ISSN 0132-3474

Номер 3

Май - Июнь 2024

ПРОГРАММИРОВАНИЕ

НАУКА
1727

СОДЕРЖАНИЕ

Номер 3, 2024

КОМПЬЮТЕРНАЯ ГРАФИКА И ВИЗУАЛИЗАЦИЯ

Особенности разработки бортовой системы визуализации для гражданских воздушных судов

Б. Х. Барладян, Н. Б. Дерябин, А. Г. Волобой, В. А. Галактионов, Л. З. Шапиро, И. В. Валиев, Ю. А. Солоделов 3

Автоматическая стилизация изображений по расширенной коллекции стилей

В. В. Понамарев, В. В. Китов 14

Влияние неравносторонних апертур цифровых фильтров лапласиана “усеченная пирамида” и “двойная пирамида” на точность работы телевизионных измерительных систем

К. А. Рылов, К. С. Куприянова, А. В. Каменский 21

Преобразование данных облака точек в 3D-модели с помощью PointNet++ и Transformer

М. И. Сорокин, Д. Д. Жданов, А. Д. Жданов 33

Метод упорядочивания облаков точек для визуализации на конвейере трассировки лучей

П. Ю. Тимохин, М. В. Михайлюк 42

Двойное представление геометрии для ускорения трассировки лучей в оптических системах с поверхностями свободной формы

Д. Д. Жданов, И. С. Потемин, А. Д. Жданов 54

АНАЛИЗ ДАННЫХ

Нейросетевой метод детектирования размытия на гистологических изображениях

Г. С. Назаренко, А. С. Крылов 67

Совместное повышение разрешения и классификация фрагментов ткани для полнослайдовых гистологических изображений

Чж. Сун, А. В. Хвостиков, А. С. Крылов, А. Сетхи, И. А. Михайлов, П. Г. Мальков 75

Contents

No. 3, 2024

COMPUTER GRAFICS AND VISUALIZATION

Specifics of the Development of an On-Board Visualization System for Civil Aircrafts

B. Kh. Barladian, N. B. Deryabin, A. G. Voloboy, V. A. Galaktionov, L. Z. Shapiro, I. V. Valiev, Yu. A. Solodelov 3

Automatic Image Style Transfer Using an Augmented Style Set

V. V. Ponamaryov, V. V. Kitov 14

Influence of Unequilateral Apertures of Digital Filters Laplacian "Trunced pyramid" and "Double pyramid" on the Accuracy of Television Measuring Systems

K. A. Rylov, K. S. Kupriyanova, A. V. Kamensky 21

Conversion of Point Cloud Data to 3D Models Using PointNet++ and Transformer

M. I. Sorokin, D. D. Zhdanov, A. D. Zhdanov 33

A Method to Order Point Clouds for Visualization on the Ray Tracing Pipeline

P. Y. Timokhin, M. V. Mikhaylyuk 42

Dual Representation of Geometry for Ray Tracing Acceleration in Optical Systems with Freeform Surfaces

D. D. Zhdanov, I. S. Potemin, A. D. Zhdanov 54

DATA ATALISYS

Neural Network Method for Detecting Blur in Histological Images

G. S. Nazarenko, A. S. Krylov 67

Joint Super-resolution and Tissue Patch Classification for whole Slide Histological Images

Zh. Sun, A. Khvostikov, A. Krylov, A. Sethi, I. Mikhailov, P. Malkov 75

ПРОГРАММИРОВАНИЕ, 2024, № 3, с. 3–13

КОМПЬЮТЕРНАЯ ГРАФИКА И ВИЗУАЛИЗАЦИЯ

УДК 004.92

ОСОБЕННОСТИ РАЗРАБОТКИ БОРТОВОЙ СИСТЕМЫ ВИЗУАЛИЗАЦИИ ДЛЯ ГРАЖДАНСКИХ ВОЗДУШНЫХ СУДОВ

© 2024 г. Б. Х. Барладян[a], *, Н. Б. Дерябин[a], **, А. Г. Волобой[a], *, В. А. Галактионов[a], ****, Л. З. Шапиро[a], *****, И. В. Валиев[a], ******, Ю. А. Солоделов[b], *******

[a] *Институт прикладной математики им. М.В. Келдыша РАН*
125047 Москва, Миусская пл., д. 4, Россия
[b] *Государственный научно-исследовательский институт авиационных систем*
125319 Москва, ул. Викторенко, д. 7, Россия
**E-mail: bbarladian@gmail.com*
***E-mail: dek@keldysh.ru*
****E-mail: voloboy@gin.keldysh.ru*
*****E-mail: vlgal@gin.keldysh.ru*
******E-mail: pls@gin.keldysh.ru*
*******E-mail: piv@gin.keldysh.ru*
********E-mail: yasolodelov@2100.gosniias.ru*

Поступила в редакцию: 19.01.2024 г.
После доработки: 19.01.2024 г.
Принята к публикации: 24.01.2024 г.

Приборные панели современных самолетов создаются по концепции “стеклянной кабины”. Эта новая идеология интерфейса позволяет улучшить восприятие важной полетной информации за счет отображения ее на одном многофункциональном дисплее. В работе рассматриваются проблемы, возникающие при разработке сертифицируемой системы визуализации дисплея пилота, предназначенной для работы на гражданских воздушных судах под управлением российской операционной системы реального времени JetOS. В статье приведено несколько алгоритмических решений, позволяющих добиться приемлемой скорости визуализации. В частности, подробно описано решение проблемы жесткого расписания разделов операционной системы, благодаря которому удалось преодолеть деградацию скорости визуализации. Намечены пути дальнейших работ.

Ключевые слова: дисплей в кабине, бортовая система визуализации, OpenGL SC, RTOS, авионика, драйвер графического процессора

DOI: 10.31857/S0132347424030018, **EDN:** QBCOYU

1. ВВЕДЕНИЕ

Комплексы бортового оборудования современных гражданских авиалайнеров относятся к системам, критическим с точки зрения безопасности (safety critical). Они работают под управлением операционной системы реального времени (ОСРВ), необходимость использования которой вытекает из современной концепции построения авиационных систем “Интегрированная модульная авионика” [1]. Ключевой особенностью этой концепции является выполнение нескольких функциональных приложений, реализующих программное обеспечение той или иной самолетной системы, на одном процессоре. Приложения при этом должны быть разделены по времени исполнения, вытесняющая многозадачность запрещена. Такой режим работы приложений и обеспечивается ОСРВ [2, 3].

Каждое разрабатываемое гражданское воздушное судно и его комплектующие должны проходить сертификацию, что является обязательным требованием для всех систем и продуктов, критических с точки зрения безопасности. Для сертификации программного обеспечения бортового оборудования воздушного судна существует стандарт DO-178C [4], определяющий, в частности, требования к процессу разработки ПО. Эта специфика не позволяет применять готовые, известные решения для той или другой функциональности. Программный интерфейс ОСРВ, предназначенный для использования на борту

воздушного судна, должен соответствовать стандарту ARINC653 [5].

До настоящего времени зарубежные ОСРВ, такие как VxWorks 653 или Thales MACS2, применялись в качестве бортовой операционной системы на разрабатываемых отечественных воздушных судах. Но в последнее время использование зарубежных программных продуктов сталкивается с серьезными проблемами или даже становится невозможным. Это послужило причиной разработки отечественной операционной системы реального времени (ОСРВ) JetOS [3]. Процесс разработки JetOS идет в соответствии с DO-178C, и также она соответствует стандарту ARINC653 для бортовых авиационных операционных систем. Так как бортовые системы для различных самолетов строятся на разных вычислительных платформах (процессорах и соответствующих GPU), то операционная система должна быть адаптирована и сертифицирована под каждую из них.

При разработке кабин современных самолетов существует тенденция использовать большие дисплеи. Эта концепция называется “стеклянная кабина”. На один дисплей можно выводить различную информацию о полетной навигации и состоянии оборудования самолета. На рис. 1 показана приборная панель МС-21, разработанная в соответствии с этой концепцией.

Таким образом, разработка системы визуализации дисплеев пилота воздушного судна является неотъемлемой частью разработки бортовой ОСРВ. Поэтому процесс ее разработки также должен соответствовать авиационным стандартам DO-178C и ARINC653. При этом задача осложняется высокими требованиями к надежности визуализации и невысокой производительностью процессоров, так как на борт обычно ставятся энергосберегающие платформы, имеющие историю надежного использования.

Статья организована следующим образом. Раздел 2 представляет архитектуру бортовой системы визуализации и особенности двух стандартов OpenGL SC, предписанных для использования в авиации. В разделе 3 рассказывается о создании чисто программной реализации системы визуализации дисплея пилота. В разделе 4 показаны особенности реализации графической компоненты с аппаратным ускорением. Раздел 5 посвящен решению проблемы замедления визуализации из-за жесткого расписания разделов операционной системы. Раздел 6 содержит заключение.

Рис. 1. Приборная панель самолета МС-21.

2. СИСТЕМА ВИЗУАЛИЗАЦИИ В КАБИНЕ ПИЛОТА

Одним из важных требований к графической компоненте ОСРВ является использование специальной версии графической библиотеки OpenGL SC (SC – safety critical), предназначенной для систем, критических с точки зрения безопасности. В настоящее время библиотека OpenGL SC представлена двумя версиями стандарта: 1.0.1 и 2.0. Стандарт OpenGL SC1.0.1 является подмножеством стандарта OpenGL 1.3.

Стандарт OpenGL SC2.0 основан на стандарте OpenGL ES2.0, но не является его подмножеством. Он разработан для графического оборудования, работающего на встроенных платформах. В нем удалены те аспекты OpenGL ES2.0, которые не соответствуют детерминированной безопасности для критических по безопасности программных приложений. В частности, удален встроенный компилятор шейдеров. Предполагается, что вершинный (vertex) и фрагментный (fragment) шейдеры отдельно компилируются и связываются с двоичным объектом программы, который генерирует исполняемый код из скомпилированных шейдеров. При этом в стандарте не оговорен способ создания и интерфейс двоичного объекта программы. Стандартизован только интерфейс функции ProgramBinary(), которая позволяет загрузить предварительно скомпилированный двоичный объект.

Стандарты OpenGL SC2.0 и OpenGL SC1.0.1 значительно отличаются. Некоторые функции версии 1.0.1 не существуют в версии 2.0 и могут быть заменены шейдерными программами. Можно сказать, что в OpenGL SC2.0 упор делается на программируемый конвейер трехмерной графики, а в OpenGL SC1.0.1 – на фиксированную функциональность.

В общем виде систему визуализации в кабине пилота можно представить в виде схемы (рис. 2).

Графическая библиотека, написанная по одному из стандартов OpenGL SC, генерирует изображения для каждого кадра и выводит их на экран, используя драйвер дисплея.

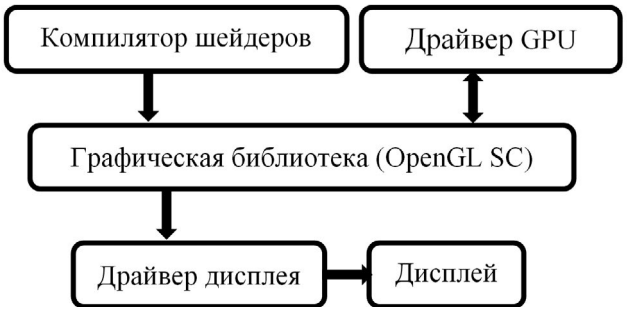


Рис. 2. Схема визуализации в кабине самолета.

При разработке программной реализации графической библиотеки, т. е. реализации, не использующей аппаратное ускорение конкретной вычислительной платформы, только драйвер дисплея должен учитывать специфику конкретного GPU. Программная версия графической библиотеки была разработана на языке C. Ее перенос на любую платформу не должен вызывать существенных затруднений.

Для использования аппаратной поддержки графического процессора для каждой вычислительной платформы необходимо разрабатывать драйвер GPU, который обеспечит доступ графической библиотеки к необходимым ресурсам графического ускорителя.

Для работы с приложениями, написанными по стандарту OpenGL SC2.0, требуется внешний компилятор шейдеров, поскольку стандарт OpenGL SC2.0 предполагает загрузку шейдеров в виде уже скомпилированного двоичного объекта.

3. ПРОГРАММНАЯ ВЕРСИЯ БИБЛИОТЕКИ OPENGL SC1.0.1

Как уже было сказано, для программной реализации графической библиотеки только драйвер дисплея должен учитывать специфику конкретной вычислительной платформы.

Разумеется, программная реализация библиотеки значительно уступает по производительности библиотеке, реализованной с использованием аппаратного ускорения. Однако у нее есть два существенных преимущества. Во-первых, сама программная версия OpenGL является платформонезависимой, а значит, может быть относительно легко перенесена на любую выбранную вычислительную платформу. Во-вторых, подготовка сертификационного пакета для программной реализации библиотеки не вызывает существенных проблем, поскольку весь код доступен и написан на языке C, а процесс его создания может быть проведен в соответствии с DO-178C.

При создании программной версии библиотеки OpenGL SC были разработаны новые алгоритмы, ускоряющие визуализацию в несколько раз. Далее, для повышения производительности мы распараллелили работу библиотеки на многоядерных процессорах, используя специальное расширение стандарта ARINC653, реализованное в JetOS. Расширение называется Asymmetric Multi-Processing (AMP). Это позволило получить дополнительное ускорение в 2.5–3 раза для типичных авиационных приложений, используя многоядерные процессоры и не отклоняясь от разрешенных стандартов. Реализация программной версии OpenGL SC1.0.1 подробно описана в работах [6, 7].

Но работа программной реализации OpenGL заканчивается генерацией изображения кадра в памяти процессора, а для вывода на экран дисплея еще необходима библиотека фрейм буфера, без которого библиотека просто не может быть использована в реальном бортовом оборудовании. И если сама программная реализация OpenGL является платформонезависимой, то драйвер дисплея, естественно, зависит от используемого оборудования. Тем не менее эта реализация существенно проще, чем полноценного драйвера GPU. Например, для платформы i.MX6 нам удалось реализовать такой драйвер с минимальным использованием кода, специфичного для GPU.

При разработке графической компоненты бортовой ОСРВ важно понимать, с какими данными она работает и какие требования предъявляются к ней. Визуализируемые приложения – это не произвольные виртуальные сцены, а типовые приложения, используемые в авиации. Главным из них является Primary flight display (PFD). На рис. 3 приведен PFD для Sukhoi Superjet 100, а на рис. 4 – для МС-21. Примером других приложений может быть индикатор состояния дверей Doors (рис. 5).

Так как типичные авиационные приложения представляют визуализацию состояния приборов и датчиков, то минимально приемлемая скорость визуализации может быть ниже, чем для обычной анимации. Уже при 15 кадрах в секунду данные достаточно оперативно отображаются на экране, картинка выглядит вполне приемлемо. Для оптимизированной программной версии библиотеки при использовании четырех ядер процессора удалось получить скорость ~14 кадров в секунду для наиболее сложного приложения SS_PFD. Остальные доступные приложения показали более высокую скорость (детали приведены в табл. 1).

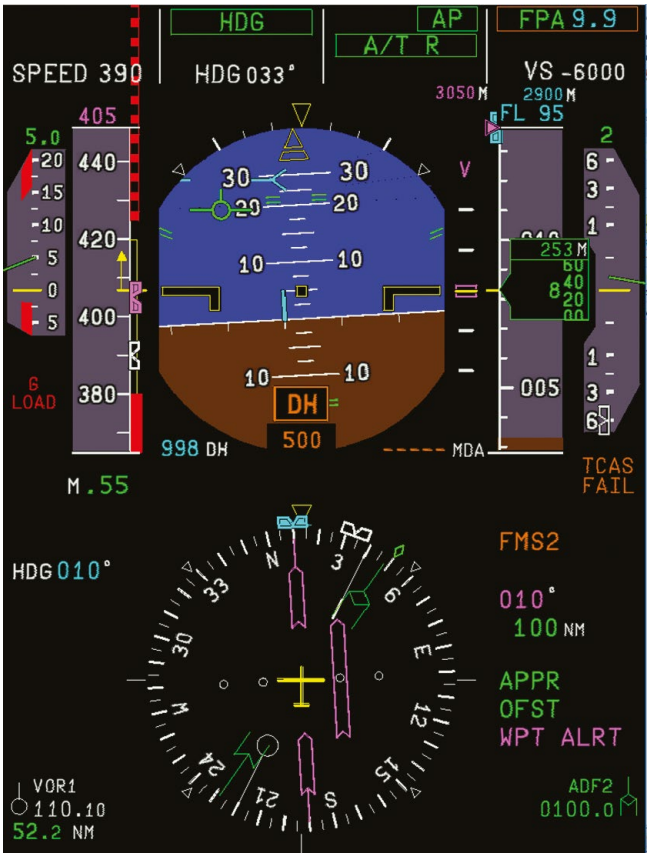


Рис. 3. Primary Flight Display SS_PFD.

Таблица 1. Скорость визуализации различных реализаций OpenGL SC в кадрах в секунду

Приложение	SWGL1	SWGL4	HWGL
SS_PFD (рис. 3)	5.9	13.8	10.8
MC_PFD (рис. 4)	6.3	15.6	20.0
Doors (рис. 5)	12.0	34.7	60
Doors + PDF (рис. 6)	4.6	6.2	16.5

Использование больших дисплеев в кабине пилота подразумевает их многооконность, чтобы различная информация была оперативно доступна пилоту. Реализация многооконной визуализации с использованием программной версии OpenGL SC позволила получить скорость 6.2 кадра в секунду для приложений (рис. 6). Детали многооконной визуализации описаны в [7].

Разработанная нами программная реализация библиотеки OpenGL SC1.0.1 при использовании многоядерности процессора оказалась достаточно эффективной для многих практических авиационных приложений, но не для всех. Кроме того, в ряде случаев возникает необходимость использования ядер процессора для других компонент комплекса бортового оборудования. В связи

Рис. 4. Primary Flight Display MC_PFD.

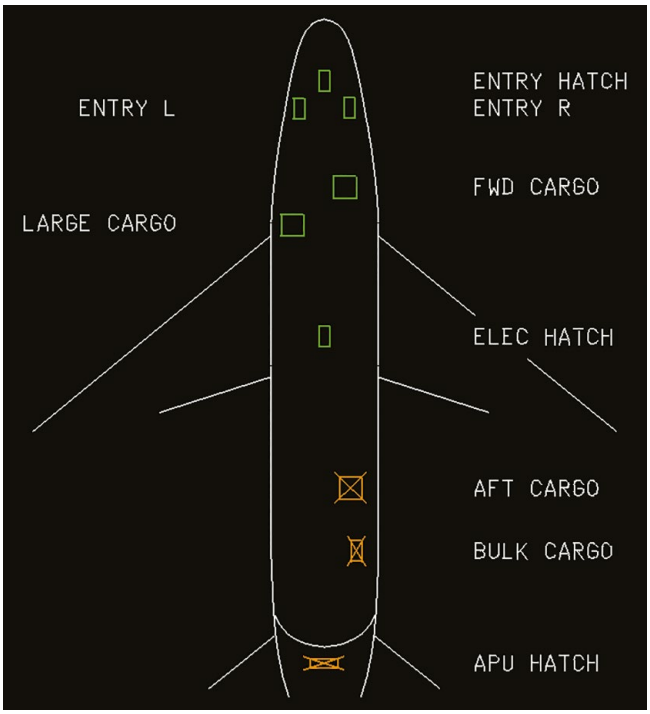


Рис. 5. Индикатор состояния дверей Doors.

с этим вопрос использования GPU для ускорения визуализации остается актуальным, тем более, что именно для этих целей GPU в основном и разрабатывались.

4. РЕАЛИЗАЦИЯ БИБЛИОТЕКИ OPENGL SC С АППАРАТНЫМ УСКОРЕНИЕМ

Реализация различных версий OpenGL SC на базе коммерческих драйверов GPU рассматривается в работах [8–10]. Однако необходимость разработки сертификационного пакета для графической компоненты в соответствии с требованиями КТ-178С требует доступа ко всем ее кодам. В большинстве случаев драйверы, предоставляемые производителями GPU, не имеют открытого

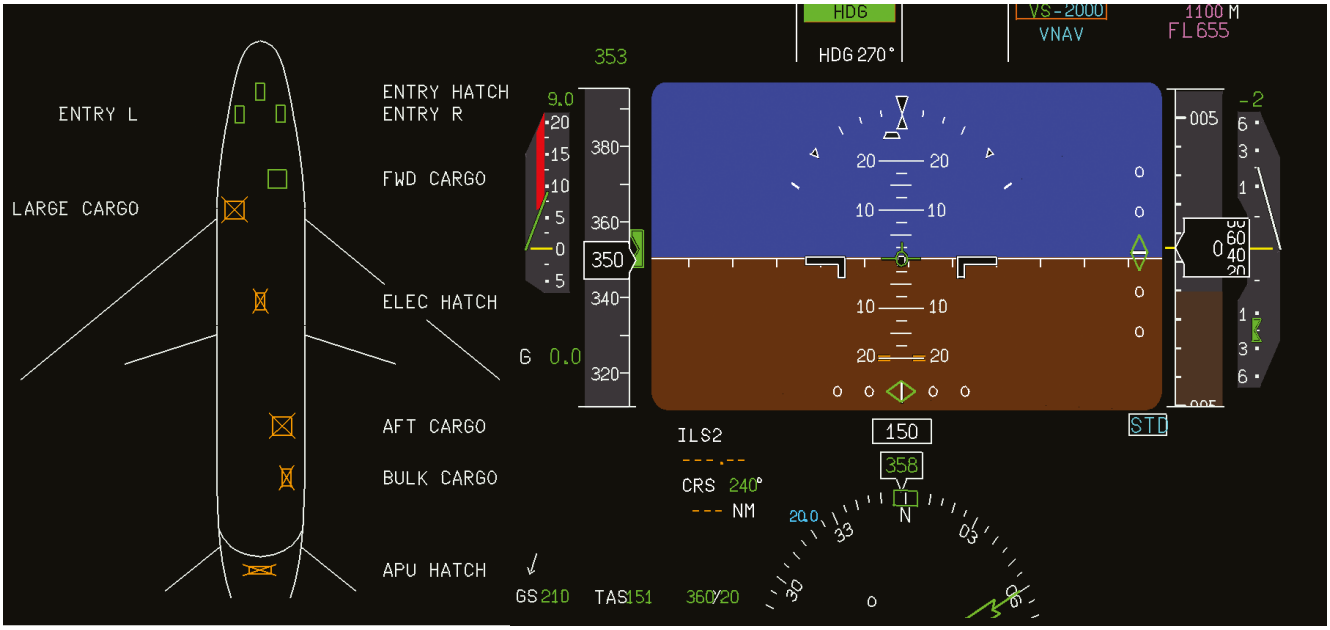


Рис. 6. Многооконная визуализация Doors и PFD.

программного кода, поэтому такое ПО не может быть сертифицировано для использования в авиации. Также драйвер необходимо адаптировать для работы под управлением ОСРВ JetOS. Поэтому наша разработка базировалась на пакете Mesa 3D [11].

Mesa 3D представляет собой программную реализацию с открытым исходным кодом различных спецификаций графического API, включая OpenGL, Vulkan и др. Mesa 3D переводит эти спецификации в драйверы конкретного графического оборудования. Некоторые производители процессоров, такие как, например, AMD или Intel, сами разрабатывают драйверы для этого пакета с открытым кодом для производимых ими процессоров. Другие производители, такие как Nvidia или Vivante, полностью заменяют драйверы в Mesa 3D, обеспечивая собственную реализацию библиотеки OpenGL. Для такого оборудования сообщество разработчиков Mesa создает альтернативные открытые драйверы методом обратной разработки (reverse engineering), такие как Nouveau для Nvidia или Etnaviv для Vivante. При использовании платформы i.MX6 с графическим процессором Vivante единственной возможностью разработки сертифицируемой библиотеки OpenGL SC является разработка драйвера на основе Etnaviv.

Детально процесс реализации графической библиотеки OpenGL SC с аппаратной поддержкой на базе пакета Mesa 3D описан в работах [12, 13]. Отметим здесь, что реализация многооконного режима потребовала разработать принципиально другой алгоритм по сравнению с программной версией библиотеки, поскольку при использовании аппаратной поддержки может быть использован только один экземпляр OpenGL. Аппаратная поддержка GPU позволила в большинстве случаев увеличить скорость визуализации и для стандартных приложений, и для многооконной визуализации (см. табл. 1). В таблице SWGL1 – программная версия библиотеки при использовании одного ядра процессора, SWGL4 – программная версия библиотеки при использовании четырех ядер процессора, HWGL – версия библиотеки с аппаратной поддержкой.

В реальности при распараллеливании графической компоненты с программной версией OpenGL SC в многооконном режиме было задействовано только три ядра процессора: два ядра для каждой составляющей изображения (Doors и PFD) и одно ядро для компоновщика всего изображения.

Отдельно следует отметить реализацию графической библиотеки, поддерживающей стандарт OpenGL SC2.0, который в настоящее время является наиболее используемым в авиационных приложениях. Реализация программной версии этого стандарта не имеет практического смысла, поскольку шейдеры в настоящее время эффективно реализованы в современных GPU, а их программная реализация будет неэффективна.

Но для работы с приложениями, написанными по этому стандарту, необходима реализация внешнего компилятора шейдеров, поскольку стандарт OpenGL SC2.0 предполагает загрузку

шейдеров в виде уже скомпилированного двоичного объекта. Код компилятора шейдеров в пакете Mesa 3D написан на языке C++, который практически невозможно сертифицировать по авиационным нормам. Получается, что этот код трудно исключить для стандарта 1.0.1, но относительно просто для стандарта 2.0, поскольку компилятор шейдеров вынесен во внешнюю программу, которая используется только на этапе подготовки приложений. Таким образом, сертификация реализации библиотеки OpenGL SC стандарта 2.0 с аппаратной поддержкой получается значительно проще, чем стандарта 1.0.1.

4.1. Проблемы использования аппаратного ускорения

В некоторых случаях использование аппаратного ускорения может привести не к ускорению, а к замедлению скорости визуализации. Это видно на примере приложения SS_PFD: в табл. 1 скорость визуализации с аппаратной поддержкой ниже, чем чисто программной. Это часто связано с тем, что приложения для бортового оборудования создаются в соответствии со стандартами и с помощью автоматизированных систем, которые не учитывают специфику GPU. Более серьезная проблема возникла (и была решена нами) с приложением, созданным по стандарту ARINC661 [14].

Система отображения в кабине экипажа Cockpit display system (CDS) предоставляет видимую и звуковую информацию о воздушном судне и окружающей среде и получает команды управления от экипажа. Через нее экипаж управляет современным самолетом. Интерфейсы этой системы должны удовлетворять авиационному стандарту ARINC661. Некоторые приложения, взаимодействующие с CDS, создаются с помощью программ автоматизированного проектирования, гарантирующих соответствие этому стандарту [15].

Приложение, представляющее карту движения по аэродрому с взлетно-посадочной полосой и рулежными дорожками (рис. 7), использует сервер ARINC661, разработанный компанией Ansys Inc. [15].

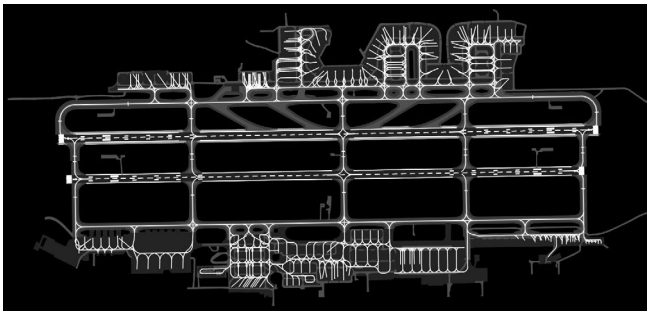


Рис. 7. Карта рулежных дорожек аэродрома.

Приложение оказалась неэффективным с точки зрения использования графической компоненты с аппаратным ускорением. Оно визуализировало каждый отрезок линии и каждый треугольник треугольной сетки отдельным набором команд библиотеки OpenGL, в то время как эффективный интерфейс GPU подразумевает выполнение как можно большого набора команд за одно обращение. В работе [16] мы оптимизировали работу сервера ARINC661 путем добавления в конвейер визуализации препроцессора, который объединяет команды в группы рисования однородных примитивов, отрезков или треугольников. Само приложение при этом не модифицировалось. В результате этой оптимизации визуализация карты аэродрома (см. рис. 7) была ускорена с 1.7 до 15.2 кадров в секунду.

4.2. Подготовка OpenGL SC с аппаратной поддержкой к сертификации

Как уже говорилось, прототип библиотеки был разработан на основе пакета Mesa 3D. В соответствии со стандартом DO-178C в ПО бортового оборудования не должно быть неисполняемого кода. Поэтому одной из важных задач было исключение кода, который не будет использоваться в OpenGL SC. Это сложная задача в силу практического отсутствия документации в пакете Mesa 3D и большого объема сложного кода, который предполагает выполнение под управлением различных операционных систем и использование различных платформ и GPU.

Для этой цели был использован итерационный процесс, на каждом шаге которого достигается частичное снижение сложности и избыточности пакета. Сначала использовались, в основном, формальные методы – это методы, которые не требуют понимания работы кода, а только понимания общих принципов программирования. Простейшим применением формального метода является удаление неиспользуемой функции. Если какая-то функция не используется, ее описание и определение удаляются. После удаления неиспользуемой функции могут появиться другие неиспользуемые функции. В определении их помогает компилятор, который диагностирует большинство неиспользуемых статических функций.

Аналогично, можно удалить неиспользуемую переменную или неиспользуемый член структуры. Переменная не используется, если она нигде не читается. Может оказаться, что переменной

(члену структуры) присваивается только одно значение. В этом случае ее тоже можно удалить, заменив чтение этой переменной ее значением.

Если в процессе адаптации функция стала пустой, ее и ее вызовы можно удалить. Если она стала возвращать единственное значение, то можно заменить вызовы этой функции данным значением. Если в файле не осталось внешних функций и переменных, то он удаляется, и т. д.

Для анализа неиспользуемого кода оказалась полезным использование такой функциональности как получение покрытия кода при работе приложения. Эта функциональность [17] была разработана и обычно применяется при тестировании программного обеспечения. Ее основное назначение – определение насколько хорошо разработанные тесты покрывают код, его функции и ветвления. Использование этой функциональности оказалась эффективно для исключения неиспользуемого кода библиотеки.

Методы, использованные в этом итерационном процессе, подробно описаны в работе [13]. В результате удалось на порядок снизить объем кода.

5. ПРОБЛЕМЫ РЕАЛЬНОГО ВРЕМЕНИ И ЖЕСТКОГО РАСПИСАНИЯ

Каждое программное приложение, работающее на борту гражданского судна, обязано следовать стандарту ARINC653, где прописаны жесткие требования безопасности. Для обеспечения реального времени система должна иметь возможность достаточно часто переключать использование процессора для выполнения разных задач, поэтому невозможно использовать все ресурсы процессора только для графической компоненты. В соответствии со стандартом каждое приложение выполняется в отдельном разделе (partition) ОСРВ. Диспетчеризация работы разделов является строго детерминированной по времени. В зависимости от конфигурации разделов в модуле, общих требований к ресурсам, наличия ресурсов и требований отдельных разделов формируется расписание времени активации разделов, в котором отдельным разделам выделяются их окна. Каждый раздел получает процессорное время в соответствии с этим расписанием.

Для обеспечения реального времени операционной системы необходимо, чтобы активация соответствующего раздела происходила не реже чем через заданный интервал. Как правило, время реакции системы не должно превышать десятков, максимум сотни миллисекунд. Кроме графического приложения, которое отображает поступающую информацию, в системе работают другие приложения. Как минимум должно быть еще одно приложение, которое обрабатывает поступающую информацию и передает ее графическому приложению. Таким образом, длительность окна раздела, в котором выполняется графическое приложение, не может превышать нескольких десятков миллисекунд (желательно не более 20 мс). Следует отметить, что переключение разделов – довольно затратная процедура. Требования по безопасности регламентируют, что при каждом переключении должны быть очищены кэши и регистры, сохранено их содержимое и затем восстановлено при продолжении работы данного раздела. Поэтому полезное процессорное время для каждого приложения меньше, чем время выделенного раздела, как это показано на рис. 8, где полезное время в каждом разделе представлено закрашенным прямоугольником. В результате получается, что при работе дополнительных разделов скорость визуализации снижается.

