачах, где доступна только бинарная маска, и сопоставима по качеству с MSBM, обученной на данных с полной разметкой.

1 Введение

Модели формы объекта используются в различных задачах компьютерного зрения: сегментации [3, 4], восстановлении части изображения [5], детектировании объекта [6]. Учитывая знания о форме объекта в задаче сегментации можно значительно улучшить качество решения [2].

Существует множество способов задавать форму объекта [4, 7, 8], однако большинство из них либо недостаточно гибкие, либо требуют сложной дополнительной разметки обучающей выборки. SBM позволяет задавать форму объекта с помощью глубокой сети определенной структуры [1]. В [2] предложена многоклассовая модель, в которой класс имеет смысл некоторой части объекта. Многоклассовая модель обладает большей выразительной способностью по сравнению с бинарной, однако для ее обучения требуются данные с полной разметкой.

В данной работе мы предлагаем метод обучения MSBM, использующий лишь бинарные маски объектов и семена их частей (точки на изображении, в окрестностях которых лежат пиксели заданного класса). Для обучения модели с такими входными данными нами предложена EM-подобная процеду-

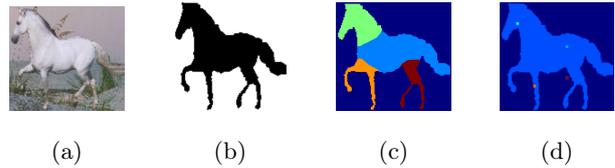


Рис. 1: (a) – изображение, (b) – бинарная разметка \mathbf{b} , (c) – многоклассовая разметка \mathbf{m} , (d) семена \mathbf{s} для 4 частей: головы, передних ноги, задних ноги, крупа.

ра. Семена частей могут размечаться как вручную, так и автоматически с помощью детектора частей объекта (part-based detector, [8]). Использование детектора частей объектов позволяет обучать MSBM на основе лишь исходных изображений и их бинарных разметок.

В разделе экспериментов мы показали, что MSBM, обученная нашим алгоритмом, превосходит SBM и сопоставима с MSBM, обученной стандартным способом. Также показано, что модель, обученная нашим алгоритмом, лучше MSBM, для которых многоклассовая разметка была получена из бинарной маски и семян на основе простых эвристических предположений.

2 Нотация

Пусть нам дано D центрированных и масштабированных изображений с объектами одного типа (рис. 1а, тип: лошади в профиль). За P обозначим число частей объекта, а N – число пикселей на изображении. Пусть $B = \{\mathbf{b}^d\}_{d=1, \dots, D}$ – множество бинарных разметок изображений (рис. 1b), где $\mathbf{b}^d \in \{0, 1\}^N$, а $M = \{\mathbf{m}^d\}_{d=1, \dots, D}$ – множество многоклассовых разметок (рис. 1c), где $\mathbf{m}^d \in \{0, \dots, P\}^N$, 0 соответствует фону. Также будем использовать бинарные переменные $m_{ip}^d \in \{0, 1\}$ для обозначения принадлежности пикселя i на изображении d классу p и векторные переменные $\mathbf{m}_p^d = \{m_{ip}^d\}_{i=1, \dots, N}$. Введем функцию $f_{coord}(i)$, ставящую в соответствие номеру пикселя его координаты на изображении и множество семян $S = \{\mathbf{s}^d\}_{d=1, \dots, D}$, где \mathbf{s}^d – это упорядоченный набор семян для изображения d , каждое семя задается своими координатами на изображении.

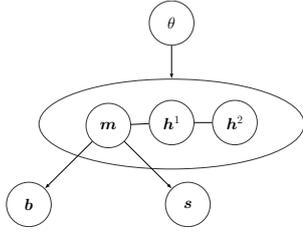


Рис. 2: Графическая модель $p(\mathbf{b}, \mathbf{s}, \mathbf{m}, \mathbf{H} | \boldsymbol{\theta})$.

3 Модели формы Больцмана

SBM — модель бинарной формы, представляющая собой трехслойную глубинную модель Больцмана (DBM, [9]) с дополнительными ограничениями. Наблюдаемый слой, соответствующий изображению, делится на 4 части с перекрытием и каждая часть соединяется только со своим подмножеством переменных на первом скрытом слое. Более того, веса для каждой из этих частей одинаковы. Такие ограничения позволяют существенно уменьшить число параметров модели, что позволяет избежать переобучения и ускорить процесс обучения. Будем считать, что эти ограничения неявно учтены в соответствующей матрице связей.

SBM с наблюдаемым слоем \mathbf{b} и скрытыми $\mathbf{H} = \{\mathbf{h}^1, \mathbf{h}^2\}$ задается с помощью распределения Гиббса:

$$p(\mathbf{b}, \mathbf{H} | \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \exp(-E(\mathbf{b}, \mathbf{H} | \boldsymbol{\theta})), \quad (1)$$

где $Z(\boldsymbol{\theta}) = \sum_{\mathbf{b}, \mathbf{H}} \exp(-E(\mathbf{b}, \mathbf{H} | \boldsymbol{\theta}))$ — нормировочная константа, $\boldsymbol{\theta} = \{\mathbf{a}, \mathbf{c}^1, \mathbf{c}^2, \mathbf{W}^1, \mathbf{W}^2\}$ — набор параметров модели и энергия:

$$E(\mathbf{b}, \mathbf{H} | \boldsymbol{\theta}) = \mathbf{b}^T \mathbf{W}^1 \mathbf{h}^1 + \mathbf{h}^{1T} \mathbf{W}^2 \mathbf{h}^2 + \mathbf{h}^{1T} \mathbf{c}^1 + \mathbf{b}^T \mathbf{a} + \mathbf{h}^{2T} \mathbf{c}^2. \quad (2)$$

MSBM является обобщением SBM на многоклассовый случай. Она задается аналогичным распределением со следующей энергией:

$$E(\mathbf{m}, \mathbf{H} | \boldsymbol{\theta}) = \sum_{p=1}^P \mathbf{m}_p^T \mathbf{W}_p^1 \mathbf{h}^1 + \mathbf{h}^{1T} \mathbf{W}^2 \mathbf{h}^2 + \mathbf{h}^{1T} \mathbf{c}^1 + \sum_{p=1}^P \mathbf{m}_p^T \mathbf{a}_p + \mathbf{h}^{2T} \mathbf{c}^2 \quad (3)$$

Описание классического алгоритма обучения моделей Больцмана приведено в [10].

4 Совместная модель

Введем совместное распределение на переменные $\mathbf{b}, \mathbf{m}, \mathbf{s}, \mathbf{H}$ предполагая, что \mathbf{b} и \mathbf{s} условно независимы при известном \mathbf{m} :

$$p(\mathbf{b}, \mathbf{s}, \mathbf{m}, \mathbf{H} | \boldsymbol{\theta}) = p(\mathbf{b} | \mathbf{m}) p(\mathbf{s} | \mathbf{m}) p(\mathbf{m}, \mathbf{H} | \boldsymbol{\theta}). \quad (4)$$

Соответствующая графическая модель представлена на рис. 2. Под распределением $p(\mathbf{m}, \mathbf{H} | \boldsymbol{\theta})$ здесь понимается распределение Гиббса для MSBM, $p(\mathbf{b} | \mathbf{m})$ задается как:

$$p(\mathbf{b} | \mathbf{m}) = \prod_{i=1}^N p(b_i | m_i) = \prod_{i=1}^N [b_i = [m_i \neq 0]], \quad (5)$$

а вид $p(\mathbf{s} | \mathbf{m})$ основан на предположении, что пиксель i «тянет на себя» семя m_i :

$$p(\mathbf{s} | \mathbf{m}) \propto \prod_{i: m_i \neq 0} \mathcal{N}(s_{m_i} | f_{coord}(i), \sigma^2) \quad (6)$$

где σ — внешний параметр модели.

5 EM-алгоритм

Для обучения параметров $\boldsymbol{\theta}$ будем с помощью EM-алгоритма решать следующую задачу максимизации:

$$\log P(B, S | \boldsymbol{\theta}) = \sum_{d=1}^D \log p(\mathbf{b}^d, \mathbf{s}^d | \boldsymbol{\theta}) \rightarrow \max_{\boldsymbol{\theta}}. \quad (7)$$

При этом будем считать переменные \mathbf{b}, \mathbf{s} наблюдаемыми, а \mathbf{m}, \mathbf{H} — скрытыми.

На E-шаге будем находить вариационную оценку апостериорного распределения на скрытые переменные в семействе полностью факторизованных распределений $q^d(\mathbf{m}, \mathbf{H}) = \prod_{i=1}^I q_i^d(m_i) \prod_{j=1}^J q_j^d(h_j^1) \prod_{k=1}^K q_k^d(h_k^2)$ минимизируя дивергенцию Кульбака-Лейблера:

$$\min_{q^d} \text{KL}(q^d(\mathbf{m}, \mathbf{H}) || p(\mathbf{m}, \mathbf{H} | \mathbf{b}^d, \mathbf{s}^d, \boldsymbol{\theta})). \quad (8)$$

Решение данной задачи сновано на [11, р. 466]. Оно представляет собой итеративную процедуру со следующими формулами пересчета:

$$q_i^d(m_i = p) \propto [b_i^d \cdot m_i \neq 0] \exp\left(-\frac{\|s_{m_i}^d - f_{coord}(i)\|_2^2}{2\sigma^2} + a_{i, m_i} + \sum_j W_{i, j, m_i}^1 q_j(h_j^1 = 1)\right) [b_i^d = 0] [m_i^d = 0] \quad (9)$$

$$q_j^d(h_j^1 = 1) = \sigma\left(c_j^1 + \sum_{i, p=1}^P q_i(m_i = p) W_{i, j, p}^1 + \sum_k W_{j, k}^2 q_k(h_k^2 = 1)\right), \quad (10)$$

$$q_k^d(h_k^2 = 1) = \sigma\left(c_k^2 + \sum_j q_j(h_j^1 = 1) W_{j, k}^2\right), \quad (11)$$

где $\sigma(x) = 1/(1 + e^{-x})$. Данная процедура также обобщается на случай, когда часть семян неизвестна.

На M-шаге обновляем параметры модели $\boldsymbol{\theta}$ решая следующую задачу:

$$\max_{\boldsymbol{\theta}} \sum_{d=1}^D \left[\sum_{\mathbf{m}, \mathbf{H}} q^d(\mathbf{m}, \mathbf{H}) \log p(\mathbf{b}^d, \mathbf{s}^d, \mathbf{m}, \mathbf{H} | \boldsymbol{\theta}) \right]$$

Учитывая полную факторизованность распределения $q(\mathbf{m}, \mathbf{H})$ получаем следующую задачу максимизации:

$$\max_{\boldsymbol{\theta}} \sum_{d=1}^D \left(\sum_{i=1}^I \sum_{p=1}^P a_{i, p} q_i(m_i^d = p) + \sum_{j=1}^J c_j^1 q_j(h_j^1 = 1) + \sum_{i=1}^I \sum_{j=1}^J \sum_{p=1}^P q_i(m_i^d = p) W_{i, j, p}^1 q_j(h_j^1 = 1) + \sum_{j=1}^J \sum_{k=1}^K q_j(h_j^1 = 1) W_{j, k}^2 q_k(h_k^2 = 1) + \sum_{k=1}^K c_k^2 q_k(h_k^2 = 1) - \log Z(\boldsymbol{\theta}) \right) \quad (12)$$

Данную максимизационную проблему можно решить используя стохастический градиентного метод.

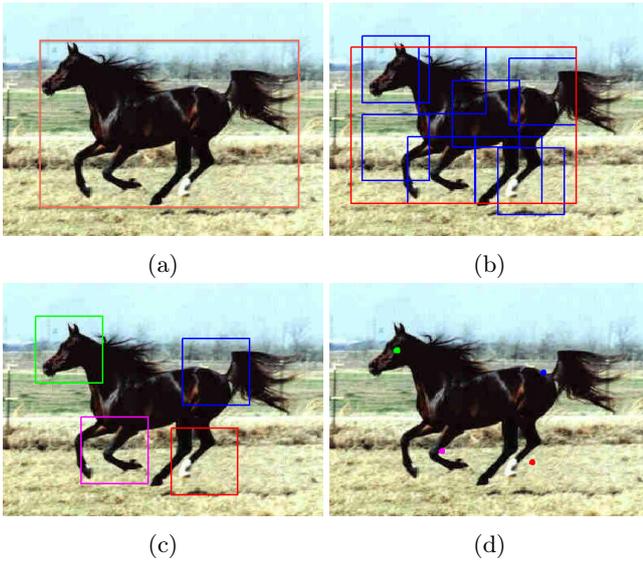


Рис. 3: (а) изображение с выделенным ограничивающим прямоугольником, (б) результат применения детектора, (в) выбранные части, (д) семена для выбранных частей.

6 Использование детектора частей

Семена на изображениях можно получать как посредством ручной разметки, так и автоматически с помощью детектора частей [8]. Для обучения такого детектора требуются изображения с размеченными ограничивающими прямоугольниками для объектов (рис. 3, а). Детектор сам выделяет части на изображении, имеющие сходную структуру на обучающей выборке. На этапе применения к тестовому изображению детектор выделяет ограничивающие прямоугольники для объекта и его частей (рис. 3, б). Далее выбираются только некоторые из частей (рис. 3, в), центры которых и принимаются за семена (рис. 3, г).

7 Эксперименты

Сравним MSBM, обученную предложенным алгоритмом, с SBM и с MSBM, обученными стандартным образом. Сравнение будем проводить на двух наборах данных: лошади ([3], 328 изображений) и мотоциклы ([12], 798 изображений). Предварительно обучим детектор и получим семена для этих выборок, масштабируем (для лошадей: 32 на 32 пикселя, для мотоциклов: 40 на 40) и центрируем бинарные маски объектов и разделим выборки на обучение и контроль (для лошадей: 278 и 50 изображений, для мотоциклов: 600 и 198). Для стандартного обучения MSBM выполним многоклассовую ручную разметку на наборе данных с лошадьми. Также получим многоклассовую разметку данных из бинарных масок и семян на основе двух эвристик: пиксель объекта принадлежит той части, к которой он ближе всего с точки зрения евклидова расстояния (Euc1), и с точки зрения длины кратчайшего пути, проходящего внут-

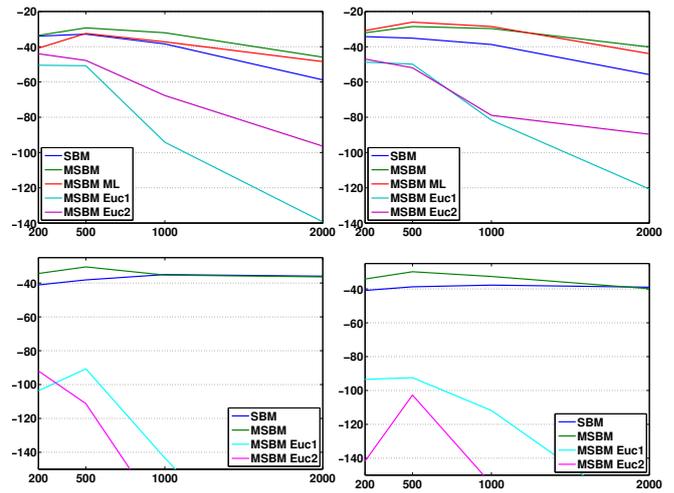


Рис. 4: Качество восстановления формы. Сверху – данные с лошадьми, снизу – с мотоциклами. Слева – $|h^2| = 100$, справа – $|h^2| = 200$. По оси абсцисс отложено $|h^1|$. Наша модель – зеленый график. Чем выше значение значения меры, тем лучше.

ри объекта (Euc2). Для сравнения моделей будем использовать две меры качества описанные ниже.

7.1 Восстановление формы

Первая мера оценивает качество восстановления части изображения по всему остальному изображению. Рассмотрим бинарную маску объекта с известными семенами. Разделим все пиксели этой маски на два класса случайных переменных: наблюдаемые b^o и скрытые b^u , семена при этом также делятся на два соответствующих класса. В случае с SBM можно оценить вероятность $p(b^u|b^o) \approx \sum_{r=1}^R p(b^u|\mathbf{H}_r)/R$, а в случае MSBM – вероятность $p(b^u|b^o, s^o) \approx \sum_{r=1}^R p(b^u|\mathbf{H}_r, s^o)/R$, где \mathbf{H}_r , $r = \overline{1, R}$ – результаты R итераций семплирования Гиббса из распределений $p(\mathbf{H}|b^o)$ и $p(\mathbf{H}|b^o, s^o)$ соответственно.

Будем вычислять данные оценки для всех изображений тестовой выборки. Деление пикселей на классы зададим следующим образом: разделим изображение на 9 равных частей сеткой 3 на 3 и будем поочередно брать в качестве скрытых переменных один из полученных сегментов. Итоговая мера качества представляет собой усредненный по разбиениям пикселей и тестовым изображениям логарифм оцененных вероятностей. Результаты приведены на рис. 4.