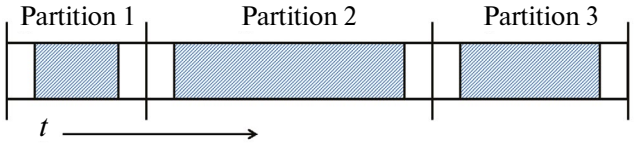


Рис. 8. Расписание выполнения разделов в ОСРВ. Закрашенные прямоугольники соответствуют полезному времени выполнения приложения, пустые прямоугольники в начале и конце каждого раздела – сохранение и восстановление кэшей и регистров.

5.1. Схема работы алгоритма

Для решения этой проблемы мы выделили работу библиотеки OpenGL SC в отдельный раздел ОСРВ (сервер OpenGL). Она выполняется на отдельном ядре процессора с учетом всех ограничений, налагаемых стандартом. Из этого же раздела организовано взаимодействие с GPU. Этот подход является дальнейшим развитием идей, предложенных в работах [13, 16]. В работе [13] драйвер OpenGL, использующий GPU, работал в отдельном разделе и принимал команды от других приложений, но все разделы для различных окон работали на одном ядре процессора. В работе [16] драйвер OpenGL работал в отдельном разделе на отдельном ядре, но при этом использовалось сжатие команд, специфичное для сервера ARINC661, которое будет неэффективно и не всегда возможно в общем случае.

В общем случае разработанный алгоритм позволяет обеспечить эффективную реализацию вывода результатов работы различных

графических приложений на дисплей пилота [13]. Приложения при этом могут работать в различных разделах ОСРВ JetOS и на различных ядрах процессора. В частности, алгоритм обеспечивает эффективную реализацию многооконного режима. В настоящей работе исследовалась только эффективность использования предложенной схемы алгоритма для работы в режиме жесткого расписания.

Мы реализовали драйвер OpenGL общего вида, который работает на отдельном ядре процессора и обрабатывает команды OpenGL, поступающие из графических приложений, которые работают на других ядрах процессора. Подход обеспечивает определенное распараллеливание работы графического приложения. Часть работы (подготовка данных и команд) происходит на одном ядре процессора, а другая часть (сервер OpenGL) – на другом ядре процессора и на GPU. Пока данный кадр визуализируется на GPU, раздел на другом ядре может подготавливать данные и команды для следующего кадра.

Схема работы алгоритма, при котором сервер OpenGL выполняется на отдельном ядре процессора, показана на рис. 9. Основное графическое приложение, использующее для рендеринга библиотеку OpenGL, представлено как Application 1. Application 2 – это дополнительное приложение, которое эмулирует выполнение других необходимых задач параллельно с основным графическим приложением. Оба приложения работают в одном модуле на одном ядре процессора с разделением по времени. Каждое из них реализовано в своем разделе JetOS и работает согласно расписанию. В наших исследованиях это приложение использовалось для моделирования загрузки модуля неграфическими задачами. В тестах оно не выполняло никакой содержательной работы, но в расписании на него отводился определенный квант времени.

5.2. Алгоритм использования сервера OpenGL

Поскольку сервер OpenGL и графическое приложение работают независимо друг от друга на разных ядрах процессора в двух отдельных копиях ОСРВ JetOS, то их работу необходимо синхронизировать. Для простоты изложения мы опустим здесь специфические для JetOS процедуры запуска отдельных копий JetOS и запуска разделов в них. Опишем только передачу данных между графическим приложением и OpenGL сервером и синхронизацию их работы.

Графическое приложение только подготавливает необходимые данные и команды в соответствии с заданным стандартом OpenGL. Мы использовали стандарт OpenGL SC2.0. Для передачи данных и команд между приложением и сервером были реализованы две специальные библиотеки **OGLOUT** и **OGLIN**, которые используют общую память для приложения и сервера. Первая библиотека заменяет (эмулирует) все команды стандарта OpenGL SC2.0 и используется в графическом приложении вместо реальной библиотеки OpenGL. Вместо выполнения соответствующих команд она просто записывает все необходимые данные и идентификаторы команд в специальные массивы, расположенные в общей памяти (Buffer). Библиотека **OGLIN** используется в OpenGL-сервере. С ее помощью сервер читает информацию, записанную в буфере, и последовательно выполняет команды OpenGL, подготовленные графическим приложением.

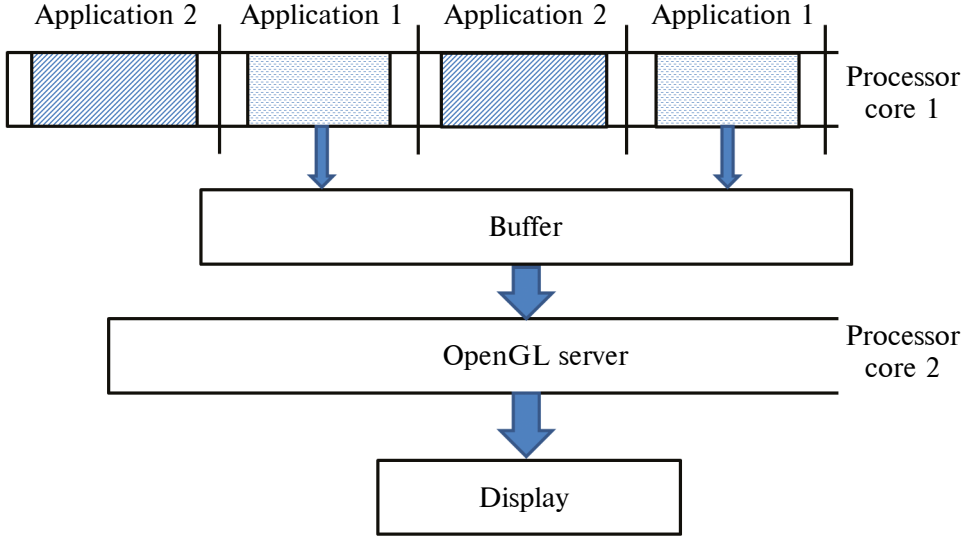


Рис. 9. Схема работы графического приложения в режиме жесткого расписания.

Синхронизация работы приложения и сервера OpenGL выполняется с помощью специальных объектов, называемых событиями, которые реализуются через небольшие общие блоки памяти между модулями. Доступ к этим объектам реализован с помощью атомарных операций (atomic operations). В нашем случае используются два события:

StartRend – устанавливается в сигнальное состояние в приложении в библиотеке **OGLOUT**, когда данные, хранящиеся в общей памяти, готовы к обработке сервером OpenGL, т. е. записаны все данные и команды, необходимые для генерации данного кадра;

EndRend – устанавливается в сигнальное состояние сервером OpenGL, когда закончена обработка заданного фрагмента данных, хранящегося в общей памяти, данный кадр выведен на экран и сервер OpenGL готов обрабатывать следующий кадр.

Первоначально событие **StartRend** устанавливается в несигнальное состояние, а **EndRend** – в сигнальное состояние.

5.3. Результаты тестов

В качестве графического приложения в тестах использовалось два приложения: SS_PFD (см. рис. 3) и более простой прототип PFD. В первом тесте Application 1 и Application 2 работали в одном модуле, но в разных разделах, при этом OpenGL использовалась непосредственно из графического приложения. Длительность окна для работы раздела Application 1 (графическое приложение) была зафиксирована в 10 мс, а длительность окна для работы раздела второго приложения варьировалась от 0 до 20 мс. Длительность 0 означает, что раздел со вторым приложением не использовался. Результаты первого теста для разных вариантов длительности второго, неграфического, приложения приведены в табл. 2.

Из табл. 2 видно, что работа второго приложения в том же модуле с разделением времени с графическим приложением заметно уменьшает скорость визуализации. Степень замедления зависит от расписания и специфики приложения, но в любом случае оно достаточно существенно.

Таблица 2. Скорости визуализации в кадрах в секунду в зависимости от длительности окна второго приложения (одно ядро процессора)

Приложение	0 mc	3 mc	5 mc	10 mc	20 mc
SS_PFD	20	16.5	15	13.3	8.9
proto-pfd	30	27	27	20	13.4

Таблица 3. Скорости визуализации в кадрах в секунду в зависимости от длительности окна второго приложения (с сервером OpenGL)

Приложение	0 mc	30 mc	40 mc	50 mc
SS_PFD	20	20.0	20.0	16.7
proto-pfd	30	30.0	26.8	24.2

Во втором тесте библиотека OpenGL была реализована в виде сервера, как показано на рис. 9. Те же два приложения по-прежнему работали в одном модуле в двух разделах, а библиотека OpenGL – в отдельном модуле на отдельном ядре процессора. Длительность окна для работы раздела графического приложения была также зафиксирована в 10 мс, а длительность окна для работы второго приложения варьировалась от 10 до 50 мс. Результаты второго теста для разных вариантов длительности второго, неграфического, приложения приведены в табл. 3.

Из табл. 3 видно, что работа второго приложения с разделением времени с графическим приложением при использовании сервера OpenGL также уменьшает скорость визуализации, но степень замедления значительно меньше. До длительности раздела второго приложения 30 мс (т. е. в три раза больше, чем длительность раздела графического приложения) замедления практически не видно. Оно начинает проявляться, только если длительность раздела второго приложения существенно превышает время, отводимого в расписании для раздела графического приложения.

6. ЗАКЛЮЧЕНИЕ

Создание бортовой системы визуализации для кабины пилота гражданского воздушного судна является сложной и многофакторной задачей. Главным требованием остается безопасность полетов. Это накладывает принципиальные ограничения, значительно усложняющие создание графической системы. Первое ограничение – это надежная, энергосберегающая вычислительная платформа. На борт воздушного судна не ставят современные и достаточно мощные графические процессоры Nvidia или AMD из-за их высоких энергетических затрат и недостаточной наработки на отказ. Надежность процессоров проверяется со временем, поэтому на борт ставятся не самые новые процессоры, производительность которых невысока по современным меркам. Вторым фактором разработки системы визуализации

является ее использование под операционной системой реального времени, которая во многом отличается от общепринятых ОС Linux или Windows. Сложность создания графической системы также связана с необходимостью ее сертификации как технологии, критической с точки зрения безопасности. Процесс ее создания должен удовлетворять жестким стандартам, предъявляемым к программному обеспечению комплекса бортового оборудования гражданского воздушного судна. Поэтому невозможно использование готовых существующих реализации графических библиотек и драйверов. При этом должна быть гарантирована приемлемая скорость визуализации авиационных приложений.

В процессе создания графической компоненты многооконной визуализации дисплея пилота нами было разработано множество алгоритмов и подходов, позволивших визуализировать авиационные приложения с приемлемой скоростью. Особенностью реализации также является работа компоненты под ОСРВ с жестким расписанием. Надо сказать, что некоторые из решений, успешно работающих для одной вычислительной платформы, могут не иметь эффекта для другой. Также авиационные приложения становятся все более сложными, что повышает безопасность полетов, предоставляя пилотам дополнительную информацию. Поэтому дальнейшие разработки будут связаны с повышением производительности графической компоненты и адаптацией к новым платформам.

СПИСОК ЛИТЕРАТУРЫ

1. *Федосов Е.А., Косьянчук В.В., Сельвесюк Н.И.* Интегрированная модульная авионика // Радиоэлектронные технологии. 2015. № 1. С. 66–71.
2. *Федосов Е.А., Ковернинский И.В., Кан А.В., Солоделов Ю.А.* Применение операционных систем реального времени в интегрированной модульной авионики. OSDAY2015. http://osday.ru/solodelov.html
3. *Солоделов Ю.А., Горелиц Н.К.* Сертифицируемая бортовая операционная система реального времени JetOS для российских проектов воздушных судов // Труды ИСП РАН. 2017. Т. 29. № 3. С. 171–178. https://doi.org/10.15514/ISPRAS-2017-29(3)-10
4. DO-178C Software Considerations in Airborne Systems and Equipment Certification. http://www.rtca.org/store_product.asp?prodid=803
5. Avionics application software standard interface (ARINC653). SAE-ITC, 2015. https://aviation-ia.sae-itc.com/standards/arinc653p0-3-653p0-3-avionics-application-software-standard-interface-part-0-overview-arinc-653
6. *Барладян Б.Х., Волобой А.Г., Галактионов В.А., Князь В.В., Ковернинский И.В., Солоделов Ю.А., Фролов В.А., Шапиро Л.З.* Эффективная реализация OpenGL SC для авиационных встраиваемых систем // Программирование. 2018. № 4. С. 3–10. https://doi.org/10.31857/S013234740000519-5
7. *Барладян Б.Х., Шапиро Л.З., Малачиев К.А., Хорошилов А.И., Солоделов Ю.А., Волобой А.Г., Галактионов В.А., Ковернинский И.В.* Система визуализации для авиационной ОС реального времени JetOS // Труды Института системного программирования РАН. 2020. Т. 32. № 1. С. 57–70. https://doi.org/10.15514/ISPRAS-2020-32(1)-3
8. *Baek N. and Lee H.* OpenGL ES1.1 Implementation Based on OpenGL // Multimedia Tools and Applications. V. 57. No. 3 (2012). P. 669–685.
9. *Baek N., Lee H.* OpenGL SC Implementation over an OpenGL ES1.1 Graphics Board // 2012 IEEE International Conference on Multimedia & Expo Workshops (ICMEW 2012). P. 671–671. https://doi.org/10.1109/ICMEW.2012.127
10. *Baek N. and Kim K.J.* Design and implementation of OpenGL SC2.0 rendering pipeline // Cluster Computing (2019). 22: S931–S936. https://doi.org/10.1007/s10586-017-1111-1
11. The Mesa 3D Graphics Library. https://www.mesa3d.org/
12. *Barladian B. Kh., Deryabin N.B., Voloboy A.G., Galaktionov V.A., Shapiro L.Z.* High speed visualization in the JetOS aviation operating system using hardware acceleration // CEUR Workshop Proceedings. 2020. V. 2744. P. 107:1–107:9. https://doi.org/10.51130/graphicon-2020-2-4-3
13. *Barladian B.K., Deryabin N.B., Shapiro L.Z., Solodelov Yu.A., Voloboy A.G. and Galaktionov V.A.* Multiwindow Rendering on a Cockpit Display Using Hardware Acceleration // Programming and Computer Software. 2021. V. 47. № 6. P. 457–465. https://doi.org/10.1134/S0361768821060025
14. ARINC Standards. https://www.aviation-ia.com/products/661p1-8-cockpit-display-system-interfaces-user-systems-part-1-avionics-interfaces-basic
15. Ansys SCADE Solutions for ARINC661 Compliant Systems, 2021. https://www.ansys.com/products/embedded-software/solutions-for-arinc-661
16. *Barladian B.K., Shapiro L.Z., Deryabin N.B., Solodelov Yu.A., Voloboy A.G. and Galaktionov V.A.* Efficient Rendering for the Cockpit Display System Designed in Compliance with the ARINC661 Standard // Programming and Computer Software. 2022. V. 48. № 3. P. 147–154. https://doi.org/10.1134/S0361768822030021
17. *Brian Gough.* An Introduction to GCC – for the GNU compilers gcc and g++ – Coverage testing with gcov. https://www.linuxtopia.org/online_books/an_introduction_to_gcc/gccintro_81.html

SPECIFICS OF THE DEVELOPMENT OF AN ON-BOARD VISUALIZATION SYSTEM FOR CIVIL AIRCRAFTS

© 2024 B. Kh. Barladian[a], N. B. Deryabin[a], I. V. Valiev[a], A. G. Voloboy[a], V. A. Galaktionov[a], L. Z. Shapiro[a], Yu. A. Solodelov[b]

[a] *Keldysh Institute of Applied Mathematics, Russian Academy of Sciences, Miusskaya pl. 4, Moscow, 125047 Russia*

[b] *State Scientific Research Institute of Aviation Systems, ul. Viktorenko 7, Moscow, 125319 Russia*

The instrument panels of modern aircraft are created using the "glass cockpit" concept. This new interface philosophy improves the perception of important flight information by displaying it on a single multi-function display. The paper considers the problems that arise when developing a certified pilot display visualization system designed for operation on civil aircraft under the Russian real-time operating system JetOS. The paper presents several algorithmic solutions that allow achieving acceptable visualization speed. In particular, a solution to the problem of rigid scheduling of operating system partitions is described in detail. This solution allows to overcome the degradation of rendering speed. Directions for further work have been outlined.

Keywords: cockpit display, on-board visualization system, OpenGL SC, RTOS, avionics, GPU driver

REFERENCES

1. *Fedosov E.A., Kos'yanchuk V.V., Sel'vesyuk N.I.* Integrated modular avionics // Radioelektron. techn. 2015. № 1. P. 66–71.
2. *Fedosov E.A., Koverninskiy I.V., Kan A.V., Solodelov Y.A.* Application of real-tim oerating systems in integrated modular avionics. OSDAY2015. http://osday.ru/solodelov.html
3. *Solodelov Yu.A. and Gorelits N.K.* Certifiable onboard real-time operation system JetOS for Russian aircrafts design // Proceedings of the Institute for System Programming of the RAS. 2017. V. 29. № 3. P. 171–178. https://doi.org/10.15514/ISPRAS-2017-29(3)-10
4. DO-178C Software Considerations in Airborne Systems and Equipment Certification. http://www.rtca.org/store_product.asp?prodid=803
5. Avionics application software standard interface (ARINC653). SAE-ITC, 2015. https://aviation-ia.sae-itc.com/standards/arinc653p0-3-653p0-3-avionics-application-software-standard-interface-part-0-overview-arinc-653
6. *Barladian B.Kh., Voloboy A.G., Galaktionov V.A., Knyaz' V.V., Koverninskii I.V., Solodelov Yu.A., Frolov V.A., Shapiro L.Z.* Efficient Implementation of OpenGL SC for Avionics Embedded Systems // Programming and Computer Software. 2018. V. 44. № 4. P. 207–212. https://doi.org/10.1134/S0361768818040059
7. *Barladyan B.H., Shapiro L.Z., Mallachiev K.A., Khoroshilov A.V., Solodelov Yu.A., Voloboy A.G., Galaktionov V.A., Koverninsky I.V.* Rendering System for the Aircraft Real-Time OS JetOS // Proceedings of the Institute for System Programming of the RAS. 2020. V. 32. № 1. P. 57–70. https://doi.org/10.15514/ISPRAS-2020-32(1)-3
8. *Baek N. and Lee H.* OpenGL ES1.1 Implementation Based on OpenGL // Multimedia Tools and Applications. V. 57. No. 3 (2012). P. 669–685.
9. *Baek N., Lee H.* OpenGL SC Implementation over an OpenGL ES1.1 Graphics Board // 2012 IEEE International Conference on Multimedia & Expo Workshops (ICMEW 2012). P. 671–671. https://doi.org/10.1109/ICMEW.2012.127
10. *Baek N. and Kim K.J.* Design and implementation of OpenGL SC2.0 rendering pipeline // Cluster Computing (2019). 22: S931–S936. https://doi.org/10.1007/s10586-017-1111-1
11. The Mesa 3D Graphics Library. https://www.mesa3d.org/
12. *Barladian B.Kh., Deryabin N.B., Voloboy A.G., Galaktionov V.A., Shapiro L.Z.* High speed visualization in the JetOS aviation operating system using hardware acceleration // CEUR Workshop Proceedings. 2020. V. 2744. P. 107:1–107:9. https://doi.org/10.51130/graphicon-2020-2-4-3
13. *Barladian B.K., Deryabin N.B., Shapiro L.Z., Solodelov Yu.A., Voloboy A.G. and Galaktionov V.A.* Multiwindow Rendering on a Cockpit Display Using Hardware Acceleration // Programming and Computer Software. 2021. V. 47. № 6. P. 457–465. https://doi.org/10.1134/S0361768821060025
14. ARINC Standards. https://www.aviation-ia.com/products/661p1-8-cockpit-display-system-interfaces-user-systems-part-1-avionics-interfaces-basic
15. Ansys SCADE Solutions for ARINC661 Compliant Systems, 2021. https://www.ansys.com/products/embedded-software/solutions-for-arinc-661
16. *Barladian B.K., Shapiro L.Z., Deryabin N.B., Solodelov Yu.A., Voloboy A.G. and Galaktionov V.A.* Efficient Rendering for the Cockpit Display System Designed in Compliance with the ARINC661 Standard // Programming and Computer Software. 2022. V. 48. № 3. P. 147–154. https://doi.org/10.1134/S0361768822030021
17. *Brian Gough.* An Introduction to GCC – for the GNU compilers gcc and g++ – Coverage testing with gcov. https://www.linuxtopia.org/online_books/an_introduction_to_gcc/gccintro_81.html

ПРОГРАММИРОВАНИЕ, 2024, № 3, с. 14–20

КОМПЬЮТЕРНАЯ ГРАФИКА И ВИЗУАЛИЗАЦИЯ

УДК 519.6

АВТОМАТИЧЕСКАЯ СТИЛИЗАЦИЯ ИЗОБРАЖЕНИЙ ПО РАСШИРЕННОЙ КОЛЛЕКЦИИ СТИЛЕЙ

© 2024 г. В. В. Понамарев[a,*], В. В. Китов[a,b,]**

[a] *МГУ им. М.В. Ломоносова, ф-т ВМК*
119991 Москва, ГСП-1, Ленинские горы, д. 1
[b] *РЭУ им. Г.В. Плеханова, лаборатория искусственного интеллекта*
119333 Москва, Стремянный пер., д. 36
**E-mail: valera.pon.vp@gmail.com*
***E-mail: v. v.kitov@yandex.ru*

Поступила в редакцию: 20.01.2024 г.
После доработки: 24.01.2024 г.
Принята к публикации: 24.01.2024 г.

Стилизация изображений представляет собой актуальную прикладную задачу по автоматической перерисовке исходного изображения (контента) в стиле другого изображения (задающего целевой стиль). Традиционные методы стилизации изображений предоставляют единственный вариант стилизации. Если он пользователя не устраивает, например, в силу возникающих в ходе стилизации артефактов, то ему приходится выбирать другой стиль. В работе предлагается модификация алгоритма стилизации, дающая разнообразные результаты стилизации одним стилем, а также повышающая среднее качество стилизации за счет использования не только стилевой информации с исходного стилевого изображения, но и информации с изображений, имеющих похожий стиль.

Ключевые слова: генерация изображений, обработка изображений, нейронные сети

DOI: 10.31857/S0132347424030029, **EDN:** QAXHSF

ВВЕДЕНИЕ

Алгоритму нейросетевой стилизации изображений (neural style transfer) подаются на вход два изображения – конетное изображение (контент), и стилевое изображение (стиль). Алгоритм решает задачу автоматической перерисовки контентного изображения в стиле стилевого изображения. Под стилем понимается цветовая гамма и характерные паттерны рендеринга, такие как мазки кисти художника. Указанная задача актуальна для создания ярких иллюстраций в книгах, на сайтах, в рекламе, дизайне, а также в индустрии развлечений.

Изначально указанная задача была известна под названием non-photorealistic rendering [1–3] и решалась эвристическими методами обработки изображений, подбираемыми под каждый стиль. Нейросетевой перенос стиля, предложенный в работе [4], позволил переносить стиль с произвольного стилевого изображения-образца, как показано на примере стилизации контентного изображения двумя стилями на рис. 1.

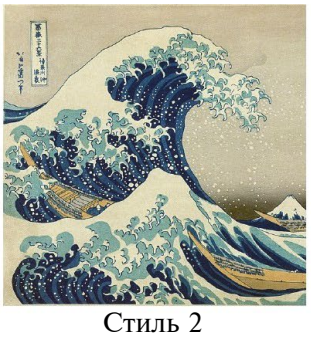
На рис. 1 видно, что качество стилизации существенно зависит от сочетаемости стилей контентного и стилевого изображений. Если они сочетаемы (стиль 2), то результат стилизации получается приемлемым. Если же стили контентного и стилевого изображений существенно различаются, например, по четкости, как при стилизации первым стилем (см. рис. 1), который более размыт, чем контент, то и стилизация получается неудовлетворительная. Получается, что для создания эффектных стилизаций пользователь вынужден долго перебирать вручную различные стили, пока не найдет стиль, который хорошо сочетается с контентом.

Чтобы упростить процедуру создания качественных стилизаций в работе предлагается производить стилизацию не одним пользовательским стилем, а совокупностью изображений, похожих по стилю на входное пользовательское стилевой изображение. Для расчета стилевой похожести предлагается специальная процедура расчета стилевых характеристик. За счет агрегации стилевой информации с нескольких изображений повышается устойчивость алгоритма к возможным индивидуальным несоотвествиям контентного и стилевого изображений. Проведенные опросы респондентов показывают, что получаемые стилизации действительно получаются более качественными.

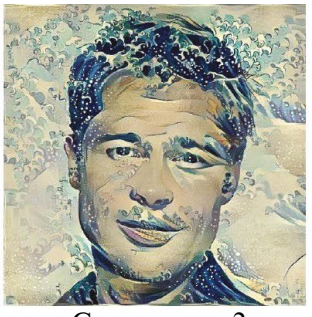
Контент

Стиль 1

Стилизация 1



Стиль 2



Стилизация 2

Рис. 1. Применение переноса стиля методом [4]. Первый стиль наложился хуже, чем второй, из-за сильной несочетаемости с контентным изображением (по четкости).

Другим преимуществом предлагаемой модификации является возможность более широко смотреть на стиль. Теперь он не задается одним лишь исходным стилевым изображением, а задан целой совокупностью изображений, имеющих стиль, похожий на целевой. Тем самым, беря разные подмножества из этих изображений, можно получать различные варианты стилизации. Это удобно, если пользователю хотелось бы получить иной вариант стилизации исходным стилем.

Стоит отметить, что базовый подход к стилизации [4], над котором предлагается надстройка, производит оптимизацию в исходном цветовом пространстве стилизуемого изображения, что занимает несколько десятков секунд на видеокарте и неприменимо, например, к стилизации видеопотоков в реальном времени. Много последующих работ, например [5, 6], было посвящено ускорению стилизации за счет преобразования контентного изображения через специальную нейронную сеть. Поскольку предлагаемый метод представляет собой надстройку над алгоритмом стилизации, то он применим и к ним, а также к другим методам стилизации.

БАЗОВЫЙ АЛГОРИТМ СТИЛИЗАЦИИ ИЗОБРАЖЕНИЙ

Рассмотрим оригинальный подход, указанный в статье [4]. Процесс переноса стиля оперирует со следующими входными данными:

- S – изображение, содержащее желаемый художественный стиль;
- Y – изображение, содержимое которого требуется отобразить в нужном стиле (контент).

Задача алгоритма стилизации – сгенерировать изображение X (стилизацию), на котором объекты контента Y будут изображены в стиле стилевого изображения S. Стиль задает цветовую гамму и характерные паттерны стилевого изображения – углы, переливы между цветами, характерные паттерны, такие как мазки кисти художника.

Для переноса стиля решается оптимизационная задача в пространстве пикселов результирующего стилизованного изображения X:

$$\mathcal{L}_{cont}(X,Y) + \alpha\mathcal{L}_{style}(X,S) \to \min_{X}, \qquad (1)$$

где потери $\mathcal{L}_{cont}(X,Y)$ штрафуют расхождение контентного изображения и стилизации по смыслу (что именно изображено), а $\mathcal{L}_{style}(X,S)$ штрафуют расхождение стилевого изображения и стилизации по стилю (как изображено). Гиперпараметр $\alpha > 0$ управляет противоречием между более точной передачей контента (смысла) и более полной передачей стиля.

Расчет контентных и стилевых потерь производится по представлениям изображений X, Y, S на промежуточных слоях сети VGG [7], обученной классифицировать изображения, на выборке ImageNet [8]. Причем для расчета контентных потерь используется один промежуточный слой k, а для расчета стилевых потерь – совокупность более ранних и более поздних слоев, как показано на рис. 2.

Функция потерь для контента:

$$\mathcal{L}_{cont}(X,Y) = \frac{1}{H_k W_k C_k}\sum_{c=1}^{C_k}\sum_{i=1}^{H_k}\sum_{j=1}^{W_k}(X_{cij}^k - Y_{cij}^k)^2,$$

где X^k, $Y^k \in \mathbb{R}^{C_k \times H_k \times W_k}$ – тензоры промежуточных представлений на некотором слое k (гиперпараметр) изображений X и Y после того, как их

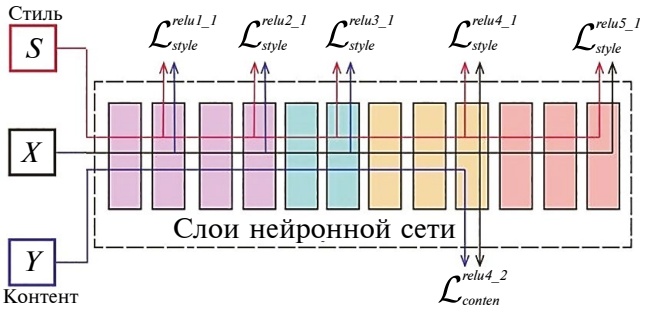


Рис. 2. Используемые при стилизации слои сети VGG.

пропустили через классификационную нейросеть; C_k – число каналов; $H_k \times W_k$ – пространственное разрешение карты признаков. Суть функции заключается в том, что если по смыслу на изображениях X и Y изображены разные объекты, то и их промежуточные представления в сети VGG будут различаться. Штрафуются расхождения именно в промежуточных признаковых представлениях (отвечающих за семантику), а не в исходных RGB-представлениях, поскольку иначе это служило бы слишком сильной привязкой стилизации к исходному контентному изображению и стилизации бы не получилось.

Стилевые потери представляют собой сумму стилевых потерь по отдельным слоям сети VGG:

$$\mathcal{L}_{style} = \sum_k \mathcal{L}^k_{style},$$

где

$$\mathcal{L}^k_{style}(X,S) = \frac{1}{C_k^2 H_k^2 W_k^2} \sum_{i=1}^{H_k} \sum_{j=1}^{W_k} \left(G^{X^k}_{ij} - G^{S^k}_{ij} \right)^2. \quad (2)$$

Здесь $G^{Z^k} \in \mathbb{R}^{C_k \times C_k}$ обозначает матрицу Грама, состоящую из скалярных произведений между всевозможными каналами промежуточного представления Z^k в сети VGG на слое k для изображения Z (равного либо стилевому изображению S либо результату стилизации X):

$$G^{Z^k}_{ij} = \sum_{i=1}^{W_k} \sum_{j=1}^{H_k} Z^k_{cij} Z^k_{cij}. \quad (3)$$

Контент определяет пространственное расположение объектов на изображении, поэтому при расчете контентной функции потерь производится сравнение промежуточных представлений в привязке к пространственным координатам (i и j).

Стиль же задает общее распределение цветов и более общих паттернов (границ, углов, переливов, мазков кисти и т. д.). Поэтому сначала извлекается это распределение в виде скалярных произведений между каналами (при расчете скалярных произведений пространственная информация теряется, так как происходит агрегация по всевозможным пространственным координатам i, j), а затем штрафуется уже расхождение распределений признаков между стилизацией и стилем. Такая структура стилевых потерь приближает стиль, но не приводит к переносу контентной (смысловой) информации со стилевого изображения.

ПРЕДЛАГАЕМЫЙ ПОДХОД РАСШИРЕННОЙ СТИЛИЗАЦИИ

Поскольку привяка к определенному пользовательскому изображению стиля может быть слишком жестким ограничением, приводящем к артефактам стилизации, вызванной слабым исходным соответствием контента и стиля (см. рис. 1), предлагается трехэтапный алгоритм стилизации, более устойчивый к индивидуальной несовместимости исходного контентного и стилевого изображений:

1. Найти для исходного стилевого изображения наиболее близкие по стилю изображения в широкой базе художественных изображений, таких как Wikiart [9] или Pandora [10] (которая использовалась в работе). Опционально: перекрасить похожие стилевые изображения в цветовую гамму стиля.

2. Усреднить матрицы Грама по похожим изображениям.

3. Применить алгоритм стилизации (1), заменив в стилевых потерях (2) матрицы Грамма стилевого изображения на усредненные матрицы Грама по набору изображений, задающих похожий стиль на заданный.