7.2 Порождение формы из семян

Вторая мера оценивает качество порождения формы из семян. Пусть даны семена s и соответствующие им пиксели b^s : $f_{coord}(b_p^s) = s_p \forall p = \overline{1, P}$ принадлежат объекту. Сгенерируем значения скрытых переменных из распределений $p(\mathbf{H}|b^s)$ для SBM и из $p(\mathbf{H}|b^s, s)$ для MSBM. Далее сгенерируем форму \mathbf{b} или \mathbf{m} соответственно. Повторим эти две операции 100 раз и получим итоговые формы объекта. Приме-

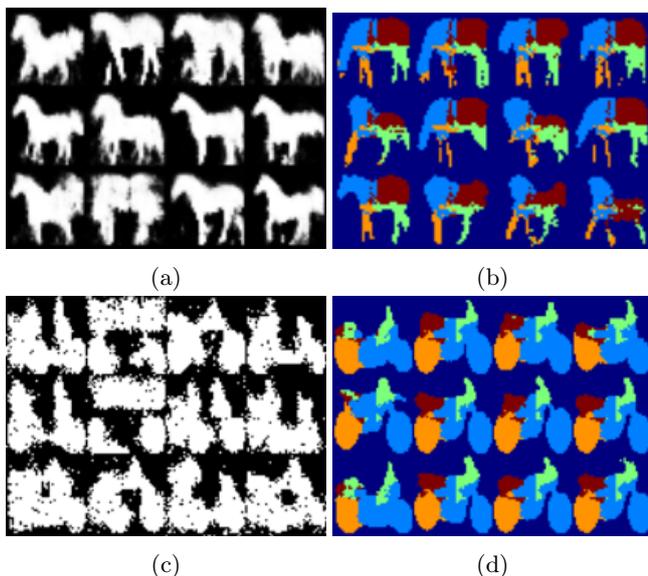


Рис. 5: Порождение формы из семян. Слева – SBM, справа – MSBM, обученная нашим алгоритмом.

ры получаемых форм представлены на рис. 5. Итоговая мера качества представляет собой усредненное по тестовой выборке расстояние Хэмминга между получившейся бинарной формой (в многоклассовом случае также бинаризуем результат) и исходной, для которой брались семена. Результаты приведены на рис. 6.

8 Заключение

В данной работе был предложен новый алгоритм обучения MSBM, требующий лишь бинарные маски объектов и семена для частей объектов. Также был рассмотрен метод автоматического получения семян с помощью детектора частей. На этапе экспериментов показано, что MSBM, обученная предлагаемым алгоритмом, превосходит по качеству SBM и сопоставима с MSBM, обученной стандартным способом. Также показано, что предлагаемый метод обучения дает лучшие результаты, чем эвристическое получение полной разметки из бинарной и семян и последующее классическое обучение MSBM.

Список литературы

- [1] Eslami, S.M.A., Heess, N., Williams, C.K.I., Winn, J.: The shape boltzmann machine: a strong model of object shape. In: International Journal of Computer Vision. (2013)
- [2] Eslami, S.M.A., Williams, C.: A generative model for parts-based object segmentation. In Bartlett, P., Pereira, F., Burges, C., Bottou, L., Weinberger, K., eds.: Advances in Neural Information Processing Systems 25. (2012) 100–107
- [3] Borenstein, E.: Combining top-down and bottom-up segmentation. In: In Proceedings IEEE workshop on Perceptual Organization in Computer Vision, CVPR. (2004) 46

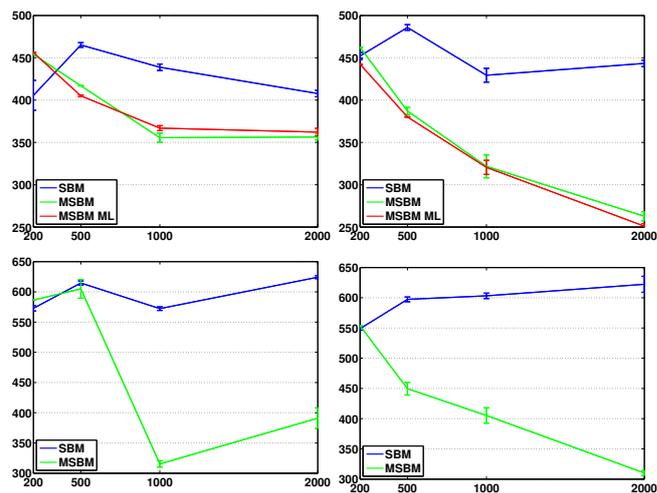


Рис. 6: Порождение формы из семян. Сверху – данные с лошадьми, снизу – с мотоциклами. Слева – $|h^2| = 100$, справа – $|h^2| = 200$. По оси абсцисс отложено $|h^1|$. Наша модель – зеленый график. Чем ниже значение меры, тем лучше.

- [4] Van Ginneken, B., Frangi, A.F., Staal, J.J., ter Haar Romeny, B.M., Viergever, M.A.: Active shape model segmentation with optimal features. IEEE Transactions on Medical Imaging **21**(8) (2002) 924–933
- [5] Chan, T.F., Shen, J.: Non-texture inpainting by curvature-driven diffusions (cdd). J. Visual Comm. Image Rep **12** (2001) 436–449
- [6] Ferrari, V., Jurie, F., Schmid, C.: From images to shape models for object detection. International Journal of Computer Vision **87**(3) (2010) 284–303
- [7] Yangel, B., Vetrov, D.: Image segmentation with a shape prior based on simplified skeleton. In: Energy Minimization Methods in Computer Vision and Pattern Recognition. Volume 6819 of Lecture Notes in Computer Science. (2011) 247–260
- [8] Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. IEEE Transactions on Pattern Analysis and Machine Intelligence **32**(9) (2010) 1627–1645
- [9] Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Computation **18**(7) (2006) 1527–1554
- [10] Salakhutdinov, R., Hinton, G.: An efficient learning procedure for deep boltzmann machines. Neural Computation **24**(8) (2012) 1967–2006
- [11] Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag New York, Inc., Secaucus, NJ, USA (2006)
- [12] Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. IEEE Transactions on Pattern Analysis and Machine Intelligence **28**(4) (2006) 594–611

Алгоритм поиска областей резкости на стереоизображениях

А. Афанасьева, А. Игнатенко

Факультет вычислительной математики и кибернетики

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

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

Аннотация

Ряд современных исследований в области человеко-машинного интерфейса посвящен улучшению восприятия оператором информации в стереопотоке. Для повышения удобства работы со стереоданными разрабатываются системы дополненной реальности, позволяющие получать дополнительную информацию о рассматриваемых объектах в реальном времени и в удобном для них виде. Подобные системы предполагают разработку человеко-машинного интерфейса как системы инструментов, работающих со стереоданными и экспортирующими в систему набор некоторой информации, полезной для восприятия. Для получения информации данные инструменты, как правило, нуждаются в специальных алгоритмах, заточенных под работу со стереоизображениями. В работе рассматривается один из возможных алгоритмов для инструмента поиска областей резкости в стереоизображении. Цель работы – разработка надежного алгоритма поиска областей резкости на стереоизображении, учитывающего особенности восприятия человеком стереоданных, обладающего высокой скоростью, потенциально пригодного для работы в режиме реального времени. Предложенный алгоритм использует ранее изученные методы для измерения резкости моноизображений, а также, в отличие от многих других работ на эту тему, учитывает особенности человеческого восприятия стереоданных для поиска областей резкости на обоих изображениях стереопары. Разработанный алгоритм может применяться к ряду стереоизображений, обладающих относительной детализированностью и определенной степенью сходства. Реализация алгоритма позволяет использовать его для разработки инструмента для работы в дополненной реальности и дальнейшей интеграции его в систему человеко-машинного интерфейса.

Ключевые слова: дополненная реальность, стереопоток, стереоизображение, области резкости, метрика резкости, стереосоответствие, диспаратитет.

1. ВВЕДЕНИЕ

Ввиду сложности восприятия человеческим глазом некоторых объектов для более детального изучения их подвергают съемке со стереокамер, поскольку стерео облегчает восприятие глубины данных объектов. В настоящее время активно развиваются технологии работы со стереоданными в дополненной реальности. Оператор, работающий со стерео, получает стереоизображение в совокупности с некоторой информацией, облегчающей восприятие объекта и навигацию по нему. Для работы со стереоданными в режиме дополненного стереопотока требуется особый интерфейс, улучшающий качество восприятия оператором стереоданных. Среди набора инструментов, образующих данный интерфейс, особое место

занимает инструмент выделения областей резкости на стереоизображении. Обладая данным инструментом, оператор может контролировать текущее состояние фокуса, рассматривая объект с разных сторон. Инструмент нахождения областей резкости облегчил бы работу оператора по взаимодействию с аппаратным обеспечением, поскольку результаты алгоритма нахождения областей резкости можно интерпретировать соответствующим образом для указаний по фокусировке (установке) для подвижки камер, с которых рассматривается объект.

2. ПОСТАНОВКА ЗАДАЧИ И ТРЕБОВАНИЯ К АЛГОРИТМУ

Алгоритм нахождения областей резкости предполагается использовать в системе человеко-машинного интерфейса для работы со стереопотоком в режиме реального времени. Поэтому важным требованием к алгоритму является высокая скорость работы. Первостепенным требованием также является достаточная точность результатов. Главным критерием качества будет считаться соответствие полученного результата восприятию человеческого глаза. На вход алгоритму подается стереопара, на выходе необходимо выделить пиксели на каждом изображении, которые оператор, работающий с объектом в стерео, воспримет в резкости.

3. ВОСПРИЯТИЕ РЕЗКОСТИ В СТЕРЕО

Главная проблема восприятия стереоизображения заключается в том, что, в отличие от моноизображения, человек воспринимает объект сразу с двух изображений. Предположим, объект визуально расположен в плоскости экрана. При рассмотрении схемы восприятия объекта со стереомонитора можно убедиться в том, что наиболее комфортно для глаза будут восприниматься пиксели изображения с относительно небольшим диспаратитетом.

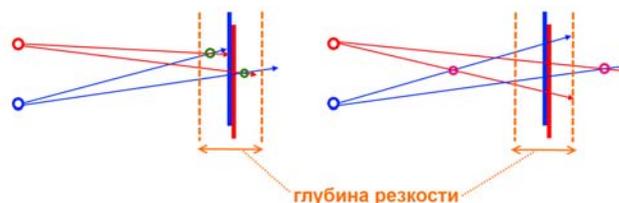


Рис. 1. Слева – допустимые для резкости соответствия, справа – соответствия, некомфортные для глаза (красным и синим – показаны левое и правое изображения)

В дальнейшем будем отталкиваться от того предположения, что точка объект воспринимается в стерео наилучшим образом, если выполняются два условия:

- 1) соответствующий пиксель находится в резкости на каждом изображении стереопары;
- 2) соответствующий пиксель обладает относительно небольшим диспаратитом, приемлимым для восприятия человеческим глазом.

Только в совокупности оба этих условия обеспечат резкость области и комфортное ее восприятие на стереоизображении. Алгоритм, реализующий нахождение областей резкости на изображении, обязан выдавать области, пиксели которых удовлетворяют обоим поставленным условиям.

4. ОБЗОР СУЩЕСТВУЮЩИХ МЕТОДОВ

Для поиска диспаратита пикселей изображения требуется решать задачу поиска стереосоответствий. Методы поиска стереосоответствий условно подразделяются на две категории: *локальные*, рассматривающие некоторую окрестность пикселя, и *глобальные*. К локальным методам относятся корреляционные методы, при которых соответствия ищутся при помощи корреляционных окон, а также методы, основанные на использовании локальных особенностей изображения (так называемых ключевых точек). В статье [5] представлен метод классической корреляции, усовершенствованный для качественной работы с границами внутри изображения. Авторы [12] используют так называемые двунаправленные стереосопоставления для более точного результата. Итеративные методы вычисления диспаратита на основе использования корреляционного окна представлены в статьях [9] и [15]. Авторы [16] используют для нахождения соответствий на изображениях ключевые особенности изображения, инвариантные к аффинным преобразованиям.

Глобальные методы предполагают расчет стереосоответствия для всех пикселей изображения с последующей постобработкой полученной карты диспаратитом с целью уменьшить выбросы диспаратита а также получить более качественные стереосоответствия на границах объектов. Примеры таких методов можно обнаружить, например, в работах [17] и [6]. Некоторые алгоритмы стереосопоставления основаны на использовании алгоритма распространения доверия [18], [14], [10], а также на сегментации изображения для улучшения результатов классической корреляции [7], [10]. К сожалению, применять подобные алгоритмы к решению поставленной задачи неприемлемо ввиду их низкой скорости работы.

Помимо нахождения диспаратита в точках изображения, необходимо также решать задачу поиска областей резкости на отдельных изображениях стереопары. В настоящий момент существует целый ряд методов поиска резкости на моноизображении. Возможно несколько подходов к измерению резкости изображения. Простейший подход основан на использовании общепринятых метрик резкости: оператор Лапласиана [8], [11], [13], модифицированный Лапласиан [11], [8], Тененград [8], абсолютное значение градиента [13], [4]. Данные метрики основаны на измерении отклика границ в каждом пикселе изображения. Некоторые методы используют информацию о границах на изображении, измеряя их толщину и основываясь на ней в выводах о резкости соответствующей области изображения (данные алгоритмы представлены в статьях [17], [18]). Отдельного

рассмотрения заслуживает группа методов, основанных на частотных разложениях изображения и его частей. В работе [1] для измерения резкости изображение разбивается на блоки, и для каждого блока, содержащего количество граничных пикселей выше заданного порога, происходит Двумерное дискретное преобразование Фурье. Более резкие области изображения характеризуются большими коэффициентами при высоких частотах. Основываясь на некоторой метрике, отдающей предпочтение высоким частотам, можно сделать выводы о резкости изображения. Подобный подход ввиду своей простоты и надежности был применен также и в данной работе.

5. ОПИСАНИЕ РЕАЛИЗОВАННОГО АЛГОРИТМА

Разработанный алгоритм состоит из двух этапов, которые будут рассмотрены последовательно:

1. Построение маски резкости для левого изображения стереопары.
2. Поиск областей с низким диспаратитом среди областей резкости, найденных на первом этапе.

5.1. Построение маски резкости

На данном этапе всё изображение разбивается на блоки фиксированного размера. Для каждого блока применяется Быстрое двумерное дискретное преобразование Фурье. Затем из спектра по некоторому порогу также выбрасываются все низкие частоты. Для полученного спектра вычисляется сумма коэффициентов при оставшихся частотах.



Рис. 2. Схема работы с блоком изображения (коэффициенты Фурье-преобразования)

В том случае, если рассчитанное значение полученной величины выше некоторого заданного порога, блок причисляется к фокусному, в противном случае пиксели блока исключаются из дальнейшего рассмотрения. С целью убрать из рассмотрения частоты с низкими значениями коэффициентов также возможна пороговая бинаризация полученного спектра. Таким образом, характеристика блока, используемая для вычисления резкости блока по Фурье-коэффициентам, может быть основана также и на суммировании коэффициентов частот в разложении выше некоторого заданного порога.

Размер области должен быть достаточным для того, чтобы высокие частоты отобразились в разложении именно как высокие. Из этого соображения (а также для уменьшения времени работы) в текущей реализации выбран достаточно большой размер блока.

Описанная процедура используется в качестве предобработки исходных изображений стереопары для отсеивания из рассмотрения размытых областей. Отбор резких блоков позволяет достаточно быстро убрать из рассмотрения области изображений, некомфортные для восприятия оператором.

5.2. Поиск областей с низким диспаратетом

5.2.1. Базовый алгоритм поиска областей с низким диспаратетом

с низким диспаратетом

Для нахождения области изображения с низким диспаратетом расчет диспаратетов пикселей изображения осуществляется на некоторой сетке. Среди точек, для которых был произведен расчет, ищутся точки с диспаратетом по модулю ниже некоторого порога. Среди них выделяются граничные точки, и рассчитывается граница области резкости на левом изображении. Область резкости на правом изображении будет, соответственно, состоять из точек, являющихся стереосоответствиями к точкам на левом изображении, принадлежащем его области резкости.

Для расчета диспаратета в заданной точке применяется классический метод поиска соответствия на основе корреляционного окна с L2-метрикой.

5.2.2. Ускоренный алгоритм поиска областей с низким диспаратетом

Для получения более точного результата, т.е. для получения более четких контуров областей резкости, требуется увеличивать размер сетки, что приводит к большому числу вычислений и, соответственно, большему времени работы алгоритма. С целью ускорения расчет производился итеративно на сетках разного размера. Первоначально расчет диспаратета проводится по точкам на крупной сетке (с достаточно большим – 50 - 100 пикселей, в зависимости от размера изображения, шагом). На следующей итерации шаг сетки уменьшался в два раза и производился расчет диспаратета в точках, соседствующих тем, которые обнаружили небольшой (меньше заданного порога) диспаратет на предыдущем шаге. Подобный подход позволил уменьшить количество вычислений. Подобную оптимизацию можно применять лишь в том случае, если надежность вычисления диспаратета достаточно высока. В случае если при расчете диспаратета на грубой сетке будет допущена существенная ошибка, данная ошибка повлечет за собой потерю из рассмотрения области на изображении, или же наоборот, будет рассмотрен ряд лишних точек сетки на следующей итерации.



Рис. 3. Схема работы алгоритма

6. РЕЗУЛЬТАТЫ

Ввиду того, что алгоритм предназначен для работы со стереоизображениями, часть которых находится в резкости, результаты работы целесообразно демонстрировать на стереоизображениях, обладающих низкой диспаратетностью. Пример результата приведен на рисунке ниже. Зеленым выделены точки, ограничивающие область резкости на стереоизображении, синим – непосредственно точки, находящиеся в резкости.



Рис. 4. Результат работы базового алгоритма (шаг сетки 21 пиксель, допустимый диспаратет 7, время работы ~350 мс)

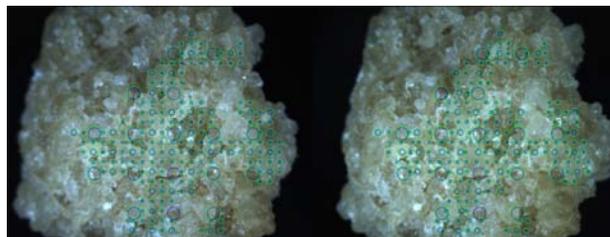


Рис. 5. Результат работы ускоренного варианта алгоритма для 3-х итераций (начальный шаг сетки 80, допустимый диспаратет 7, время работы ~340 мс, время работы базового алгоритма с теми же параметрами – 660 мс)

7. ВЕРИФИКАЦИЯ

В процессе тестирования алгоритма на различных стереоизображениях был установлена приблизительная величина диспаратета, при которой человеческий глаз воспринимает стереоизображение (или его область) в резкости. Для изображений разрешения порядка 2500x800 допустимое отклонение диспаратета от нулевого для рассматриваемых изображений составило около 15 – 20. На основе этого вывода была проведена верификация алгоритма. Вначале некоторый набор стереоизображений был размечен вручную (разметка осуществлялась оператором на левом изображении). Затем разработанный алгоритм поиска областей резкости запускался для нескольких значений параметра, задающего максимальный диспаратет.