Для поиска похожих изображений по стилю извлекаются вектора поканальных средних для промежуточных представлений в сети VGG. Например, для изображения Z и его промежуточного представления Z^k на слое k получим компоненты C_k-мерного вектора средних следующим образом:

$$v^k_c(Z) = \frac{1}{H_k W_k} \sum_{i=1}^{W_k} \sum_{j=1}^{H_k} Z^k_{cij},\ c = \overline{1, C_k}. \quad (4)$$

Далее указанные векторы конкатенируются по более ранним и более поздним слоям VGG, чтобы выравнивать соответствие по разным слям сети. Сравнение изображений по стилю производится сравнением результирующих векторов по Евклидовой норме. Указанное векторное представление содержит информацию о стиле, а не о контенте, поскольку при поканальном усреднении стирается информация о пространственном расположении объектов, а сохраняется только

статистика присутствия тех или иных признаков (цветов и более общих локальных паттернов), которые и характеризуют стиль.

Чтобы повысить соответствие исходному стилю, рекомендуется перекраска наиболее похожих стилевых изображений на исходный стиль в цвета исходного стиля, используя алгоритм гистограммной перекраски (histogram matching) [11].

Несмотря на то что в следующей секции приводятся результаты запусков для стилизационной модели [4], предлагаемый подход достаточно общий, чтобы быть применимым и для других методов стилизации. Код предложенного подхода доступен на github[1].

СРАВНИТЕЛЬНЫЕ ЭКСПЕРИМЕНТЫ

Определим параметры, при которых будет производиться стилизация. Максимальное количество предлагаемых изображений взято равным 3, в качестве метрики близости будем использовать Евклидову метрику, порог близости объектов, при которой будут добавляться похожие изображения, возьмем равным *threshold* = 500. Стилизацию будем проводить по агрегированному (с помощью усреднения матриц Грама) стилю. Сопоставим результаты полученной стилизации со стилизацией с добавленным перекрашиванием найденных стилей в цвет исходного пользовательского стиля.

На рис. 3, 4 показаны некоторые примеры стилизаций с перекрашиванием и без. В обоих случаях видно, что результирующая стилизация немного отлична от исходной, хотя и повторяет ее общие черты. Отстутствие перекрашивания снижает соответствие исходному стилю, но может добавлять яркости результату, что особенно заметно на рис. 3.

Количество стилей в группе является настраиваемым параметром. Используя группы разных размеров, можно получать отличные друг от друга стилизации. Обозначим за k число стилей в группе. На рис. 5, 6 изображены стилизации по группам стилей при разных k. Большее количество изображений позволяет выразительнее передавать детали стилей и контента, но существует условный предел на k, после которого групповая стилизация начинает размываться и терять преимущества относительно стилизации по исходному изображению. Чем больше k, тем менее похожие стили добавляются в группу. Это также зависит от размера всей коллекции изображений. При небольшом объеме уже третий стиль может сильно отличаться от первого. Результаты экспериментов показывают достаточность использования двух-трех дополнительных стилей для повышения качества стилизации.

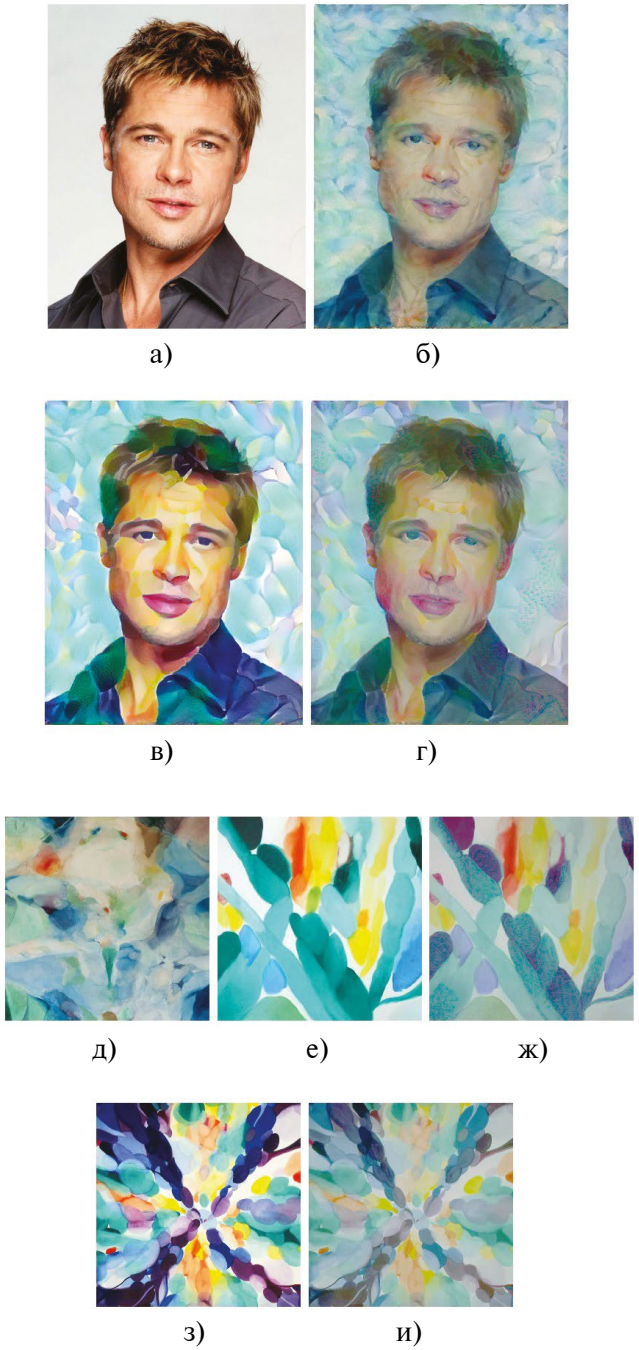
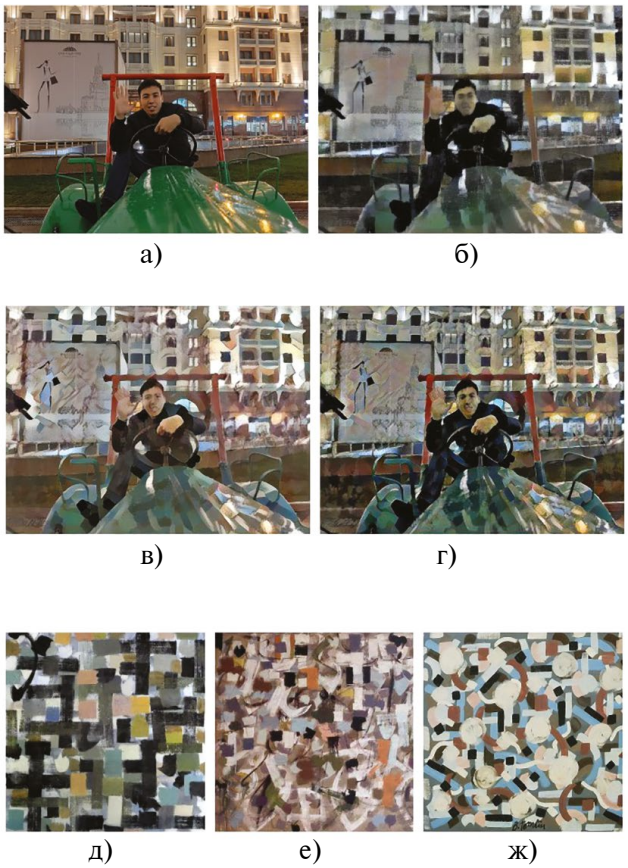


Рис. 3. Пример работы алгоритма стилизации по группе найденных стилей с перекрашиванием исходного стиля: а) контент; б) стандартная стилизация контента а) по стилю д); в) предложенная стилизация по группе найденных стилей е) и з); г) предложенная стилизация по группе стилей с перекрашиванием ж) и и); д) исходный стиль; е) 1-й наденный стиль; ж) стиль е) перекрашенный в д); з) 2-й найденный стиль; и) стиль з) перекрашенный в д).

[1] httpps://github.com/valerapon/Style-transfer-UESC



а) б) в) г) д) е) ж)

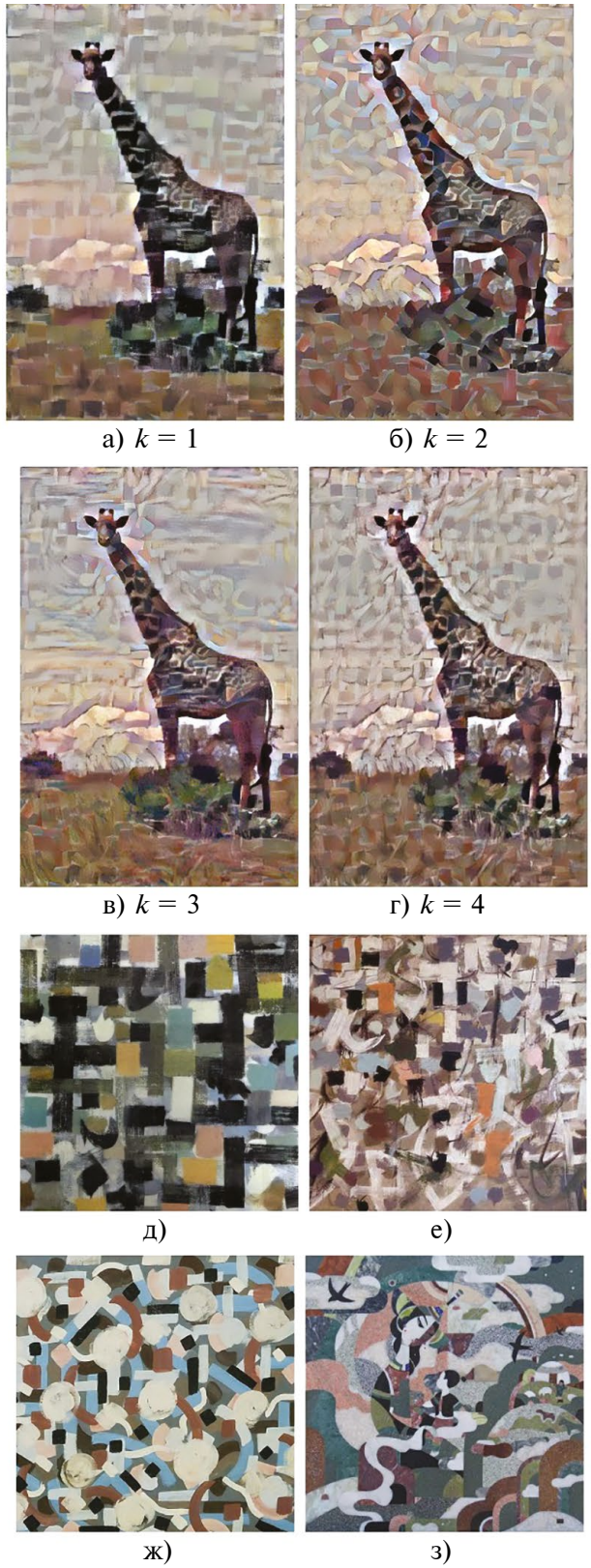
Рис. 4. Пример работы алгоритма стилизации по группе найденных стилей с переносом цветовой схемы исходного стиля: а) контент; (б) стандартная стилизация контента а) по стилю д); в) предложенная стилизация по группе найденных стилей е), ж); г) предложенная стилизация по группе стилей с перекрашиванием е), ж); д) пользовательский стиль; е), ж) похожие на (е) стили.

Таблица 1. Сравнение метрик существующих популярных моделей с предложенным. Значения подсчитывались на тестовом наборе из 1000 пар контент-стиль

Мера качества	Gatys et al. [4]	AdaIN [15]	WCT [16]	**Предложенный**
Deception rate [14]	0.213	0.117	0.082	**0.323**
Avg. Gradient Magnitude [12]	160.3	153.9	165.1	**177.8**
Avg. 90% Quantile FFT [13]	0.488	0.467	0.508	**0.541**

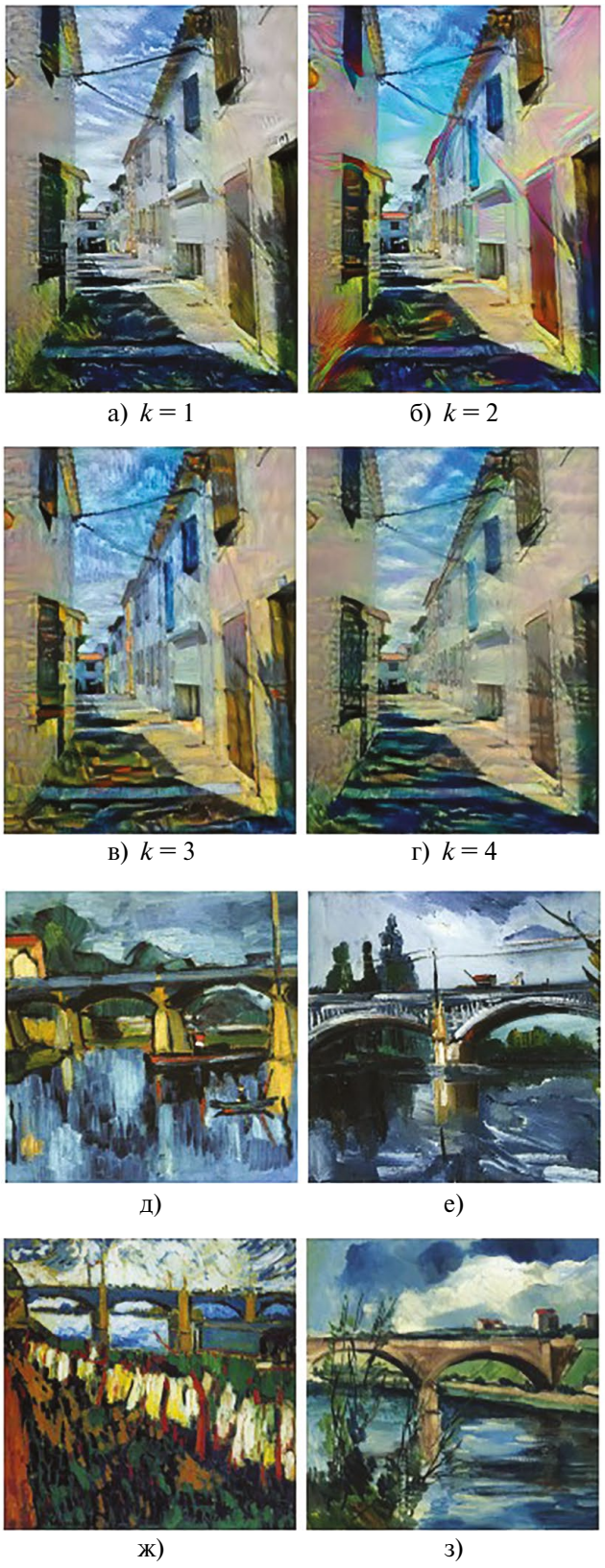
Таблица 2. Результаты опроса пользователей по сравнению оригинального способа стилизации [4] и предложенного способа стилизации

Число вопросов	Число респондентов	Число ответов	% голосов за новый метод
25	58	1450	71%



а) $k = 1$ б) $k = 2$ в) $k = 3$ г) $k = 4$ д) е) ж) з)

Рис. 5. Работа предложенным алгоритмом стилизации по группе найденных стилей при различном размере группы k: а) стилизация по д), $k = 1$; б) стилизация по д) и е), $k = 2$; в) стилизация по д) – ж), $k = 3$; г) стилизация по д) – з), $k = 4$.



а) $k = 1$ б) $k = 2$ в) $k = 3$ г) $k = 4$ д) е) ж) з)

Рис. 6. Работа предложенным алгоритмом стилизации по группе найденных стилей при различном размере группы k: а) стилизация по д), $k = 1$; б) стилизация по д) и е), $k = 2$; в) стилизация по д) – ж), $k = 3$; г) стилизация по д) – з), $k = 4$.

Для численного сравнения стилизаций (см. табл. 1), выполненных разными моделями, были рассмотрены метрики: Average Gradient Magnitude [12] – средняя магнитуд градиентов изображения с окном 3×3 пикселов (оператор Собеля); Average 0.90 Quantile FFT [13] – стилизация делится на 8×8 частей, для каждого блока вычисляются преобразование Фурье и среднее амплитуд, лежащих выше квантили 90%; Deception rate [14] – на основе сети VGG строится классификатор художественных стилей (600 классов), каждый из алгоритмов стилизации тестируется на способность сохранять исходный стиль с помощью построенного классификатора; если он правильно определяет художественный стиль у стилизации, то она считается корректной; оценивается доля корректных стилизаций классификатором на наборе из 1000 пар контент–стиль для каждого алгоритма. По всем мерам качества достигается улучшение предложенного метода по сравнению с базовым [4], использующим лишь один исходный стиль.

Также был проведен опрос среди респондентов, которым для всевозможных пар контент-стиль предлагались на выбор два варианта стилизаций – базовым методом [4] и предложенным (группой похожих стилей с их перекраской под исходный стиль). Респонеднтам предлагалось выбрать, на их взгляд, более удачный вариант стилизации. Варианты стилизаций каждый раз предлагались в случайном порядке, а респонденты не были ознакомлены с деталями алгоритмов стилизации. Всего участвовало 58 респондентов, каждый из которых сравнивал 25 пар стилизаций. Результаты опросов представлены в табл. 2.

Результат опроса пользователей показал, что пользователи в 71% случаев выбирали новый метод стилизации, что показывает его преимущество по отношению к базовому методу стилизации [4].

ЗАКЛЮЧЕНИЕ

В работе исследовалась задача автоматической стилизации изображений. Было предложено расширить исходное стилевое изображение выборкой похожих по стилю изображений. Для стилевого сопоставления использовались вектора поканальных средних в представлениии VGG, которые хорошо себя показали на практике. Стилизация группой похожих стилей, а не только исходным, дает более устойчивый и качественный результат, что показали формальные меры качества и опросы респондентов, чаще предпочитавшие стилизации предложенным методом.

Дополнительным преимуществом метода является возможность изменять группу похожих стилевых изображений, что позволит пользователю получать различные варианты стилизаций, если первоначальная стилизация пользователя не устраивает.

БЛАГОДАРНОСТЬ

Работа выполнена в рамках государственного задания в сфере научной деятельности Министерства науки и высшего образования РФ на тему “Модели, методы и алгоритмы искусственного интеллекта в задачах экономики для анализа и стилизации многомерных данных, прогнозирования временных рядов и проектирования рекомендательных систем”, номер проекта FSSW-2023-0004.

СПИСОК ЛИТЕРАТУРЫ

1. *Gooch B., Gooch A.* Non-photorealistic rendering. CRC Press, 2001.
2. *Strothotte T., Schlechtweg S.* Non-photorealistic computer graphics: modeling, rendering, and animation. Morgan Kaufmann, 2002.
3. *Rosin P., Collomosse J.* Image and video-based artistic stylisation. Springer Science & Business Media, 2012. V. 42.
4. *Gatys L.A., Ecker A.S., Bethge M.* A neural algorithm of artistic style // arXiv preprint arXiv:1508.06576, 2015.
5. *Huang X., Belongie S.* Arbitrary style transfer in real-time with adaptive instance normalization // Proceedings of the IEEE international conference on computer vision. 2017. P. 1501–1510.
6. *Li Y. et al.* Universal style transfer via feature transforms // Advances in neural information processing systems. 2017. V. 30.
7. *Simonyan K., Zisserman A.* Very deep convolutional networks for large-scale image recognition // arXiv preprint arXiv:1409.1556, 2014.
8. *Deng J. et al.* Imagenet: A large-scale hierarchical image database // 2009 IEEE conference on computer vision and pattern recognition. IEEE. 2009. P. 248–255.
9. *Phillips F., Mackintosh B.* Wiki Art Gallery, Inc.: A case for critical thinking // Issues in Accounting Education. 2011. V. 26. № 3. P. 593–608.
10. *Florea C. et al.* Pandora: Description of a painting database for art movement recognition with baselines and perspectives // 2016 24th European Signal Processing Conference (EUSIPCO). IEEE. 2016. P. 918–922.
11. *Buzuloiu V. et al.* Adaptive-neighborhood histogram equalization of color images // Journal of Electronic Imaging. 2001. V. 10. № 2. P. 445–459.
12. *de Villiers J.P.* A comparison of image sharpness metrics and real-time sharpening methods with GPU implementations // Proceedings of the 7th International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa. 2010. P. 53–62.
13. *Hassen R., Wang Z., Salama M.M.A.* Image sharpness assessment based on local phase coherence // IEEE Transactions on Image Processing. 2013. V. 22. № 7. P. 2798–2810.
14. *Sanakoyeu A. et al.* A style-aware content loss for real-time hd style transfer // Proceedings of the European conference on computer vision (ECCV). 2018. P. 698–714.
15. *Huang X., Belongie S.* Arbitrary style transfer in real-time with adaptive instance normalization // Proceedings of the IEEE international conference on computer vision. 2017. P. 1501–1510.
16. *Li Y. et al.* Universal style transfer via feature transforms // Advances in neural information processing systems. 2017. V. 30.

AUTOMATIC IMAGE STYLE TRANSFER USING AN AUGMENTED STYLE SET

© 2024 V. V. Ponamaryov[a], V. V. Kitov[b]

[a] *Lomonosov Moscow State University, f-ty CMC, Moscow, 119991 Russia*

[b] *Plekhanov Russian University of Economics, Artificial Intelligence Laboratory, Moscow, 117997 Russia*

Image style transfer is an applied task for automatic rendering of the original image (content) in the style of another image (specifying the target style). Traditional image stylization methods provide only a single stylization result. If the user is not satisfied with it due to stylization artifacts, he has to choose a different style. The work proposes a modified stylization algorithm, giving a variety of stylization results, and achieves improved stylization quality by using additional style information from similar styles.

Keywords: image generation, image processing, diversity of generations, neural networks

ПРОГРАММИРОВАНИЕ, 2024, № 3, с. 21–32

КОМПЬЮТЕРНАЯ ГРАФИКА И ВИЗУАЛИЗАЦИЯ

УДК 004.92

ВЛИЯНИЕ НЕРАВНОСТОРОННИХ АПЕРТУР ЦИФРОВЫХ ФИЛЬТРОВ ЛАПЛАСИАНА “УСЕЧЕННАЯ ПИРАМИДА” И “ДВОЙНАЯ ПИРАМИДА” НА ТОЧНОСТЬ РАБОТЫ ТЕЛЕВИЗИОННЫХ ИЗМЕРИТЕЛЬНЫХ СИСТЕМ

© 2024 г. К. А. Рылов[a], *, К. С. Куприянова[a], **, А. В. Каменский[a], ***

[a] *Томский Государственный университет систем управления и радиоэлектроники ТУСУР*
634050 Томск, ул. Ленина, д. 40, Россия

**E-mail: tstr70@mail.ru*

***E-mail: kuprianovak8@gmail.com*

****E-mail: andru170@mail.ru*

Поступила в редакцию: 19.01.2024 г.
После доработки: 19.01.2024 г.
Принята к публикации: 24.01.2024 г.

В современном мире обработка цифровых изображений требует увеличения быстродействия используемых методов и алгоритмов обработки. Одним из способов повышения быстродействия является преобразование пространственных фильтров в фильтры с рекурсивно-сепарабельной формой реализации. Свойство рекурсии подразумевает использование предыдущих выходных значений функции для формирования текущего отсчета. Под свойством сепарабельности понимается разделение на обработку по столбцу и по строке матрицы значений цифрового изображения. Преобразование пространственных фильтров заключается в изменении апертуры масок в неортогональную (неравностороннюю) форму, что позволяет сократить количество вычислительных операций и ускорить процесс обработки при сохранении ее эффективности. В работе представлено описание неравносторонних апертур разработанных ранее цифровых фильтров лапласиан “усеченная пирамида” и “двойная пирамида”. Для неравносторонних апертур впервые получены результаты по их применению для телевизионных измерительных систем, из которых видно, что для этого использования рекомендуется фильтр лапласиан “усеченная пирамида” с неравносторонними апертурами обработки, поскольку он повышает эффективность измерения дальности до объектов интереса при сокращении времени обработки. По результатам обработки модифицированными фильтрами были получены наборы обработанных изображений для каждого из 10-ти исходных изображений. Для каждого набора обработанных изображений проводились измерения пикового отношения сигнал-шум, среднеквадратического отклонения и выбор оптимального центрального коэффициента маски фильтра для последующей оценки эффективности обработки модифицированными фильтрами. Оценка влияния рекурсивно-сепарабельных фильтров лапласиана “усеченная пирамида” и “двойная пирамида” с масками неравносторонней апертуры на точность работы телевизионных измерительных систем заключалась в рассмотрении их влияния на измерение расстояния (от камеры) до объекта интереса на изображении при контроле времени обработки. По результатам оценки можно сделать вывод о том, что за счет применения предварительной обработки изображений модифицированными цифровыми фильтрами улучшается точность измерения расстояния от камеры до объекта измерения при сокращении времени обработки.

Ключевые слова: цифровые фильтры, рекурсивно разделяемые фильтры, “усеченная пирамида” Лапласа, “двойная пирамида” Лапласа, цифровая обработка изображений, телевизионная измерительная система, пиковое соотношение сигнал/шум, среднеквадратичное отклонение, двумерная дискретная свертка

DOI: 10.31857/S0132347424030033, **EDN:** QATTKZ

1. ВВЕДЕНИЕ

В связи с быстрым развитием компьютерных технологий и широким внедрением видеоинформационных технологий усовершенствование способов и методов обработки изображений является весьма актуальной задачей.

Примером таких технологий являются телевизионные измерительные системы (ТИС),

которые нашли достаточно широкое применение в устройствах наблюдения, ориентации, навигации, системах технического зрения роботов, в медицине и других областях [1]. ТИС предназначены для дистанционного автоматического контроля за состоянием пространства в угле поля зрения телевизионных датчиков, для измерения параметров совокупности объектов, находящихся в контролируемом пространстве, и при необходимости для управления состоянием и взаимодействием этих объектов [2].

Классические алгоритмы фильтрации изображений осуществляются (производятся) с помощью операции двумерной дискретной свертки, которая представляет собой сумму произведений значений пикселов изображения и значений маски фильтра. Данный метод фильтрации известен как пространственная фильтрация [3, 4]. При использовании фильтров больших размеров такой способ требует большого числа вычислений, так как функцию свертки необходимо вычислить для каждого пиксела изображения.

Для улучшения классических алгоритмов цифровых фильтров в данной работе применяются свойства рекурсии и сепарабельности. Под свойством рекурсии понимается использование предыдущих выходных значений функции свертки для формирования текущего отсчета. Свойство сепарабельности позволяет реализовать разделение на обработку по столбцу и по строке матрицы значений входного изображения [5].

Рекурсивно-сепарабельные фильтры – “масочные” фильтры, так как для обработки используются маски, представленные в виде двумерных матриц. Примером таких фильтров является фильтр лапласиан “усеченная пирамида” (ЛУП) и фильтр лапласиан “двойная пирамида” (ЛДП) [6]. Применение данных фильтров позволяет сократить количество вычислительных операций, тем самым ускоряя процесс фильтрации.

Целью работы является оценка влияния модифицированных цифровых фильтров с масками неравносторонней апертуры на быстродействие работы алгоритма предобработки цифровых изображений, формируемых ТИС и влияния на точность определения расстояния до объекта в поле зрения ТИС.

Для достижения поставленной цели необходима программная реализация неравносторонних апертуры цифровых фильтров ЛУП и ЛДП. Для оценки быстродействия необходимо сравнить скорость выполнения операций в фильтрах равносторонней и неравносторонней апертуры на ряде тестовых изображений. Для оценки влияния фильтров на точность работы ТИС необходимо после обработки преобразованными фильтрами измерить расстояние до объектов интереса на тестовых изображениях.

2. РАЗРАБОТКА НЕРАВНОСТОРОННИХ АПЕРТУР ЦИФРОВЫХ ФИЛЬТРОВ ЛУП И ЛДП

Разработка неравносторонних апертур рекурсивно-сепарабельных фильтров заключается в преобразовании их равносторонней формы в неравностороннюю [7]. Данное преобразование осуществляется путем расчета и изменения значений коэффициентов кадровых и строчных рециркуляторов. При условии: форма преобразованного фильтра должна быть схожа с изначальным трехмерным видом исходной матрицы 7×7 элементов. Рециркуляторами называют образующие рекурсивные ячейки, которые соответствуют обработке по строке (СР) и по кадру (КР) матрицы входного изображения, а заданные в них значения служат для формирования размера требуемой маски.

Структурная схема цифрового фильтра ЛУП [8], формирующая маски равносторонней формы, представлена на рис. 1, где $x(n_1, n_2)$ –

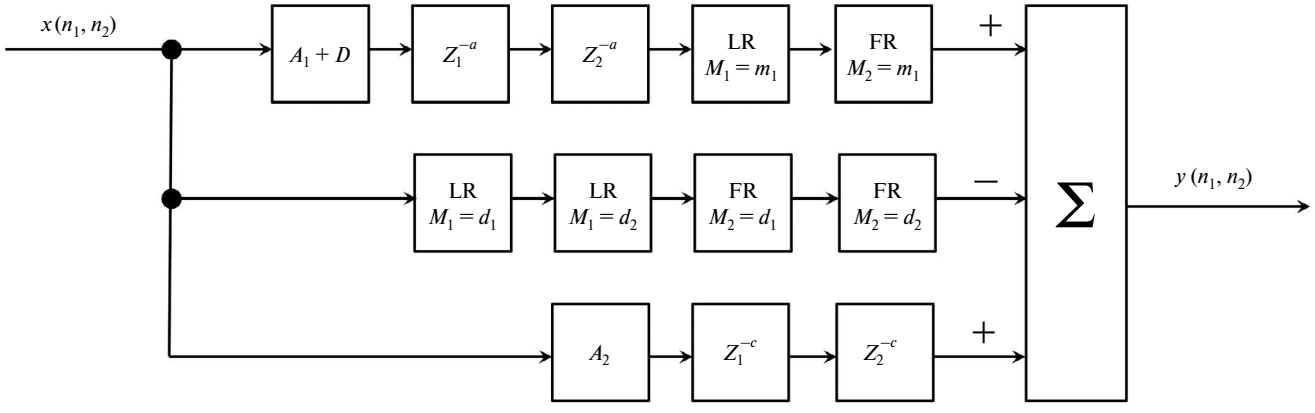


Рис. 1. Структурная схема ЛУП.

входная информация; $y(n_1, n_2)$ – выходная информация; D – дополнительный параметр для маски свертки; a – дополнительный параметр сдвига маски; m_1 – коэффициент строчных и кадровых рециркуляторов для дополнительной маски; d_1 и d_2 – параметры строчного и кадрового рециркулятора для основной маски; c – коэффициент смещения маски увеличения центрального элемента; A_1 – коэффициент подъема центральной апертуры финальной маски; A_2 – коэффициент увеличения центрального элемента; LR – строчный рециркулятор; FR – кадровый рециркулятор; z – элемент задержки.

Данная структурная схема состоит из трех ветвей. В первой ветви фильтра формируется маска размером $m_1 \times m_1$ элемента, которая необходима для создания области положительных элементов (при наличии окаймления в виде отрицательных элементов) внутри основной маски, это осуществляется для нормализации яркости обработанного изображения. Вторая ветвь фильтра формирует основную маску фильтра размерностью $d_1 \times d_2$ элементов, которая подается на сумматор с отрицательным знаком. Третья ветвь используется для добавления к итоговой апертуре дополнительного коэффициента фильтрации (A_2). Далее происходит суммирование полученных масок первой и второй ветвей фильтра, и за счет полученной отрицательной основной маски происходит вычитание первой положительной маски, в результате чего на выходе получается итоговая маска фильтра размерностью $d_1 \times d_2$ элементов, эквивалентная обработанному изображению.

Для формирования маски неравносторонней формы, размерностью 7×5 элементов, изменялись размеры СР и КР. Для маски апертуры 7×5 элементов в структурной схеме задается начальный дополнительный коэффициент 16.67. Данный коэффициент вводится для корректного соотношения сумм внутренней и внешней частей итоговой матрицы, его вычисление производится из условия равенства нулю суммы этих частей маски фильтра. После этого, в соответствии со структурной схемой, происходит перемножение коэффициентов рециркуляторов, где в первой ветви фильтра используются строчный рециркулятор с коэффициентом 3 и кадровый рециркулятор с коэффициентом 3, во второй ветви фильтра используются два строчных рециркулятора с коэффициентами 5 и 3 и два кадровых рециркулятора с коэффициентами 4 и 2. Подробный алгоритм работы фильтров описан в ранее опубликованной работе [9].

Для фильтра ЛДП формирование маски неравносторонней формы размерностью 7×5 элементов происходит аналогичным образом, как и в случае фильтра ЛУП. Структурная схема цифрового фильтра ЛДП, формирующего маски любой размерности, представлена на рис. 2, где $x(n_1, n_2)$ – входная информация; $y(n_1, n_2)$ – выходная информация; D – дополнительный параметр для маски свертки; a – дополнительный параметр сдвига маски; m_1 – параметр строчных и кадровых рециркуляторов для дополнительной маски; d_1 и d_2 – параметы строчного и кадрового рециркулятора для основной маски; c – коэффициент смещения для маски увеличения центрального элемента; A_1 – коэффициент возвышения центральной апертуры; A_2 – коэффициент увеличения центрального элемента.

В ходе преобразования масок для каждого фильтра в неравностороннюю форму были получены значения начального добавочного коэффициента и значения коэффициентов СР и КР, представленные в табл. 1.

Полученные итоговые маски цифровых рекурсивно-сепарабельных фильтров ЛУП и ЛДП размерностью 7×5 элементов показаны на рис. 3 и 4.

В результате впервые были получены неравносторонние апертуры фильтров ЛУП и ЛДП для разработанных ранее рекурсивно-сепарабельных фильтров. Следует отметить, что благодаря этим результатам стала возможна модификация алгоритмов построения фильтров ЛУП и ЛДП,

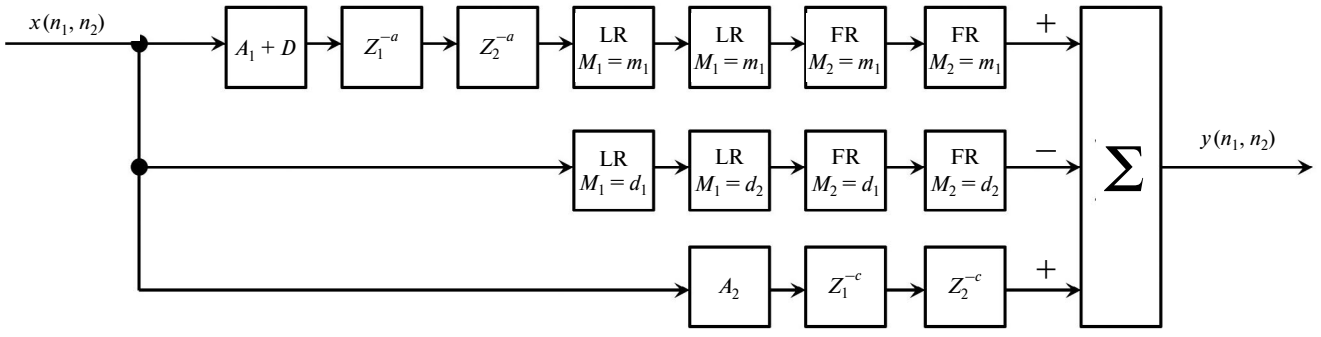


Рис. 2. Структурная схема ЛДП.

Таблица 1. Значения коэффициентов рециркуляторов для фильтров ЛУП и ЛДП

Фильтр ЛУП					
		1-я ветвь фильтра		2-я ветвь фильтра	
Размер маски	Дополнительный параметр D	Коэфф-ты СР и КР		Коэфф-ты СР	Коэфф-ты КР
7 × 7	25	3 и 3		5 и 3	5 и 3
7 × 5	16.67	3 и 3		5 и 3	4 и 2
7 × 3	20	3 и 0		5 и 3	2 и 2
Фильтр ЛДП					
		1-я ветвь фильтра		2-я ветвь фильтра	
Размер маски	Дополнительный параметр D	Коэфф-ты СР	Коэфф-ты КР	Коэфф-ты СР	Коэфф-ты КР
7 × 7	14	2 и 2	2 и 2	5 и 3	5 и 3
7 × 5	9.375	2 и 2	2 и 2	5 и 3	4 и 2
7 × 3	15	2 и 2	0 и 0	5 и 3	2 и 2

−1	−2	−3	−3	−3	−2	−1
−2	−4	10.67	10.67	10.67	−4	−2
−2	−4	10.67	10.67	10.67	−4	−2
−2	−4	10.67	10.67	10.67	−4	−2
−1	−2	−3	−3	−3	−2	−1

Рис. 3. Маска 7 × 5 фильтра ЛУП.

−1	−2	−3	−3	−3	−2	−1
−2	−4	3.375	12.75	3.375	−4	−2
−2	−4	12.75	31.5	12.75	−4	−2
−2	−4	3.375	12.75	3.375	−4	−2
−1	−2	−3	−3	−3	−2	−1

Рис. 4. Маска 7 × 5 фильтра ЛДП.

вследствие чего их алгоритмы стали способны автоматически строить неравносторонние апертуры.

2.1. Программная реализация алгоритма обработки фильтров ЛУП и ЛДП

Реализация алгоритма обработки фильтрами ЛУП и ЛДП заключается в последовательном и параллельном применении цепочек СР и КР.

СР и КР, как уже указанно выше, это базовые элементы рекурсивно-сепарабельных фильтров. Алгоритм работы рециркуляторов основан на вложенных циклах (по ширине и по высоте изображения). Алгоритмы работы СР и КР представлены в виде блок-схем на рис. 5 и 6.

В блок-схемах алгоритмов используются следующие обозначения: k – позиция текущего элемента по горизонтали; m – позиция текущего элемента по вертикали; i_1 – общее количество элементов по горизонтали во входной матрице; i_2 – общее количество элементов по горизонтали в выходной матрице; j_1 – общее количество элементов по вертикали во входной матрице; j_2 – общее количество элементов по вертикали в выходной матрице; $X(k, m)$ – входная матрица; $y(k, m)$ – выходная матрица.

Параллельное включение последовательных цепочек рециркуляторов с изменяющимися коэффициентами СР и КР приводит к реализации конфигурации разработанных фильтров.

3. ОЦЕНКА ВЛИЯНИЯ ЦИФРОВЫХ ФИЛЬТРОВ ЛУП И ЛДП НА РАБОТУ ТИС

Для проведения оценки влияния рекурсивно-сепарабельных цифровых фильтров на точность работы ТИС было взято 10 специальных изображений, отснятых на полигоне. Каждое изображение формировалось при определенных параметрах ТИС, которые позволяли ей видеть объекты интереса на определенных расстояниях.

Полигон для испытаний ТИС представляет собой неровный участок местности, на котором были расположены опознавательные знаки (опорные точки) (рис. 7).

Часть полученных изображений представлена на рис. 8.

3.1. Оценка быстродействия работы алгоритмов цифровых фильтров ЛУП и ЛДП

Рекурсивно-сепарабельный метод обработки изображений позволяет сократить количество

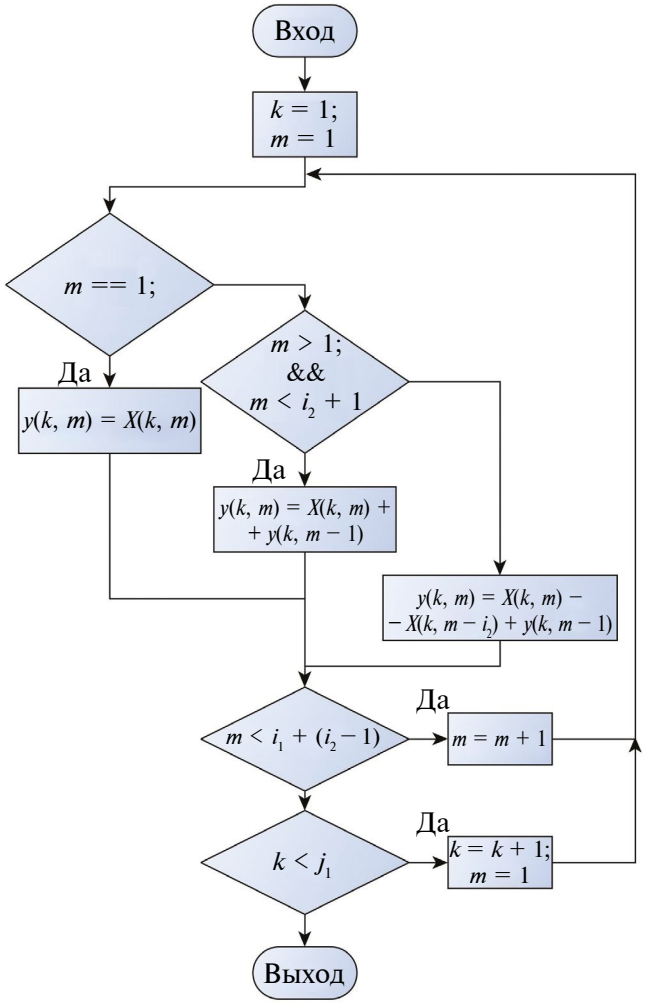


Рис. 5. Алгоритм работы СР.

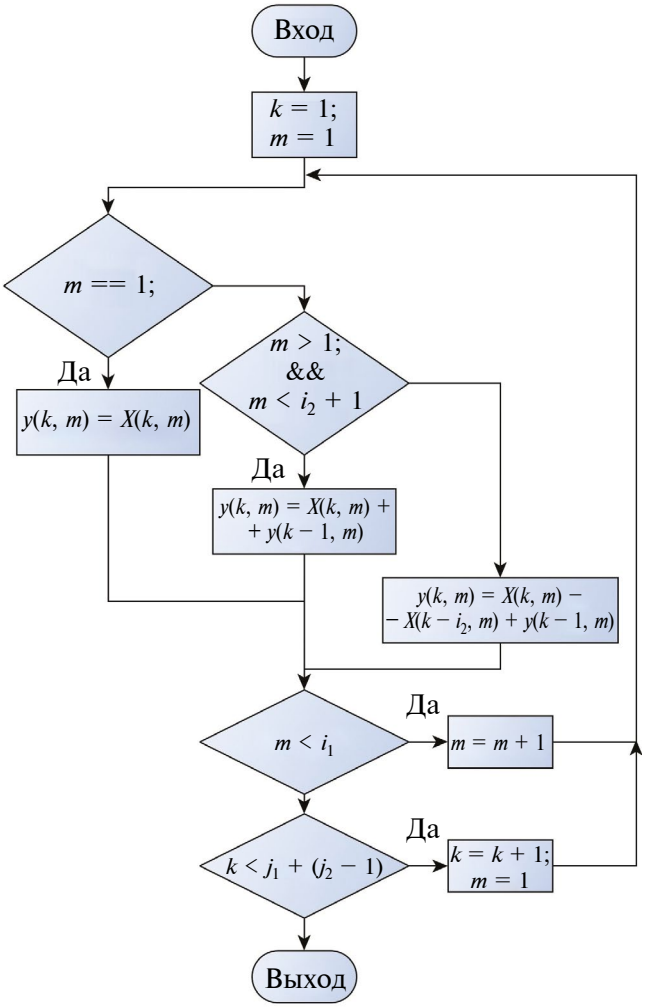


Рис. 6. Алгоритм работы КР.

вычислительных операций относительно стандартного метода двумерной свертки, тем самым он ускоряет процесс фильтрации.

Для определения ускорения работы фильтров важен размер исходных изображений, подвергаемых фильтрации. Изображения, полученные с ТИС, используемые для эксперимента, имеют размер 1032 × 772 пиксела. Скорость выполнения алгоритмов, реализующих фильтры, зависит от конфигурации (значений размеров рециркуляторов, представленных в табл. 1) и структуры фильтров (см. рис. 1 и 2). Результат ускорения работы фильтров с применением неравносторонних апертур приведен в табл. 2. Представленные результаты получены путем усреднения измерений времени работы алгоритмов для 10 изображений.

Рис. 7. Вид полигона с места расположения ТИС.

Таблица 2. Время работы и ускорение фильтров ЛУП и ЛДП

Размер апертуры	ЛУП		ЛДП	
	Время, мс	Ускорение, раз	Время, мс	Ускорение, раз
7 × 7	29.94	—	39.82	—
7 × 5	21.40	1.4	28.46	1.4
7 × 3	12.44	2.4	16.54	2.4

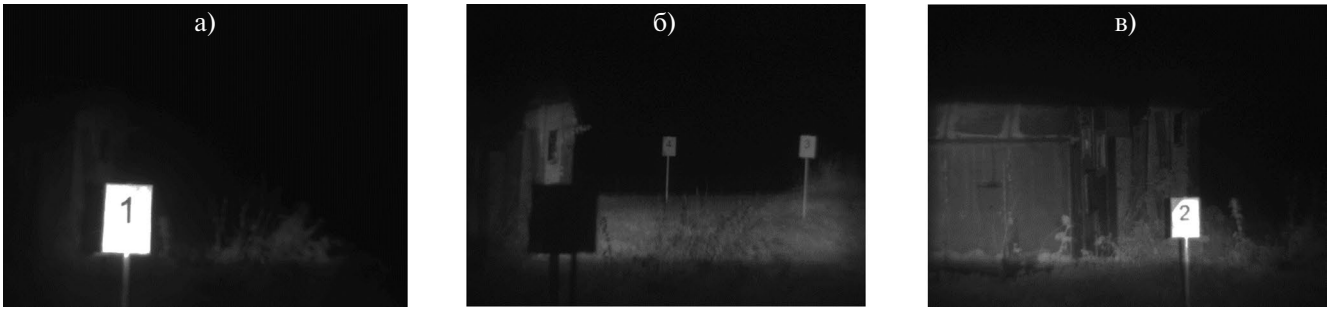


Рис. 8. Тестовые изображения: а) ТИ № 1 Транспарант-1; б) ТИ № 2 Транспарант-3; в) ТИ № 3 Транспарант-2.

Измерения скорости работы алгоритмов фильтров ЛУП и ЛДП проводились на платформе (процессоре) Core i5–9600K, 3.7 ГГц под управлением операционной системы Windows 10.

Подробное описание влияния неравносторонних апертур на характеристики цифровых фильтров ЛУП и ЛДП дано в ранее проделанной и опубликованной работе [10].

3.2. Оценка эффективности обработки модифицированными фильтрами ЛУП и ЛДП

Для оценки эффективности обработки для каждой маски фильтра производилось изменение положительной части матрицы значений путем ее пошагового увеличения за счет добавления дополнительных коэффициентов [11]. Увеличение происходит за счет добавления коэффициента поднятия от 0 до 10 к исходному значению положительной части маски. Результатом оценки эффективности обработки изображения будет минимальное отклонение полученных измерений дальности расположения тестовых объектов от эталонного измерения, выполненного лазерным дальномером. По полученным результатам будут определены оптимальные коэффициенты фильтрации. В ранее опубликованных материалах оптимальный коэффициент выбирался на основе измерения разрешения обработанных изображений [10]. В результате работы фильтров с измененными масками были получены наборы изображений.

Для каждого изображения были проведены измерения среднеквадратического отклонения (СКО), пикового отношения сигнал/шум (ПОСШ). Формулы, используемые для расчета характеристик, приведены ниже:

$$\text{СКО} = \sqrt{\frac{\sum_{i=0}^{N_1}\sum_{j=0}^{N_2}\left(x_{ij} - y_{ij}\right)^2}{\left(N_1 + 1\right)\left(N_2 + 1\right)}},$$

где x_{ij} – текущий отсчет входного изображения; y_{ij} – текущий отсчет выходного изображения; N_1, N_2 – ширина и высота изображения;

$$\text{ПОСШ} = 20\lg\frac{\text{MAX}}{\sigma},$$

где MAX – максимальная яркость изображения; σ – среднеквадратическое отклонение.

В итоге были получены результаты оценки изменения СКО и ПОСШ для 10 тестовых изображений, обработанных фильтрами ЛДП и ЛУП с различными размерами апертуры (7×3, 7×5, 7×7), при изменении коэффициента обработки от 1 до 10. Получено большое количество результатов (132 измерения для 1-го тестового изображения). Приведем ряд закономерностей из них, которые наблюдались в полученных результатах:

• при обработке фильтрами с апертурами 7×3 и 7×7 элементов наблюдается стабильный рост ПОСШ с ростом значений коэффициентов обработки, что свидетельствует о снижении шума, также значения СКО постепенно стремятся к нулю, что говорит о наличии изменений, вносимых после обработки;

• при обработке фильтрами с апертурой 7×5 элементов наблюдается подъем ПОСШ при первых 2–3 изменениях коэффициента фильтрации, однако далее с его ростом происходит снижение ПОСШ, при этом значение СКО при всех измерениях колебалось в диапазоне от 0.01 до 0.1.

В табл. 3 представлены результаты измерения ПОСШ для одного из тестовых изображений, обработанных фильтром ЛДП с разными размерами апертуры обработки, подтверждающие сформулированные выше выводы.

Для наглядности изменений, произошедших на тестовых изображениях, приведем ряд примеров зон измерения дальности, расположенных на различных расстояниях. Результаты обработки ТИ № 1 фильтрами ЛУП и ЛДП с масками размерностью 7×7 элементов представлены на рис. 9. Для фильтра ЛУП центральный коэффициент равен 17, для фильтра ЛДП – 58.

Результаты обработки ТИ № 2 фильтрами ЛУП и ЛДП с масками размерностью 7×5 элементов показаны на рис. 10. Для фильтра ЛУП центральный коэффициент равен 19, для фильтра ЛДП – 36.

Таблица 3. Результаты измерения ПОСШ при обработке фильтром ЛДП

	Размер апертуры фильтра		
Результат измерения ПОСШ	7 × 3	7 × 5	7 × 7
	47.0431	77.0365	39.1443
	42.6313	81.1998	38.1657
	46.0926	78.5284	41.5324
	48.9906	78.4868	43.9070
	50.2213	76.3267	45.3991
	52.4076	79.2550	46.9840
	52.8337	76.5747	47.9170
	55.0987	77.7039	49.2083
	55.1801	76.1263	49.8406
	57.5485	77.0323	50.9732
	57.2802	75.7893	51.3757

Результаты обработки ТИ № 3 фильтрами ЛУП и ЛДП с масками размерностью 7 × 3 элементов представлены на рис. 11. Для фильтра ЛУП центральный коэффициент равен 19, для фильтра ЛДП – 30.

На полученных в результате фильтрации изображениях наблюдаются значительное увеличение резкости изображения и повышение четкости границ объектов интереса.

3.3. Оценка влияния модифицированных фильтров ЛУП и ЛДП на точность работы ТИС

Для оценки влияния модифицированных цифровых фильтров ЛУП и ЛДП на точность работы ТИС был разработан программный модуль, который позволяет измерить расстояние от камеры до объекта интереса при контроле времени обработки изображений [12].

Рис. 9. Результат обработки ТИ № 1: а) исходное изображение; обработанные изображения: б) фильтр ЛУП, в) фильтр ЛДП.

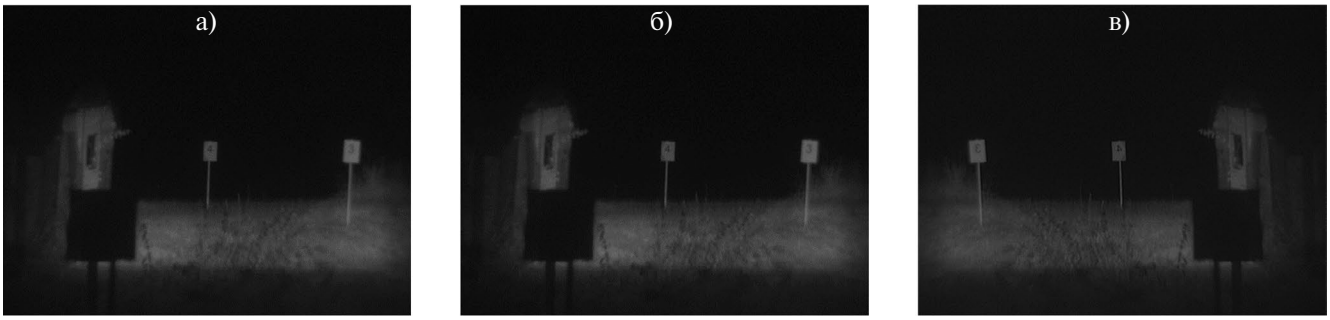


Рис. 10. Результат обработки ТИ № 2: а) исходное изображение; обработанные изображения: б) фильтр ЛУП, в) фильтр ЛДП.

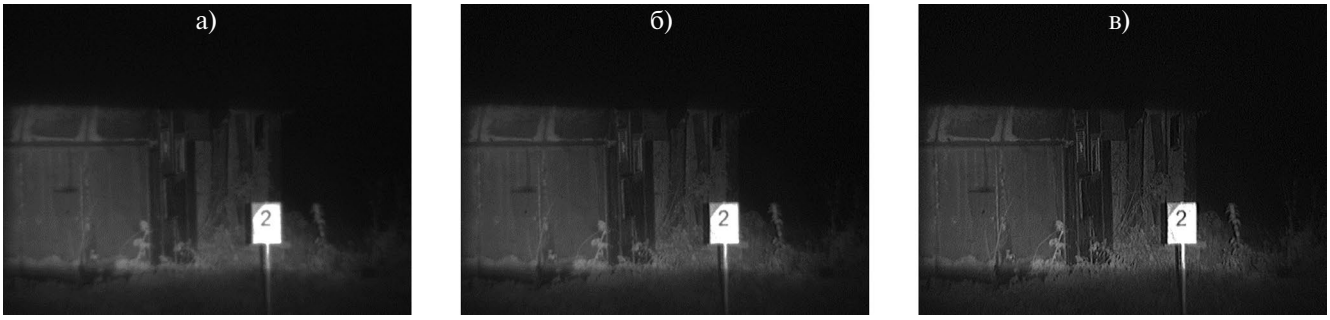


Рис. 11. Результат обработки ТИ № 3: а) исходное изображение; обработанные изображения: б) фильтр ЛУП, в) фильтр ЛДП.

По результатам измерений произведено сравнение расстояний, полученных с изображений без предобработки фильтрами, отснятых посредством ТИС и расстояний, измеренных с изображений с предобработкой модифицированными фильтрами, также полученных посредством ТИС с реальным расстоянием, полученным в ходе прямых измерений. Прямое измерение проводилось лазерным дальномером непосредственно от места расположения ТИС до объектов интереса, представленных на рис. 8.

В ходе сравнения полученных данных определяются коэффициент и размер маски фильтра, при котором удалось определить расстояние с наименьшим отклонением от результатов прямых измерений посредством лазерного дальномера.

Полученные данные измерений расстояния для ТИ № 1, расположенного на расстоянии 11 м, представлены в табл. 4.

В табл. 4 жирным шрифтом выделены значения с наименьшим отклонением от расстояния, полученного при прямых измерениях, проделанных с помощью лазерного дальномера.

Из результатов, представленных в таблице, видно, что при обработке фильтром ЛУП с любой размерностью апертуры наилучший результат достигается при значениях коэффициента фильтрации, равном 1 или 2, при этом с ростом размерности апертуры повышается точность определения расстояния. Разница между эталонным результатом, полученным с помощью лазерного дальномера и обработанным изображением фильтром ЛУП с оптимальным коэффициентом равным 1, составляет 0.002 м.

Для фильтра ЛДП наблюдается рост оптимального коэффициента обработки с ростом размерности апертуры цифрового фильтра. В итоге наилучший результат также достигается при апертуре 7 × 7 элементов, но уже с оптимальным коэффициентом, равным 10. Разница между результатами эталонного измерения и измерения по обработанному изображению составила 0.019 м.

Полученные данные измерений расстояния для ТИ № 2, расположенного на расстоянии 19.06 м, представлены в табл. 5.

На изображении, расположенном на расстоянии примерно 19 м, наблюдается разность в результатах двух исследуемых фильтров. Фильтр ЛУП наилучшие результаты по оценке дальности показывает при коэффициентах обработки от 5 до 7 вне зависимости от размеров апертуры обработки. Наилучший результат был достигнут при размере апертуры 7 × 5 элементов с оптимальным коэффициентом обработки, равным 5. Отличие от эталонного измерения составило 0.002 м, что на 0.558 м точнее, чем измерение ТИС.

Обработка фильтром ЛДП показала противоположный результат обработки объекта, расположенного на расстоянии 11 м. В данном эксперименте наблюдаются противоположные результаты, с ростом размеров апертуры обработки уменьшается значение оптимального коэффициента обработки. Отличие от эталонного

Таблица 4. Результаты измерения расстояния для ТИ № 1

Измерение дальности до объектов, м							
Зона № 1 – Транспарант № 1							
Лазерный дальномер		11 м					
ТИС		11.12 м					
Фильтр		ЛУП			ЛДП		
Размер маски		7 × 3	7 × 5	7 × 7	7 × 3	7 × 5	7 × 7
Коэффициенты	1	11.062	**11.077**	**11.002**	11.187	11.069	11.671
	2	**11.057**	11.096	11.033	11.076	11.069	11.388
	3	11.074	11.090	11.053	11.036	11.073	11.224
	4	11.071	11.091	11.068	**11.031**	11.068	11.140
	5	11.089	11.083	11.263	11.039	11.076	11.091
	6	11.080	11.093	11.047	11.035	11.082	11.035
	7	11.075	11.079	11.075	11.053	11.073	11.025
	8	11.089	11.079	11.091	11.058	11.075	11.049
	9	11.089	11.094	11.079	11.054	**11.060**	11.037
	10	11.078	11.092	11.070	11.066	11.072	**11.019**

Таблица 5. Результаты измерения расстояния для ТИ № 2

Измерение дальности до объектов, м							
Зона № 9 – Транспарант № 2							
Лазерный дальномер		19.06 м					
ТИС		18.5 м					
Фильтр		ЛУП			ЛДП		
Размер маски		7 × 3	7 × 5	7 × 7	7 × 3	7 × 5	7 × 7
Коэффициенты	1	18.787	19.148	18.901	19.041	18.974	19.455
	2	19.021	19.002	18.774	18.769	19.156	**19.099**
	3	18.962	18.937	19.009	18.812	19.213	18.678
	4	19.073	19.126	18.908	18.876	**19.046**	18.948
	5	**19.062**	**19.044**	19.078	18.839	19.149	18.933
	6	19.199	18.811	18.974	**19.056**	19.114	19.225
	7	18.810	18.968	**19.051**	18.822	19.103	18.671
	8	19.036	18.949	18.729	19.109	19.168	18.419
	9	19.117	19.078	18.925	19.016	18.928	18.399
	10	18.949	19.004	18.810	18.985	18.791	17.903

измерения составило 0.004 м, что на 0.556 м точнее, чем измерение ТИС.

Полученные данные измерений расстояния для ТИ № 3, расположенного на расстоянии 34.97 м, представлены в табл. 6.

Проанализированы полученные результаты измерения дальности до тестового объекта, располагающегося на расстоянии 34.97, и выявлен минимальный эффект от обработки тестовых изображений. Связано это с тем, что результат измерения дальности, полученный с помощью ТИС, очень близок к эталонному измерению и отличается всего на 0.19 м. Для фильтра ЛУП оптимальные коэффициенты были определены от 9 до 10 при использовании любой размерности фильтра.

Выявить зависимость изменения оптимальных коэффициентом от изменения размерности апертуры обработки для фильтра ЛДП не удается,

Таблица 6. Результаты измерения расстояния для ТИ № 3

Измерение дальности до объектов, м							
Зона № 3 – Транспарант № 3							
Лазерный дальномер		34.97 м					
ТИС		34.78 м					
Фильтр		ЛУП			ЛДП		
Размер маски		7 × 3	7 × 5	7 × 7	7 × 3	7 × 5	7 × 7
Коэффициенты	1	33.533	34.643	33.297	31.299	34.597	27.428
	2	34.067	34.659	33.806	32.820	34.528	28.674
	3	34.311	34.698	34.073	33.470	34.554	29.485
	4	34.465	34.705	34.304	33.715	34.558	30.250
	5	34.592	34.721	34.355	33.966	**34.601**	30.742
	6	34.609	34.741	34.465	34.188	34.548	31.189
	7	34.634	34.761	34.528	34.224	34.569	31.648
	8	34.631	34.767	34.557	34.266	34.561	31.995
	9	34.664	**34.800**	**34.591**	**34.542**	34.575	32.239
	10	**34.713**	34.761	34.590	34.384	34.550	**32.588**

поскольку значение оптимальных коэффициентов то снижается от маски 7 × 3 к маске 7 × 5 элементов, то обратно возрастает при увеличении размеров апертуры обработки до маски размерностью 7 × 7 элементов, что требует дополнительных исследований на расширенной базе экспериментальных данных.

4. ЗАКЛЮЧЕНИЕ

Из проделанной работы следует, что рекурсивно-сепарабельный метод обработки изображений фильтрами с неравносторонней апертурой позволяет сократить количество вычислительных операций относительно базовой размерности исходного фильтра, тем самым ускоряя процесс обработки.

Согласно полученным результатам (см. табл. 2) время выполнения фильтров с неравносторонней апертурой относительно исходной (равносторонней) сократилось в 1.4 раза для 7 × 5 (~8.5 мс) и в 2.4 раза для 7 × 3 (~17.5 мс).

В результате применения рекурсивно-сепабельных фильтров для ТИ № 1 точность определения расстояния при обработке фильтром ЛУП с апертурой размером 7 × 7 увеличилась на 0.118 м, а при обработке фильтром ЛДП с апертурой 7 × 7 – на 0.101 м.

Для ТИ № 2 точность определения расстояния при обработке фильтром ЛУП с апертурой размером 7 × 5 увеличилась на 0.544 м, а при обработке фильтром ЛДП с апертурой 7 × 5 – на 0.546 м.

Для ТИ № 3 точность определения расстояния при обработке фильтром ЛУП с апертурой размером 7 × 5 увеличилась на 0.02 м, а при обработке фильтром ЛДП с апертурой 7 × 5 – на 0.179 м.

Можно отметить, что при обработке фильтром ЛУП наблюдается закономерность: при росте расстояния до объекта интереса растет и значение оптимального коэффициента фильтрации. Это следует из того, что для объекта, расположенного на расстоянии 11 м, оптимальный коэффициент был равен 1–2, для объекта, расположенного на расстоянии 19.06 м, он составляет 5–7 и для объекта, расположенного на расстоянии 34.97 м, он равен 9–10.

Фильтр ЛУП оказался более эффективен и стабилен в результатах, чем фильтр ЛДП. Это связано с формируемой апертурой обработки. У фильтра ЛУП имеется усеченная верхняя часть, которая дает эффект сглаживания, устраняя шумы в исходном изображении. Из-за этого для применения в ТИС, при оценке дальности до объектов интереса в поле зрения системы, предлагается использовать фильтр ЛУП с неравносторонними апертурами.

5. ИСТОЧНИК ФИНАНСИРОВАНИЯ

Исследование выполнено за счет гранта Российского научного фонда № 21-79-10200 в ТУСУРе.