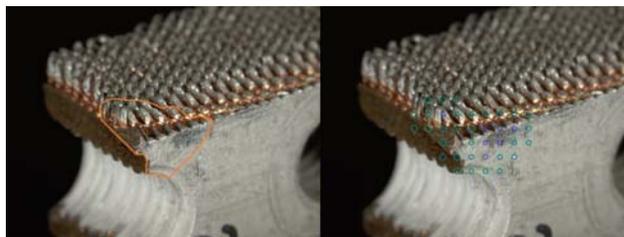


Рис. 6. Ручная разметка резкой области на изображении стереопары (слева) и результат работы алгоритма (справа)

8. ВЫВОДЫ

В процессе тестирования алгоритма на различных стереоизображениях были получены устойчивые результаты с хорошо структурированными, равномерно освещенными объектами, обладающими небольшой диспаратностью. Одним из важных требований, определяющих стабильность работы алгоритма, является сходство изображений, которое определяет применимость классического метода корреляционного окна для поиска стереосоответствия.

9. ДАЛЬНЕЙШИЕ ПЛАНЫ

Алгоритм выдает неточные результаты в том случае, если левое и правое изображения стереопары обладают большой диспаратностью. Однако увеличение максимального разброса диспаратности может привести к нестабильному результату на стереоизображениях с низкой диспаратностью, поэтому для улучшения качества работы алгоритма на подобных данных потребуются, вероятно, применять другие подходы. В дальнейшей планируется устранить указанные недостатки при помощи использования алгоритма нарастающего окна. Алгоритм также дает крайне нестабильный результат, если левое и правое изображения стереопары по-разному освещены, а также в том случае, если они обладают небольшим количеством деталей. Для подобного рода изображений планируется разработка отдельных алгоритмов поиска соответствий, основанных на учете локальных особенностей изображения. Результирующая маска резкости часто получается достаточно грубой, из-за чего в область резкости могут попасть размытые пиксели с низким диспаратетом. Для устранения этого недостатка планируется улучшение поиска областей резкости на моноизображениях (на первом этапе алгоритма).

10. ССЫЛКИ

- [1] Ferzli R., Karam L. J., Caviedes J. A robust image sharpness metric based on kurtosis measurement of wavelet coefficients //Proc. of Int. Workshop on Video Processing and Quality Metrics for Consumer Electronics. – 2005.
- [2] Ferzli R., Karam L.J. A no-reference objective image sharpness metric based on just-noticeable blur and probability summation //Image Processing, 2007. ICIP 2007. IEEE International Conference on. – IEEE, 2007. – Т. 3. – P. III-445-III-448.
- [3] Ferzli R., Karam L.J. A no-reference objective image sharpness metric based on the notion of just noticeable blur (JNB) //Image Processing, IEEE Transactions on. – 2009. – Т. 18. – №. 4. – P. 717 - 728.
- [4] Groen F. C. A., Young I. T., Ligthart G. A comparison of different focus functions for use in autofocus algorithms //Cytometry. – 1985. – Т. 6. – №. 2. – P. 81 - 91.
- [5] Hirschmüller H., Innocent P.R., Garibaldi J. Real-time correlation-based stereo vision with reduced border errors //International Journal of Computer Vision. – 2002. – Т. 47. – № 1 - 3. – P. 229 - 246.
- [6] Hirschmuller H. Accurate and efficient stereo processing by semi-global matching and mutual information //Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. – IEEE, 2005. – Т. 2. – P. 807 - 814.

- [7] Hong L., Chen G. Segment-based stereo matching using graph cuts //Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on. – IEEE, 2004. – Т. 1. – P. I-74-I-81 Vol. 1.
- [8] Huang W., Jing Z. Evaluation of focus measures in multi-focus image fusion //Pattern Recognition Letters. – 2007. – Т. 28. – №. 4. – P. 493 - 500.
- [9] Kanade T., Okutomi M. A stereo matching algorithm with an adaptive window: Theory and experiment //Pattern Analysis and Machine Intelligence, IEEE Transactions on. – 1994. – Т. 16. – №. 9. – P. 920 - 932.
- [10] Klaus A., Sormann M., Karner K. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure //Pattern Recognition, 2006. ICPR 2006. 18th International Conference on. – IEEE, 2006. – Т. 3. – P. 15 - 18.
- [11] Nayar S. K., Nakagawa Y. Shape from focus //Pattern analysis and machine intelligence, IEEE Transactions on. – 1994. – Т. 16. – №. 8. – P. 824 - 831.
- [12] Stefano L. D., Marchionni M., Mattoccia S. A fast area-based stereo matching algorithm //Image and vision computing. – 2004. – Т. 22. – №. 12. – P. 983 - 1005.
- [13] Subbarao M., Choi T. S., Nikzad A. Focusing techniques //Optical Engineering. – 1993. – Т. 32. – №. 11. – P. 2824 - 2836.
- [14] Sun J., Zheng N. N., Shum H. Y. Stereo matching using belief propagation //Pattern Analysis and Machine Intelligence, IEEE Transactions on. – 2003. – Т. 25. – №. 7. – P. 787 - 800.
- [15] Sun J. et al. Symmetric stereo matching for occlusion handling //Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. – IEEE, 2005. – Т. 2. – P. 399 - 406.
- [16] Tuytelaars T., Van Gool L. J. Wide Baseline Stereo Matching based on Local, Affinely Invariant Regions //BMVC. – 2000. – Т. 412.
- [17] Wang Z.F., Zheng Z.G. A region based stereo matching algorithm using cooperative optimization //Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. – IEEE, 2008. – P. 1 - 8.
- [18] Yang Q. et al. Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling //Pattern Analysis and Machine Intelligence, IEEE Transactions on. – 2009. – Т. 31. – № 3. – P. 492 - 504.

Система поиска минимального пути в среде с полигональными препятствиями

К. Заева, А. Семенов

Факультет прикладной математики и кибернетики

Тверской государственный университет, Тверь, Россия

kapitanka.ki@gmail.com, semenov@tversu.ru

Аннотация

В статье описан подход нахождения кратчайшего пути между двумя точками в двумерной среде с полигональными препятствиями, использующий непрерывное представление данных. Рассматривается задача векторизации и аппроксимации с контролируемой точностью бинарного черно-белого изображения для получения полигональных областей.

Ключевые слова: поиск минимального пути, среда с препятствиями, векторизация изображения, граф видимости.

1. ВВЕДЕНИЕ

Реализована система, позволяющая по карте местности, которая подается пользователем на вход, получить визуальное представление кратчайшего пути между любыми двумя заданными пунктами (точками). Карта местности представляет собой двумерную плоскость, а все препятствия на местности, сквозь которые невозможно пройти, аналогично, заданы двумерными непересекающимися фигурами (многоугольниками). Начальный и конечный пункты пути могут быть заданы любыми точками на плоскости, которые не являются частью препятствия. Заметим, что искомый путь может проходить вдоль границы объектов, но не может их пересекать. Иллюстрацию примера работы системы можно увидеть на изображении ниже (рис. 1).

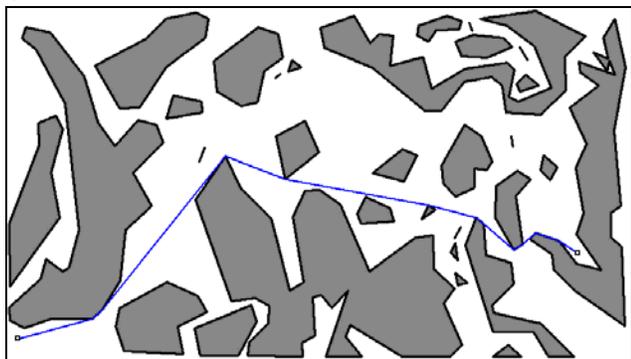


Рис. 1. Система поиска минимального пути в среде с полигональными препятствиями

Несмотря на наличие различных уже существующих способов решения задачи поиска минимального пути в двумерной среде с препятствиями, данная задача не теряет своей актуальности в связи с постоянным развитием приложений, которые требуют более быстрых и точных решений. Большинство уже существующих методов поиска пути предполагают, что пространство разбито на квадратные или шестиугольные ячейки. Для решения задачи непрерывное

пространство сводится к нескольким дискретным вариантам, к которым уже применяются различные алгоритмы поиска маршрута [3]. Однако дискретное решение имеет множество недостатков, основным из которых является возникновение ошибок накопления в ходе вычислений. Поэтому для получения более точных результатов мы используем непрерывный аналог решения задачи.

На вход системе подается бинарное черно-белое изображение – «карта» с препятствиями, где в качестве препятствий выступают простые полигоны без самопересечений. Исходное растровое изображение карты преобразовывается к непрерывному представлению с заданной точностью, после чего пользователь может выбирать начальную и конечную точки, получая кратчайший путь между ними. Минимальный путь вычисляется с помощью построения графа видимости полигональной области и нахождения маршрута на этом графе одним из классических алгоритмов [3].

2. ВЕКТОРИЗАЦИЯ ИЗОБРАЖЕНИЙ

В качестве среды с препятствиями может использоваться любое бинарное растровое изображение, которое система получает на вход. Для получения набора полигонов нам необходимо преобразовать контуры изображений из дискретного (растрового) представления в непрерывное (векторное) представление.

На вход задачи подается бинарное изображение, в котором присутствуют только два цвета: 1 – белый, 0 – черный. На выходе имеем последовательность граничных точек.

Вычисление границы растрового бинарного образа осуществляется через направленный перебор точек с целочисленными координатами вдоль границы объекта. Граничной точкой на растре называется точка, имеющая смежную точку другого цвета. Граничной парой (мостом) будем называть пару соседних точек на изображении, имеющих переход цветов. Обход границы осуществляется против часовой стрелки (рис. 2).

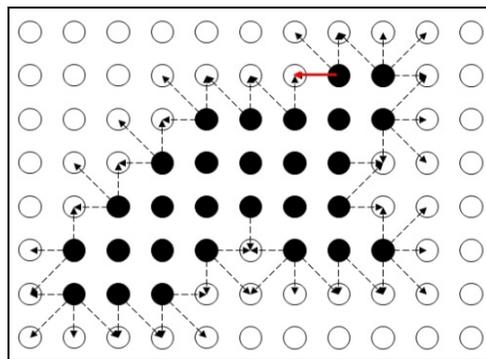


Рис. 2. Начало алгоритма обнаружения границ объекта

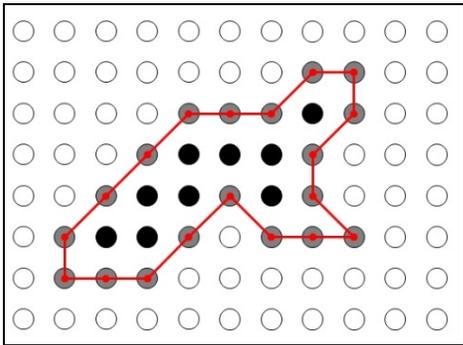


Рис. 3. Результат обнаружения границ объекта

Основные шаги выделения границ объектов [4]:

1. Проследивание начинается с обнаружения начальной граничной пары.

2. На каждом шаге один конец моста переносится в новую (пробную) точку. Основное правило перемещения моста:

- если пробная точка оказалась черной, то в нее переносится левый конец моста;
- если пробная точка оказалась белой, то в нее переносится правый конец моста.

3. Определение координат новой пробной точки:

Пусть левый (черный) конец моста имеет координаты $L = (Lx, Ly)$, правый (белый) конец моста – координаты $R = (Rx, Ry)$, пробная точка – координаты $T = (Tx, Ty)$. В зависимости от положения моста координаты точки T вычисляются следующим образом:

- вертикальное или горизонтальное положение моста

$$T = (Rx - (Ry - Ly), Ry + (Rx - Lx));$$

- диагональное положение моста

$$T = 1/2(Lx + Rx - Ry + Lx, Ly + Ry + Rx - Lx);$$

4. В граничную последовательность будет добавляться новая точка всякий раз, когда происходит перемещение левого (черного) конца моста

5. Условие завершения обхода границы – мост занял свое изначальное положение, с которого начиналось проследивание.

Алгоритм оконтуривания применяется последовательно ко всем объектам – препятствиям входного изображения. По завершению алгоритма выделения границ одного объекта все точки, участвующие в процессе, помечаются как уже посещенные, после чего мы переходим к «поиску» следующего объекта, если такой существует.



Рис. 4. Исходное входное изображение

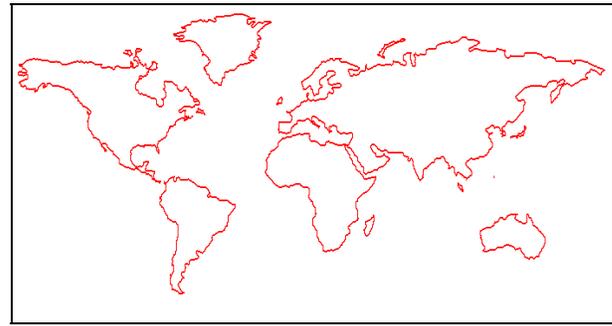


Рис. 5. Получение контуров объектов решением задачи векторизации

3. АППРОКСИМАЦИЯ ГРАНИЦ ОБЪЕКТОВ

Результатом выполнения алгоритма векторизации бинарного изображения является набор граничных точек. Однако если на вход мы подадим изображение не маленького формата, то этот набор будет иметь чрезвычайно большое количество объектов, так как рассматривается каждый пиксель изображения. Применять последующие действия к такому списку точек крайне нерационально, поэтому для дальнейших действий необходимо решить задачу аппроксимации набора граничных точек объекта.

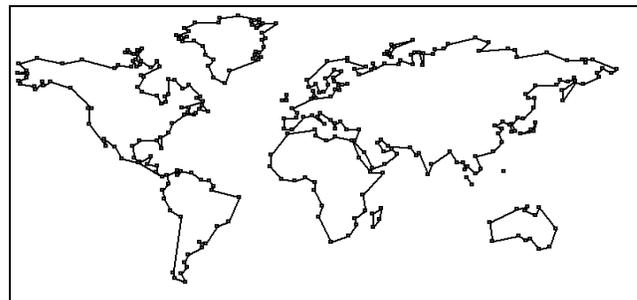


Рис. 6. Аппроксимация границ меньшим количеством точек (с точностью 3 пикселя)

На вход алгоритма мы получаем пронумерованную последовательность точек. Для решения задачи будем использовать простой метод кусочно-линейной аппроксимации, названный итеративным подбором концевых точек [5]. Идея такой кусочно-линейной аппроксимации состоит в следующем. Пусть нам дана последовательность точек $P_i, i = 0, \dots, N - 1$, являющихся граничными точками объекта, и определена точность аппроксимации ϵ .

- Решаем задачу кусочно-линейной аппроксимации отрезком прямой множества точек $P_i, i = 0, \dots, N - 1$.

- Определяем точку из множества P_i , максимально удаленную от полученного отрезка. В качестве расстояния между отрезком прямой и точкой P_i следует брать длину перпендикуляра, опущенного из точки P_i на отрезок.

- Если максимальное расстояние больше заданной наперед точности аппроксимации ϵ , то исходное множество разбивается на два подмножества, точкой максимально удаленной от построенного отрезка прямой, и задача аппроксимации рекурсивно решается отдельно для каждого из полученных подмножеств.

Ниже приведен пример подобной аппроксимации, где показаны четыре начальных итерационных шага для множества, состоящего из 19 точек (рис. 7).

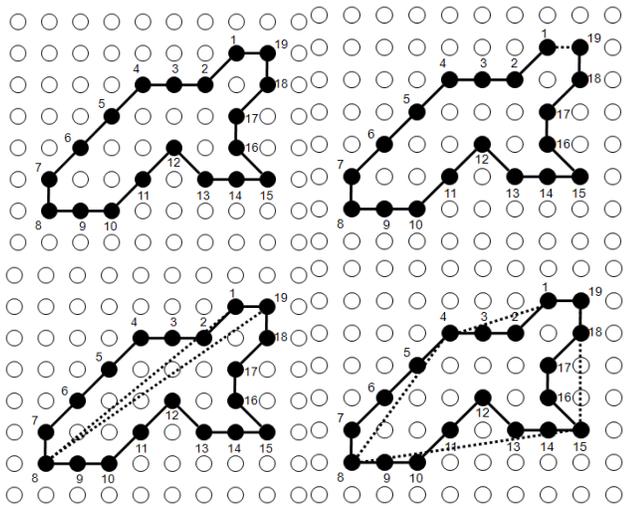


Рис. 7. Итерационный алгоритм аппроксимации

На выходе алгоритма мы получаем набор отрезков, которыми аппроксимируется заданная фигура. Начало и конец этих отрезков и составляют искомое нами множество граничных точек, по набору которых в дальнейшем будут строиться полигональные препятствия. Основное преимущество этого алгоритма состоит в его простоте.

4. ПОСТРОЕНИЕ ГРАФА ВИДИМОСТИ

Будем называть две точки взаимовидимыми, если отрезок, их соединяющий, не содержит в себе внутренних точек препятствий-многоугольников [1]. Пример взаимовидимых и не являющихся таковыми вершин многоугольника можно увидеть на иллюстрации ниже (рис. 8).