СПИСОК ЛИТЕРАТУРЫ

1. *Курячий М.И., Рудникович А.С., Семенов Е.В.* Телевизионные измерительные системы на базе процессоров TMS320C6000 // Доклады международной конференции DSPA. 2003. Т. 2. № 52. С. 1–4.
2. *Коротаев В.В., Краснящих А.В.* Телевизионные измерительные системы: Учеб. пособие. СПб.: СПбГУ ИТМО, 2008. 108 с.
3. *Гонсалес Р.С., Вудс Р.Е.* Цифровая обработка изображений. 3-е изд., испр. и доп. М.: Техносфера, 2012. 1104 с.
4. *Chun He, Ke Guo, Huayue Chen.* An Improved Image Filtering Algorithm for Mixed Noise; A special issue of Applied Sciences (ISSN 2076-3417), belongs to the section “Computing and Artificial Intelligence” // Appl. Sci. 2021. 11(21). 10358. https://doi.org/10.3390/app112110358
5. *Sironi A., Tekin B., Rigamonti R., Lepetit V., Fua P.* Learning Separable Filters // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2015. V. 37. I. 1. P. 94–106. https://doi.org/10.1109/TPAMI.2014.2343229
6. *Minh N. Do, Vetterli M.* Frame reconstruction of the laplacian pyramid. Acoustics, Speech, and Signal Processing, 1988. ICASSP-88 // 1988 International Conference on 6: 3641–3644. V. 6.
7. *Kamenskiy A.V.* High-speed recursive-separable image processing filters // Computer Optics 2022; 46(4): 659–665. https://doi.org/10.18287/2412-6179-CO-1063
8. *Kamenskiy A.V., Kuryachiy M.I., Krasnoperova A.S., Ilyin Y.V., Akaeva T.M., Boyarkin S.E.* / High-speed recursive-separable image processing filters with variable scanning aperture sizes // Computer Optics, 2023. 47(4): 605–613. https://doi.org/10.18287/2412-6179-CO-1240
9. *Сай С.В., Каменский А.В., Курячий М.И.* Современные методы анализа и повышения качества цифровых изображений: монография. Министерство науки и высшего образования Российской Федерации, Тихоокеанский государственный университет. Хабаровск: Изд-во Тихоокеан. гос. ун-та, 2020. 173 с.
10. *Куприянова К.С., Рылов К.А., Каменский А.В.* Влияние неравносторонних апертур на характеристики цифровых фильтров лапласиан “усеченная пирамида” и лапласиан “двойная пирамида” // Труды

33-й Междунар. конф. по компьютерной графике и машинному зрению. Москва, 19–21 сент. 2023 г. С. 482–491.

11. *Смирнов В.М.* Влияние размера апертуры на степень устранения импульсного шума при медианной фильтрации // Научная сессия ГУАП. 2019. С. 22–29.

12. *Kapustin V.V., Zahlebin A.S., Movchan A.K., Kuriyachiy M.I., Kruticov M.* Experimental assessment of the distance measurement accuracy using the active-pulse television measuring system and a digital terrain model // Computer Optics. 2022. V. 46(6): 948–954. https://doi.org/10.18287/2412-6179-CO-1114

INFLUENCE OF UNEQUILATERAL APERTURES OF DIGITAL FILTERS LAPLACIAN "TRUNCED PYRAMID" AND "DOUBLE PYRAMID" ON THE ACCURACY OF TELEVISION MEASURING SYSTEMS

© 2024 K. A. Rylov[a], K. S. Kupriyanova[a], A. V. Kamensky[a]

[a] Tomsk State University of Control Systems and Radioelectronics TUSUR
st. Lenina 40, Tomsk, Russia 634050

In the modern world, digital image processing requires increasing the speed of the processing methods and algorithms used. One way to improve performance is to transform spatial filters into filters with a recursively separable form of implementation. The recursion property implies the use of previous output values of a function to form the current sample. The property of separability is understood as division into processing by column and row of a matrix of digital image values. The transformation of spatial filters consists of changing the aperture of the masks into a non-orthogonal (non-equilateral) form, which reduces the number of computational operations and speeds up the processing process, while maintaining its efficiency. The paper presents a description of the non-equilateral apertures of the previously developed Laplacian digital filters "truncated pyramid" and "double pyramid". For non-equilateral apertures, results were obtained for the first time on their use for television measuring systems. From which it can be seen that a "truncated pyramid" Laplacian filter with non-equilateral processing apertures is recommended for use in TIS, since it increases the efficiency of measuring the range to objects of interest while reducing processing time. Based on the results of processing with modified filters, sets of processed images were obtained for each of the 10 original images. For each set of processed images, measurements of the peak signal-to-noise ratio, standard deviation and selection of the optimal central coefficient of the filter mask were carried out, for subsequent assessment of the effectiveness of processing with modified filters. The assessment of the influence of recursively separable Laplacian filters "truncated pyramid" and "double pyramid" with non-equilateral aperture masks on the accuracy of television measurement systems consisted of considering their influence on measuring the distance (from the camera) to the object of interest in the image, when – control of processing time. Based on the evaluation results, we can conclude that by using pre-processing of images with modified digital filters, the accuracy of measuring the distance from the camera to the measurement object is improved, while reducing processing time.

Keywords: digital filters; recursively separable filters; "truncated pyramid" Laplacian; "double pyramid" Laplacian; digital image processing; television measuring system; peak signal-to-noise ratio; standard deviation; two-dimensional discrete convolution

REFERENCES

1. *Kuryachiy M.I., Rudnikovich A.S., Semenov E.V.* [Television measuring systems based on TMS320C6000 processors]. Televizionnye izmeritel'nye sistemy na baze processorov TMS320C6000 // Reports of the international conference DSPA. 2003. V. 2. No. 52. P. 1–4.
2. *Korotaev V.V., Krasnyashchikh A.V.* [Television measuring systems]. Televizionnye izmeritel'nye sistemy; Textbook. St. Petersburg: St. Petersburg State University ITMO, 2008. 108 p.
3. *Gonzalez R.S., Woods R.E.* [Digital image processing]. Cifrovaya obrabotka izobrazhenij. 3rd edition, corrected and expanded. M.: Tekhnosphere, 2012. 1104 p.
4. *Chun He, Ke Guo, Huayue Chen.* An Improved Image Filtering Algorithm for Mixed Noise; A special issue of Applied Sciences (ISSN 2076-3417), belongs to the section "Computing and Artificial Intelligence" // Appl. Sci. 2021. 11(21). 10358. https://doi.org/10.3390/app112110358
5. *Sironi A., Tekin B., Rigamonti R., Lepetit V., Fua P.* Learning Separable Filters // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2015. V. 37. I. 1. P. 94–106. https://doi.org/10.1109/TPAMI.2014.2343229
6. *Minh N. Do, Vetterli M.* Frame reconstruction of the laplacian pyramid. Acoustics, Speech, and Signal Processing, 1988. ICASSP-88 // 1988 International Conference on 6: 3641–3644. V. 6.

7. *Kamenskiy A.V.* High-speed recursive-separable image processing filters // Computer Optics 2022; 46(4): 659–665.
https://doi.org/10.18287/2412-6179-CO-1063

8. *Kamenskiy A.V., Kuryachiy M.I., Krasnoperova A.S., Ilyin Y.V., Akaeva T.M., Boyarkin S.E.* / High-speed recursive-separable image processing filters with variable scanning aperture sizes // Computer Optics, 2023. 47(4): 605–613.
https://doi.org/10.18287/2412-6179-CO-1240

9. *Sai S.V., Kamensky A.V., Kuryachiy M.I.* [Modern methods of analyzing and improving the quality of digital images: monograph]. Sovremennye metody analiza i povysheniya kachestva cifrovyh izobrazhenij: monografiya; Ministry of Sciences and Higher Education of the Russian Federation, Pacific State University. Khabarovsk: Pacific Publishing House. state Univ., 2020. 173 p.

10. *Kupriyanova K.S., Rylov K.A., Kamenskiy A.V.* [Study of the unevenness of the aperture edge on the characteristics of digital filters Laplacian "truncated pyramid" and Laplacian "double pyramid"]. Vliyanie neravnostoronnih apertur na harakteristiki cifrovyh fil'trov laplasian "usechennaya piramida" i laplasian "dvojnaya piramida" // Proceedings of the 33rd Int. conf. computer graphics and machine vision. Moscow, September 19–21. 2023. P. 482–491.

11. *Smirnov V.M.* [Measuring the size of the aperture on the degree of limitation of impulse noise with median filtering]. Vliyanie razmera apertury na stepen' ustraneniya impul'snogo shuma pri mediannoj fil'tracii // Scientific session of the Guap. 2019. P. 22–29.

12. *Kapustin V.V., Zahlebin A.S., Movchan A.K., Kuriyachiy M.I., Kruticov M.* Experimental assessment of the distance measurement accuracy using the active-pulse television measuring system and a digital terrain model // Computer Optics. 2022. V. 46(6): 948–954.
https://doi.org/10.18287/2412-6179-CO-1114

ПРОГРАММИРОВАНИЕ, 2024, № 3, с. 33–41

КОМПЬЮТЕРНАЯ ГРАФИКА И ВИЗУАЛИЗАЦИЯ

УДК 004.92

ПРЕОБРАЗОВАНИЕ ДАННЫХ ОБЛАКА ТОЧЕК В 3D-МОДЕЛИ С ПОМОЩЬЮ POINTNET++ И TRANSFORMER

© 2024 г. М. И. Сорокин[a], *, Д. Д. Жданов[a], **, А. Д. Жданов[a], ***

[a] *Университет ИТМО*

197101 Санкт-Петербург, Кронверкский проспект, д. 49, литер А, Россия

**E-mail: vergotten@gmail.com*

***E-mail: ddzhdanov@mail.ru*

****E-mail: andrew.gtx@gmail.com*

Поступила в редакцию: 19.01.2024 г.
После доработки: 19.01.2024 г.
Принята к публикации: 24.01.2024 г.

В данной работе представлен подход к восстановлению 3D-моделей из данных облака точек, основанный на использовании современных архитектур нейронных сетей. Основу метода составляют PointNet++ и Transformer. PointNet++ играет центральную роль, обеспечивая эффективное извлечение признаков и кодирование сложных геометрий 3D-сцен. Это достигается благодаря рекурсивному применению PointNet++ к вложенным разбиениям входного набора точек в метрическом пространстве. Выпуклая декомпозиция, являющаяся важным этапом в подходе, позволяет преобразовать сложные трехмерные объекты в набор более простых выпуклых форм. Это упрощает обработку данных и делает процесс восстановления более управляемым. Transformer затем обучает модель на этих признаках, что позволяет генерировать высококачественные реконструкции. Важно отметить, что Transformer используется исключительно для определения положения стен и границ объектов. Это сочетание технологий позволяет достичь высокой точности при восстановлении 3D-моделей. Основная идея метода заключается в сегментации облака точек на малые фрагменты, которые затем восстанавливаются как полигональные сетки. Для восстановления отсутствующих точек в данных облака точек применяется метод, основанный на алгоритме L1-Median и локальных признаках облака точек. Этот подход может адаптироваться к различным геометрическим структурам и исправлять ошибки топологического соединения. Предложенный метод был сравнен с несколькими современными подходами и показал свой потенциал в различных областях, включая архитектуру, инжиниринг, цифровизацию культурного наследия и системы дополненной и смешанной реальности. Это подчеркивает его широкую применимость и значительный потенциал для дальнейшего развития и применения в различных областях.

Ключевые слова: облако точек, трехмерные модели, нейронные сети, PointNet++, трансформер, выпуклое разложение

DOI: 10.31857/S0132347424030044, **EDN:** QARZUT

1. ВВЕДЕНИЕ

В области компьютерного зрения преобразование облака точек в трехмерную сцену представляет собой сложную проблему. Это охватывает широкий спектр областей, включая робототехнику, дополненную и виртуальную реальность, а также оцифровку культурного наследия. Облака точек, которые обычно получают с помощью лидара или других датчиков глубины, часто содержат шум, пробелы и выбросы, что затрудняет процесс точной и полной реконструкции трехмерной сцены. Кроме того, облака точек требуют большого объема данных и не имеют внутренней структуры или топологии, что требует надежных и эффективных методов обработки.

В рамках данной работы разрабатывается метод реконструкции трехмерной сцены из облака точек, который позволяет создать детализированную и полную трехмерную модель сцены в виде полигональных сеток. Эти сетки, состоящие из вершин, ребер и граней, образующих сетчатую структуру на поверхности объекта, являются распространенным форматом представления трехмерных объектов. Они идеально подходят для дальнейшей обработки, поскольку предоставляют информацию о топологии и связности поверхности объекта.

Метод объединяет преимущества глубокого обучения и геометрического анализа для создания высококачественных полигональных сеток из облаков точек. Глубокое обучение отлично

подходит для извлечения функций и представлений из сложных данных, таких как облака точек, в то время как геометрический анализ дает математический подход к изучению формы и структуры объектов. Объединив эти два подхода, можно учитывать как локальные, так и глобальные свойства облака точек, одновременно эффективно создавая полигональные сетки.

Основная концепция метода заключается в первоначальном сегментировании облака точек на небольшие фрагменты с помощью нейронной сети. Затем каждый фрагмент реконструируется как полигональная сетка. Эти отдельные сетки впоследствии объединяются в единую трехмерную модель сцены. Эта процедура повторяется на различных уровнях детализации для создания многоуровневой трехмерной модели сцены.

Для улучшения процесса метод включает в себя использование герметичных сеток и выпуклого разложения. Герметичные сетки, представляющие собой многоугольные сетки без каких-либо отверстий, зазоров или свободных краев, идеально подходят для представления твердых объектов в трехмерном пространстве, поскольку они гарантируют, что каждое ребро принадлежит ровно двум граням. Это имеет решающее значение для сохранения целостности трехмерной модели.

Выпуклая декомпозиция – еще один ключевой компонент подхода, который включает в себя разбиение сложной трехмерной формы на набор более простых выпуклых фигур. С такими формами проще работать в вычислительном отношении и их можно обрабатывать более эффективно. В данном методе каждый фрагмент облака точек раскладывается на выпуклые формы, а затем реконструируется в виде полигональной сетки.

Используя герметичные полигональные сетки и выпуклую декомпозицию, метод может обрабатывать сложные трехмерные структуры и обеспечивать точность и полноту реконструированной трехмерной сцены. Это делает подход надежным и универсальным, демонстрируя сильные возможности обобщения и адаптируемости к различным типам сцен и объектов.

2. АНАЛИЗ СУЩЕСТВУЮЩИХ МЕТОДОВ

Процесс создания трехмерных полигональных сетей из облаков точек является сложной задачей, которая требует детального анализа и извлечения объектов из данных облака точек. Этот процесс также включает в себя оптимизацию процесса обучения для генерации моделей объектов и восстановление отсутствующих точек. Работа со сложной геометрией и неполными данными может представлять особые трудности, особенно в реальных условиях. Однако, благодаря применению передовых архитектур нейронных сетей и инновационных методов обучения, эти проблемы можно преодолеть, что позволяет создавать высококачественные 3D-реконструкции.

Есть множество подходов к реконструкции трехмерных форм из облаков точек, которые можно условно разделить на две категории: методы, основанные на геометрии, и методы, основанные на обучении. Геометрические методы используют разнообразные алгоритмы для создания триангуляций [1], выпуклых оболочек [2], альфа-форм [3] или поверхностей множества точек из облаков точек [4]. Эти методы обычно требуют значительных вычислительных ресурсов и могут быть сложными в реализации, но они обеспечивают высокую точность и могут быть очень эффективными в определенных условиях.

С другой стороны, методы, основанные на обучении, используют методы машинного обучения для анализа облаков точек и создания трехмерных моделей. Эти методы могут быть более гибкими и масштабируемыми, чем геометрические методы, и они особенно полезны при работе с большими наборами данных или сложными формами. Однако они также могут требовать значительного количества данных для обучения и в некоторых случаях быть менее точными. Поэтому выбор метода зависит от конкретной задачи и доступных ресурсов. В области воссоздания 3D-форм из облаков точек некоторые методы обучения показали впечатляющие результаты. Эти подходы применяют нейронные сети для выделения характеристик из облаков точек и восстановления трехмерных объектов в разнообразных форматах. Вот несколько передовых методов, которые используются для обработки и анализа облаков точек в трехмерном пространстве:

• PointNet [5] – нейронная сеть, способная напрямую работать с облаками точек. Она уникальна тем, что учитывает перестановочную инвариантность точек во входных данных. PointNet предлагает универсальную архитектуру, которая может быть применена для решения различных задач, включая классификацию объектов, сегментацию частей и семантический анализ сцен;

• AtlasNet [6] – представляет собой передовой метод, который преобразует облака точек в треугольные сетки с помощью коллекции параметрических поверхностей, называемых атласами. Эти

атласы затем проецируются на 3D-структуру объекта. AtlasNet обучается синхронизировать атласы с геометрией объекта и уменьшать промежутки между ними, что способствует созданию более точных и подробных моделей;

• PCN [7] – подход, который способен реконструировать полную 3D-структуру объекта из неполного облака точек. PCN применяет три последовательно действующих модуля: дополнение, усиление и уточнение. PCN обучается восполнять отсутствующие части объекта и увеличивать плотность точек, что позволяет создавать более полные и детализированные модели.

В контексте изучения семантики и формы объектов на основе разреженных облаков точек RfD-Net [8] представляет собой инновационный подход. Эта архитектура разделяет процесс реконструкции семантического экземпляра на две ключевые задачи: глобальную локализацию объекта и прогнозирование локальной формы. RfD-Net разработан для работы с неполными облаками точек 3D-сцены, целью является одновременное понимание структуры 3D-объектов и восстановление сеток экземпляров объектов. Он использует разреженность данных облаков точек, уделяя особое внимание прогнозированию форм с высокой степенью детализации. Одной из особенностей RfD-Net является его способность справляться с проблемой увеличения сложности модели, что часто встречается в существующих методах. Это достигается за счет использования генератора форм, поддерживающего прямое неявное обучение. В каждом сегменте объекта облака точек содержатся детали формы, которые помогают неявному обучению функций, позволяя реконструировать поверхности с высокой степенью дискретизации. Еще один подход, реализующий принцип “сквозной обработки” (end-to-end), который заслуживает особого внимания, – это RevealNet [9]. Эта архитектура нейронной сети применяет методы совместного обучения для изучения характеристик цвета и геометрии. Она специально разработана для работы с неполными сканами RGB-D-сцены в целях идентификации отдельных экземпляров объектов и определения их полной геометрической структуры. RevealNet обладает полносверточной 3D-структурой, что обеспечивает эффективное семантическое завершение экземпляра для 3D-сканирований больших помещений за один проход. В результате сканируемая сцена разбивается на отдельные и полные трехмерные объекты, включая скрытые и невидимые части объекта, что демонстрирует потенциал и гибкость при работе с большими и сложными наборами данных. Эти методы представляют собой важные шаги в направлении более эффективного и точного анализа и восстановления трехмерных форм из облаков точек. Они открывают новые возможности для исследований и разработок в области компьютерного зрения и машинного обучения.

В данной работе представлен метод, который эффективно объединяет преимущества геометрических подходов и подходов, основанных на машинном обучении. Этот метод использует принципы глубокого обучения для выделения характеристик и восстановления скрытых трехмерных форм, в то время как геометрический анализ применяется для устранения шума, восстановления отсутствующих точек и извлечения структуры объекта из сцены. Одним из основных достоинств этого подхода, по сравнению с другими методами восстановления 3D-сцены из облаков точек, является его способность воспроизводить планировку сцены. Используя сеть, основанную на трансформере, адаптированную в соответствии с такими исследованиями, как PQ-Transformer [10] и Omni-PQ [11], можно определить положение и форму стен и потолков с помощью ограничивающих рамок. Это обеспечивает более точную оценку размера объекта и его расположения в пространстве. Более того, этот подход способен создавать полигональные сетки из скрытых 3D-форм, используя адаптированный BSP-Net [12]. Однако у этого подхода есть определенные ограничения. Для обучения автоэнкодера и других компонентов сети необходим значительный объем данных. Более того, этот подход совместим только с теми категориями объектов, которые были использованы при обучении. Следовательно, он не способен реконструировать объекты из других категорий или объекты с существенно измененной формой. В рамках применяемого метода используется алгоритм L1-Median [13] для выявления ключевых точек в структуре облака точек. Затем, с использованием этих ключевых точек, определяется главное направление и рассчитывается плотность локального облака точек. Этот процесс создает условия для дополнения недостающих точек, что способствует более полному отображению структуры облака точек.

3. ПОДХОД К РЕКОНСТРУКЦИИ 3D-СЦЕНЫ ИЗ ОБЛАКА ТОЧЕК

Предложенный метод представляет собой последовательность ключевых этапов. Начальный этап – это извлечение признаков, включающее

в себя обработку данных и выделение значимых характеристик для последующей сегментации. Следующий этап – формирование скрытых трехмерных форм из объектов. После этого происходят анализ и извлечение структуры сцены. Затем, на основе скрытых векторов, создаются трехмерные модели объектов. И наконец, все данные и модели объединяются для создания окончательной 3D-модели сцены. Этот процесс обеспечивает целостный подход к реконструкции 3D-сцены из облака точек.

В начальной фазе процесса извлечения признаков с использованием PointNet++ [14], усовершенствованной версии PointNet, происходит детектирование локальных и глобальных объектов из облака точек. PointNet++ строит иерархические отношения между точками, что позволяет лучше улавливать локальные структуры и обобщать сложные сцены. Этот процесс включает в себя несколько важных шагов. Сначала из облака точек случайным образом отбираются точки до фиксированного размера N с применением алгоритма выборки самой дальней точки (FPS). Затем каждая отобранная точка и ее k ближайших соседей подвергаются обработке для получения локальных объектов. Эти локальные характеристики отражают форму и текстуру поверхности в непосредственной близости от каждой точки. Для получения глобальных объектов проводится обработка всего набора точек, при этом для сбора информации используется операция агрегирования. Эти глобальные характеристики описывают структуру и контекст всего набора. В конце локальные и глобальные объекты объединяются для формирования окончательного представления объекта в виде набора точек размером $N \times D$, где D – размерность пространства объектов. Таким образом, начальный этап извлечения признаков является критическим для дальнейшего анализа и сегментации данных набора точек.

После формирования окончательных представлений объектов в виде облака точек, следующим шагом является семантическая сегментация. В рамках этого процесса каждая точка в облаке точек классифицируется на основе извлеченных признаков. Эти сгруппированные точки проходят через ряд слоев для прогнозирования класса объекта для каждой точки. В процессе извлечения признаков и сегментации используется функция перекрестной энтропии. Эта функция потерь измеряет разницу между предсказаниями модели и истинными метками. Для оптимизации процесса используется оптимизатор Адама, который адаптивно корректирует скорость обучения для каждого параметра, основываясь на оценках первого и второго моментов градиентов.

Эксперименты осуществлялись на двух наборах данных: ScanNet и ShapeNetv2. ScanNet представляет собой набор RGB-D-данных, включающий в себя 2.5 млн кадров, полученных в результате более чем 1500 сканирований. Эти данные аннотированы позициями 3D-камер, реконструкциями поверхности и семантической сегментацией на уровне отдельных экземпляров.

Рис. 1. Наборы данных ScanNet и ShapeNetv2.

Рис. 2. Результаты семантической сегментации на уровне классов и экземпляров сцен.

В свою очередь, ShapeNetv2 является набором данных 3D-моделей, включающим в себя 57 категорий объектов (рис. 1).

На рис. 2 представлены результаты семантической сегментации на уровне классов и сегментации на уровне экземпляров сцен.

Слева направо рисунок демонстрирует:

- исходное облако точек – первичные 3D-данные, подлежащие анализу;
- семантический прогноз на уровне классов – определение класса, к которому относится каждая точка в облаке;
- семантические истинные метки классов для каждой точки, используемые для обучения и проверки модели;
- семантический прогноз на уровне экземпляров – определение, к какому конкретному отдельному объекту принадлежит каждая точка;
- семантические истинные метки объекта для каждой точки на уровне экземпляров, используемые для обучения и проверки модели.

На этапе извлечения макета облака точек преобразуются в прямоугольные блоки с помощью сети трансформера. Модуль голосования генерирует прогнозы блоков. Затем эти прогнозы обрабатываются декодером трансформера для дальнейшего уточнения. После нескольких слоев прямого распространения и подавления немаксимальных значений (NMS) эти прогнозы становятся окончательными ограничивающими рамками, которые определяют положения и размеры стен и потолков в пространстве. На рис. 3 демонстрируются результаты извлечения макета. Прямоугольники, которые символизируют стены сцен, кодируются цветом. Зеленый цвет соответствует истинным положительным прогнозам, в то время как желтый цвет указывает на ложные положительные прогнозы.

Ключевым элементом подхода является выпуклое разложение на герметичных сетках, полученных из набора данных ShapeNet с помощью TSDF Fusion [15]. Сеть, обученная представлять 3D-форму через выпуклое разложение, использует принцип бинарного разделения пространства

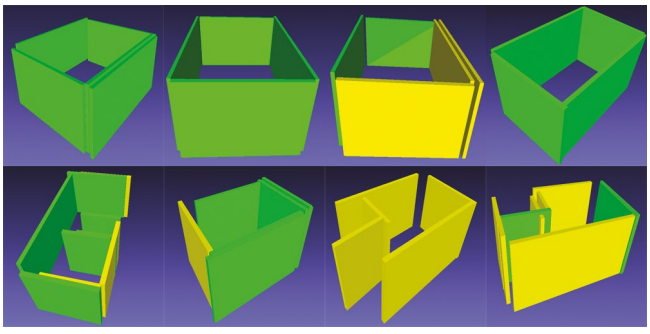


Рис. 3. Визуализация результатов извлечения макета.

(BSP), который заключается в рекурсивном разделении пространства для формирования выпуклых множеств. В процессе обучения сеть кодирует 3D-форму в скрытый вектор, который захватывает основные особенности входной формы. Декодер сети затем принимает этот скрытый вектор и декодирует его в набор параметров, которые определяют BSP-дерево. Это BSP-дерево используется для генерации набора выпуклых тел, которые аппроксимируют исходную 3D-форму. Сеть обучается восстанавливать форму, минимизируя функцию потерь между истинной формой и формой, восстановленной сетью. Потери восстановления вычисляются как среднеквадратичная ошибка между истинными и предсказанными значениями плотности. Это означает, что потери восстановления штрафуют модель за разницу между истинной формой и формой, восстановленной сетью. Градиенты функции потерь вычисляются и распространяются обратно через сеть. Затем параметры сети обновляются с использованием алгоритма оптимизации Adam для минимизации потерь. Результатом являются компактные полигональные сетки, которые идеально подходят для представления острых геометрических форм, гарантированно герметичны и могут быть легко параметризованы. На рис. 4 показаны предсказанные и истинные 3D-модели.

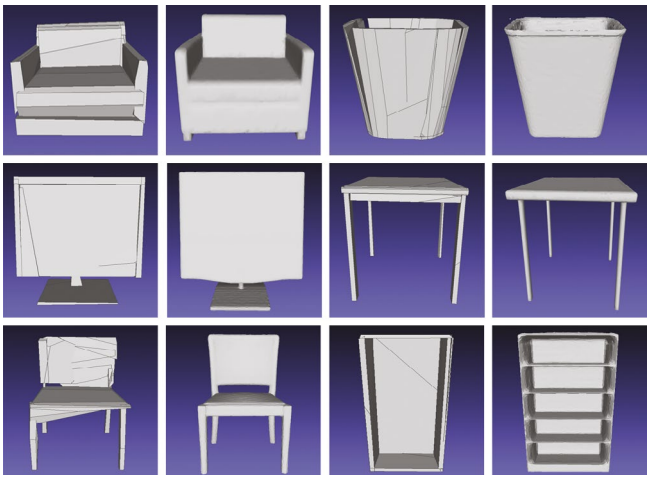


Рис. 4. Сравнение предсказанных (слева) и истинных (справа) 3D-моделей.

После определения формы, ориентации и расположения объектов на сцене эти данные собираются в полноценную 3D-модель. Модель, представляющая собой полигональную сетку, создается на основе предсказанных 3D-форм объектов и прямоугольников макета. Пример такой интегрированной 3D-модели сцены можно увидеть на рис. 5.

На рис. 6 представлены сегментированные облака точек и их восстановленные 3D-модели сцен.

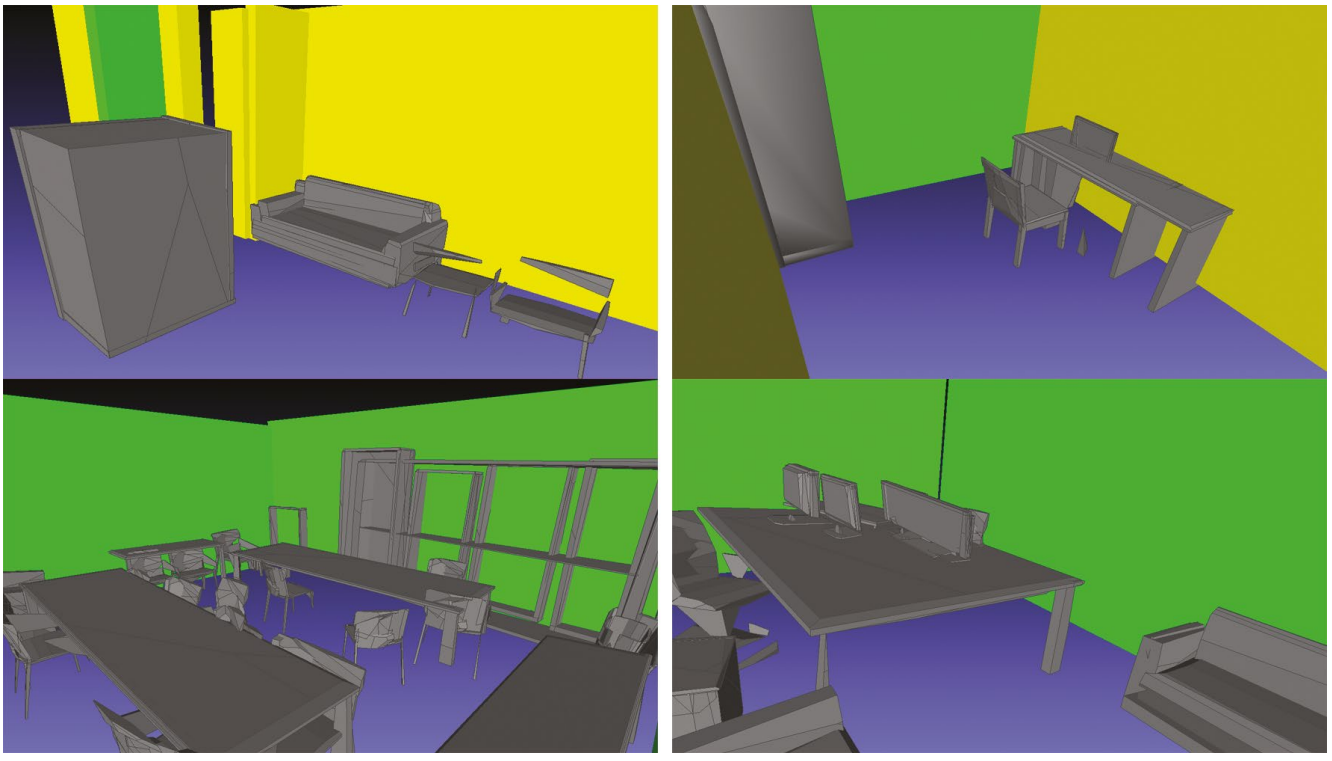


Рис. 5. Формирование 3D-модели сцены из предсказанных данных.

Рис. 6. Сегментированные облака точек и восстановленные 3D-модели сцен.

4. ОЦЕНКА ЭФФЕКТИВНОСТИ ПРЕДЛОЖЕННОГО МЕТОДА

В данной работе для оценки качества получаемых 3D-форм объектов, представленных в виде полигональных сеток, применяются метрики AP_mesh, Precision_mesh и Recall_mesh. Эти метрики позволяют оценить эффективность предложенного метода в обнаружении и восстановлении объектов из облаков точек с использованием скрытых векторов.

AP_mesh представляет собой среднюю точность для полигональных сеток, которая вычисляется как среднее арифметическое точностей для различных порогов пересечения по объединению (IoU) между предсказанными и истинными сетками. Precision_mesh определяет долю правильно обнаруженных объектов среди всех предсказанных, вычисляемую как отношение числа истинно положительных (TP) к сумме истинно положительных и ложно положительных (FP) сеток.

Recall_mesh, в свою очередь, представляет собой долю правильно обнаруженных объектов среди всех реальных, вычисляемую как отношение числа истинно положительных (TP) к сумме истинно положительных и ложно отрицательных (FN) сеток. Эти метрики позволяют оценить, насколько хорошо метод обнаруживает и восстанавливает объекты из облаков точек с использованием различных порогов пересечения по объединению (IoU) между предсказанными и истинными сетками.

В процессе тестирования использовалась выборка, включающая в себя восемь категорий объектов: стол, стул, книжная полка, диван, мусорное ведро, шкаф, дисплей и ванна (табл. 1). Для формирования этой выборки применялся метод TSDF Fusion, который позволяет восстанавливать 3D-поверхности объектов на основе набора RGB-D-изображений. Метод TSDF Fusion применяет специальное скалярное поле, которое хранит расстояние от каждой точки в пространстве до ближайшей поверхности. Этот метод объединяет несколько таких полей, полученных с разных точек зрения, в одно общее поле. Затем общее поле преобразуется в полигональную сетку с помощью алгоритма Marching Cubes. Благодаря методу TSDF Fusion возможно получение гладких и водонепроницаемых сеток из шумных и неполных данных.

Для оценки и сравнения с другими методами в контексте 3D-реконструкции объектов используется метрика mAP (средняя средняя точность).

Таблица 1. Метрики качества восстановления 3D-формы для различных типов объектов при пороге IoU0.1

Тип объекта	AP_mesh (%)	Precision_mesh (%)	Recall_mesh (%)
Стол	37.455	44.518	55.372
Стул	80.977	57.855	92.924
Книжная полка	38.209	33.333	54.067
Диван	44.099	41.727	55.238
Мусорное ведро	62.604	47.619	84.112
Шкаф	23.892	23.256	42.636
Дисплей	75.123	61.786	87.500
Ванна	50.928	55.455	52.586

Таблица 2. Сравнение предложенного метода с аналогичными работами. Оценки mAP (средняя точность) измеряются при пороге IoU для сетки 0.5

	Дисплей	Ванна	Мусорное ведро	Диван	Стул	Стол	Шкаф	Книжная полка	mAp
RevealNet	13.16	13.64	18.19	24.79	15.87	11.28	8.60	10.60	14.52
RfD-Net	26.67	27.57	23.34	15.71	12.23	1.92	14.48	13.39	16.90
Наш метод	**28.32**	**26.29**	**25.54**	**28.37**	**10.28**	**15.69**	**20.24**	**14.65**	**21.17**

Эта метрика позволяет оценить качество восстановленных сеток путем их сравнения с эталонными сетками с использованием метрики IoU (Intersection over Union) для сеток. При установке порога IoU для сеток на уровне 0.5 восстановленная сетка должна перекрывать по крайней мере 50% эталонной сетки, чтобы считаться правильно восстановленной. Это позволяет оценить, насколько точно восстановленная сетка соответствует истинной форме объекта (табл. 2).

5. ЗАКЛЮЧЕНИЕ

В данной работе предлагается подход к созданию 3D-моделей на основе данных облака точек, который успешно решает задачи расположения элементов, позиционирования объекта и восстановления полигональных сеток. Данный метод применяет передовые архитектуры нейронных сетей, включая PointNet++ для извлечения признаков. Это позволяет модели улавливать сложные зависимости и отношения между различными частями данных, что в свою очередь помогает в создании более точных и детализированных 3D-моделей. PointNet++ является эффективной нейронной сетью, способной обрабатывать облака точек напрямую, захватывая сложные геометрические структуры, характерные для 3D-данных. Transformer, в свою очередь, представляет собой модель, которая использует механизмы самовнимания для уделения внимания различным частям входных данных при вычислении представления данных. В этом контексте Transformer применяется для извлечения макета из данных. Ключевым элементом данного метода является выпуклое декомпозирование, т. е. процесс, в ходе которого сложная форма разбивается на более простые выпуклые части. Выпуклые формы обладают более простыми математическими свойствами, которые можно использовать для эффективных вычислений, что делает их более удобными для работы в различных приложениях, таких как компьютерная графика, физические симуляции, рендеринг и 3D-печать. Используя эти техники в совокупности, предложенный метод эффективно кодирует сложные геометрии 3D-сцен и генерирует реконструкции. Кроме того, он решает проблему отсутствующих точек, применяя специальные методы для восстановления этих точек. Предложенный метод продемонстрировал обещающие результаты и имеет потенциал для применения в различных областях, включая архитектуру, инженерию, цифровизацию культурного наследия, а также системы дополненной и смешанной реальности и другие области, где требуется воссоздание 3D-сцен.

6. ИСТОЧНИК ФИНАНСИРОВАНИЯ

Эта работа была поддержана Российским научным фондом, грант № 22-11-00145.

СПИСОК ЛИТЕРАТУРЫ

1. *Liu N., Lin B., Lv G., Zhu A.X., Zhou L.* A Delaunay triangulation algorithm based on dual-spatial data orga-

nization // PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science. 2019. P. 19–31.

2. *Ivanovsky S.A., Preobrazhensky A.S., Simonchik S.K.* Algorithms for computational geometry. Convex hulls: simple algorithms // Computer tools in education. 2007. P. 3–16.
3. *Attali D., Lieutier A., Salinas L.* Vietoris-Rips complexes also provide topologically correct reconstructions of sampled shapes // Proceedings of the twenty-seventh annual symposium on Computational geometry. ACM. 2011. P. 491–500.
4. *Guennebaud G., Gross M.* Algebraic point set surfaces. ACM Transactions on Graphics (TOG). 2007. V. 26. No. 3. Article 23.
5. *Qi C.R., Su H., Mo K., Guibas L.J.* Pointnet: Deep learning on point sets for 3d classification and segmentation // Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE. 2017. P. 652–660.
6. *Groueix T., Fisher M., Kim V.G., Russell B.C., Aubry M.* Atlasnet: A papier-mâché approach to learning 3d surface generation // Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE. 2018. P. 216–224.
7. *Yuan W., Khot T., Held D., Mertz C., Hebert M.* Pcn: Point completion network // 2018 International Conference on 3D Vision (3DV). IEEE. 2018. https://doi.org/10.1109/3DV.2018.00088
8. *Yinyu Nie, Ji Hou, Xiaoguang Han, Matthias Nießner.* RfD-Net: Point Scene Understanding by Semantic Instance Reconstruction // Computer Vision and Pattern Recognition. 2020.
9. *Ji Hou, Angela Dai, Matthias Nießner.* RevealNet: Seeing Behind Objects in RGB-D Scans // Computer Vision and Pattern Recognition. 2019.
10. *Xiaoxue Chen, Hao Zhao, Guyue Zhou, Ya-Qin Zhang.* PQ-Transformer: Jointly Parsing 3D Objects and Layouts from Point Clouds // Computer Vision and Pattern Recognition. 2021.
11. *Huan-ang Gao, Beiwen Tian, Pengfei Li, Xiaoxue Chen, Hao Zhao, Guyue Zhou, Yurong Chen, Hongbin Zha.* From Semi-supervised to Omni-supervised Room Layout Estimation Using Point Clouds // Computer Vision and Pattern Recognition. 2023.
12. *Zhiqin Chen, Andrea Tagliasacchi, Hao Zhang.* Learning Mesh Representations via Binary Space Partitioning Tree Networks // Computer Vision and Pattern Recognition. 2021.
13. *Wei Cao, Jiayi Wu, Yufeng Shi, Dong Chen.* Restoration of Individual Tree Missing Point Cloud Based on Local Features of Point Cloud. Terrestrial and Mobile Mapping in Complex Indoor and Outdoor Environments. 2022.
14. *Qi C.R., Yi L., Su H., Guibas L.J.* PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space // Proceedings of the Computer Vision and Pattern Recognition. 2017.
15. *Zhaofeng Niu, Yuichiro Fujimoto, Masayuki Kanbara, Taishi Sawabe, Hirokazu Kato.* DFusion: Denoised TSDF Fusion of Multiple Depth Maps with Sensor Pose Noises // Computer Vision and Machine Learning for Intelligent Sensing Systems. 2022.

CONVERSION OF POINT CLOUD DATA TO 3D MODELS USING POINTNET++ AND TRANSFORMER

© 2024 M. I. Sorokin[a], D. D. Zhdanov[a], A. D. Zhdanov[a]

[a] *ITMO University*

Kronverkskii pr. 49A, St. Petersburg, 197101 Russia

This work presents an approach to reconstructing 3D models from point cloud data, based on the use of modern neural network architectures. The basis of the method is PointNet++ and Transformer. PointNet++ plays a central role, providing efficient feature extraction and encoding of complex 3D scene geometries. This is achieved by recursively applying PointNet++ to nested partitions of the input point set in metric space. Convex decomposition, an important step in the approach, allows transforming complex three-dimensional objects into a set of simpler convex shapes. This simplifies data processing and makes the reconstruction process more manageable. The Transformer then trains the model on these features, allowing for the generation of high-quality reconstructions. It is important to note that the Transformer is used exclusively to determine the position of walls and object boundaries. This combination of technologies allows achieving high accuracy in the reconstruction of 3D models. The main idea of the method is to segment the point cloud into small fragments, which are then restored as polygonal meshes. To restore missing points in point cloud data, a method based on the L1-Median algorithm and local point cloud features is used. This approach can adapt to various geometric structures and correct topological connection errors. The proposed method was compared with several modern approaches and showed its potential in various fields, including architecture, engineering, digitization of cultural heritage, and augmented and mixed reality systems. This underscores its wide applicability and significant potential for further development and application in various fields.

Keywords: point cloud, 3D models, neural networks, PointNet++, transformer, convex decomposition

ПРОГРАММИРОВАНИЕ, 2024, № 3, с. 42–53

КОМПЬЮТЕРНАЯ ГРАФИКА И ВИЗУАЛИЗАЦИЯ

УДК 004.925.3

МЕТОД УПОРЯДОЧИВАНИЯ ОБЛАКОВ ТОЧЕК ДЛЯ ВИЗУАЛИЗАЦИИ НА КОНВЕЙЕРЕ ТРАССИРОВКИ ЛУЧЕЙ

© 2024 г. П. Ю. Тимохин[a], *, М. В. Михайлюк[a], **

[a]ФГУ "ФНЦ Научно-исследовательский институт системных исследований РАН"
117218 Москва, Нахимовский пр-т, 36/1
**E-mail: p_tim@bk.ru*
***E-mail: mix@niisi.ras.ru*

Поступила 19.01.2024 г.
После доработки 19.01.2024 г.
Принята к публикации 24.01.2024 г.

В настоящее время активно развивается оцифровка объектов окружающей среды (растительности, рельефа, архитектурных сооружений и др.) в виде облаков точек. Интеграция таких оцифрованных объектов в системы виртуального окружения позволяет повысить качество моделируемой обстановки, однако требует эффективных методов и алгоритмов визуализации больших объемов точек в реальном времени. В данной статье исследуется решение этой задачи на современных многоядерных графических процессорах с поддержкой аппаратного ускорения трассировки лучей. Предлагается модифицированный метод разбиения исходного неупорядоченного облака точек на группы точек, визуализация которых эффективно распараллеливается на ядрах трассировки лучей. В работе описан алгоритм построения таких групп с помощью чередующихся массивов индексов точек, который работает быстрее альтернативных решений на связных списках, а также имеет меньшие накладные расходы памяти. Предложенные метод и алгоритм были реализованы в программном комплексе визуализации облаков точек и апробированы на ряде оцифрованных объектов окружающей среды. Результаты апробации подтвердили эффективность предложенных решений и возможность их применения в системах виртуального окружения, видеотренажерных и геоинформационных системах, виртуальных лабораториях и др.

Ключевые слова: виртуальное окружение, визуализация, реальное время, трассировка лучей, облако точек

DOI: 10.31857/S0132347424030054, **EDN:** QAOZYR

1. ВВЕДЕНИЕ

В настоящее время во многих областях человеческой деятельности (архитектура, промышленность, геология, археология, лесное хозяйство и др.) активно внедряется оцифровка объектов окружающей среды с помощью дистанционного 3D-сканирования [1, 2]. Результатом такой оцифровки является набор несвязанных между собой точек (*облако точек*), каждой из которых присвоено одно или несколько скалярных значений (цвет, температура, коэффициент отражения инфракрасного излучения и др.). В системах 3D-сканирования наряду со стационарными устройствами также применяется аэросъемка [3, 4], что дает возможность качественно оцифровывать архитектурные сооружения, растительность, рельеф местности и другие сложные и труднодоступные компоненты окружающей среды [5].

Одним из актуальных направлений является интеграция таких оцифрованных объектов в системы виртуального окружения [6–8]. Это позволяет повысить реалистичность и разнообразие моделируемой обстановки (например, растительности), что особенно важно для видеотренажерных комплексов, а также снизить трудоемкость создания виртуальной сцены. При этом для получения качественного представления объекта его оцифровка должна выполняться по большому числу точек (миллионам и выше), в связи с чем возникает задача визуализации таких объемов данных в реальном времени (со скоростью не менее 25 кадров в секунду).

Эффективным путем является решение описанной задачи с помощью современных многоядерных графических процессоров (GPU), поддерживающих аппаратно-ускоренную трассировку лучей [9]. В данной работе предлагается модифицированный метод разбиения исходного неупорядоченного облака точек на группы точек, обработка которых эффективно распараллеливается

на ядрах трассировки лучей. Предлагаемый метод существенно снижает время препроцессинга облака точек и накладные расходы памяти по сравнению с альтернативными решениями.

2. ПРЕДЫДУЩИЕ ИССЛЕДОВАНИЯ

Развитие методов визуализации облаков точек, как и многих других задач компьютерной графики, исторически шло по двум направлениям: растеризация и трассировка лучей. Подробный обзор работ по этим направлениям можно найти в недавнем исследовании [10]. Анализ этих работ показывает тесную связь прогресса в том или ином направлении с эволюцией графических вычислительных средств.

Первые GPU имели небольшое число ядер, которые различались по специализации (вершинные и пиксельные конвейеры) и предназначались для отображения полигональных моделей. Исследования этого периода посвящены поиску представлений облаков точек, обеспечивающих визуальную непрерывность поверхности, сопоставимую по качеству с полигональными моделями: круглые и эллиптические диски (сплэттинг) [11–13], неявные и полиномиальные поверхности [14–16]. Построение таких моделей вначале выполнялось на стороне центрального процессора (CPU), однако с ростом числа ядер и возможностей по их программированию оно стало постепенно переноситься с CPU на GPU [17, 18]. Для ускорения графических расчетов также применялись иерархические структуры данных (вложенные октодеревья [19], *k-d*-дерево [20]), построение которых выполнялось на CPU и занимало немало времени (минуты, часы).

Следующим этапом эволюции стал переход GPU к унифицированной шейдерной архитектуре, сопровождаемый появлением сотен (а затем и тысяч) программируемых вычислительных ядер общего назначения (CUDA-ядер). Это открыло путь к переносу на GPU не только всех графических построений (OpenCL [21], вычислительные шейдеры [22]), но и ускоряющих иерархических структур данных [23, 24]. В этот период активно исследуются техники управления уровнем детализации (LOD) облаков точек (адаптивный сплэттинг [25], непрерывный LOD [26]), а также влияние порядка точек в облаке на скорость визуализации [22]. Наиболее продвинутые решения представляют собой системы распараллеленных на GPU LOD-техник и иерархических структур данных, которые позволяют визуализировать в реальном времени миллионы точек [27, 28]. Общей чертой таких решений является интенсивный расход вычислительного ресурса CUDA-ядер, что ограничивает возможности их совместного использования с другими методами визуализации в системах виртуального окружения.

Появление в 2018 г. аппаратного ускорения трассировки лучей в GPU NVidia ознаменовало наступление новой эры в компьютерной графике реального времени [29]. В архитектуру GPU были введены *RT-ядра* – аппаратные ядра нового типа, предназначенные только для просчета трассировки лучей, что позволило значительно повысить ее скорость, а также разгрузить универсальные CUDA-ядра, необходимые для широкого круга методов визуализации. Для реализации распараллеливания вычислений на RT-ядрах в новую архитектуру также был добавлен специальный конвейер трассировки лучей (*RT-конвейер*), включающий в себя закрытый аппаратный блок и программируемые (шейдерные) стадии, доступные разработчикам. Наличие таких аппаратно-программных возможностей дало мощный импульс к ревизии существующих методов визуализации и разработке новых решений [30, 31].

Чтобы использовать аппаратное ускорение трассировки лучей, вся геометрия виртуальной сцены должна быть объединена в *дерево ограничивающих объемов* (Bounding Volume Hierarchy, BVH) [32]. Построение BVH-дерева выполняется автоматически драйвером видеокарты на основе данных о треугольных/процедурных примитивах сцены, которые указывает разработчик. В случае процедурных примитивов (вычисляемых в процессе визуализации) разработчик должен явно задать для них листья BVH-дерева – ограничивающие параллелепипеды (Axis-Aligned Bounding Boxes, AABBs). Построенное таким образом BVH-дерево привязывается к RT-конвейеру и в дальнейшем используется его аппаратным блоком для быстрого поиска пересечений лучей с примитивами сцены.

Для того чтобы связка “BVH-дерево–RT-конвейер” эффективно функционировала, необходимо перед построением BVH-дерева правильно сгруппировать примитивы сцены: их AABBs должны как можно меньше пересекаться между собой и содержать пустот [33]. Применительно к облакам точек это вызывает затруднение, так как их точки в общем случае расположены в произвольном порядке. Простым решением является создание отдельного AABB для каждой точки (ее процедурной 3D-модели, например сферы) [34], однако это переводит основную вычислительную нагрузку (поиск пересечений луча с AABBs) на аппаратный блок RT-конвейера, что

делает его “бутылочным горлышком” при увеличении размера облака точек. Кроме того, хранение AABB для каждой точки приводит к ощутимым накладным расходам видеопамяти, что также ограничивает их число, которое можно обрабатывать на видеокарте.

В целях обхода описанных ограничений в нашем предыдущем исследовании [9] мы предложили объединять точки облака в группы, соответствующие листьям разреженного воксельного октодерева (далее *октантные группы точек*, ОГТ), что дало заметный прирост скорости визуализации на RT-конвейере и позволило сократить накладные расходы видеопамяти. В настоящей работе описывается модификация данного подхода, направленная на сокращение времени препроцессинга облака точек, а именно формирования ОГТ. В разделе 3 предлагаются метод и алгоритм построения ОГТ с помощью чередующихся массивов индексов точек, которые превосходят по скорости и затратам памяти альтернативные решения, основанные на связных списках. В разделе 4 приводятся результаты апробации созданного решения на облаках точек, полученных с помощью 3D-сканирования реальных объектов.

3. МЕТОД ОКТАНТНЫХ ГРУПП ТОЧЕК

В предлагаемом методе поиск непересекающихся групп близлежащих точек осуществляется на основе обхода октодерева. Мы строим вокруг исходного облака точек ограничивающий объем, разбиваем его на 8 октантов и проверяем, сколько точек попадает в каждый октант. Если число точек в октанте превышает заданное пороговое значение $K_{\max}$, то октант снова разбивается на 8 октантов, и процедура повторяется до тех пор, пока не будут найдены все ОГТ, удовлетворяющие условию $K_{\max}$. Несмотря на кажущуюся простоту идеи, ее реализация включает в себя ряд “подводных камней”.

Во-первых, в качестве исходного ограничивающего объема мы используем не параллелепипед, а куб размера $d_{\max}$:

$$d_{\max} = \max(\max(x_{\max} - x_{\min}, y_{\max} - y_{\min}), z_{\max} - z_{\min}) + 10\varepsilon, \quad (1)$$

где $x_{\min}$, $y_{\min}$, $z_{\min}$ и $x_{\max}$, $y_{\max}$, $z_{\max}$ – наименьшие и наибольшие из координат точек облака, а ε – машинная погрешность представления вещественных чисел (в данной работе $\varepsilon = 10^{-6}$). Разбиение куба на октанты (также кубы) в результате дает *кучные* группы точек, т. е. сосредоточенные в большом количестве на небольшом пространстве. Наши эксперименты показали, что если вместо куба использовать параллелепипед, то при его разбиении будут получаться вытянутые группы точек, AABBs которых содержат больше пустот и, согласно [33], менее эффективно обрабатываются на RT-конвейере.

Во-вторых, в процессе разбиения ограничивающего куба на октанты может наступить случай, при котором размер $d_{\max}$ куба достигнет погрешности ε, и станет невозможно отделить точки друг от друга (случай “*слипшихся*” точек). В этом случае мы создаем для каждой такой точки свою отдельную ОГТ и прекращаем дальнейшее разбиение октанта. Согласно нашим исследованиям вероятность возникновения “слипшихся” точек наиболее высока при $K_{\max} = 1$ и стремительно убывает с ростом $K_{\max}$. Например, в облаке точек “Beautiful Autumn” (табл. 1) число “слипшихся” точек составляет 0.44% (от общего числа точек) при $K_{\max} = 1$, 0.003% при $K_{\max} = 2$ и 0% при $K_{\max} = 3$.

В-третьих, как показывает практика, выбор значения $K_{\max}$ зависит не только от аппаратных характеристик видеокарты, но и от сложности виртуальной сцены (размера облака точек, числа объектов в сцене и др.), ввиду чего необходимы эффективные структуры данных и алгоритмы, обеспечивающие формирование ОГТ на этапе загрузки системы визуализации. Рассмотрим это более подробно.

Введем обозначения следующих основных структур данных, используемых в данной работе:

- $\boldsymbol{S_P}$ – структура атрибутов точки облака, хранящая позицию (x, y, z) и цвет (r, g, b) точки;
- $\boldsymbol{M_{IN}}$ и $\boldsymbol{M_{OUT}}$ – входной неупорядоченный и выходной упорядоченный массивы точек облака (структур S_P), длиной n каждый;
- $\boldsymbol{S_G}$ – структура-дескриптор ОГТ вида $\{i_{FIRST}, N_{POINTS}\}$, где i_{FIRST} – индекс первой точки ОГТ из массива M_{OUT}, а N_{POINTS} – число точек в ОГТ;
- $\boldsymbol{G_{OUT}}$ – выходной массив дескрипторов октантных групп точек (структур S_G) длиной n (в худшем случае число ОГТ будет совпадать с числом точек в облаке).

Рассмотрим задачу извлечения ОГТ из массива M_{IN}, т. е. получения выходных массивов M_{OUT} и G_{OUT}. Для этого мы будем рекурсивно разбивать ограничивающий куб облака точек на 8 октантов и формировать выборки элементов из массива M_{IN} для каждого k-го номера октанта согласно выражению

$$k = 4\left(\lfloor z/d_{oct} \rfloor \% 2\right) + 2\left(\lfloor y/d_{oct} \rfloor \% 2\right) + \left(\lfloor x/d_{oct} \rfloor \% 2\right), \quad (2)$$

где x, y, z – координаты позиции точки облака (исходно нормализованы и сдвинуты в неотрицательную область); $\lfloor\ \rfloor$ – целая снизу часть числа;

% – остаток от деления; d_{oct} – размер куба-октанта, а $k \in [0, 7]$.

Для хранения и обработки таких выборок введем следующие вспомогательные структуры данных:

• $\boldsymbol{I_0}$ и $\boldsymbol{I_1}$ – пара массивов индексов точек облака из входного массива M_{IN}, длиной n каждый;

• $\boldsymbol{K}$ – байтовый массив номеров октантов точек облака из входного массива M_{IN}, длиной n.

Каждая выборка представляет собой последовательность отобранных индексов точек облака из массива M_{IN}, записанную в массиве I_0 или I_1 (зависит от четности уровня рекурсии), и задается тройкой параметров (b, u, q), где b – флаг четности уровня рекурсии (0 – четный, 1 – нечетный), u – смещение первого элемента выборки в I_b-м массиве, а q – число элементов в выборке.

Вначале мы создаем *стартовую выборку* вида $\{0, 1, ..., n-1\}$, которая совпадает с порядком элементов в исходном массиве M_{IN}. На 0-уровне рекурсии мы получаем до 8-ми выборок (по числу октантов) из стартовой выборки, на 1-м уровне – до 8-ми выборок из каждой выборки 0-го уровня и т. д., пока длина выборки не перестанет превышать $K_{\max}$ или не наступит случай “слипшихся” точек (см. выше). Отметим, что сумма длин выборок (i+1)-го уровня, получаемых из выборки i-го уровня, равна длине этой выборки, что позволяет перезаписывать выборки (i+1)-го уровня на место выборки i-го уровня. Это свойство позволяет хранить выборки в массивах I_0 и I_1 фиксированного размера n и, чередуя эти массивы в качестве источника и приемника данных, продвигаться в глубь рекурсии. На основе описанной идеи был разработан следующий алгоритм.

Алгоритм извлечения октантных групп точек с помощью чередующихся массивов

1. Создадим массивы M_{OUT}, G_{OUT}, I_0, I_1 и K, а также счетчики m и g элементов в M_{OUT} и G_{OUT}.
2. Обнулим счетчики m и g, а в массив I_0 запишем стартовую выборку индексов (см. выше).
3. Вычислим размер $d_{\max}$ ограничивающего куба облака точек согласно выражению (1).
4. Выполним рекурсивную обработку стартовой выборки ($b = 0$, $u = 0$, $q = n$).

Если $d_{\max} \le \varepsilon$, то создадим ОГТ для каждой из “слипшихся” точек.

Цикл по индексу i в массиве I_b от u до $u + q - 1$

$M_{OUT}[m] = M_{IN}[I_b[i]]$; $G_{OUT}[g] = \{m, 1\}$; $m = m + 1$; $g = g + 1$;

Конец цикла.

В противном случае, если $q \le K_{\max}$, то входная выборка (b, u, q) становится ОГТ:

$G_{OUT}[g] = \{m, q\}$; $g = g + 1$;

Цикл по индексу i массива I_b от u до $u + q - 1$

$M_{OUT}[m] = M_{IN}[I_b[i]]$; $m = m + 1$;

Конец цикла.

В противном случае перейдем к обработке 8-ми выходных выборок.

Запомним текущее значение $d_{\max}$ и разделим его пополам: $d_{save} = d_{\max}$; $d_{\max} = 0.5 d_{\max}$.

Создадим массив Q длин 8-ми выходных выборок и вычислим эти длины.

Обнулим все элементы массива Q.

Цикл по индексу i в массиве I_b от u до $u + q - 1$.

Вычислим номер k октанта согласно (2) для точки $M_{IN}[I_b[i]]$ и $d_{oct} = d_{\max}$.

Запомним номер k октанта: $K[I_b[i]] = k$.

$Q[k] = Q[k] + 1$;

Конец цикла.

Создадим массив U смещений 8-ми выходных выборок и вычислим эти смещения.

Запишем $U[0] = u$, а остальные элементы массива U обнулим.

Цикл по номеру k октанта от 1 до 7

$U[k] = U[k-1] + Q[k-1]$;

Конец цикла.

Распределим точки из входной выборки (b, u, q) по 8-ми выходным выборкам.

Вычислим флаг b_{NEXT} четности следующего уровня рекурсии:

$b_{NEXT} = (b + 1)\% 2$;

Создадим массив C смещений в 8-ми выходных выборках.

Инициализируем массив C: $C[0] = U[0]$, $C[1] = U[1]$, ..., $C[7] = U[7]$.

Цикл по индексу i в массиве I_b от u до $u + q - 1$.

Получим номер k октанта: $k = K[I_b[i]]$.
$I_{b,\,NEXT}[C[k]] = I_b[i]$; $C[k] = C[k] + 1$;
Конец цикла.
Выполним обработку 8-ми выходных выборок.
Цикл по номеру k октанта от 0 до 7.
Если $Q[k] > 0$, то обработаем выборку (b_{NEXT}, $U[k]$, $Q[k]$) согласно п. 4.
Конец цикла.
Восстановим текущее значение $d_{\max}$: $d_{\max} = d_{save}$.
Конец алгоритма.

В результате выполнения приведенного алгоритма в массиве M_{OUT} формируется g октантных групп точек ($g \le n$) размера не более $K_{\max}$ каждая, а в массиве $G_{OUT} - g$ дескрипторов, позволяющих считывать эти группы из массива M_{OUT}.

Как можно заметить, при реализации предложенного алгоритма вместо массивов I_0, I_1 и K можно также использовать структуру данных типа "связный список", например, двусвязный список *std:: list* или односвязный список *std:: forward_list* из стандартной библиотеки шаблонов C++ [35]. Преимуществом списков является константная сложность операций вставки и удаления элемента, а также возможность переставить элемент внутри списка без его пересоздания (с помощью функций *splice* и *splice_after*, см. [35]). Это позволяет сократить число шагов при обработке последнего случая в п. 4 алгоритма и, в частности, избежать отдельного шага вычисления длин выборок. Однако, элементы списка, в отличие от массива, располагаются в памяти не последовательно друг за другом, а произвольным образом, вследствие чего время перехода от элемента к элементу списка в циклах будет выше. Чтобы оценить эффект от этих факторов, в данной работе были реализованы два дополнительных варианта предложенного алгоритма, основанные на *std:: list* и *std:: forward_list*, и произведено сравнение времен извлечения ОГТ для всех трех реализаций (см. раздел "Результаты").

4. РЕЗУЛЬТАТЫ

Предложенные метод и алгоритм были реализованы в прототипе визуализатора облаков точек, созданном в рамках нашего предыдущего исследования [9]. Данный программный комплекс разработан на языке C++ с применением языка GLSL программирования шейдеров и API Vulkan [36] и выполняет визуализацию облаков точек в реальном времени с помощью аппаратно-ускоренной трассировки лучей.

Апробация модифицированного визуализатора проводилась при разрешении HD (720p) области вывода на персональном компьютере (Intel Core i7-6800K 6x3.40 ГГц, 16 Гб RAM), оборудованном графической картой NVidia GeForce RTX 2080 (8 Гб VRAM, 46 RT-ядер, 2944 CUDA-ядер, драйвер NVidia DCH 536.40). Для апробации использовались облака точек реальных объектов окружающей среды из банка данных Sketchfab [5] (номера 1–7 в табл. 1)[1] и проекта CAVES NASA [37] (номера 8, 9).

Апробация выполнялась в два этапа. На *первом этапе* проводилось сравнение временных затрат на формирование октантных групп точек с помощью предложенного решения и альтернативных вариантов реализации (см. конец раздела 3). Замеры времени для всех трех вариантов реализации выполнялись на одном и том же оборудовании при значении $K_{\max} = 8$. Результаты замеров приведены в табл. 1.

Из табл. 1 видно, что предложенный в данной работе вариант (*a*) реализации алгоритма извлечения ОГТ работает быстрее альтернативных реализаций (*b*) и (*c*) в среднем в 1.5 и 1.9 раза. Кроме того, ввиду отсутствия необходимости хранения указателей на соседние элементы (один или два), чередующиеся массивы (*a*) требуют меньше памяти (минимум на 30%), чем связные списки (*b*) и (*c*).

На *втором этапе* апробации исследовались зависимости скоростей визуализации облаков точек из табл. 1 и накладных расходов видеопамяти (числа AABBs октантных групп точек) от значения $K_{\max}$. Для этого была проведена серия экспериментов, в которой значение $K_{\max}$ увели-

[1] (1) "Beautiful Autumn", автор Epic_Tree_Store, https://skfb.ly/VrYw; (2) "Lanyon Quoit – point cloud", автор Penwith Landscape Partnership, https://skfb.ly/6SOGS; (3) "tree "maybe last" – point cloud", автор Jer Bot, https://skfb.ly/6BZUP; (4) "Scotland: Dundee, The McManus (5M point cloud)", автор Daniel Muirhead, https://skfb.ly/6SVFM; (5) "Scan Unscannable: Grass Cloud 1", автор Epic_Tree_Store, https://skfb.ly/TAF9; (6) "Scan Unscannable: Bush Cloud", автор Epic_Tree_Store, https://skfb.ly/THSr; (7) "Point Cloud – Plaza Mayor de Almagro, Spain", автор Global Digital Heritage, https://skfb.ly/6RZrD. Данные модели распространяются по лицензии Creative Commons Attribution (http://creativecommons.org/licenses/by/4.0/).

Таблица 1. Сравнение производительности вариантов реализации алгоритма извлечения ОГТ: (*a*) чередующиеся массивы; (*b*) двусвязный список *std:: list*; (*c*) односвязный список *std:: forward_list*

№	Облако точек	Число точек	Число ОГТ	Время извлечения октантных групп точек, с		
				(*a*)	(*b*)	(*c*)
1	Beautiful Autumn	1 580 297	530 841	0.35	0.52	0.72
2	Lanyon Quoit	1 800 000	531 671	0.79	1.27	1.58
3	Stump	1 839 043	575 533	0.48	0.75	1.01
4	The McManus	5 000 000	1 610 157	2.68	4.31	4.81
5	Grass	5 133 067	1 595 571	1.13	1.63	2.17
6	Bush	5 333 200	1 891 213	1.37	1.72	2.25
7	Plaza Almagro	12 736 674	3 411 935	5.19	8.19	9.33
8	Sheepridge Site	23 882 848	7 407 168	6.28	9.72	11.46
9	King's Bowl	37 508 760	12 042 023	10.49	16.03	19.59

чивалось от 1 до 1024 (по степеням двойки) с сохранением параметров виртуальной камеры для каждого облака точек. Результаты экспериментов показаны на графиках на рис. 1. Рассмотрим их более подробно.