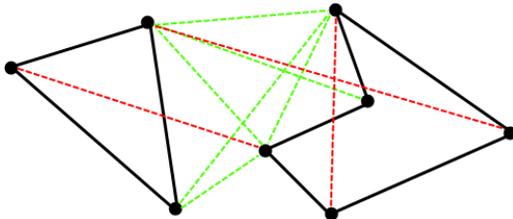


Рис. 8. Взаимовидимые (соединены зеленым цветом) и не взаимовидимые (соединены красным цветом) вершины многоугольника

Заметим, что соседние вершины всегда являются взаимовидимыми (на рис. 8 отмечены только видимые не соседние вершины).

Вершинами графа видимости являются вершины препятствий-многоугольников. Рёбрами соединены только взаимовидимые вершины графа. Кратчайший путь на полученном графе видимости и будет являться искомым минимальным маршрутом между двумя точками [2].

Таким образом, для построения графа видимости полигональной области необходимо найти все пары взаимовидимых вершин из множества, состоящего из вершин всех полигональных препятствий на плоскости. То есть для каждой пары вершин мы должны проверить, пересекает ли соединяющих их отрезок какую-либо сторону из всех существующих многоугольников. Однако такой подход

выполняется большим количеством итераций и не является эффективным. Для уменьшения времени работы, затрачиваемого на построения графа видимости, можно внести некоторые дополнительные проверки. Как уже было упомянуто выше, соседние вершины многоугольника всегда взаимовидимы, и ребро графа будет совпадать со стороной фигуры, соединяющей эти вершины. Два ребра, исходящие из одной вершины, образуют некий сектор (рис. 9), и все вершины, лежащие внутри него, не могут являться взаимовидимыми с вершиной сектора. Таким образом, проверке подвергаются только вершины, лежащие вне сектора, образуемого исследуемой вершиной, что позволяет сократить количество проверок.

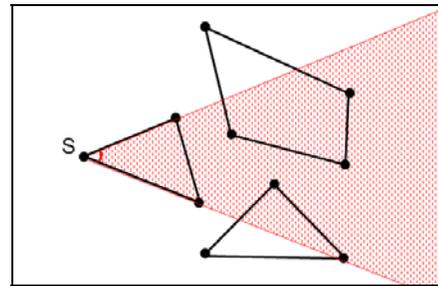


Рис. 9. Сектор обзора для вершины S

Заметим, что построение кратчайшего пути по графу видимости будет вестись лишь по выпуклым вершинам многоугольника (вершинам, внутренний угол которых меньше 180 градусов). То есть проверки можно сократить до проверок отрезков, соединяющих точки из множества всех выпуклых вершин, что существенно уменьшит количество общих проверок на взаимовидимость.

Проиллюстрируем пример построения графа видимости для областей, состоящих из нескольких полигональных объектов (рис. 10, 11).

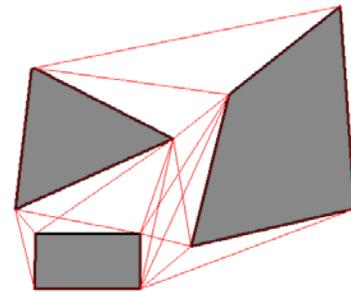


Рис. 10. Граф видимости для нескольких простых полигональных фигур



Рис. 11. Граф видимости для различных полигональных областей

5. ВЫЧИСЛИТЕЛЬНЫЕ ЭКСПЕРИМЕНТЫ

Для решения задачи поиска минимального пути между двумя точками граф видимости расширяется – добавляются начальная и конечная точки поиска пути к списку вершин, после чего отрезками соединяются видимые с ними вершины, аналогично описанному выше алгоритму. По полученному графу ищется кратчайший путь одним из стандартных алгоритмов поиска пути на графе (рис. 12, 13).

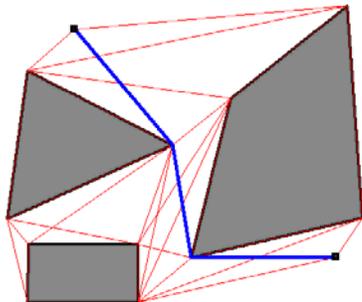


Рис. 12. Минимальный путь между двумя точками, найденный по графу видимости

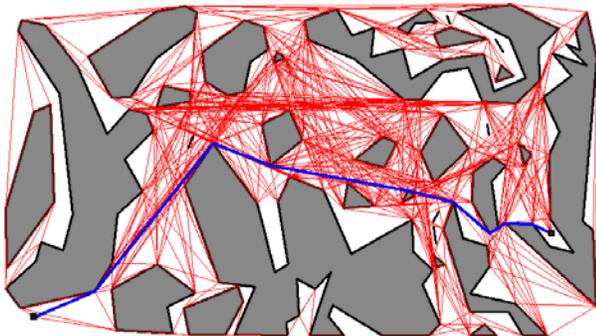


Рис. 13. Минимальный путь между двумя точками, найденный по графу видимости

В качестве проверки и апробации разрабатываемой системы были проведены вычислительные эксперименты для нахождения кратчайшего пути в произвольном лабиринте. В систему было загружено изображение лабиринта, после чего оно путем векторизации было приведено к набору полигональных областей, по которым был построен граф видимости. Найденный минимальный путь между заданными точками по графу (рис. 14, 15) будет являться кратчайшим маршрутом в лабиринте. Система решает данную задачу с высокой точностью и за короткий интервал времени.

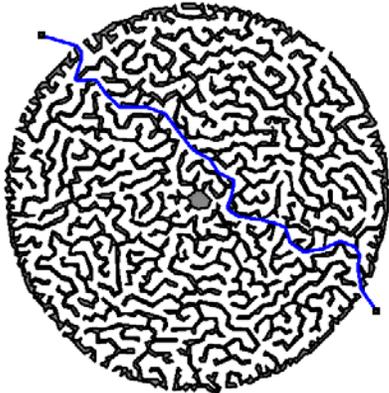


Рис. 14. Нахождение кратчайшего пути между двумя точками в лабиринте

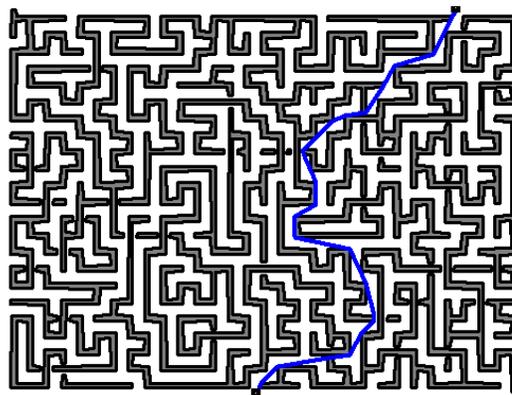


Рис. 15. Нахождение кратчайшего пути между двумя точками в лабиринте

6. ЗАКЛЮЧЕНИЕ

Описанная в статье система поиска минимального пути в среде с полигональными препятствиями представляет собой эффективный и простой инструмент, который легко может быть использован в различных приложениях.

7. БЛАГОДАРНОСТИ

Работа была выполнена при частичной поддержке гранта РФФИ № 12-07-00622-а.

8. ССЫЛКИ

- [1] Asano T., Guibas L.J., Hershberger J., Imai H. Visibility of disjoint polygons. *Algorithmica*, 1: 49 - 63, 1986.
- [2] Berg M., Cheong O., Kreveld M., Overmars M. *Computational Geometry: Algorithms and Applications*. Springer-Verlag Berlin Heidelberg, Third Edition: 323 - 333, 2008.
- [3] Stout V. (оригинальная статья), 1997, Maxim Kamensky (перевод), 2000). Алгоритмы поиска пути. <http://pmg.org.ru/ai/stout.htm>
- [4] Местецкий Л.М. Непрерывная морфология бинарных изображений: фигуры, скелеты, циркуляры. – М., Физматлит, 2009.
- [5] Семенов А.Б. Методы моделирования «живого» персонального почерка // Труды международной конференции «ГРАФИКОН-2009». – М.: МГУ, 2009. – С. 390 – 391.

Об авторах

Заева Камила – студентка магистратуры кафедры информационных технологий Тверского государственного университета.

Email: kapitanka.ki@gmail.com.

Семенов Андрей – кандидат физико-математических наук, доцент кафедры информационных технологий Тверского государственного университета.

E-mail: semenov@tversu.ru.

Дополнительные селективные признаки минеральных объектов для оптического метода сепарации

Д. Петухова, Д. Шитов, А. Чертов
Кафедра оптико-электронных приборов и систем
Университет ИТМО, Санкт-Петербург, Россия
PetuxovaDarja@yandex.ru

Аннотация

Настоящая работа посвящена исследованию возможности создания автоматического классификатора минеральных объектов на основе оптического метода сепарации с реализацией анализа дополнительных селективных признаков. В ходе работы формализован каждый из рассматриваемых селективных признаков, представлены полученные результаты.

Ключевые слова: оптический метод сепарации, минеральный объект, селективный признак, цвет, блестка, прожилка.

1. ВВЕДЕНИЕ

На сегодняшний день одним из перспективных методов обогащения руд твердых полезных ископаемых является оптический метод, основанный на регистрации цветовых различий минеральных объектов. Данный метод активно применяется во многих странах для сепарации различных продуктов и материалов, однако реализующее его обогатительное оборудование обладает рядом недостатков, сказывающихся на эффективности его работы и ограничивающих сферу применения метода в целом. Одним из таких недостатков является сложность работы со слабоконтрастным материалом, то есть материалом с незначительными цветовыми отличиями между составляющими его минеральными компонентами.

Решением данной проблемы может стать анализ дополнительных селективных признаков, которые в настоящий момент не регистрируются современными оптическими сепараторами. Такими признаками являются, например, особенности поверхности минеральных объектов типа «блестка» и «прожилка». Разработка методологии автоматического поиска и анализа дополнительных селективных признаков могла бы значительно повысить эффективность работы оптического метода обогащения, а также существенно расширить область его применения.

Таким образом, целью данной статьи является исследование возможности создания автоматического классификатора минеральных объектов, объединяющего возможности оптического метода сепарации и анализ указанных дополнительных селективных признаков.

2. МЕТОД БЛИЖАЙШИХ СОСЕДЕЙ

Разработка любого классификатора основывается на специфике входных данных: группы объектов, уже разделенных на классы (в данном случае, минеральных объектов, разделенных по степени обогащения), и группы объектов, класс которых не известен. Принадлежность к тому или иному классу определяется при помощи оценки признаков объектов. Таким образом, в задаче классификации выделяются два основных момента:

непосредственно сам алгоритм классификации и признаки разделения, с которыми он работает.

Существует огромное количество различных алгоритмов классификации [1 – 2]. Они отличаются, как по принципу работы, так и по способу их обучения. В данной работе использовался алгоритм, основанный на методе ближайших соседей. Данный алгоритм базируется на оценивании сходства объектов – объект относится к тому классу, к которому принадлежат ближайшие к нему объекты из обучающей выборки. Также следует отметить, что в работе используется несколько измененный метод взвешенных ближайших соседей. Для предотвращения случаев возникновения неоднозначности, обусловленной наличием более двух классов, i -му соседу анализируемого объекта приписывается вес w_i (как правило, убывающий с ростом ранга соседа i). Тогда объект относится к классу, который набирает больший суммарный вес среди k ближайших соседей (рис. 1)

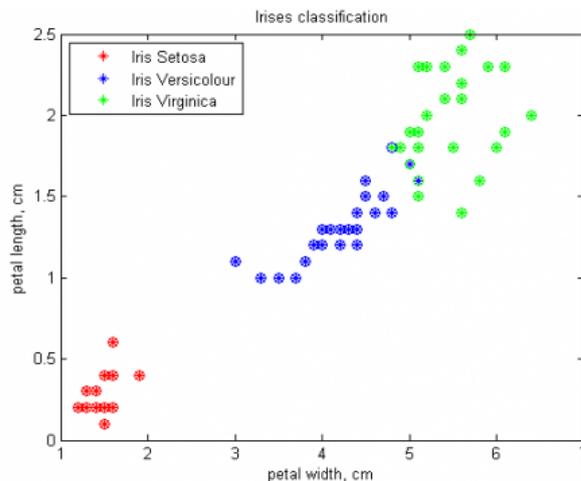


Рис. 1. Классификация по методу ближайших соседей. Классический пример реализации алгоритма – разделение «ирисов Фишера» по длине и ширине лепестка

Второй важной частью любого классификатора является формализация селективных признаков. В случае минеральных объектов поиск отличительных особенностей представляет собой трудоемкую задачу. Причина кроется в том, что в большинстве случаев между группами различной степени обогащения практически невозможно визуально определить разницу. Тем не менее, было выделено три основополагающих признака, которые, судя по опыту специалистов минералогов, являются наиболее достоверными при контроле руды. Обратимся к описанию каждого из них.

3. АНАЛИЗ СЕЛЕКТИВНЫХ ПРИЗНАКОВ МИНЕРАЛЬНЫХ ОБЪЕКТОВ

Формализация каждого из селективных признаков реализована в C# с подключением модуля обработки изображений EmguCV.

3.1. Цветовые параметры

Для решения задачи выделения необходимого цвета использовалась интерактивная цветовая сегментация. Анализ особенностей представления цвета при видеообработке и способов его количественной оценки показал, что используемые в существующих сепараторах цветовые модели характеризуются неравномерностью цветовых пространств, поэтому не всегда объективны для решения задач, связанных с анализом цвета [3]. На основании этого был сделан вывод о том, что цветовой анализ должен проводиться сразу в нескольких цветовых моделях с последующим выбором оптимальной из них.

Основываясь на сравнении таких модификаций известного метода Magic Wand, как пороговое выделение, выделение с использованием «чистых цветов», пороговое выделение с использованием радиуса в цветовом пространстве, для системы RGB было решено реализовать выделение цвета следующим образом. Оператор ставит метку на цветовой области, которую необходимо выделить (рис. 2). Определив координаты метки, алгоритм вычисляет среднее значение по всем трем каналам в области 5x5 вокруг данной метки. Далее каждый пиксель анализируемого изображения проверяется на выполнение следующего условия:

$$\sqrt{(r_{cp} - r)^2 + (g_{cp} - g)^2 + (b_{cp} - b)^2} > R,$$

где R – радиус окружности вокруг метки в цветовой системе RGB, а x_{cp} – цветовые координаты центра этой окружности.

В случае неудовлетворения условию пиксель обнуляется. Таким образом, на изображении остается лишь пиксели, которые попадают в окружности с радиус R в кубе цветовой системы RGB. Радиус, по умолчанию, равен 20, но оператор в случае необходимости может изменить это значение.

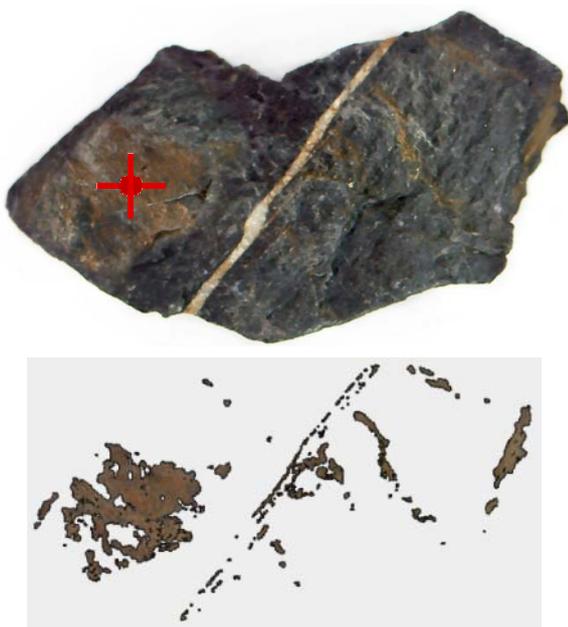


Рис. 2. Пример цветовой сегментации

В случае использования других цветовых моделей (HLS, LUV), в разной степени удовлетворяющих условию независимости цветовых составляющих, применяется обычное пороговое выделение. При этом каждому цветовому каналу назначается индивидуальное значение порога, что делает настройку более точной.

3.2. Блестки

При анализе такого селективного признака, как «блестки», важно отличать его от характерных для некоторых руд вкраплений светлых оттенков. Блестки характеризуются зашкаливающим сигналом в канале яркости. Для нахождения данного признака на образце использовалась следующая методика:

- изображение переводится в систему HSV и выделяется канал VALUE (дальнейшая обработка производится только над этим каналом);
- над одноканальным изображением производится пороговая бинаризация с целью выделения наиболее ярких участков изображения;
- на изображении производится поиск замкнутых контуров и их фильтрация по размеру (отбрасываются все контуры, размеры которых более 10*10 пикселей);
- производится подсчет оставшихся контуров и рассчитанное количество принимается за результат (рис. 3).



Рис. 3. Пример поиска и подсчета блесток

Таким образом, результатом работы предложенного алгоритма является подсчитанное количество небольших ярких участков («блесток») на объекте. Апробация алгоритма показала, что он не включил в конечный результат светлые вкрапления, на основании чего можно сделать вывод об эффективности его работы.

3.3. Прожилки

Поиск прожилков осложнен тем фактом, что их форма может значительно варьироваться от одного объекта к другому. У некоторых образцов прожилка имеет ярко выраженный цвет и простую вытянутую форму, у других – выглядит словно паутина, и толщина каждой линии слишком мала для качественного детектирования. Таким образом, было решено выделять только те прожилки, которые соответствуют первому из описанных случаев.

Для выделения прожилков выполняется следующая последовательность действий:

- изображение переводится в систему HSV и выделяется канал VALUE (дальнейшая обработка производится только над этим каналом);
- далее проводится нормализация гистограммы одноканального изображения с целью увеличения контрастности и улучшения яркости изображения;

- так как прожилки, за редким исключением, обычно представлены светлыми оттенками, обнуляются все пиксели, не попадающие в диапазон (250, 255);
- проводится обработка анализируемого одноканального изображения медианным фильтром с окном размытия 5x5;
- выполняется пятикратная обработка морфологическими операциями эрозии и расширения с окном 3x3;
- на изображении производится поиск замкнутых контуров и их фильтрация по двум параметрам (отбрасываются все контуры, площадь которых больше значения 0,01 (при этом площадь контура нормируется на площадь всего изображения) и контуры, отношение сторон которых меньше значения 3,5);
- производится подсчет оставшихся контуров и рассчитанное количество принимается за результат (рис. 4).



Рис. 4. Пример поиска прожилок

Как было отмечено выше, предложенная методика корректно анализирует только ярко выраженные прожилки. Следовательно, нуждается в доработке для учета и более сложных вариантов данного селективного признака.

4. ЗАКЛЮЧЕНИЕ

Таким образом, каждый из рассмотренных признаков формализован, и реализован алгоритм нахождения некоторой количественной оценки для каждого из них. В дальнейшем планируется разработать линейный метрический классификатор, специализирующийся на классификации минеральных объектов, оперирующий вышеописанными признаками разделения.

5. ССЫЛКИ

- [1] Vapnik V.N. Statistical learning theory. – N.Y.: John Wiley & Sons, Inc., 1998.
- [2] Witten I.H., Frank E. Data Mining: Practical Machine Learning Tools and Techniques (Second Edition). – Morgan Kaufmann, 2005.
- [3] Вакуленко А.Д., Горбунова Е.В., Чертов А.Н. Критерии применимости цветовых пространств в оптико-электронных системах цветового анализа минералов // Научно-технический вестник информационных технологий, механики и оптики. – 2012. – № 3 (79). – С. 156 – 157.

Об авторах

Петухова Дарья – аспирант кафедры ОЭПиС Университета ИТМО.

E-mail: PetukovaDarja@yandex.ru.

Шитов Денис – магистрант кафедры ОЭПиС Университета ИТМО.

E-mail: qdenisq@gmail.com.

Чертов Александр – старший научный сотрудник кафедры ОЭПиС Университета ИТМО.

E-mail: a.n.chertov@mail.ru.

Алгоритм нахождения отрезков в задаче анализа аэрофотоснимков

Л. Новоторцев, А. Волобой
Институт прикладной математики им. М.В. Келдыша, Москва, Россия
torets13@gmail.com, voloboy@gin.keldysh.ru

Аннотация

На данный момент в области геоинформатики и фотометрии большое значение имеют задачи, связанные с обнаружением, обработкой и построением различных объектов на аэрофотоснимках. В частности, это имеет отношение к обнаружению построек, для которых в большинстве случаев характерны линейные черты. По этой причине целесообразно сначала выделить отрезки на изображении. Потом, на основе найденных отрезков, проводить дальнейший анализ. Данная работа посвящена вопросу выделения линейных черт, таких, как отрезки на изображениях для последующего объединения их в связанные контуры.

Ключевые слова: Компьютерное зрение, выделение линий, выделение углов, преобразование Хафа.

ВВЕДЕНИЕ

В задачах компьютерного зрения часто возникают задачи, связанные с реконструкцией всей сцены или отдельных объектов на ней по множеству фотографий этой сцены, сделанных с разных ракурсов. Алгоритмы решения этих задач активно используются в картографии и смежных областях.

Внимание данной статьи направлено на рассмотрение задачи нахождения и реконструкции отдельных объектов по заданному набору шаблонов, представляющих собой трёхмерную каркасную модель.

Существует большое количество алгоритмов, связанных с реконструкцией сцены по набору фотографий, но все они имеют определённые недостатки при применении к описанной выше задаче. Первый подход связан с построением плотной модели ЦМР (Цифровая модель рельефа). Потом ЦМР обрабатывается с целью получить границы строений [3, 5, 7, 8]. Но общий метод недостаточно точен для построения зданий. Поэтому применяются методы, использующие априорную информацию о форме зданий, к примеру, плоские крыши у зданий [2]. Второй подход, который наиболее часто используют для задач реконструкции строений по набору аэрофотоснимков, опирается на выделение линейных участков на изображении (отрезков) и применение подхода перебора плоскостей [1]. Однако этот алгоритм на изображениях большого размера может работать неоправданно долго из-за сопоставления прямых на большом количестве изображений (рекомендуемое количество пересекающихся изображений – 5 – 6). По этой причине возникает задача сужения рассматриваемых областей для уменьшения вычислительных затрат.

Для реализации подобного алгоритма необходимо сначала разработать метод обнаружения прямых линий и отрезков на изображении.

Первым шагом решения задачи нахождения прямых является задача обнаружения контуров. Наиболее распространённым методом решения этой задачи является алгоритм Кэнни [4] нахождения граней. Этот алгоритм решает целый ряд проблем, которые возникают в подобных задачах, таких, как

дублирование контура. Результатом работы этого алгоритма является двоичное изображение, которое имеет нули во всех точках, где контура не обнаружено и ненулевые значения в остальных точках.

Наиболее распространённым и эффективным алгоритмом выделения прямых на изображении является преобразование Хафа [6, 9] и его модификации [10, 11], призванные ускорить расчёты, улучшить результаты или адаптировать его к конкретным постановкам задач. В качестве основы алгоритма, приведённого в этой статье, было решено использовать иерархический алгоритм Хафа, описанный в [11]. Его суть заключается в том, что изображение разбивается на блоки, в каждом из которых применяется преобразование Хафа, после чего прямые, полученные в каждом блоке объединяются, используя иерархический подход. Однако этот алгоритм имеет определённые недостатки. В случае участков с низкой контрастностью часть линии может быть не найдена, в результате чего линия будет сегментирована, что отрицательно может сказаться на последующих вычислениях.

Метод, предоставленный в данной статье, направлен на устранение данного изъяна.

ОБЩАЯ КОНЦЕПЦИЯ

В рассматриваемой статье предлагается использовать регулярное разбиение изображения и последующее нахождение прямых и их концов. Алгоритм можно разделить на три стадии:

1. Определение размера одного блока изображения, исходя из значения допустимой погрешности угла наклона прямых (отрезков).
2. Нахождение прямых линий в каждом блоке.
3. Нахождение концов найденных прямых.

Рассмотрим каждую стадию подробнее.

Определение размера блока

Первая задача, которую необходимо решить – выбрать размер блока. Ввиду дискретности изображения, при слишком малых размерах сегментов, невозможно будет различить прямые с малой разницей углов наклона. Минимальным требованием для возможности различить прямые, имеющие разницу в угле наклона $\Delta\theta$, является несовпадение их конечных пикселей. Иными словами, необходимо, чтобы $\Delta x \geq 1$ или $\Delta y \geq 1$. Ввиду симметричности синуса и косинуса, достаточно рассмотреть область от 0 до 45° . В этом случае $\Delta x \geq \Delta y$; это означает, что достаточно удовлетворить только неравенство $\Delta y \geq 1$. Тогда, учитывая, что

$$(1) y = L \times \sin(\theta),$$

где L – сторона сегмента, получим:

$$(2) 1 \leq \Delta y = y_1 - y_2 = L \times \sin(\theta + \Delta\theta) - L \times \sin(\theta),$$

$$(3) 1 \leq 2 \times L \times (\sin(\Delta\theta/2) \times \cos(\theta + \Delta\theta/2))$$

Учитывая, что $\theta \in [0; \pi/4]$, $-\cos(\theta + \Delta\theta/2) \geq \sqrt{2}/2$. Подставляя это в неравенство (3), получим:

$$(4) L \geq 1 / (\sqrt{2} \times \sin(\Delta\theta/2)).$$

В таблице приведены значения L для различных $\Delta\theta$.

$\Delta\theta$	1	2	3	4	5	6	7	8	9	10
L	82	41	28	21	17	14	12	11	10	9

Допустимая погрешность и минимальные размеры сегмента для обеспечения этой погрешности

Нахождение прямых линий

На данном этапе применяется непосредственно алгоритм Хафа к каждому блоку. Суть этого алгоритма заключается в том, что для каждой точки сегмента перебираются все углы наклона прямой с заданным шагом. Для каждой точки сегмента, помеченной, как граница, и для каждого предполагаемого угла наклона θ высчитывается длина перпендикуляра ρ , опущенного на предполагаемую прямую из точки начала координат. Потом в ячейку, соответствующую паре (θ, ρ) зачисляется один голос. В результате получается карта голосования. Далее, на основе подобного голосования находятся параметры прямых: те пары (θ, ρ) , для которых количество голосов превышает заданную планку, считаются параметрами найденной прямой.

В рассматриваемом методе предлагается выделять только одну прямую, набравшую наибольшее количество голосов, превышающее заданную планку. Предполагается, что потери на этом этапе компенсируются на третьем этапе при построении отрезков.

Выбор только одной прямой обусловлен тем, что при достаточно малых размерах блоков (в разделе 2.1 полученные размеры можно вполне считать малыми) потеря информации минимальна.

Нахождение концов найденных прямых

Следующим шагом является окончательное нахождение отрезков по найденным прямым. Основная сложность этого этапа заключается в том, что на предыдущем этапе линии могли быть сегментированы в результате низкого контраста участка изображения или в результате того, что линия в сегменте была слишком короткая.

Для решения этой проблемы предлагается использовать следующий подход. Для найденной прямой рассматриваются сегменты, полученные на первом и втором этапах, через которые она проходит. В зависимости от конкретных целей применения рассматриваемого алгоритма и реализации можно или рассматривать все сегменты, или рассматривать последовательно, начиная от того, где прямая была найдена. В каждом сегменте, через который проходит прямая, вычисляются пересечения прямой с его границами. Далее, вычисляется длина отрезка прямой внутри сегмента и длина перпендикуляра к ней в данном сегменте (угол наклона не изменяется):

$$(5) \begin{cases} y_0 = \frac{x_0 \cdot \sin(\theta) + \rho}{\sin(\theta)} \\ x_0 = \frac{\rho}{\sin(\theta)} \end{cases}$$

где y_0 – точка пересечения с левой стороной сегмента или её продолжением; x_0 – точка пересечения с нижней стороной сегмента или её продолжением.

В результате мы получаем параметры θ и ρ , описывающие прямую в данном сегменте.

После этого рассматриваются результаты голосования в данном сегменте. В этом случае достаточно рассмотреть только точки (θ, ρ) на карте голосования или её окрестность. Во втором случае окрестность достаточно взять равной единице по каждому из параметров. Это определяется тем,

что мы и так строили сегменты, для гарантирования необходимой нам точности. На этот раз в качестве планки для определения наличия прямой используется её длина внутри сегмента (возможно, какой-то процент от длины). По аналогии со вторым этапом прямая считается найденной, если количество голосов превышает эту планку.

Если прямая найдена в данном сегменте, то он помечается как сегмент, содержащий эту прямую, и алгоритм повторяется для следующего сегмента, через который проходит прямая, а конец прямой обновляется. В случае если рассматриваемая прямая совпадает с уже найденной в рассматриваемом сегменте, то найденная удаляется или помечается, как уже обработанная. В некоторых случаях бывает нужно также требовать не полное совпадение, а малое отличие двух прямых.

Если прямая не найдена, то в зависимости от допустимой длины разрыва алгоритм продолжается для следующего сегмента или останавливается.

РЕЗУЛЬТАТЫ



Рис. 1. Исходное изображение

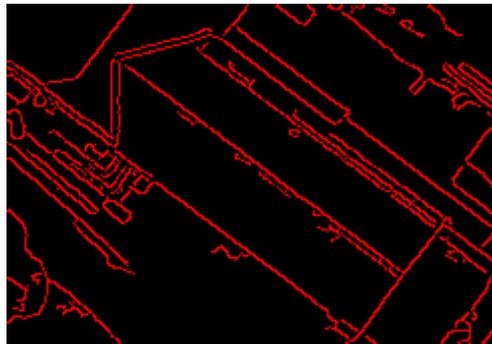


Рис. 2. Результат работы фильтра Кэнни

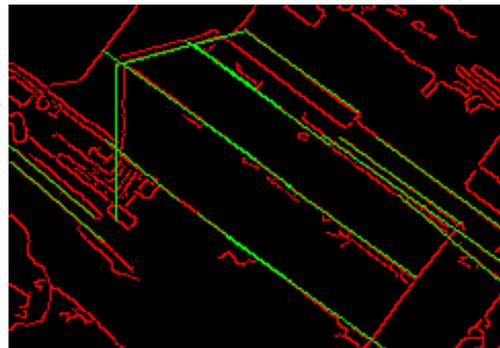


Рис. 3. Результаты, полученные алгоритмом Хафа, реализованным в OpenCV

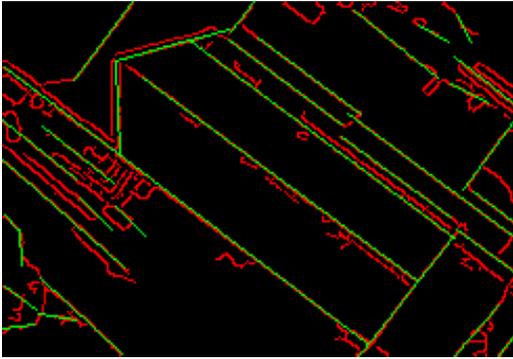


Рис. 4. Результаты, полученные рассматриваемым алгоритмом

На рис. 1 приведено исходное изображение. На рис. 2 представлен результат его обработки алгоритмом Кэнни. Красным обозначены найденные грани.

На рис. 3 приведён результат, полученный алгоритмом Хафа, который реализован в OpenCV. На рис. 4 – результат работы рассматриваемого алгоритма. Красным выделены грани, найденные фильтром Кэнни, зелёным – выделенные отрезки.

Рассматриваемый алгоритм имеет ряд преимуществ и недостатков по сравнению с другими вариантами реализации алгоритма Хафа.

К преимуществам можно отнести то, что показывают результаты (рис. 3, 4). Рассматриваемый алгоритм более точно выделяет отрезки по сравнению с алгоритмом Хафа, предложенным в OpenCV. На рис. 3 найдено гораздо меньше отрезков и присутствует сегментация отрезков, что негативно может сказаться на последующих вычислениях. Рассмотренный в данной статье метод даёт намного более приемлемый результат, что значительно облегчает дальнейшее связывание отрезков.

К недостаткам можно отнести то, что скорость работы, по сравнению с иерархическим алгоритмом Хафа, заметно меньше и то, что погрешность определения концов отрезков пропорциональна размерам сегментов.

ССЫЛКИ

1. Baillard C., Zisserman A. A plane-sweep strategy for the 3D reconstruction of buildings from multiple images //International Archives of Photogrammetry and Remote Sensing. – 2000. – Т. 33. – №. B2; PART 2. – С. 56 - 62.
2. 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. – С. 135 - 144.
3. Brunn A., Weidner U. Extracting buildings from digital surface models //International Archives of Photogrammetry and Remote Sensing. – 1997. – Т. 32. – №. 3 SECT 4W2. – С. 27 - 34.
4. Canny J. A computational approach to edge detection //Pattern Analysis and Machine Intelligence, IEEE Transactions on. – 1986. – №. 6. – С. 679 - 698.
5. 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. – Т. 32. – С. 49 - 56.
6. Duda R. O., Hart P. E. Use of the Hough transformation to detect lines and curves in pictures //Communications of the ACM. – 1972. – Т. 15. – №. 1. – С. 11 - 15.
7. Fernandes L. A. F., Oliveira M. M. Real-time line detection through an improved Hough transform voting scheme //Pattern Recognition. – 2008. – Т. 41. – №. 1. – С. 299 - 314.
8. Girard S. et al. Building detection from high-resolution color images //Remote Sensing. – International Society for Optics and Photonics, 1998. – С. 278 - 289.
9. Hough P. V. C. Method and means for recognizing complex patterns : пат. 3069654 США, 1962.
10. Wang Z., Wu F., Hu Z. MSLD: A robust descriptor for line matching //Pattern Recognition. – 2009. – Т. 42. – №. 5. – С. 941 - 953.
11. Yacoub S.B., Jolion J.M. Hierarchical line extraction // IEE Proceedings-Vision, Image and Signal Processing. – 1995. – Т. 142. – №. 1. – С. 7 - 14.

Корреляционный алгоритм распознавания символьной информации в телевизионных изображениях

Д. Григорович

ЗАО «Московский научно-исследовательский телевизионный институт», Москва, Россия
greendm@mail.ru

Аннотация

Автоматизация распознавания и анализа полетных данных, представленных в виде символьной информации на телевизионных изображениях, является актуальной на данный момент для ряда задач управления космическими аппаратами.

В данной работе рассматриваются вопросы выделения, обработки и распознавания символьной информации, передаваемой в телевизионном формате, предлагается корреляционный алгоритм ее распознавания с использованием бинарных шаблонов и дается описание его работы, приводятся результаты экспериментов на реальных изображениях.

Ключевые слова: распознавание символьной информации, телевизионные изображения, корреляция.

1. ВВЕДЕНИЕ

При решении специальных задач мониторинга и управления космическими аппаратами (КА) осуществляют совместную передачу телевизионного (ТВ) сигнала с изображениями КА и телеметрической информации (ТМИ). Такая ТМИ представляется в виде символьной информации, которая характеризует отдельные параметры движения КА, и отображается на экране видеоконтрольных устройств для их анализа операторами [3], [4].

Требуется обеспечить высокую достоверность анализа поступающей информации для принятия соответствующего правильного решения по управлению процессом движения КА в реальном масштабе времени. Также необходимо проводить оперативные мероприятия послеполетного анализа, устранение и анализ возможных внештатных ситуаций. Вследствие быстрой смены символьной информации одновременно по нескольким параметрам движения КА на оператора возлагается обработка значительного объема данных, представленных в ТВ изображениях. В связи с этим возникает вопрос автоматизации процессов обработки символьной информации, и ее сохранения в удобной форме для последующего чтения и проведения анализа, т.е. в виде графиков, таблиц и т.д.