Из левого графика $f_{avg}(K_{\max})$ видно, что предложенный метод упорядочивания облаков точек позволяет получить прирост скорости визуализации до 47% (чем больше размер облака точек, тем больше прирост) по сравнению с референсными значениями при $K_{\max} = 1$ (AABB на точку, как в [34]). Правый график $N_{\text{AABBs}}(K_{\max})$ показывает монотонное убывание числа AABBs (и связанных с ними накладных расходов видеопамяти) с ростом $K_{\max}$. При значениях $K_{\max}$, соответствующих пикам производительности на графике $f_{avg}(K_{\max})$, число AABBs сокращается в 1.9–5.4 раз по сравнению с референсными значениями при $K_{\max} = 1$. Отметим, что даже в случае небольшого прироста производительности (см. кривую *a* на графике $f_{avg}(K_{\max})$) предложенный метод обеспечивает существенное сокращение числа AABBs.

Еще одним интересным наблюдением является то, что с ростом размеров облаков точек пики производительности на графике $f_{avg}(K_{\max})$ становятся менее ярко выраженными и смещаются в область более высоких значений $K_{\max}$. Данный факт может указывать на то, что с точки зрения

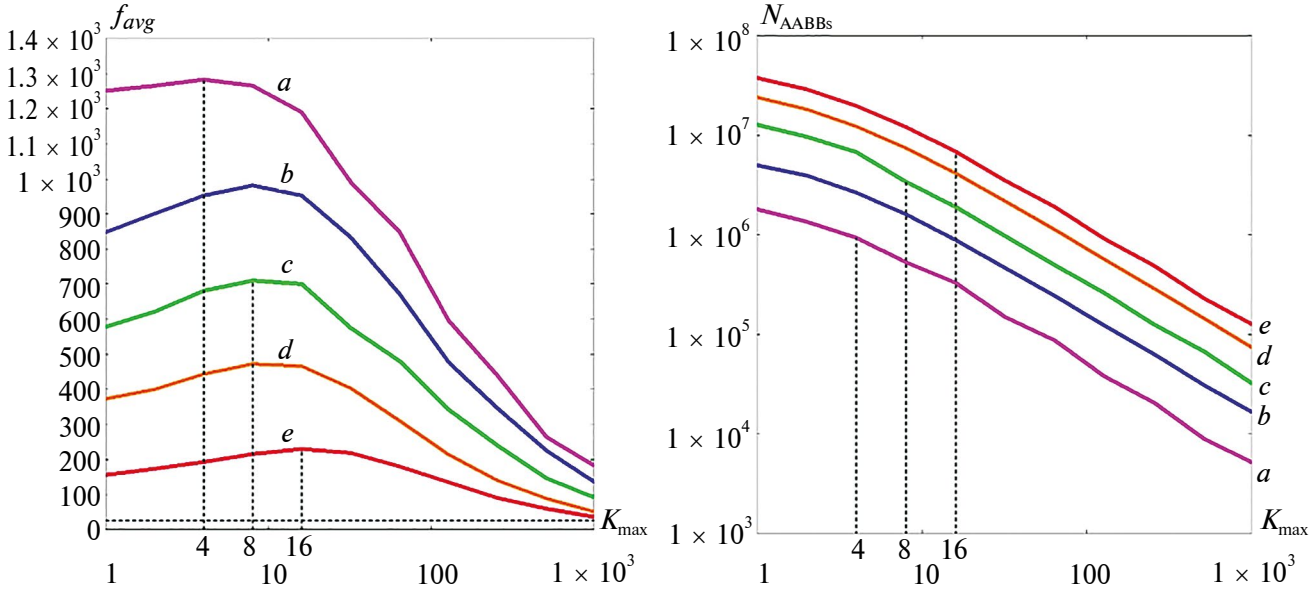
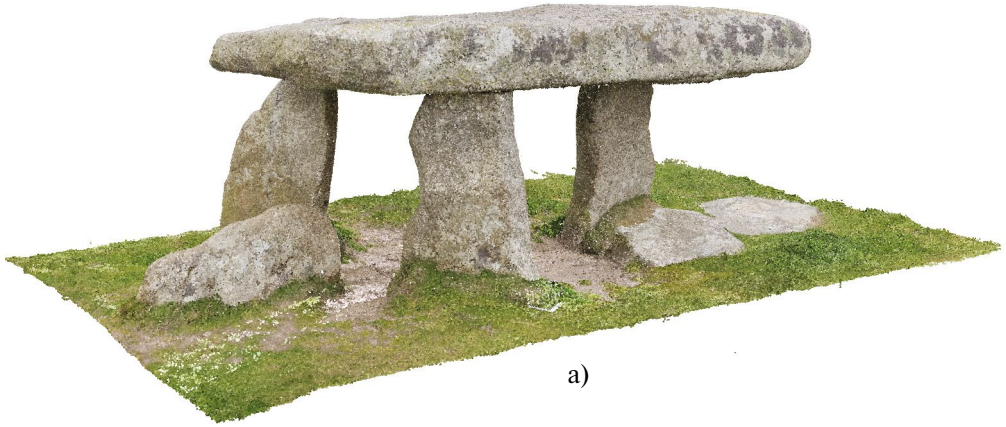
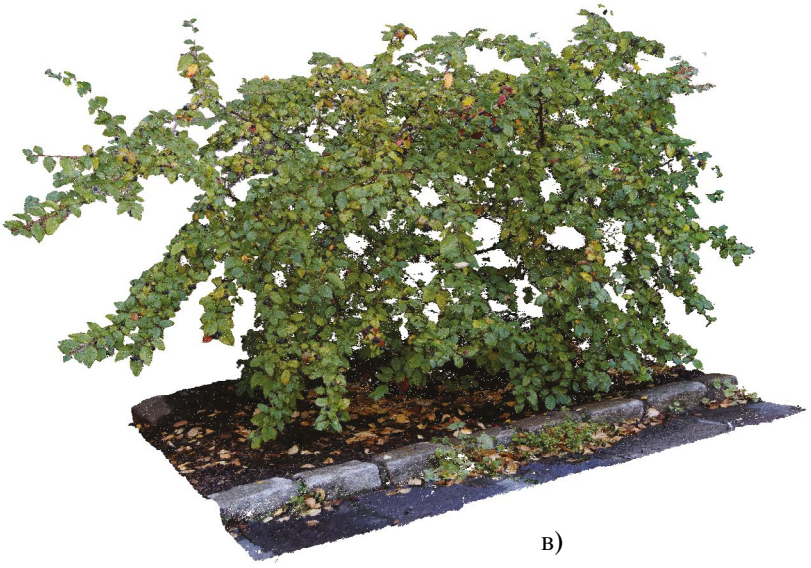


Рис. 1. Влияние параметра $K_{\max}$ на среднюю частоту f_{avg} синтеза изображений, в кадрах в секунду (слева), и на число N_{AABBs} ограничивающих параллелепипедов (справа). На приведенных графиках для осей $K_{\max}$ и N_{AABBs} используется логарифмическая шкала, а кривые *a*, *b*, *c*, *d*, *e* соответствуют облакам точек № 2, 4, 7, 8, 9 из табл. 1. На левом графике пунктирная горизонтальная линия обозначает нижний порог частоты 25 кадров в секунду, соответствующей визуализации в реальном времени.

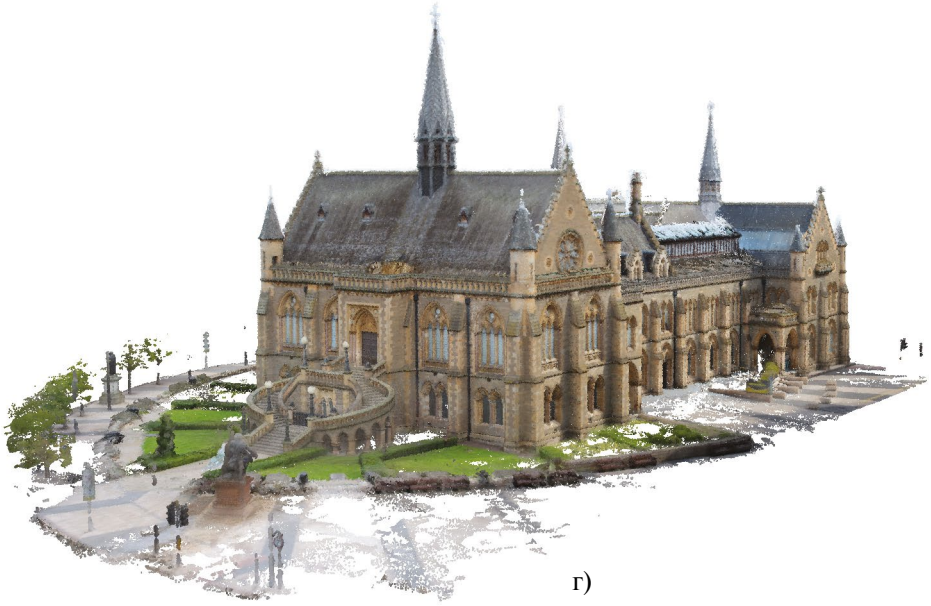


а)

б)



в)



г)

Рис. 2. Примеры кадров визуализации, полученных с помощью нашего модифицированного визуализатора облаков точек. Изображения а)–г) соответствуют облакам точек № 2, 1, 6, 4 из табл. 1.

оптимизации накладных расходов видеопамяти выгоднее объединять несколько небольших облаков точек в одно крупное, чем визуализировать каждое облако точек по отдельности. В этой связи разработка эффективной стратегии визуализации мультиобъектных сцен из облаков точек с помощью аппаратно-ускоренной трассировки лучей представляется нам интересным направлением для будущих исследований.

5. ЗАКЛЮЧЕНИЕ

Современные технологии позволяют создавать подробные цифровые 3D-модели реальных объектов окружающей среды в виде облаков точек, состоящих из миллионов элементов. Интеграция таких объектов в системы виртуального окружения позволяет существенно повысить реалистичность и разнообразие моделируемой обстановки, однако требует эффективных решений для обработки больших объемов точек в реальном времени. В данной работе предложен метод упорядочивания облаков точек, обеспечивающий прирост скорости визуализации до 47% на видеокартах с аппаратным ускорением трассировки лучей, а также эффективное снижение расходов видеопамяти на построение ускоряющих структур данных. Предложенный метод основан на объединении точек облака в октантные группы точек, которые хорошо вписываются в BVH-дерево – ускоряющую иерархическую структуру данных, на основе которой работает конвейер трассировки лучей. В работе предложен алгоритм извлечения октантных групп точек из исходного неупорядоченного облака точек с помощью чередующихся массивов индексов точек, который работает до 1.9 раз быстрее альтернативных решений, основанных на связных списках, и требует минимум на 30% меньше памяти. Разработанный алгоритм может выполняться на этапе загрузки системы визуализации, что позволяет оперативно настраивать пороговый размер $K_{\max}$ октантных групп точек для достижения пиковой производительности на имеющихся исходных данных и оборудовании. Предложенное решение было реализовано в разработанном ранее прототипе визуализатора облака точек и апробировано на ряде облаков точек реальных объектов окружающей среды (на рис. 2 показаны примеры полученных кадров визуализации таких объектов). Полученные результаты подтвердили эффективность разработанного решения и возможность его применения в системах виртуального окружения, видеотренажерных комплексах, научной визуализации, геоинформационных системах и др. В качестве будущей работы планируются исследование и разработка эффективной стратегии визуализации мультиобъектных сцен из облаков точек с помощью аппаратно-ускоренной трассировки лучей.

6. БЛАГОДАРНОСТИ

Публикация выполнена в рамках государственного задания ФГУ ФНЦ НИИСИ РАН по теме № FNEF-2024-0002 “Математическое моделирование многомасштабных динамических процессов и системы виртуального окружения”.

СПИСОК ЛИТЕРАТУРЫ

1. *Guo M., Sun M., Pan D., Wang G., Zhou Y., Yan B., Fu Z.* High-precision deformation analysis of yingxian wooden pagoda based on UAV image and terrestrial LiDAR point cloud // Heritage Science. 2023. V. 11. P. 1–18. https://doi.org/10.1186/s40494-022-00833-z
2. *Adamopoulos E., Papadopoulou E.-E., Mpia M., Deligianni E.-O., Papadopoulou G., Athanasoulis D., Konioti M., Koutsoumpou M., Anagnostopoulos C.N.* 3D Survey and Monitoring of Ongoing Archaeological Excavations via Terrestrial and Drone LIDAR // ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences. 2023. V. X-M-1-2023. P. 3–10. https://doi.org/10.5194/isprs-annals-X-M-1-2023-3-2023
3. *Weiser H., Schäfer J., Winiwarter L., Krašovec N., Fassnacht F.E., Höfle B.* Individual tree point clouds and tree measurements from multi-platform laser scanning in German forests // Earth System Science Data. 2022. V. 14. № 7. P. 2989–3012. https://doi.org/10.5194/essd-14-2989-2022
4. *Risbøl O., Gustavsen L.* LiDAR from drones employed for mapping archaeology – Potential, benefits and challenges // Archaeological Prospection. 2018. V. 25. P. 329–338. https://doi.org/10.1002/arp.1712
5. Sketchfab – The leading platform for 3D & AR on the web. 2023. https://sketchfab.com/
6. *Casado-Coscolla A., Sanchez-Belenguer C., Wolfart E., Sequeira V.* Rendering massive indoor point clouds in virtual reality // Virtual Reality. 2023. V. 27. P. 1859–1874. https://doi.org/10.1007/s10055-023-00766-3
7. *Kharroubi A., Hajji R., Billen R., Poux F.* Classification and Integration of Massive 3D Points Clouds in a Virtual Reality (VR) Environment // The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. 2019. V. XLII-2/W17. P. 165–171. https://doi.org/10.5194/isprs-archives-XLII-2-W17-165-2019
8. *Discher S., Masopust L., Schulz S., Richter R., Döllner J.* A Point-Based and Image-Based Multi-Pass Render-

ing Technique for Visualizing Massive 3D Point Clouds in VR Environments // Journal of WSCG. 2018. V. 26. № 2. P. 76–84.
https://doi.org/10.24132/JWSCG.2018.26.2.2

9. *Тимохин П.Ю., Михайлюк М.В.* Моделирование особенностей ландшафта с помощью облаков точек в системах виртуального окружения // Труды 33-й Международной конференции по компьютерной графике и машинному зрению (GraphiCon 2023). 2023. С. 157–168.
https://doi.org/10.20948/graphicon-2023-157-168
10. *Kivi P.E.J., Mäkitalo M.J., Žádník J., Ikkala J., Vadakital V.K.M., Jääskeläinen P.O.* Real-Time Rendering of Point Clouds With Photorealistic Effects: A Survey // IEEE Access. 2022. V. 10. P. 13151–13173.
https://doi.org/10.1109/ACCESS.2022.3146768
11. *Kobbelt L., Botsch M.* A survey of point-based techniques in computer graphics // Computers & Graphics. 2004. V. 28. № 6. P. 801–814.
https://doi.org/10.1016/j.cag.2004.08.009
12. *Botsch M., Hornung A., Zwicker M., Kobbelt L.* High-Quality Surface Splatting on Today's GPUs // Proc. Eurographics / IEEE VGTC Symposium Point-Based Graphics. 2005. P. 17–24.
https://doi.org/10.2312/SPBG/SPBG05/017-024
13. *Linsen L., Müller K., Rosenthal P.* Splat-based Ray Tracing of Point Clouds // Journal of WSCG. 2007. V. 15. P. 51–58.
https://dspace5.zcu.cz/bitstream/11025/1426/1/Linsen.pdf
14. *Wald I., Seidel H.-P.* Interactive ray tracing of point-based models // Proceedings Eurographics / IEEE VGTC Symposium Point-Based Graphics (Jun. 2005). 2005. P. 9–16.
https://doi.org/10.1145/1187112.1187176
15. *Adamson A., Alexa M.* Ray tracing point set surfaces // Proceedings of the Shape Modeling International (SMI '03). 2003. P. 272–279.
https://doi.org/10.1109/SMI.2003.1199627
16. *Hubo E., Mertens T., Haber T., Bekaert P.* Self-similarity based compression of point set surfaces with application to ray tracing // Computers & Graphics. 2008. V. 32. № 2. P. 221–234.
https://doi.org/10.1016/j.cag.2008.01.012
17. *Tejada E., Gois J.P., Nonato L.G., Castelo A., Ertl T.* Hardware-accelerated Extraction and Rendering of Point Set Surfaces // Proceedings of the 8th Joint Eurographics – IEEE VGTC Symposium on Visualization (EuroVis '06). 2006. P. 21–28.
https://doi.org/10.2312/VisSym/EuroVis06/021-028
18. *Zhang Y., Pajarola R.* Deferred blending: Image composition for single-pass point rendering // Computers & Graphics. 2007. V. 31. № 2. P. 175–189.
https://doi.org/10.1016/j.cag.2006.11.012
19. *Wimmer M., Scheiblauer C.* Instant points: Fast rendering of unprocessed point clouds // Eurographics Symposium on Point-Based Graphics (eds. Botsch M., Chen B., Pauly M., Zwicker M.). Geneva, Switzerland: The Eurographics Association. 2006. P. 129–136.
https://doi.org/10.2312/SPBG/SPBG06/129-136
20. *Hubo E., Mertens T., Haber T., Bekaert P.* The Quantized kd-Tree: Efficient Ray Tracing of Compressed Point Clouds // 2006 IEEE Symposium on Interactive Ray Tracing, Salt Lake City, UT, USA. 2006. P. 105–113.
https://doi.org/10.1109/RT.2006.280221
21. *Günther C., Kanzok T., Linsen L., Rosenthal P.* A GPGPU-based Pipeline for Accelerated Rendering of Point Clouds // Journal of WSCG. 2013. V. 21. № 2. P. 153–162.
https://dspace5.zcu.cz/bitstream/11025/6978/1/Gunther.pdf
22. *Schütz M., Kerbl B., Wimmer M.* Rendering Point Clouds with Compute Shaders and Vertex Order Optimization // Computer Graphics Forum. 2021. V. 40. № 4. P. 115–126.
https://doi.org/10.1111/cgf.14345
23. *Kashyap S., Goradia R., Chaudhuri P., Chandran S.* Implicit Surface Octrees For Ray Tracing Point Models // ICVGIP'10: proceedings of the 7th Indian Conference on Computer Vision, Graphics and Image Processing (December 2010). 2010. P. 227–234.
https://doi.org/10.1145/1924559.1924590
24. *Karras T.* Maximizing Parallelism in the Construction of BVHs, Octrees, and k-d Trees // Eurographics/ ACM SIGGRAPH Symposium on High Performance Graphics. Eurographics Association. 2012. P. 33–37.
https://doi.org/10.2312/EGGH/HPG12/033-037
25. *Kim H.-J., Cengiz Öztireli A., Gross M., Choi S.-M.* Adaptive surface splatting for facial rendering // Computer Animation Virtual Worlds. 2012. V. 23. № 3–4. P. 363–373.
https://doi.org/10.1002/cav.1463
26. *Schütz M., Krösl K., Wimmer M.* Real-Time Continuous Level of Detail Rendering of Point Clouds // IEEE VR2019: the 26th IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Osaka, Japan. IEEE. 2019. P. 103–110.
https://doi.org/10.1109/VR.2019.8798284
27. *Kulik A., Kunert A., Beck S., Matthes C.-F., Schollmeyer A., Kreskowski A., Fröhlich B., Cobb S., D'Cruz M.* Virtual Valcamonica: Collaborative Exploration of Prehistoric Petroglyphs and Their Surrounding Environment in Multi-User Virtual Reality // Presence: Teleoperators and Virtual Environments. 2017. V. 26. № 3. P. 297–321.
https://doi.org/10.1162/pres_a_00297
28. *Schütz M., Herzberger L., Wimmer M.* SimLOD: Simultaneous LOD Generation and Rendering // ArXiv, abs/2310.03567. 2023. P. 1–12.
https://doi.org/10.48550/arXiv.2310.03567
29. NVIDIA Turing GPU Architecture // NVIDIA Corporation. 2018.
https://images.nvidia.com/aem-dam/Solutions/design-visualization/technologies/turing-architecture/NVIDIA-Turing-Architecture-Whitepaper.pdf

30. *Sanzharov V.V., Frolov V.A., Galaktionov V.A.* Survey of Nvidia RTX Technology // Programming and Computer Software. 2020. V. 46. № 4. P. 297–304. https://doi.org/10.1134/S0361768820030068
31. *Timokhin P.Y., Mikhaylyuk M.V.* An Efficient Technology of Real-time Modeling of Height Field Surface on the Ray Tracing Pipeline // Programming and Computer Software. 2023. V. 49. № 3. P. 178–186. https://doi.org/10.1134/S0361768823030064
32. *Rusch M., Bickford N., Subtil N.* Introduction to Vulkan Ray Tracing // Ray Tracing Gems II. NVIDIA. 2021. P. 213–255. https://doi.org/10.1007/978-1-4842-7185-8_16
33. *Sjoholm J.* Best Practices for Using NVIDIA RTX Ray Tracing (Updated) // NVIDIA Technical Blog. Jul 25, 2022. https://developer.nvidia.com/blog/best-practices-for-using-nvidia-rtx-ray-tracing-updated/
34. *Lefrançois M.-K.* Intersection Shader // NVIDIA Vulkan Ray Tracing Tutorials. 2020–2023. https://github.com/nvpro-samples/vk_raytracing_tutorial_KHR/tree/master/ray_tracing_intersection
35. C++ reference. Containers library. Sequence containers. 2023. https://en.cppreference.com/w/cpp/container
36. Vulkan 1.3.275 – A Specification (with all ratified extensions) // The Khronos Vulkan Working Group. 2024. https://registry.khronos.org/vulkan/specs/1.3-khr-extensions/pdf/vkspec.pdf
37. *Wong U., Whittaker W., Jones H., Whittaker R.* NASA Planetary Pits and Caves Analog Dataset. 2014. https://ti.arc.nasa.gov/dataset/caves/

A METHOD TO ORDER POINT CLOUDS FOR VISUALIZATION ON THE RAY TRACING PIPELINE

© 2024 P. Y. Timokhin[a], M. V. Mikhaylyuk[a]

[a] *Federal State Institution "Scientific Research Institute for System Analysis of the Russian Academy of Sciences" Nakhimovskii pr. 36/1, Moscow, 117218 Russia*

Currently, the digitization of environment objects (vegetation, terrain, architectural structures, etc.) in the form of point clouds is actively developing. The integration of such digitized objects into virtual environment systems allows the quality of the modeled environment to be improved, but requires efficient methods and algorithms for real-time visualization of large point volumes. In this paper the solution of this task on modern multicore GPUs with support of hardware-accelerated ray tracing is researched. A modified method is proposed where the original unordered point cloud is split up into point groups which visualization is effectively parallelized on ray tracing cores. The paper describes an algorithm for constructing such groups using swapping arrays of point indices, which works faster than alternative solutions based on linked lists, and also has lower memory overhead. The proposed method and algorithm were implemented in the point cloud visualization software complex and approbated on a number of digitized environment objects. The results of the approbation confirmed the efficiency of proposed solutions as well as their applicability for virtual environment systems, video simulators and geoinformation systems, virtual laboratories, etc.

Keywords: virtual environment, visualization, real time, ray tracing, point cloud

REFERENCES

1. *Guo M., Sun M., Pan D., Wang G., Zhou Y., Yan B., Fu Z.* High-precision deformation analysis of yingxian wooden pagoda based on UAV image and terrestrial LiDAR point cloud // Heritage Science. 2023. V. 11. P. 1–18. https://doi.org/10.1186/s40494-022-00833-z
2. *Adamopoulos E., Papadopoulou E.-E., Mpia M., Deligianni E.-O., Papadopoulou G., Athanasoulis D., Konioti M., Koutsoumpou M., Anagnostopoulos C.N.* 3D Survey and Monitoring of Ongoing Archaeological Excavations via Terrestrial and Drone LIDAR // ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences. 2023. V. X-M-1-2023. P. 3–10. https://doi.org/10.5194/isprs-annals-X-M-1-2023-3-2023
3. *Weiser H., Schäfer J., Winiwarter L., Krašovec N., Fassnacht F.E., Höfle B.* Individual tree point clouds and tree measurements from multi-platform laser scanning in German forests // Earth System Science Data. 2022. V. 14. № 7. P. 2989–3012. https://doi.org/10.5194/essd-14-2989-2022
4. *Risbøl O., Gustavsen L.* LiDAR from drones employed for mapping archaeology – Potential, benefits and challenges // Archaeological Prospection. 2018. V. 25. P. 329–338. https://doi.org/10.1002/arp.1712
5. Sketchfab – The leading platform for 3D & AR on the web. 2023. https://sketchfab.com/
6. *Casado-Coscolla A., Sanchez-Belenguer C., Wolfart E., Sequeira V.* Rendering massive indoor point clouds in virtual reality // Virtual Reality. 2023. V. 27. P. 1859–1874. https://doi.org/10.1007/s10055-023-00766-3
7. *Kharroubi A., Hajji R., Billen R., Poux F.* Classification and Integration of Massive 3D Points Clouds in a Virtual

Reality (VR) Environment // The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. 2019. V. XLII-2/W17. P. 165–171. https://doi.org/10.5194/isprs-archives-XLII-2-W17-165-2019

8. *Discher S., Masopust L., Schulz S., Richter R., Döllner J.* A Point-Based and Image-Based Multi-Pass Rendering Technique for Visualizing Massive 3D Point Clouds in VR Environments // Journal of WSCG. 2018. V. 26. № 2. P. 76–84. https://doi.org/10.24132/JWSCG.2018.26.2.2
9. *Timokhin P. Yu., Mikhaylyuk M.V.* Modeling of Landscape Features by Means of Point Clouds in Virtual Environment Systems // Proc. GraphiCon. Proceedings of the 33rd International Conference on Computer Graphics and Vision (GraphiCon 2023). Moscow, 2023. P. 157–168. https://doi.org/10.20948/graphicon-2023-157-168.
10. *Kivi P.E.J., Mäkitalo M.J., Žádník J., Ikkala J., Vadakital V.K.M., Jääskeläinen P.O.* Real-Time Rendering of Point Clouds With Photorealistic Effects: A Survey // IEEE Access. 2022. V. 10. P. 13151–13173. https://doi.org/10.1109/ACCESS.2022.3146768
11. *Kobbelt L., Botsch M.* A survey of point-based techniques in computer graphics // Computers & Graphics. 2004. V. 28. № 6. P. 801–814. https://doi.org/10.1016/j.cag.2004.08.009
12. *Botsch M., Hornung A., Zwicker M., Kobbelt L.* High-Quality Surface Splatting on Today's GPUs // Proc. Eurographics/IEEE VGTC Symposium Point-Based Graphics. 2005. P. 17–24. https://doi.org/10.2312/SPBG/SPBG05/017-024
13. *Linsen L., Müller K., Rosenthal P.* Splat-based Ray Tracing of Point Clouds // Journal of WSCG. 2007. V. 15. P. 51–58. https://dspace5.zcu.cz/bitstream/11025/1426/1/Linsen.pdf
14. *Wald I., Seidel H.-P.* Interactive ray tracing of point-based models // Proceedings Eurographics / IEEE VGTC Symposium Point-Based Graphics (Jun. 2005). 2005. P. 9–16. https://doi.org/10.1145/1187112.1187176
15. *Adamson A., Alexa M.* Ray tracing point set surfaces // Proceedings of the Shape Modeling International (SMI '03). 2003. P. 272–279. https://doi.org/10.1109/SMI.2003.1199627
16. *Hubo E., Mertens T., Haber T., Bekaert P.* Self-similarity based compression of point set surfaces with application to ray tracing // Computers & Graphics. 2008. V. 32. № 2. P. 221–234. https://doi.org/10.1016/j.cag.2008.01.012
17. *Tejada E., Gois J.P., Nonato L.G., Castelo A., Ertl T.* Hardware-accelerated Extraction and Rendering of Point Set Surfaces // Proceedings of the 8th Joint Eurographics – IEEE VGTC Symposium on Visualization (EuroVis '06). 2006. P. 21–28. https://doi.org/10.2312/VisSym/EuroVis06/021-028
18. *Zhang Y., Pajarola R.* Deferred blending: Image composition for single-pass point rendering // Computers & Graphics. 2007. V. 31. № 2. P. 175–189. https://doi.org/10.1016/j.cag.2006.11.012
19. *Wimmer M., Scheiblauer C.* Instant points: Fast rendering of unprocessed point clouds // Eurographics Symposium on Point-Based Graphics (eds. Botsch M., Chen B., Pauly M., Zwicker M.). Geneva, Switzerland: The Eurographics Association. 2006. P. 129–136. https://doi.org/10.2312/SPBG/SPBG06/129-136
20. *Hubo E., Mertens T., Haber T., Bekaert P.* The Quantized kd-Tree: Efficient Ray Tracing of Compressed Point Clouds // 2006 IEEE Symposium on Interactive Ray Tracing, Salt Lake City, UT, USA. 2006. P. 105–113. https://doi.org/10.1109/RT.2006.280221
21. *Günther C., Kanzok T., Linsen L., Rosenthal P.* A GPGPU-based Pipeline for Accelerated Rendering of Point Clouds // Journal of WSCG. 2013. V. 21. № 2. P. 153–162. https://dspace5.zcu.cz/bitstream/11025/6978/1/Gunther.pdf
22. *Schütz M., Kerbl B., Wimmer M.* Rendering Point Clouds with Compute Shaders and Vertex Order Optimization // Computer Graphics Forum. 2021. V. 40. № 4. P. 115–126. https://doi.org/10.1111/cgf.14345
23. *Kashyap S., Goradia R., Chaudhuri P., Chandran S.* Implicit Surface Octrees For Ray Tracing Point Models // ICVGIP'10: proceedings of the 7th Indian Conference on Computer Vision, Graphics and Image Processing (December 2010). 2010. P. 227–234. https://doi.org/10.1145/1924559.1924590
24. *Karras T.* Maximizing Parallelism in the Construction of BVHs, Octrees, and k-d Trees // Eurographics/ACM SIGGRAPH Symposium on High Performance Graphics. Eurographics Association. 2012. P. 33–37. https://doi.org/10.2312/EGGH/HPG12/033-037
25. *Kim H.-J., Cengiz Öztireli A., Gross M., Choi S.-M.* Adaptive surface splatting for facial rendering // Computer Animation Virtual Worlds. 2012. V. 23. № 3–4. P. 363–373. https://doi.org/10.1002/cav.1463
26. *Schütz M., Krösl K., Wimmer M.* Real-Time Continuous Level of Detail Rendering of Point Clouds // IEEE VR2019: the 26th IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Osaka, Japan. IEEE. 2019. P. 103–110. https://doi.org/10.1109/VR.2019.8798284
27. *Kulik A., Kunert A., Beck S., Matthes C.-F., Schollmeyer A., Kreskowski A., Fröhlich B., Cobb S., D'Cruz M.* Virtual Valcamonica: Collaborative Exploration of Prehistoric Petroglyphs and Their Surrounding Environment in Multi-User Virtual Reality // Presence: Teleoperators and Virtual Environments. 2017. V. 26. № 3. P. 297–321. https://doi.org/10.1162/pres_a_00297
28. *Schütz M., Herzberger L., Wimmer M.* SimLOD: Simultaneous LOD Generation and Rendering // ArXiv, abs/2310.03567. 2023. P. 1–12. https://doi.org/10.48550/arXiv.2310.03567

29. NVIDIA Turing GPU Architecture // NVIDIA Corporation. 2018.
https://images.nvidia.com/aem-dam/Solutions/design-visualization/technologies/turing-architecture/NVIDIA-Turing-Architecture-Whitepaper.pdf
30. *Sanzharov V.V., Frolov V.A., Galaktionov V.A.* Survey of Nvidia RTX Technology // Programming and Computer Software. 2020. V. 46. № 4. P. 297–304.
https://doi.org/10.1134/S0361768820030068
31. *Timokhin P.Y., Mikhaylyuk M.V.* An Efficient Technology of Real-time Modeling of Height Field Surface on the Ray Tracing Pipeline // Programming and Computer Software. 2023. V. 49. № 3. P. 178–186.
https://doi.org/10.1134/S0361768823030064
32. *Rusch M., Bickford N., Subtil N.* Introduction to Vulkan Ray Tracing // Ray Tracing Gems II. NVIDIA. 2021. P. 213–255.
https://doi.org/10.1007/978-1-4842-7185-8_16
33. *Sjoholm J.* Best Practices for Using NVIDIA RTX Ray Tracing (Updated) // NVIDIA Technical Blog. Jul 25, 2022.
https://developer.nvidia.com/blog/best-practices-for-using-nvidia-rtx-ray-tracing-updated/
34. *Lefrançois M.-K.* Intersection Shader // NVIDIA Vulkan Ray Tracing Tutorials. 2020–2023.
https://github.com/nvpro-samples/vk_raytracing_tutorial_KHR/tree/master/ray_tracing_intersection
35. C++ reference. Containers library. Sequence containers. 2023.
https://en.cppreference.com/w/cpp/container
36. Vulkan 1.3.275 – A Specification (with all ratified extensions) // The Khronos Vulkan Working Group. 2024.
https://registry.khronos.org/vulkan/specs/1.3-khr-extensions/pdf/vkspec.pdf
37. *Wong U., Whittaker W., Jones H., Whittaker R.* NASA Planetary Pits and Caves Analog Dataset. 2014.
https://ti.arc.nasa.gov/dataset/caves/

ПРОГРАММИРОВАНИЕ, 2024, № 3, с. 54–66

КОМПЬЮТЕРНАЯ ГРАФИКА И ВИЗУАЛИЗАЦИЯ

УДК 04.092

ДВОЙНОЕ ПРЕДСТАВЛЕНИЕ ГЕОМЕТРИИ ДЛЯ УСКОРЕНИЯ ТРАССИРОВКИ ЛУЧЕЙ В ОПТИЧЕСКИХ СИСТЕМАХ С ПОВЕРХНОСТЯМИ СВОБОДНОЙ ФОРМЫ

© 2024 г. Д. Д. Жданов[a], *, И. С. Потемин[a], **, А. Д. Жданов[a], ***

[a] *Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики*
197101 Санкт-Петербург, Кронверкский пр., 49, Россия

**E-mail: ddzhdanov@mail.ru,*

***E-mail: ipotemin@yandex.ru,*

****E-mail: andrew.gtx@gmail.com,*

Поступила в редакцию: 19.01.2024 г.
После доработки: 19.01.2024 г.
Принята к публикации: 24.01.2024 г.