В данной статье рассматриваются вопросы выделения, обработки и распознавания символьной информации, передаваемой в ТВ формате, предлагается корреляционный алгоритм ее распознавания с использованием бинарных шаблонов и дается описание его работы.

2. ПОСТАНОВКА ЗАДАЧИ ИССЛЕДОВАНИЙ

Решение описанной задачи можно осуществить с помощью поиска и распознавания на ТВ изображениях фрагментов, соответствующих одному из множества шаблонов [2]. Общепринятым подходом к решению подобных задач является корреляционное совмещение, которое сводится к

поиску максимума двумерной функции корреляции анализируемого (I) и эталонного (T) изображений [1], [5], [8]. Для получения корреляционной матрицы используется кросскорреляционная функция. Данная функция, записанная в терминах преобразования Фурье, будет иметь вид:

$$R(x, y) = \frac{J^{-1} [J[WT - WT] J^* [WI - WI]]}{\sqrt{\sigma^2(WT) \sigma^2(WI)}},$$

где J и J^{-1} – прямое и обратное преобразование Фурье, соответственно; $W = W(x, y)$ – весовая функция; σ^2 – дисперсия значений интенсивности внутри расчетной области. При вычислении преобразования Фурье используется алгоритм быстрого преобразования Фурье (БПФ), который дает преимущества во времени обработки перед прямым расчетом корреляционной функции.

Если размер изображения $W \times H$ и размер шаблона $w \times h$, то размер корреляционной матрицы $(W - w + 1) \times (H - h + 1)$. Оптимальный алгоритм сравнения, обеспечивающий наибольшую вероятность правильного распознавания, заключается в вычислении нормализованной корреляционной матрицы [9], причем $x' = 0 \dots w - 1$, $y' = 0 \dots h - 1$:

$$R(x, y) = \frac{\sum_{x', y'} (T'(x', y') \cdot I'(x + x', y + y'))}{\sqrt{\sum_{x', y'} T'^2(x', y') \cdot \sum_{x', y'} I'^2(x + x', y + y')}} ,$$

где $T'(x', y') = T(x', y') - 1 / (w \cdot h) \cdot \sum_{x'', y''} T(x'', y'')$,

$$I'(x + x', y + y') = I(x + x', y + y') - 1 / (w \cdot h) \cdot \sum_{x'', y''} I(x'', y'').$$

Нормализованная функция корреляции автоматически устраняет яркостные различия двух изображений. Идеальное совпадение анализируемого и шаблонного изображений будет при значении корреляции 1, полное различие изображений соответствует -1, значение 0 означает отсутствие корреляции между изображениями.

Далее после вычисления корреляционной функции производится поиск максимального значения, отвечающего наиболее вероятному смещению в области. Так как изображение представляется в виде конечного числа элементов, найденное положение максимума будет иметь целочисленные координаты в пикселях.

3. ИСХОДНЫЕ ДАННЫЕ

Исходные ТВ изображения имеют формат 720×576 пикселей (элементов), и представляются в виде 8-ми битных полутоновых изображений с 256-ю градациями яркости. Отдельное изображение является кадром, т. к. фоновая составляющая получается с внешней по отношению к КА ТВ

камеры. ТМИ в ТВ изображениях представлена в виде символьной информации в 52-х прямоугольных областях, называемых знакоместами. Указанные области имеют фиксированное расположение в ТВ кадре. Размерность знакоместа, состоящего из пяти символов, составляет 100×35 пикселей. Необходимо распознавать цифры, русские, некоторые английские и греческие буквы, и набор других символов. Символьная информация представлена в кадре определенным шрифтом на произвольном фоне КА. Отдельный символ занимает 14×23 пикселей.

При передаче символьной информации может ухудшаться ее качество за счет того, что:

- ТВ изображения могут подвергаться различным случайным искажениям, маскирующим характерное отличие символьной информации;
- ТВ сигнал может иметь малое значение отношения сигнал-шум;
- может изменяться фоновая составляющая ТВ изображения в местах расположения символьной информации и др.

Распознаваемые фрагменты ТВ изображений приведены на рис. 1.

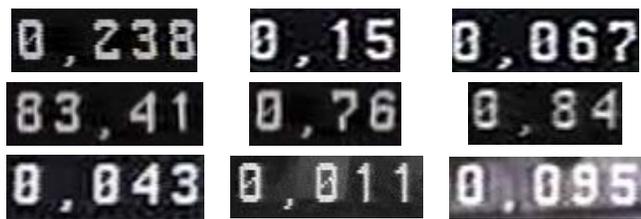


Рис. 1. Реальные фрагменты ТВ изображений с символьной информацией

Исходные данные о каждом символе должны быть представлены в отдельном файле, структура и формат которого определяется алгоритмом распознавания. Для каждого параметра движения КА задан диапазон допустимых значений.

4. ОПИСАНИЕ АЛГОРИТМА

В разработанном алгоритме используются специально подготовленные матрицы бинарных шаблонов символов, встречающихся в ТВ изображениях. Использование подобных эталонов позволяет повысить быстродействие и снизить влияние шумов.

Образцы символьной информации в виде бинарных шаблонов цифр представлены на рис. 2.



Рис. 2. Образцы бинарных шаблонов цифр

Корреляционный алгоритм распознавания символьной информации, представленной в ТВ изображениях, с использованием бинарных шаблонов (см. Алгоритм 1) включает несколько этапов обработки информации и работает следующим образом.

Вначале осуществляется получение ТВ изображения. Фиксированная структура ТВ кадра значительно облегчает и ускоряет сегментацию областей с отдельными параметрами движения КА. Координаты верхней левой вершины и размеры областей содержатся в априорных данных. С

помощью этой информации полученный кадр разбивается на отдельные интересующие области (ROI – Region of Interest).

Далее производится пороговое преобразование. На этом этапе обработки информации происходит преобразование полутонового изображения в бинарное изображение (рис. 3) с помощью порогового преобразования с фиксированной величиной порога.



Рис. 3. Бинарные фрагменты ТВ изображений с символьной информацией

<i>Корреляционный алгоритм распознавания символьной информации в ТВ изображениях с использованием бинарных шаблонов</i>	
Начало	
1.	Цикл: Для каждого полученного изображения
2.	Цикл: Для каждой интересующей области
3.	Пороговое преобразование
4.	Поиск контуров на изображении
5.	Аппроксимация контуров полигонами
6.	Цикл: Для каждого найденного контура
7.	Создание минимального ограничивающего прямоугольника
8.	Сегментация символа
9.	Инвертирование матрицы символа
10.	Нормализация символа к размеру шаблона
11.	Цикл: Для каждого шаблона
12.	Вычисление корреляционной функции между анализируемым символом и шаблоном
13.	Определение максимального значения корреляционной функции
14.	Конец цикла
15.	Определение максимума из найденных максимальных значений корреляционных функций. Максимум соответствует лучшему совпадению анализируемого символа с шаблоном
16.	Конец цикла
17.	Формирование результата из распознанных символов
18.	Анализ полученного результата на основе априорных данных
19.	Сохранение корректного результата в базе данных
20.	Построение графика и обновление таблицы параметра ТМИ
21.	Конец цикла
22.	Конец цикла
Конец алгоритма	
Алгоритм 1. Корреляционное распознавание символьной информации	

Одной из важных задач обработки и анализа ТВ изображений является сегментация, т.е. разделение его на области, для которых выполняется определенный критерий однородности, в данном случае, выделение на изображении областей приблизительно одинаковой яркости. Один из основных, простых и надежных при малом разрешении исходных изображений способов – это построение сегментации с помощью порогового преобразования. Порог – это признак, позволяющий разделить исходное ТВ изображение на два уровня яркости, каждый из которых будет отображать свой

класс объектов. Яркость пиксела бинарного изображения принимает два значения, черный и белый цвета. При уменьшении числа значений яркости исчезают слабоконтрастные детали, плавные переходы тона становятся более резкими.

Следующим этапом обработки ТВ изображения является поиск контуров символов на изображении знакоместа. Под контуром обычно понимается местоположение локального изменения или резкого перепада яркости на изображении. При этом подразумевается, что такие перепады возникают на границах объектов. Для кодирования контура используется цепной код Фримена [7], с помощью которого представляются границы в виде последовательности отрезков прямых линий определенной длины и определенного направления. Суть цепного кода заключается в том, что для любой связанной линии на изображении кодируются координаты начального пиксела, а для каждого последующего пиксела цепочки в качестве кода используется его приращение, определяющее переход на один из смежных пикселей. Поскольку таких смежных пикселей всего 8, то для кодирования каждого пикселя необходимо 3 бита информации. В данном алгоритме выполняется поиск только крайних внешних контуров символов (рис. 4).



Рис. 4. Контуров символов, найденные во фрагментах ТВ изображений

Далее найденные контуры аппроксимируются полигонами методом Дугласа-Пекера [6]. Идея обработки по указанному методу состоит в построении по контуру полигона с меньшим числом точек. Алгоритму задается желаемая точность аппроксимации, которая определяет максимальное расстояние между исходным контуром и полигоном.

Затем создается минимальный прямоугольник, ограничивающий контур, и осуществляется сегментация символа путем извлечения из изображения матрицы, соответствующей ограничивающему прямоугольнику [9] (рис. 5). Каждый бит матрицы символа инвертируется (рис. 6). Дополнительно производится нормализация символа, т.е. масштабирование каждого символа до определенного размера, для сравнения с шаблонами символов, хранящихся в файлах. Вопрос масштабирования требует отдельных исследований для определения оптимального масштаба шаблонов символов, при котором точность распознавания будет максимальной. При увеличении масштаба символов, с одной стороны, усиливаются их характерные отличия друг от друга, с другой стороны, увеличиваются искажения и дефекты символьной информации, подвергаемой распознаванию. Для этой задачи был взят размер шаблона символа 14×23 пикселей. При этом анализируемый символ масштабируется до того же размера.



Рис. 5. Фрагменты ТВ изображений с ограничивающими символами прямоугольниками

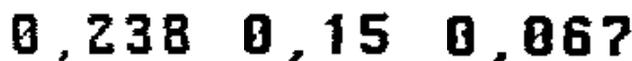


Рис. 6. Фрагменты инвертированных ТВ изображений с символьной информацией

Следующим этапом данного алгоритма является вычисление корреляционной матрицы для изображений символа и каждого шаблона в отдельности. Для каждой корреляционной функции определяется максимум, соответствующий наиболее вероятным координатам смещения в области анализируемого изображения. Затем определяется лучшее совпадение по максимуму из найденных максимальных значений корреляционной функции для каждого шаблона. Если размеры распознаваемого изображения I и шаблона T равны, то результирующая матрица состоит из одного элемента. В данном алгоритме за счет выделения каждого отдельного анализируемого символа с помощью минимального ограничивающего прямоугольника значительно сокращается объем вычислений, поскольку размер символа приводится к размеру шаблона, и уменьшается область поиска.

Из распознанных символов формируется результат в виде строки. Результат распознавания анализируется, является ли он корректным числом или строкой текста на основе априорных данных возможных значений параметра ТМИ.

В результате выполнения указанных операций по обработке ТВ изображений в каждом знакоместе осуществляется распознавание символьной информации, содержащей полетные данные КА. Распознанная информация сохраняется в базе данных. Впоследствии полученные данные используются для построения графиков и таблиц, характеризующих параметры движения КА за определенный отрезок времени.

5. ЭКСПЕРИМЕНТАЛЬНЫЕ РЕЗУЛЬТАТЫ

Для реализации указанного алгоритма обработки и распознавания ТВ изображений было написано программное обеспечение (ПО) с графическим интерфейсом на языке C++. Для разработки программы использовалась кросс-платформенная библиотека Qt [10]. Ряд этапов данного алгоритма был реализован с применением библиотеки алгоритмов компьютерного зрения, обработкой изображений и численных алгоритмов общего назначения с открытым кодом OpenCV [9]. Настройки графического интерфейса, априорная информация, и вспомогательные данные процесса распознавания ТВ изображений хранятся в XML-файлах, и считываются программой при ее загрузке. Корректные результаты распознавания символьной информации, содержащейся во всех подвергаемых распознаванию знакоместах ТВ изображений, записываются в базу данных SQL в реальном времени. Графики параметров движения КА строятся и отображаются с использованием набора Qt-виджетов и вспомогательных классов Qwt [11] в процессе работы ПО. В программе реализована многопоточная обработка всех знакомест, что позволяет распознавать большое количество символьной информации параллельно.

Проверка корреляционного алгоритма осуществлялась на компьютере с процессором Intel(R) Core(TM) i5-2410M 2.30 ГГц, оперативной памятью DDR3 800 МГц, под управлением 64-битной операционной системы Windows 7.

Результаты экспериментальных исследований показали, что время распознавания одного знакоместа в ТВ изображении, содержащего пять символов составило 10 мс. Было проведено распознавание 300 тестовых ТВ изображений с символьной информацией. Точность распознавания символьной информации в ТВ изображениях составила 85%.

6. ЗАКЛЮЧЕНИЕ

Для распознавания символьной информации в ТВ изображениях с использованием бинарных шаблонов разработан корреляционный алгоритм обработки ТВ изображений и написано ПО. Сегментация символов осуществляется с помощью контуров, что позволяет значительно сократить размер корреляционной матрицы и уменьшить время поиска лучшего совпадения с шаблоном. Для определения степени близости входного изображения и шаблонов используется вычисление кросс-корреляции с помощью алгоритма БПФ, при этом уменьшается общая вычислительная сложность алгоритма. Недостатком корреляционной меры сходства является ее чувствительность к геометрическим искажениям видимых размеров символов при изменении ракурса съемки.

Алгоритм способен распознавать символьную информацию в ТВ изображениях с малым значением отношения сигнал-шум, невысоким разрешением и работать в реальном масштабе времени с параллельной обработкой до 100 исходных изображений ТМИ.

По результатам экспериментальных исследований корреляционный алгоритм распознавания символов с использованием бинарных шаблонов обладает достаточно высокой точностью распознавания и высоким быстродействием, что делает его применение оправданным при распознавании символьной информации в ТВ изображениях. Дальнейшее улучшение точности распознавания связано с применением более гибкого порогового преобразования, а также использованием дополнительных признаков, обеспечивающих достоверную селекцию символьной информации на произвольном фоне.

7. ССЫЛКИ

- [1] Гонсалес Р. Цифровая обработка изображений. – М., 2005. – 1072 с.
- [2] Григорович Д.В. Распознавание символьной информации в телевизионных изображениях с использованием бинарных шаблонов. Сб. докл. XX международной научно-технической конференции «Радиолокация, навигация, связь» (RLNC-2014). – Т. 1. – С. 159 – 164, Воронеж, Россия, апрель 2014.
- [3] Григорович Д.В., Смирнов А.И. Оперативная селекция телеметрической информации в телевизионных изображениях. Сб. мат. XI Международной научно-технической конференции «Распознавание-2013». – Курск, Россия, сентябрь 2013. – С. 382 – 384
- [4] Кравец В.Г., Любинский В.Е. Основы управления космическими полетами. – М., 1983. – 224 с.
- [5] Сойфер В.А. Методы компьютерной обработки изображений. – М., 2001. – 784 с.
- [6] Douglas D.H., Peucker T.K. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature, Journal «The Canadian Cartographer», vol. 10, 1973. – No 2. – P. 112 – 122.
- [7] Freeman H. On the encoding of arbitrary geometric configurations, IEEE Electronic Computers, IRE Transactions, vol. 10, 1961. – No 2. – P. 260 – 268.
- [8] Nikhil Gupta, Rahul Gupta, Amardeep Singh, Matt Wytock. Object Recognition using Template Matching, December 2008. <http://cs229.stanford.edu/proj2008/GuptaGuptaSinghWytock-ObjectRecognitionUsingTemplateMatching.pdf>.
- [9] OpenCV API Reference. <http://docs.opencv.org/modules/refman.html>

[10] Qt Project Documentation. <http://qt-project.org/doc/>.

[11] Qwt User's Guide. <http://qwt.sourceforge.net/>.

Об авторе

Григорович Дмитрий – начальник сектора 333 ЗАО «МНИТИ».

E-mail: greendm@mail.ru.

Abstract

At present an automation of recognition and analysis of the flight data as the character information in TV images is an urgent issue for some control tasks of space vehicles. This article is devoted to the character information extraction, processing and recognition in TV format.

The correlation algorithm of character information recognition with use of binary patterns is considered, its operation description is performed, and the experimental results with real images are described.

Keywords: *character information recognition, TV images, correlation.*

Estimation of Human Age and Facial Expression Using Biologically Inspired Features

A. Spizhevoy, A. Bovyryn

Lobachevsky State University of Nizhni Novgorod and Itseez Inc., Russia
{alexey.spizhevoy, alexander.bovyryn}@itseez.com

Abstract

In this paper we deal with the problems of human age and facial expression estimation. We propose to use Biologically Inspired Features (BIF) as a facial descriptor in the both tasks to improve performance. We developed a complete pipeline for solving those problems including geometric normalization step and CLAHE method for photometric normalization. To the best of our knowledge using BIF for facial expression classification, as well as combining BIF together with ordinal hyperplanes ranking for age prediction haven't been studied before. The proposed pipeline was tested on the standard datasets such as the FG-NET aging and the Extended Cohn-Kanade (CK+) expression databases and demonstrated high accuracy compared to the state-of-the-art methods.

Keywords: *human age estimation, face normalization, bio-inspired features, ordinal hyperplanes ranking.*

1. INTRODUCTION

Interest in solving such problems as human age estimation and facial expression classification has been growing for many years. Those tasks arise in many areas such as: human-computer interaction, surveillance monitoring, video content analysis, targeted advertising, biometrics, and entertainment.

A typical solution that recognizes facial expression or human age by image/video automatically usually includes a few basic blocks: face detection and cropping, image geometric and photometric normalization, computing face descriptors, reducing optionally feature vector dimensionality, and applying a machine learning method for estimation of age and expression.

Both problems are quite sophisticated because of high variability in face appearance due to such factors as head rotation, emotions, illumination conditions, face makeup, and many others. All of these issues should be resolved, to some extent, to build an automatic solution which would be reasonable in practice. Some of the factors are taken into account during face normalization, while the others are resolved via enriching training dataset to model age/expression function more accurately.

Contribution: the current work is application guided and describes key elements of the developed system. Our contribution includes the following items:

- For the sake of achieving time performance gain we proposed to use the same features for solving both tasks.
- We built a fully automatic system that is able to estimate human age and facial expressions in real time by frames from video.
- The developed system showed accuracy higher than the state-of-the-art methods on the standard CK+ expression and FG-NET aging databases.

2. RELATED WORK

The goal of human age estimation is to predict age or age range by person's face image. All existing approaches can be roughly split into two classes by convention, depending on what type of machine learning algorithm is used. That is either regression [8, 6] or classification [2]. The former one considers age as a continuous variable, and the latter one as a discrete variable, i.e. age group index. There are also mixed approaches, for instance [6, 7]. Such factors as illumination conditions when taking photo, face makeup, facial expressions and other might lead to age estimation errors, while a human can cope with those issues quite easily.

The problem of automatic facial expression recognition is challenging as well. The same factors that might worsen age estimation results are applicable here, i.e. illumination conditions, face makeup, genetics, etc. Moreover different human expressions can be presented at the same time which makes the problem even more difficult. As noted in [10] facial expressions in humans are dynamic in nature, consisting of an onset, peak and an offset phase. Some approaches incorporate temporal information, for instance see [10, 13, 15]. However typical methods rely on appearance features mostly, we call such methods static.

Active Appearance Models are widely used for both age estimation and expression classification as facial descriptors, see [2, 6, 11, 4] for they combine both appearance and shape information. There are some works devoted to using Local Binary Patterns for age estimation and expression classification, see [14, 5] as LBP features are quite successful in face recognition and can be computed very efficiently. One of the best results demonstrated for age estimation was achieved using descriptors based on Gabor filters, see [8, 7].

In this work we investigate behavior of Biologically Inspired Features [8, 9] on the problem of facial expression classification as such approach hasn't been studied before. We show that it provides very high accuracy compared to the best known methods. Finally we propose a pipeline that uses the same features for solving both problems: age estimation and expression classification -- that reduces the average computational time in 23%, compared to the version where facial descriptors are not shared between the problems and computed independently.

2.1. Problem Statement

The problem of age estimation can be formulated as follows: given training dataset consisting of m samples $\{(X_i, y_i), i = 1..m\}$, where $X_i = X(I_i)$ is the i -th face descriptor, computed from the face image I_i , and $y_i = y_{age}(X_i)$ is the ground truth age value, either exact age or age group index. The goal is to estimate age $\hat{y}_{age}(X(I))$ by new unknown face image I , which is not presented in the training database.

The goal of facial expression classification is to determine facial expression by person's face image. We tract the task as a machine

learning classification problem. So it is formulated as follows: given training dataset consisting of m samples $\{(X_i, y_i) | i=1..m\}$, where $X_i = X(I_i)$ is the i -th face descriptor, computed from the face image I_i , and $y_i = y_{ex}(X_i) \in C_{ex} = \{An, Co, Di, Fe, Ha, Sa, Su\}$ is the ground truth expression label. In our work we deal with such facial expressions as angry, contempt, disgust, fear, happy, sadness, and surprise. The facial expression labeling of that kind is provided with the Extended Cohn-Kanade (CK+) dataset [11]. The goal is to estimate facial expression label $\hat{y}_{ex}(X(I))$ by new unknown face image I , which is not presented in the training database.

3. FACE DESCRIPTORS

As a base approach for face descriptors computation we use Biologically Inspired Features (BIF) proposed in the paper [8] for both age estimation and facial expression classification problems. Using the same descriptors saves about 23% of the computational time, since we don't have to compute different features independently -- all we need is just to compute BIF once per face. See section 3.4 for details about descriptors parameters.



Fig. 1. Results of face normalization on the FG-NET aging database images. Top row shows original images, bottom row shows normalized images

3.1. Geometric Normalization

The goal of this step is to remove uninformative geometric transformations from images such as face scale variations and in-plane head rotations. We use eye centers and map them, using similarity transformation, into two predefined points. By doing this we eliminate in-plane head rotations and scale variations, so they don't affect facial descriptors anymore. In our experiments we used eye center positions provided with the databases.

Here is the description of the method we propose. Let $p_{le} = (x_{le}, y_{le})^T$ and $p_{re} = (x_{re}, y_{re})^T$ be the coordinates of left and right eyes respectively on source image I , where W and H are its width and height respectively. Using similarity transformation, i.e. combination of rotation, scale, and translation, we map the points p_{le}, p_{re} into fixed points $\hat{p}_{le}, \hat{p}_{re}$.

We use the following fixed points which were adjusted experimentally: $\hat{p}_{le} = (0.78W, 0.25H)^T$ and

$\hat{p}_{re} = (0.22W, 0.25H)^T$. Image width and height (W and H) are the same for all images -- that's insured via image resizing after face detection and image cropping. We used the values $W=66$ and $H=66$ that gave the best results in our experiments.

Experiments showed that geometric normalization is a vital step for achieving high accuracy results. The described method decreases error on the FG-NET aging database from 5.5 to 4.56 years, see table 1.

3.2. Photometric Normalization

Input of the photometric normalization step is geometrically normalized image I . This step is important for achieving accurate results as well, and is aimed to reduce uninformative illumination variations that might worsen face recognition accuracy. We proposed two methods for experiments: Histogram Equalization (HE) and Contrast Limited Adaptive Histogram Equalization (CLAHE). For detailed descriptions we refer to [12].

While HE method improves overall contrast, it often makes sense to improve local contrast, e.g. when one part of face is much lighter/darker than the other. Our experiments showed that CLAHE provides better results than HE. You can see the accuracy achieved using HE and CLAHE for age estimation problem on the FG-NET aging database in table 1. While HE together with the geometric normalization, described in the previous section, gave 4.32 years error, using CLAHE we achieved 4.1 years error. Examples of face normalization results are shown in figure 2.

3.3. Biologically Inspired Features

In working with BIF we follow to the work [8]. The method takes as input normalized image I of some fixed predefined size. Then a number of Gabor filters are applied to the image and then the response maps are processed further. The whole feature computation algorithm can be described as follows:

1. For each angle $\theta \in [0, \pi)$ with some step:
 - a. Apply a set of Gabor filters with orientation θ to image I .
 - b. Split the response maps into pairs and merge them using per-pixel maximum operation.
 - c. Computer statistical features (standard deviations) passing over the images with sliding window.
2. Combine all features into one vector.

The results of steps 1.1 and 1.1 form so called simple (S) layer, when results of step 1.3 form complex (C) layer from S layer elements.

3.4. Dimensionality Reduction

As the total number of features after C layer elements are computed and merged into one vector can be very high, there is a need of performing dimensionality reduction. To reduce the number of features we apply Principal Component Analysis (PCA) method. Applying this technique we reduce feature vector size to 881 -- the best value from our experiments with age estimation on the FG-NET aging database. For the problem of expression classification the reduced feature vector size is 654 -- that is the number of samples in the CK+ database after augmenting it with the vertically flipped images.

4. AGE ESTIMATION

The authors of the paper [2] presented a ranking method for age estimation. They called it Ordinal Hyperplanes Ranker -- OHRank and used it together with Active Appearance Models. In this section we describe our modification of this method that we use together with BIF descriptors.

Let X_i be the i -th feature vector from the training set, which corresponding ground truth age is $y_i \in \{0, 1, \dots, K\}$, and K is the

maximum possible age (e.g. the maximum age observed in the database). For each possible age year k we split the training dataset into two parts according to the following rule:

$$X_k^+ = \{(X_i, 1) | y_i > k\},$$

$$X_k^- = \{(X_i, 0) | y_i \leq k\}.$$

Then these sets are used to train binary classifiers $\{f_k\}$. These classifiers estimate answers (as discrete variable from the $\{0,1\}$ set) on the following queries: «is age of given person greater than k years or not»? To get age estimation all of the K binary classifiers are applied to the input feature vector X . Final value is aggregated via computing sum of all the binary classifier

$$\text{answers: } y(X) = \sum_{k=0}^{K-1} f_k(X).$$

5. FACIAL EXPRESSION CLASSIFICATION

For facial expression classification we propose merging texture and geometric features (i.e. face shape landmarks) via aggregation of probability distributions from different SVM models. Image samples with the face shape landmarks from the CK+ database are shown in figure 2.

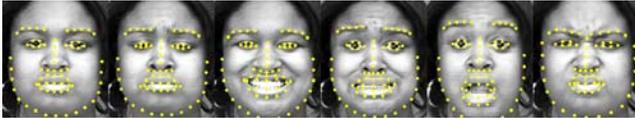


Fig. 2. Sample images from the CK+ database with the face shape landmarks

Face shape features are obtained via combining 68 face landmark points resulting into 136 dimensional feature vector. Before merging points into one feature vector they are transformed (shifted and scaled) into the face rectangle system of coordinates.

In detail the process looks as follows: given $P_1(y|X_1)$ and $P_2(y|X_2)$ -- the probability distributions obtained from two different SVM models, where X_1 and X_2 are two different feature vectors corresponding to the same image, the score function is computed as their product: $S(y|X_1, X_2) = P_1(y|X_1)P_2(y|X_2)$. Final result of classification is determined as $y_{ex}(X_1, X_2) = \arg \max_{y \in C_{ex}} S(y|X_1, X_2)$.

On the CK+ database merging probability distributions from two different SVM models with RBF kernels improves the average recognition rate from 94.1% (BIF only) to 96.8% (BIF & shape).

6. EXPERIMENTS

6.1. Standard Datasets

6.1.1. The FG-NET Aging Database

For human age estimation validation we used the standard FG-NET aging database [1]. The database contains 1002 face photos of 82 persons taken under uncontrolled conditions. The images have some light intensity and head pose variations. For each person there are more than 10 images in the range from 0 to 69 years. For each face image there are 68 landmarks available, from which we use only eyed coordinates for geometric normalization.

6.1.2. The Extended Cohn-Kanade Database

For facial expression validation we used the Extended Cohn-Kanade (CK+) database [11]. The database contains 593 face image sequences, but only 327 are labeled with expression classes. Those 327 images cover 118 out of 123 persons. Each sequence begins with the neutral face and ends with the peak intensity expression. We used only images with the peak expression intensity. The database provides the 68 face landmarks labeling.

6.2. Accuracy Metrics

There are two widely used age estimation accuracy metrics. The first one is the mean absolute error, which is computed as follows:

$$MAE = \frac{1}{m} \sum_{i=1}^m |y_i - \hat{y}_i|,$$

where m is the total number of the

test samples, y_i is the ground truth age value for the i -th test sample, and \hat{y}_i is the predicted age value. The second accuracy metric is the cumulative score, which reflects how prediction errors are distributed over years. For every error level L it is

$$\text{computed as } CS(L) = \frac{m_{e \leq L}}{m} 100\%,$$

where $m_{e \leq L}$ is the number of

the test sample prediction errors less or equal to L .

Method	MAE	CS(5)
BIF+SVR [8]	4.77	≈69
AAM+OHRank1 [2]	4.48	74.4
AAM+OHRank2 [2]	4.56	74.7
C-IsRCS+IsLPP [3]	4.38	≈74
Ours (no norm)	5.5	68.7
Ours (geom)	4.56	74.3
Ours (geom+HE)	4.32	74.7
Ours (geom+CLAHE)	4.1	76.4

Table 1. Comparison of the age estimation results on the FG-NET aging database. The proposed approach outperforms the state-of-the-art methods on the same database. Also the table shows how geometric and photometric normalization steps are important.

6.3. Results

In table 1 we compare the results of our human age estimation approach with the best reported ones so far on the FG-NET aging database. From the table you can see that our approach shows high accuracy in comparison to the best methods. The table also shows superior accuracy of the CLAHE based photometric normalization over the HE based one. Leave-one-person-out (LOPO) cross-validation technique was used to get accuracy estimate.

Method	Avg. Rec. Rate, %	Static	Valid. Protocol
Baseline [11]	83.3	Yes	LOPO
CLM [4]	74.4	Yes	LOPO
Shape+SVM [10]	84.06	Yes	4-fold
Shape+LDCRF [10]	95.79	No	4-fold
Cov3D [13]	92.3	No	5-fold
CPL [16]	88.42	Yes	10-fold
ITBN [15]	86.3	No	15-fold
Ours (no norm)	91.9	Yes	LOPO

Ours (geom)	92.7	Yes	LOPO
Ours (geom+HE)	94.4	Yes	LOPO
Ours (geom+CLAHE)	96.8	Yes	LOPO

Table 2. Comparison of recognition rates for 7 facial expressions classification on the CK+ database. The 3rd column denotes whether only one image is used in the method to estimate expression (we call such methods static) or a few frames are used. Here again the proposed approach outperforms the state-of-the-art methods on the same database.

For facial expression classification problem we compute the confusion matrix in exactly the same way as the authors of paper [11] did, i.e. LOPO cross-validation and 7 facial expressions (anger, contempt, disgust, fear, happy, sadness, surprise).

7. CONCLUSION

We proposed to use a combination of BIF together with ordinal hyperplanes ranking for human age estimation that hasn't been studied before. The developed approach was tested on the standard FG-NET aging database and MAE of 4.1 years was achieved. We investigated face normalization methods and the results showed that a huge improvement in accuracy can be made using a combination of geometric normalization and CLAHE as photometric normalization, see table 1. Haven't been applied to the problem of facial expression classification before, the proposed pipeline was tested on the the CK+ database and the result of 96.8% average recognition rate was achieved, see table 2.

	An	Co	Di	Fe	Ha	Sa	Su
Recogn. Rate, %	95.6	88.9	98.3	96	100	100	98.8

Table 3. Recognition rates for each of 7 facial expressions on the CK+ database obtained using our approach.

The obtained results show that our framework provides high accuracy for both age estimation and expression classification problems compared to the best known methods. The pipeline is also perspective for using in applications as the same features are used for solving the both problems, hence BIF must be computed only once per image. Sharing BIF descriptors between both age estimation and expression classification saves about 23% of the computational time, compared to the version where facial descriptors are not shared between the problems and computed independently.

8. REFERENCES

[1] The FG-NET Aging Database. <http://www.fgnet.rsunit.com>, <http://www.prima.inrialpes.fr/FGnet/>.

[2] Chang, K.-Y., Chen, C.-S., and Hung, Y.-P. Ordinal hyperplanes ranker with cost sensitivities for age estimation. *In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on* – P. 585 - 592. IEEE (2011).

[3] Chao, W.-L., Liu, J.-Z., and Ding, J.-J. Facial age estimation based on label-sensitive learning and age-specific local regression. *In Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on.* – P. 1941 - 1944. IEEE (2012).

[4] Chew, S.-W., Lucey, P., Lucey, S., Saragih, J., Cohn, J.-F., Sridharan, S. Person-independent facial expression detection using constrained local models. *In Automatic Face & Gesture*

Recognition and Workshops (FG 2011), 2011 IEEE International Conference on. – P. 915 - 920. IEEE (2011).

[5] Gunay, A., Nabyev, V.-V. Automatic age classification with LBP. *In Computer and Information Sciences, 2008. ISCI'08. 23rd International Symposium on* – P. 1 - 4. IEEE (2008).

[6] Guo, G., Fu, Y., Dyer, C.-R., and Huang, T.-S. Image-based human age estimation by manifold learning and locally adjusted robust regression. *Image Processing, IEEE Transactions on* 17, 7, 1178 - 1188 (2008).

[7] Guo, G., Fu, Y., Huang, T.-S., and Dyer, C.-R. Locally adjusted robust regression for human age estimation. *In Applications of Computer Vision, 2008. WACV 2008. IEEE Workshop on.* – P. 1 - 6. IEEE (2008).

[8] Guo, G., Mu, G., Fu, Y., and Huang, T.-S. Human age estimation using bio-inspired features. *In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on* p. 112 - 119. IEEE (2009).

[9] Spizhevoy, A.S., Ogolikhina, A.I., Bovyryn, A.V. Automatic Facial Age Estimation Using Adaptive Brightness Equalization and Biologically Inspired Features. *Vestnik of Lobachevsky State University of Nizhni Novgorod*, Issue 1. – P. 273 - 279 (2014).

[10] Jain, S., Hu, C., and Aggarwal, J.-K. Facial expression recognition with temporal modeling of shapes. *In Computer Vision Workshops (ICCV Workshops), 2011. IEEE International Conference on*, pp. 1642--1649. IEEE (2011).

[11] Lucey, P., Cohn, J.-F., Kanade, T., Saragih, J., Ambadar, Z., and Matthews, I. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. *In Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, p. 94 - 101. IEEE (2010).

[12] Pizer, S.-M., Amburn, E.-P., Austin, J.-D., Cromartie, R., Geselowitz, A., Greer, T., ter Haar-Romeny, B., Zimmerman, J.-B., and Zuiderveld, K. Adaptive histogram equalization and its variations. *Computer vision, graphics, and image processing* 39, 3. – P. 355 - 368 (1987).

[13] Sanin, A., Sanderson, C., Harandi, M.-T., and Lovell, B.-C. Spatio-temporal covariance descriptors for action and gesture recognition. *arXiv preprint arXiv:1303.6021* (2013).

[14] Shan, C., Gong, S., and McOwan, P.-W. Facial expression recognition based on local binary patterns: A comprehensive study. *Image and Vision Computing* 27, 6. – P. 803 - 816. (2009).

[15] Wang, Z., Wang, S., and Ji, Q. Capturing complex spatio-temporal relations among facial muscles for facial expression recognition.

[16] Zhong, L., Liu, Q., Yang, P., Liu, B., Huang, J., and Metaxas, D.-N. Learning active facial patches for expression analysis. *In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, p. 2562 - 2569. IEEE (2012).

Abstract

The STAR on first part presents comprehensive overview of popular applications demands and their mapping to graphics and computing architectures of recent Graphics Processing Units (GPU) and Accelerated Processing Units (APU). Heterogeneous System Architecture (HSA) progress and merge of multicore CPUs with GPU cores in AMD product line is analyzed from potential user point of view. Semiconductor technology progress and power reduction challenges with their influence to graphics and compute architecture evolution have been considered as well.

On the second part we present HSA architecture principles comprehensive overview and demonstrate the example of ASTC texture compression algorithm mapping to modern GPU/APU architecture and OpenCL-HSA software stack. The performance measurements show significant improvement with applied algorithms adjustments and modifications.

Keywords: GPU, CPU, APU, graphics architecture, heterogeneous architecture, compute architecture, texture compression, ASTC.

1. KEY INDUSTRY CHALLENGES FOR GRAPHICS ARCHITECTURE IN 2013-2017

Graphics and computing architecture progress has several inflection points based on different industry branches, media content creation and massive entertainment industry merging with communication and computing domains. We may briefly define programming platforms and application programming interfaces (API) with direct influence to graphics and compute capabilities of modern hardware. Implementation of certain requested functionality hardly depends on semiconductor industry technology progress as well as on general computation technology advances. Power budget reduction for the same application execution is considered to be one of critical features of all new designs in all range from handheld mobile computing to supercomputing in data centers. Extremely high cost of semiconductor manufacturing in small size nodes 14nm and below requires new approaches on system architecture using multichip configurations.

Very important influence also comes from independent software vendors (ISV) developing game engines and visual computing applications. Game and movie content creators always enquire for new visual effects of processing capabilities implemented in both software and hardware levels. Below is listed brief overview of platforms and technology development.

Platforms and APIs:

- OpenGL ES 3.0, OpenGL 4.4, Mantle (AMD), OpenGL (common)
- Windows 8.1 with DirectX 11.2, Windows 2015 “Threshold” with DirectX 12 and SVM lite support, Windows 2017 with DirectX 13 and full SVM support,
- OpenCL 2.0 (2014-15), OpenCL 2.1 (2015-16) and OpenCL 3.0 (2016-17)

Technology major trends:

- Interposer technology including advanced semiconductor and systems packaging with interposing on silicon (organic and glass as well)
- Using HBM or High Bandwidth Memory in GPU, CPU and APU
- Sequential transitions to 20 nm (2014), 14 nm (2016) and 10 nm (2017+) semiconductor manufacturing processes on foundries
- Virtual page migration (2014-15), Low power HSA-based DSP (2015)
- Chiplets (tiny chips) combined on multichip module (MCM)

Major CPU/GPU and SoC vendors develop their product roadmap responding to challenges in platforms and technology:

AMD response on product line (public info limited by 2015):

- Discrete GPUs: Bonaire and Hainan (28nm/2013), Hawaii (28nm/2013-2014), Tonga and Iceland(28nm/2014), Bermuda and Fiji (28nm/2015)
- APUs: Kabini (28nm/2013), Kaveri and Mullins(28nm/2014), Carrizo (28nm/2015) and Amur/Nolan (20nm/2015).

Intel's response on product line (public info limited by 2016):

- CPU-GPU SoC: Haswell and Silvermont(22nm/2013), Broadwell and Airmont (14nm/2014), Braswell and Goldmont (14nm/2015), Cannonlake (10nm/2016).

Nvidia's response on product line:

- DGPU Kepler II(GK11x) (28nm/2013), Maxwell (28nm/2014) and project Denver with custom ARM 64-bit core (28nm/2014)
- Mobile SoC and application processors: Logan (28nm/2013), Tegra K1 (28nm/2014) Tegra M1 (20nm/2015).

2. SOFTWARE VENDORS VISIBLE CHALLENGES

Graphics ISVs traditionally have their own set of requests and challenges which may enable new applications. We may consider following list which can be complemented any time by new ideas:

1. Virtual Reality Holographic Rendering for head-mounted displays (HMD).
2. Global illumination rendering in real time.
3. Decoupled shading to process highly detailed scenes.
4. Object+texture space combined memory hierarchy.

VR for HMD generates a lot of attention, like product of Oculus Rift and Valve startup companies. They have significantly higher requirements for processing speed due to zero latency tolerance problem. Head movement demands smooth and soft image update, visible to eye and not causing movement artifacts. It requires high refresh rates and high resolution stereo image generation comparing to existing game consoles. In addition it requires image warp and post rendering to account for simulated lens optics. Next few years will be spent to find optimized solutions for HMD VR image generation. It may require significant computational power growth for GPU

considering high resolution frame rate doubled or tripled versus latest game consoles.

Global illumination is one of favorite applications applied by researchers to GPU since they become more programmable in mid-2000s. Traditional local illumination shading and texturing algorithms implementation in popular game titles have been used as architecture optimization anchor while leaving serious capabilities for general computing which support global illumination models. We may group target applications and with their influence to architecture specifications.

First group with dense stream compute pattern (more suitable for GPU cores):

- 3D graphics in games and engineering
- High performance libraries for compute problems suitable for GPU acceleration

Second group with sparse compute pattern (more suitable for CPU/ latency optimizing system)

- Compiled OpenCL/C++ code for sparse problems
- Ray tracers for global illumination
- Other *Khronos* group platforms and based applications

Third group with special signal and image processing pattern (more suitable for DSP cores + fixed function blocks)

- Media processing

Such application groups create different vectors on architecture trends and are challenging designers to create computing machines which may fulfill several requirements. Some of them may look quite opposite and generate several issues on architecture optimization. Modern complex Systems-on-Chip (SoC) with different types of processing cores could be potential platforms. But simply putting together on the piece of silicon multiple cores does not solve the problem of programmability; even it makes the problem worse. New architecture concept of Heterogeneous System Architecture (HSA) could solve potential problem of creating multipurpose and power/cost efficient computing machines.

3. INTRODUCTION TO HETEROGENEOUS SYSTEM ARCHITECTURE (HSA)

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

target system configuration with or without GPU, DSP, video hardware and other types of specialized compute accelerators.

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

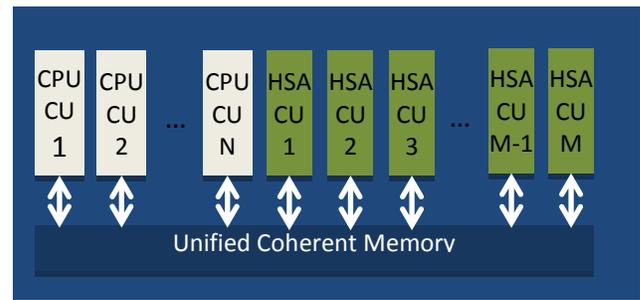


Figure 1: Generic HSA Accelerated Processing Unit (APU)

4. HSA OVERVIEW

Essential HSA features include:

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

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

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

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

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

Coherent memory regions. In traditional GPU devices, even when the CPU and GPU are using the same system memory region, the GPU uses a separate address space from the CPU, and the graphics driver must flush and invalidate GPU caches at required intervals in order for the CPU and GPU to share results.

HSA embraces a fully coherent shared memory model, with unified addressing. This provides programmers with the same coherent memory model that they enjoy on SMP CPU systems. This enables developers to write applications that closely couple CPU and GPU CU codes in popular design patterns like producer-consumer. The coherent memory heap is the default heap on HSA and is always present. Implementations may also provide a non-coherent heap for advance programmers to request when they know there is no sharing between processor types.

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

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

HSA architecture example platform is depicted on Figure 2.

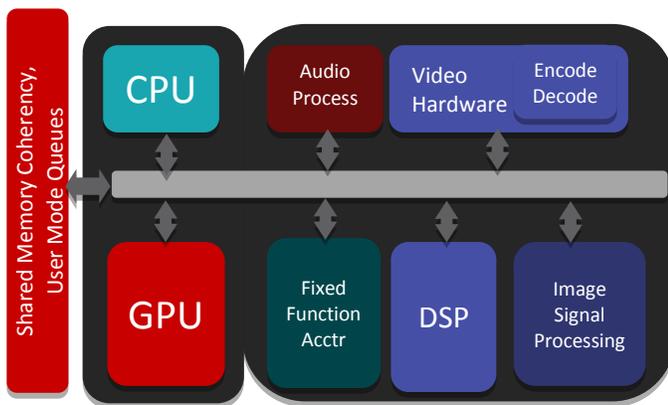


Figure 2: HSA architecture example platform.

5. HSA IMPLEMENTATION AND CONCEPTS

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

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

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

HSA provides a programming interface containing queue and notification functions. This interface allows devices to access load-balancing and device-scaling facilities provided by the higher-level task queuing library. The overall goal is to allow developers to leverage both CPU and CU devices by writing in task-parallel languages, like the ones they use today for multicore CPU systems. HSA's goal is to enable existing task and data-parallel languages and APIs and enable their natural evolution without requiring the programmer to learn a new HSA-specific programming language. The programmer is not tied to a single language, but rather has available a world of possibilities that can be leveraged from the ecosystem.

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

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

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

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

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

6. ASTC OVERVIEW

As an example, we have modified ASTC compression algorithm to utilize HSA features. ASTC is a modern texture compression format developed by ARM and AMD. As the other texture compression formats, it aims on reducing requirements to both memory size and bandwidth. In such case, textures are stored in memory and transferred to GPU in a compressed form. Unpacking only occurs inside GPU, usually between L1\$ and L2\$ caches. Such approach also reduces power consumption, because overall GPU↔VRAM traffic could be directly converted to power consumption. This is especially important for mobile devices such as notebooks, tablet PCs and smartphones.

Texture access pattern is highly random and texture access time is a critical factor affecting the overall performance of the graphics subsystem. Because of it, most of the texture compression schemes provide fixed rate compression and decoders are implemented in hardware. This, in turn, obviously means lossy compression. Therefore, almost all known texture compressors are block based: an image is divided into blocks of a small size and each block gets compressed and accessed independently.

The ASTC format offers an unusual degree of flexibility and supports both 2D and 3D textures, at both standard (LDR) and high dynamic range (HDR), while providing better image quality than most formats in common use today. It also provides a rich set

of compression bit rates from eight bits per pixel down to less than one bit per pixel in very fine steps. ASTC has fixed block size of 128 bits and supports 2D tiles from 4x4 to 12x12 pixels.

The very basic idea is as following: interpolation weights and up to four color endpoint pairs are stored in a compressed block, the decoder picks one color pair for every pixel and blends colors using interpolation weight to produce output color. Color and weight data could be encoded using various modes. Moreover, block layout and special bounded integer sequence encoding (BISE) allows flexible allocation of bits between different types of information. Still, decoding is rather efficient and fast.

However, that is not true for encoding. Achieving decent quality at a reasonable speed is a non-trivial task, because of format complexity. Currently there is only one ASTC encoder – ASTC Evaluation Codec which is a part of Mali Texture Compression Tool written by ARM. Our implementation is based on this codec.

Base algorithm tries to find the best colors and weights encoding for every possible block mode. It stops searching once a compression error for current block gets lower than the error limit. For the performance reasons, heuristics and early exit conditions are heavily used. There are predefined speed/quality settings named veryfast, fast, medium, thorough and exhaustive, which limits search space. Compression times for the test texture¹ are shown in Table 1.

Quality Settings	Peak signal-to-noise ratio (PSNR), dB	Compression time, seconds
Veryfast	41.931500	0.8
Fast	44.712035	1.9
Medium	45.716011	12.2
Thorough	46.072663	47.1
Exhaustive	46.203190	109.3

Table 1: Compression times for different speed/quality settings (Mali TCT 4.2, APU – AMD A10-7850K²)

The thorough and exhaustive settings are of a particular interest, because high quality modes are most demanded in 3D content development.

7. MAPPING ASTC TO HSA

As well as other block compression schemes; the ASTC could be easily parallelized on a block level (which in fact is done in ASTC Evaluation Codec). However, moving entire algorithm to a GPU would be very inefficient, because of high threads divergence caused by early exit condition heuristics and branches. Nevertheless, some computation steps can be performed in parallel.

So compression of a single block consists of a sequence of stages, where some stages could be effectively implemented on GPU cores. Schematically it is shown on Fig. 3, where boxes represent parallel steps and circles represent sequential steps of compression of a single block.

Figure 3: Compression stages for a single image block

By observing source code and profiling original codec, we have chosen three candidates for GPU offloading, which was implemented in OpenCL kernels:

- realign_weights()
- find_best_partitionings()
- compute_angular_endpoints()

Still, parallel parts of a single block compression process cannot create reasonable load level for a single GPU core. Therefore, image blocks could be compressed in batches: CPU thread executes sequential stages for a batch of blocks and prepares data for GPU. Schematically this approach is depicted on Fig. 4.

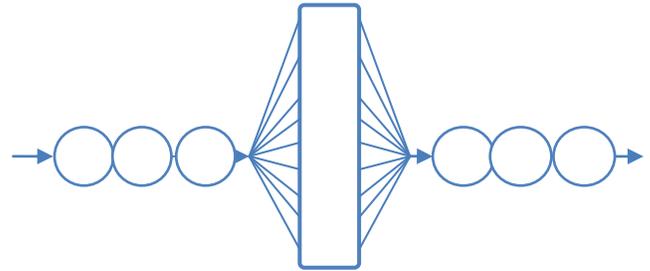


Figure 4: Compression stages for a batch compression

Our experiments show that a batch size of 512-1024 blocks provides reasonable tradeoff between memory consumption and better GPU utilization.

Offloading some work to GPU cores could increase overall performance in two ways:

- Executing parallel stages on GPU core is often more efficient in terms of time and power.
- CPU threads could process another batch while waiting results from GPU.

However, traditional GPGPU approach with discrete CPU and GPU devices faces following restrictions:

- Data should be transferred between CPU and GPU over PCIe, which has much lower bandwidth than RAM or video framebuffer. Coping time for small tasks is comparable with execution time. Sparse access to large buffers is also causes difficulties. It is possible to directly access such buffers from GPU over PCIe without copying data. However, all that host memory should be prepinned, even if many pages will never be used. It causes OS overhead and may lead to unnecessary paging activity.
- High number of kernel invocations results in high driver overhead.

In contrast, HSA platform lacks such restrictions by providing features such as hardware scheduling, user-level command queuing and coherent shared virtual memory. The last one is also greatly simplifies acceleration of existing applications.

Another approach we have used to accelerate ASTC encoding is JIT (just in time) compilation. Some compression parameters, such as tile size and searching limits, are constant for a chosen

¹ Sample texture turret_diffuse_map.png (512x512 pixels) from the Mali TCT 4.2 was used in all tests.

² AMD APU A10-7850K – 4 cores @3.7 GHz, 8 GPU cores @720 MHz



image and quality settings. As OpenCL kernels are naturally compiled at a runtime, we are able to replace such variables with macro definitions and pass actual values at runtime. This allows OpenCL compiler to make additional optimizations, reduce binary size and register pressure. It also helps increasing kernel occupancy level, which in turns allows better hiding memory latency. Results for one of the implemented kernels are shown in Table 2.

	Static compilation	JIT compilation
Binary size	24124 bytes	7628 bytes
Scalar GPRs	76	50
Vector GPRs	65	36

Table 2: **Comparing static and JIT compilation**

8. RESULTS AND FUTURE WORK

As a proof of concept we have implemented HSA accelerated ASTC encoding for LDR images without alpha channel. Currently, HSA software stack remains in development state, so some features are not yet available or optimized. Still the results (Table 3) are rather promising: HSA provides up to 5x speedup.

Quality Settings	Compression time in seconds		Speedup
	Original codec	HSA accelerated codec	
Medium	12.2	3.4	3.59x
Thorough	47.1	10.6	4.44x
Exhaustive	109.3	21.3	5.13x

Table 3: **Comparing compression times for original and modified codecs**

Note that original codec (Mali TCT 4.2) goes with 32bit binary. Our implementation was compiled for x64 target and also benefits from larger register file and SIMD instructions.

Currently CPU threads just wait while GPU executes kernels. It results in overall CPU utilization of 60-90%. Therefore, there is a lot of room for increasing performance even further by implementing dynamic load balancing between CPU and GPU cores. HSA profiling and instrumentation tool progress may give a chance to use heterogeneous cores more efficiently providing better load balance between GPU and CPU cores.

9. CONCLUSION

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

HSA is a unified computing framework. It provides a single address space accessible to both CPU and GPU (to avoid data copying), user-space queuing (to minimize communication overhead), and preemptive context switching (for better quality of service) across all computing elements in the system. HSA unifies CPUs and GPU/DSPs into a single system with common computing concepts, allowing the developer to solve a greater variety of complex problems more easily. There is a long way ahead on migration of classic sequential programming algorithms and tasks to HSA platforms to provide power/cost efficient computing in several domains of human activity.

10. REFERENCES

- [1] Heterogeneous System Architecture: A Technical Review, Advanced Micro Devices, Rev. 1.0.
- [2] <http://developer.amd.com/resources/heterogeneous-computing/>
- [3] <http://malideveloper.arm.com/develop-for-mali/tools/astc-evaluation-codec/>
- [4] Paltashev T.T., Perminov I.V., Texture compression techniques, Scientific Visualization, National Research Nuclear University "MEPhI". - 2014. - Т. 6, вып. 2014-1. - С. 96-136. - ISSN 2079-3537

About the authors

Timour Paltashev is Senior Manager of Graphics IP Engineering at Advanced Micro Devices and Professor at Northwestern Polytechnic University, College of Engineering. His contact email is [timpal\(at\)mail.npu.edu](mailto:timpal(at)mail.npu.edu).

Ilya Perminov is a PhD candidate at St. Petersburg National Research University of Information Technologies, Mechanics and Optics. His contact email is [studentikispam\(at\)gmail.com](mailto:studentikispam(at)gmail.com)