В статье исследуется возможность использования двойного представления геометрии для повышения скорости трассировки лучей и обеспечения устойчивости результатов моделирования распространения света в сложных оптических системах, содержащих поверхности свободной формы, заданные полиномами высокого порядка (до 34-го порядка) или полиномами Якоби. Был проведен анализ традиционных методов представления данной геометрии как в виде треугольной сетки, так и в виде аналитического выражения. Проведенный анализ продемонстрировал недостатки традиционных подходов, которые заключаются в недостаточной точности вычисления координат точки встречи луча с треугольной сеткой, а также в неустойчивости результатов поиска точки встречи касательных лучей с аналитической поверхностью при использовании существующих методов расчета. В результате было предложено использовать двойное представление геометрии в виде грубого приближения поверхности треугольной сеткой, которое в дальнейшем используется как начальное приближение для поиска точки встречи луча с поверхностью, заданной аналитическим выражением. Это решение позволило существенно ускорить сходимость аналитических методов и повысить устойчивость их решений. Использование библиотеки Intel® Embree для быстрого поиска точки встречи луча с грубой треугольной сеткой и векторной модели вычислений для уточнения координат точки пересечения луча с геометрией, представленной аналитическим образом, позволило разработать и реализовать алгоритм трассировки лучей в оптической системе, содержащей поверхности с двойным представлением геометрии. Эксперименты, проведенные с использованием разработанного и реализованного алгоритма, показывают значительное ускорение трассировки лучей при сохранении точности вычислений и высокой стабильности результатов. Результаты были продемонстрированы на примере расчета функции рассеивания точки и бликов для двух объективов с поверхностями свободной формы, заданными полиномами Якоби. Кроме того, для двух данных объективов был произведен расчет изображения, формируемого RGB-D-объектом, имитирующим реальную сцену.

Ключевые слова: трассировка лучей, рендеринг, треугольная сетка, поверхности произвольной формы

DOI: 10.31857/S0132347424030067, **EDN:** QANTVA

1. ВВЕДЕНИЕ

Для уменьшения габаритных размеров и числа линз при сохранении или повышении качества изображения в современных оптических системах применяются линзы с поверхностями свободной формы. Такие объективы имеют, как правило, малый размер и применяются в носимых устройствах, например, камерах смартфонов, системах сканирования, устройствах виртуальной, дополненной и смешанной реальностей. На рис. 1 представлен пример объектива камеры дополненной реальности.

Способ задания поверхностей свободной формы может быть различен. Это могут быть, например, асферические поверхности, заданные полиномами высокого порядка, поверхности, заданные сплайном, поверхности, заданные полиномами Якоби. Для эффективного проектирования и компьютерного моделирования таких систем используются средства фотореалистичной

визуализации, которые позволяют формировать не только реалистичное изображение сцены, но и изображение, наблюдаемое через объектив камеры с учетом всех его аберрационных искажений. Существуют два основных подхода для формирования таких изображений. Во-первых, это применение передаточной функции объектива к точкам изображения сцены. При этом передаточная функция должна быть вычислена для всех видимых точек сцены с учетом их пространственного положения относительно объектива камеры. Второй подход основан на трассировке лучей, и в этом случае объектив камеры становится неотъемлемой частью сцены. Точки объектов сцены становятся источниками каустического освещения для изображения, формируемого камерой. Каждый из подходов имеет свои преимущества и недостатки. Преимущество первого подхода заключается в возможности физически корректного расчета функции рассеивания отдельных точек объекта. Передаточная функция, используемая при этом расчете, может быть вычислена с учетом дифракции на краях линз. Основным недостатком этого подхода является его трудоемкость, сложность сопряжения со стохастическими методами трассировки лучей в сцене и определение видимости точек трехмерной сцены со стороны объектива камеры. Все это не позволяет применять данный подход к построению изображений трехмерных сцен и ограничивает его применение синтезом изображений плоских объектов, ортогональных оси объектива. Основным недостатком второго подхода является невозможность учета дифракционных явлений в объективе камеры и шум, вызванный стохастическим методом трассировки лучей в сцене. Однако этот подход позволяет физически корректно передать трехмерность изображения сцены и глубину резкости объектива.

Основой обоих подходов является трассировка лучей в объективах, которая может быть использована как для расчета передаточных функций, так и для передачи яркости точек объекта на матрицу изображения объектива. Поскольку поверхности объективов имеют достаточно сложную форму, описываемую полиномами высокой степени, например, полиномами Якоби [1], которые в последнее время все чаще стали использоваться для описания форм асферических поверхностей свободной формы, то задача поиска точки встречи луча с такими поверхностями становиться крайне актуальной. В работе [2] предложены параметры разложения полиномов Якоби, которые позволяют избежать ошибок при вычислении значений полиномов высокого порядка, поскольку в ряде случаев степень полинома может достигать нескольких десятков. Кроме того, предложенные параметры разложения позволяют оценить проблемы, связанные с технологией изготовления асферических поверхностей [3].

В вычислительной оптике для поиска точки встречи луча с асферической поверхностью используется метод Федера [4]. Несмотря на то что метод был предложен более 70 лет назад, он не претерпел серьезных изменений до настоящего времени. Суть этого метода заключается в последовательном поиске точки встречи луча с касательной плоскостью к асферической

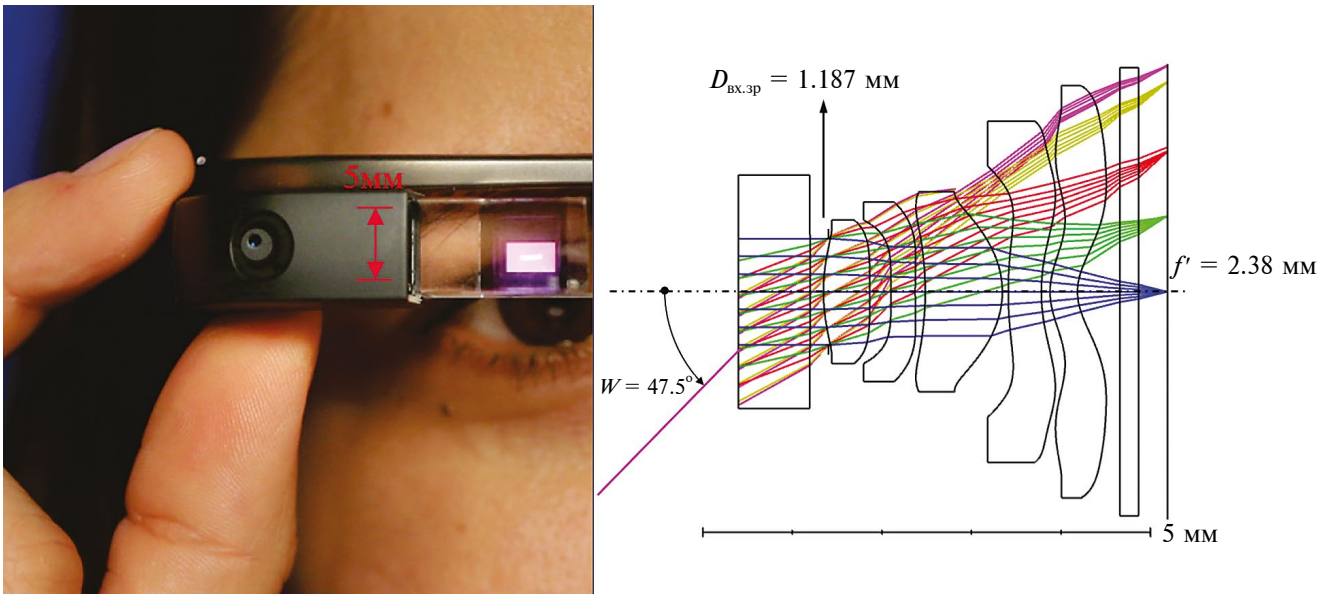


Рис. 1. Пример объектива камеры дополненной реальности.

поверхности. Параметры касательной плоскости уточняются после каждого найденного пересечения в соответствии с параметрами асферической поверхности для координат в найденной точке. В качестве первого приближения касательная плоскость устанавливается

$$z = 0, \tag{1}$$

где ось z – оптическая ось, а асферическая поверхность задана в локальной системе координат с вершиной в точке (0, 0, 0).

Как правило, в компьютерных системах вычислительной оптики асферическая поверхность задается двумя типами полиномов. Это стандартное полиномиальное представление:

$$z(\rho) = \frac{c\rho^2}{1 + \sqrt{1 - \varepsilon c^2 \rho^2}} + \sum_{m=0}^{M} a_m \rho^{2m+4}, \tag{2}$$

или так называемая асферическая поверхность Q типа (Q-type), заданная полиномом Якоби:

$$z(\rho) = \frac{c\rho^2}{1 + \sqrt{1 - \varepsilon c^2 \rho^2}} + D_{con}\left(\frac{\rho}{\rho_{\max}}\right), \tag{3}$$

где c, ε, a_m – коэффициенты асферической поверхности; $D_{con}(u) = u^4 \sum_{m=0}^{M} a_m Q_m^{con}\left(u^2\right)$, $\rho^2 = x^2 + y^2$, а Q_m^{con} – задаются полиномами Якоби следующего вида:

$$Q_m^{con}(u) = P_m^{(0.4)}(2u - 1), \tag{4}$$

где $P_0^{(0.4)}(u) = 1$, $P_1^{(0.4)}(u) = -(5 - 6u)$, $P_2^{(0.4)}(u) = 15 - 14u(3 - 2u)$ и т. д.

Оба типа полиномиального представления используют одинаковый способ подгонки луча к поверхности. Для нецентрированных асферических поверхностей метод Федера претерпел незначительные изменения [5]. Данная модификация метода, предложенного Федером, касалась возможности его применения не только к нецентрированным поверхностям, но и к различного рода неосесимметричным поверхностям, включая поверхности дифракционных решеток [6].

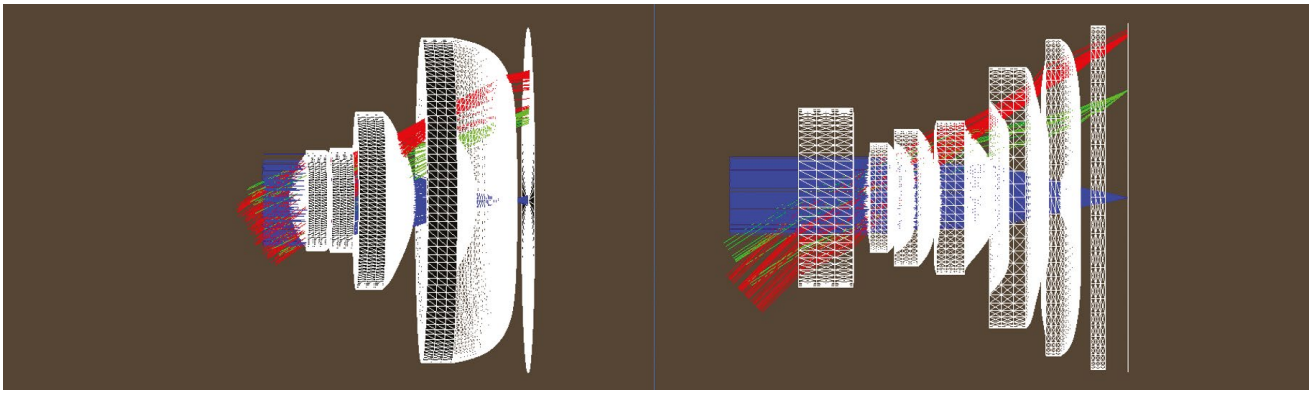
Несмотря на очевидную простоту методов, предложенных Федером, многократное вычисление полиномов высокой степени, особенно полиномов Якоби, является очень затратной процедурой и требует оптимизации. Кроме того, при вычислении значений полиномов высокой степени возможна потеря точности, а в ряде случаев это может привести к переполнению данных. Такая ситуация возникает, если начальная точка и направление луча значительно отличаются от расчетных значений, например, при расчете бликов в линзовых объективах.

Для ускорения поиска точки встречи луча с поверхностью и повышения стабильности вычислений можно использовать подходы, применяемые в компьютерной графике. Простейшим решением является аппроксимация асферической поверхности треугольной сеткой. При этом вершины треугольников совпадают с точками на асферической поверхности, а нормали в вершинах – с нормалями в соответствующих точках поверхности. Однако для достижения требуемой точности вычислений треугольная сетка должна иметь треугольники малого размера, что приводит к большому числу треугольников в сетке, т. е. увеличению объема памяти для хранения сетки, и замедлению вычислений. Для устранения этих недостатков в работе предлагается использовать двухуровневое представление геометрии, когда на первом уровне находится треугольная сетка, грубо аппроксимирующая асферическую поверхность и являющаяся ее начальным приближением для метода Федера, а на втором уровне – собственно асферическая поверхность.

2. ОЦЕНКА ТОЧНОСТИ ПРЕДСТАВЛЕНИЯ ГЕОМЕТРИИ

В качестве метрики, позволяющей оценить требуемую точность представления геометрии, использовалось среднеквадратичное отклонение функции рассеивания точки (ФРТ, Point Spread Function – PSF) от эталонной ФРТ, вычисленной классическим методом по формулам Федера [4]. Оценка проводилась для двух объективов, содержащих асферические поверхности, представленные формулами (2) и (3) соответственно. На рис. 2а показана схема четырехлинзового объектива с асферическими поверхностями, представленными формулой (2), а на рис. 2б показана схема пятилинзового объектива с асферическими поверхностями, представленными формулой (3). Диаметр входных зрачков обоих объективов составлял величину порядка 1 мм.

Для двух объективов был вычислен ряд ФРТ. ФРТ вычислялись для бесконечно удаленной точки, находящейся на оси объектива, для разных параметров модели геометрии объективов. Эталонные ФРТ, вычисленные по формулам Федера [4], представлены на рис. 3 для четырех- и пятилинзовых объективов соответственно. Поверхности исследуемой модели линзового



а) б)

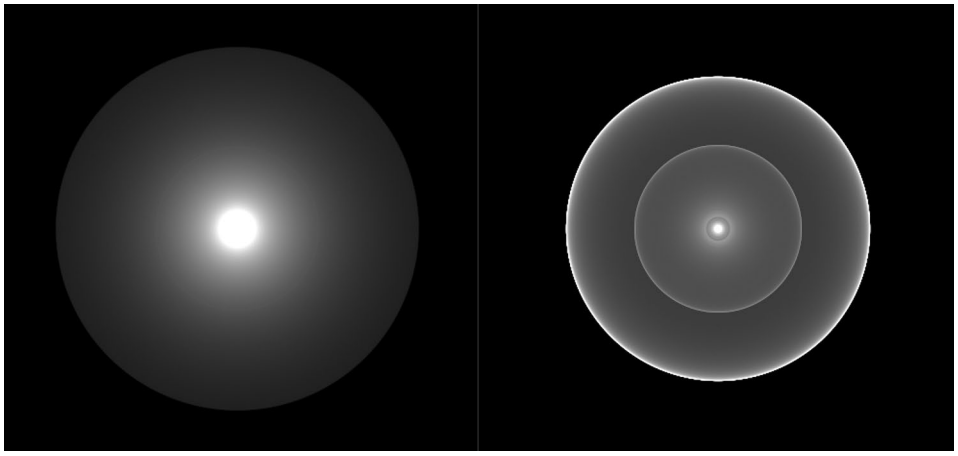
Рис. 2. Четырехлинзовый объектив с асферическими поверхностями, представленными формулой (2) – (а), пятилинзовый объектив с асферическими поверхностями, представленными формулой (3) – (б).

объектива представляли собой радиальную треугольную сетку, показанную на рис. 2, а исследуемый параметр модели – разрешение сетки, т. е. число треугольников. Пучок лучей от бесконечно удаленной точки равномерно заполнял входной зрачок объектива. Лучи в пучке распределялись по квадратной сетке с разрешением 10.000×10.000. Вычисления производились на компьютере с восьмиядерным процессором AMD Ryzen 9 и 32 Гб оперативной памяти.

Были рассмотрены три варианта задания параметров радиального и углового разрешения треугольной сетки. Первый вариант – 24×54 (1.296 треугольников), второй вариант – 240×540 (129.600 треугольников), третий вариант – 2.400×5.400 (12.960.000 треугольников). Первый параметр – радиальное разрешение, второй – угловое. Были выполнены расчеты и проведена оценка отклонений ФРТ от эталонных значений. Поиск точки встречи луча с треугольной сеткой выполнялся средствами компьютерной графики [7]. Соответствующие изображения отклонений ФРТ от эталонных представлены на рис. 4. Размер изображения для четырехлинзового объектива, представленного на рис. 3а, был равен 100 мкм, для пятилинзового объектива, показанного на рисунке 3(б), – 5 мкм. Разрешение изображения в обоих случаях составляло 500×500 точек. Относительное среднеквадратичное отклонение ФРТ от эталона представлено в табл. 1.

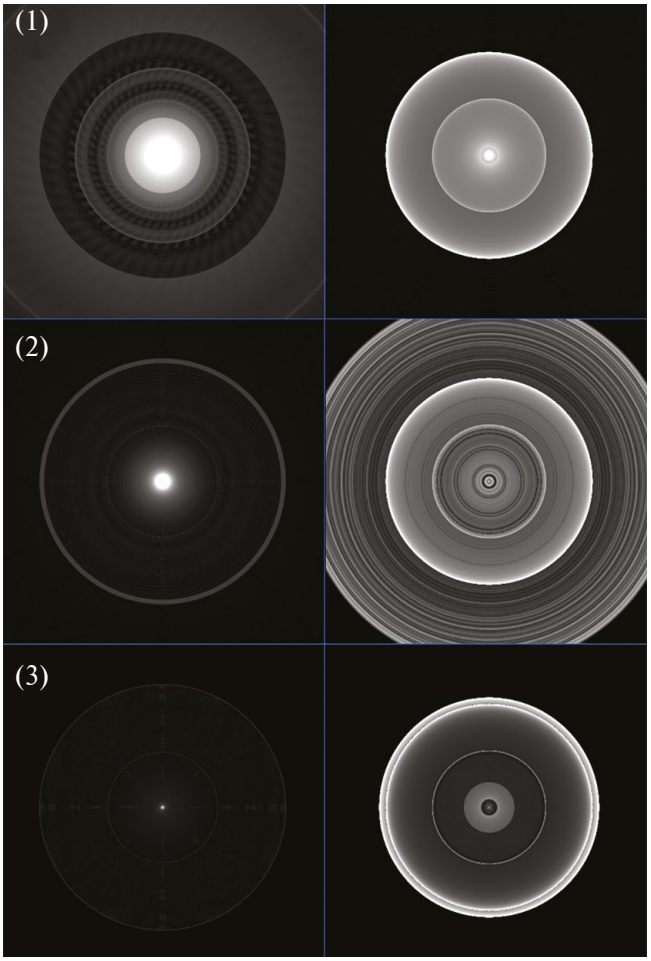
Таблица 1. Относительное среднеквадратичное отклонение ФРТ от эталона

Параметры модели (разрешение)	RMS% модель рис. 2а	RMS% модель рис. 2б
24×54	133%	97%
240×540	96%	70%
2.400×5.400	1%	55%



а) б)

Рис. 3. ФРТ четырехлинзового объектива с асферическими поверхностями, представленными формулой (2) – (а), ФРТ пятилинзового объектива с асферическими поверхностями, представленными формулой (3) – (б).



а) б)

Рис. 4. Отклонения ФРТ от эталонов для четырехлинзового (а) и пятилинзового (б) объективов и трех вариантов разрешения треугольной сетки: 24 × 54 (1), 240 × 540 (2) и 2400 × 5400 (3).

По результатам, представленным на рис. 4 и в табл. 1, видно, что высокое разрешение треугольной сетки не всегда может обеспечить требуемую точность вычислений. Особенно это касается высококачественных объективов, у которых ФРТ имеет радиус порядка 1 мкм.

Для повышения точности, стабильности и ускорения вычисления точки встречи луча с поверхностью было реализовано комбинированное решение, заключающееся в двойном представлении геометрии, когда поиск точки встречи луча с поверхностью разбивается на два этапа, сперва происходит быстрый поиск точки встречи луча с грубой треугольной сеткой, от которой далее осуществляется подгонка к асферической поверхности по алгоритмам [4] и [1]. Это решение аналогично алгоритму “displacement mapping” [8], однако цель данного решения не уменьшение объема данных, а повышение стабильности, точности и скорости расчета.

3. ДВОЙНОЕ ПРЕДСТАВЛЕНИЕ ГЕОМЕТРИИ

Для исследования преимуществ и недостатков метода, основанного на двойном представлении геометрии, была реализована программа синтеза изображений, основанная на детерминистическом и квазидетерминистическом методе трассировки лучей. Хотя эта программа не является полноценной рендер системой, она может в детерминистическом режиме выполнять расчет ФРТ и изображений RGB-D-объектов, формируемых линзовыми или зеркально-линзовыми объективами. В квазидетерминистическом режиме эта программа позволяет выполнять расчет бликов, т. е. ФРТ, формируемую в результате переотражений между поверхностями линз. Ядром данной программы являлся метод расчета точки встречи луча с поверхностью, для реализации которого использовалась библиотека Embree от компании Intel [9]. Основные причины выбора API Embree заключались в следующем:

1. Исключительная ориентация на методы поиска точки встречи луча с поверхностью.
2. Встроенные высокопроизводительные методы поиска точки встречи луча с треугольной сеткой.
3. Поддержка векторных инструкций процессора (принцип Одиночного потока Команд, Множественного потока Данных – ОКМД, Single Instruction, Multiple Data – SIMD).
4. Поддержка параллельных вычислений на базе Intel oneTBB API [10].
5. Поддержка пользовательской геометрии и специальных фильтров, позволяющих модифицировать форму поверхности.
6. Интуитивный программный интерфейс, наличие большого числа примеров.

Трассировка лучей в линзовых объективах, формирующих изображение, предполагает строго фиксированный порядок следования поверхностей, т. е. траекторий луча, например, после первой поверхности должна идти вторая поверхность, после второй – третья и т. д. Если идет расчет бликов, то порядок следования может быть нарушен, однако он остается фиксированным, т. е. в случае блика порядок следования меняет знак, например, после отражения на третьей поверхности следующая поверхность будет вторая. Кроме того, фиксированный порядок следования поверхностей позволяет избежать неоднозначности при поиске точки встречи луча с поверхностью в случае их близкого расположения, например, при наличии воздушного зазора между двумя поверхностями. Такая специфика формирования

трасс лучей в линзовых объективах позволяет эффективно использовать возможности API Embree, которое позволяет иметь неограниченное число сцен в одном вычислительном устройстве. Исходя из специфики трассировки лучей в объективе, формирующем изображение, и возможностей API Embree каждая поверхность объектива представляет собой отдельную сцену, а порядок следования поверхностей заменяется на порядок следования сцен в API Embree. Данный подход решает сразу несколько проблем. Во-первых, поиск точки пересечения для Embree ограничен одной поверхностью, например, одной треугольной сеткой. Для сложных объективов это позволяет значительно уменьшить число объектов (в десятки раз), с которыми происходит поиск точки встречи луча с поверхностью, и, соответственно, повысить скорость поиска. Во-вторых, это автоматически решает проблему неоднозначности поиска точки встречи луча с близкорасположенными или соприкасающимися поверхностями, поскольку эти поверхности принадлежат разным сценам. Данная организация сцены объектива проиллюстрирована на рис. 5.

Детерминистическая трассировка лучей в объективе может использоваться не только для расчета ФРТ, но и для расчета волновых аберраций на его выходном зрачке, что позволяет вычислить передаточную функцию объектива. Кроме того, квазидетерминистическая трассировка лучей может использоваться не только для расчета бликов, где порядок следования поверхностей определяется случайным образом по коэффициенту зеркального отражения линзовой поверхности, но и для расчета оптических характеристик многоканальных и мультиконфигурационных оптических систем. Для данных систем задается несколько траекторий лучей, соответствующих разным каналам (конфигурациям).

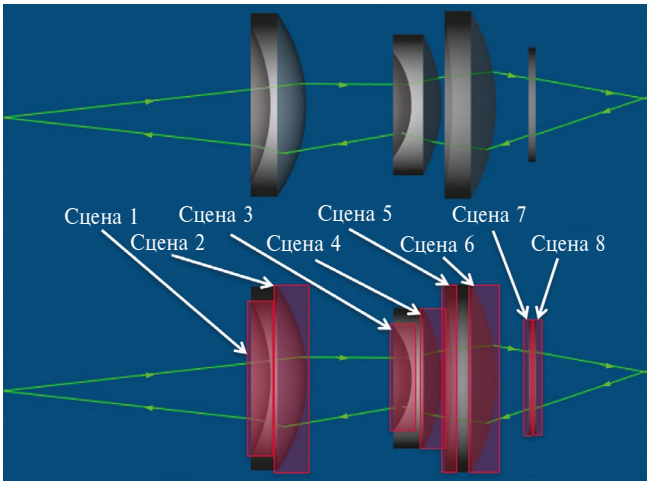


Рис. 5. Оптическая система объектива (верхний рисунок), представленная в виде набора Embree сцен (нижний рисунок).

Одним из недостатков API Embree с точки зрения вычислительной оптики является то, что все расчеты с вещественными числами выполняются с вещественными числами одинарной точности, тогда как в вычислительной оптике используются вещественные числа с двойной точностью. Одинарная точность может быть недостаточна, особенно если степень полинома асферической поверхности достигает нескольких порядков, а луч сильно отклоняется от своей расчетной траектории, например, это возникает при расчете бликов. Поэтому для повышения точности вычислений используются специальные фильтрующие callback-функции, устанавливаемые средствами API Embree (**rtcSetGeometryIntersectFilterFunction**). Функция фильтрации вызывается при каждом выполнении операции поиска точки встречи луча с поверхностью (**RTCIntersect...**) и может изменить параметры точки встречи луча с поверхность, т. е. точка пересечения, найденная с одинарной точностью, может быть пересчитана в двойную точность. Для вычислений, использующих векторные инструкции процессора, обработка несколько усложняется, поскольку вектор, состоящий из восьми вещественных чисел одинарной точности, необходимо преобразовать в два вектора из четырех вещественных чисел двойной точности, которые будут последовательно обрабатываться функцией фильтрации. Данный процесс проиллюстрирован на рис. 6.

Однако основное применение функций фильтрации – это реализация двойного представления геометрии. Основная идея метода заключается в том, что геометрия асферической поверхности должна иметь два представления. Первое, грубое представление, – это треугольная сетка низкого разрешения. Средства API Embree (функция **RTCIntersect...**) позволяют найти точку встречи луча с некоторой аппроксимацией истиной формы поверхности, заданной в виде треугольной сетки. Для небольшого числа треугольников в сетке поиск точки встречи выполняется исключительно быстро. Далее от найденной точки происходит итерационный поиск точки встречи луча с асферической поверхностью алгоритмами [4] и [1]. Итерационный поиск выполняется функцией фильтрации API Embree, которая возвращает точку пересечения луча с асферической поверхностью, и функция **RTCIntersect...** возвращает полученное значение как собственное пересечение луча с поверхностью. Очевидно, чем бли-

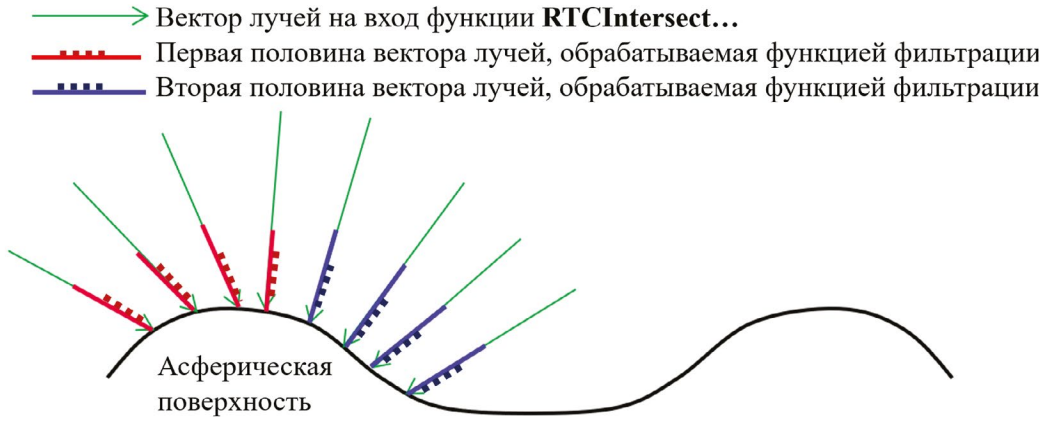


Рис. 6. Разбиение ОКМД-вектора из восьми вещественных чисел одинарной точности на два вектора из четырех вещественных чисел двойной точности в функции фильтрации.

же первое приближение оказалось к истинной поверхности, тем меньшее число итераций будет выполнено в процессе подгонки. Рис. 7 иллюстрирует этот алгоритм.

На рис. 7 также можно увидеть, что основное преимущество данного алгоритма появляется при наклонном падении лучей на поверхность. Если луч идет параллельно оптической оси, то преимущество двойного представления геометрии пропадает, поскольку начальная точка также оказывается близко от истинной поверхности, а время, затрачиваемое на поиск точки пересечения луча с треугольной сеткой, не является нулевым. С точки зрения практического применения преимущество двойного представления геометрии возникает при расчете внеполевых точек

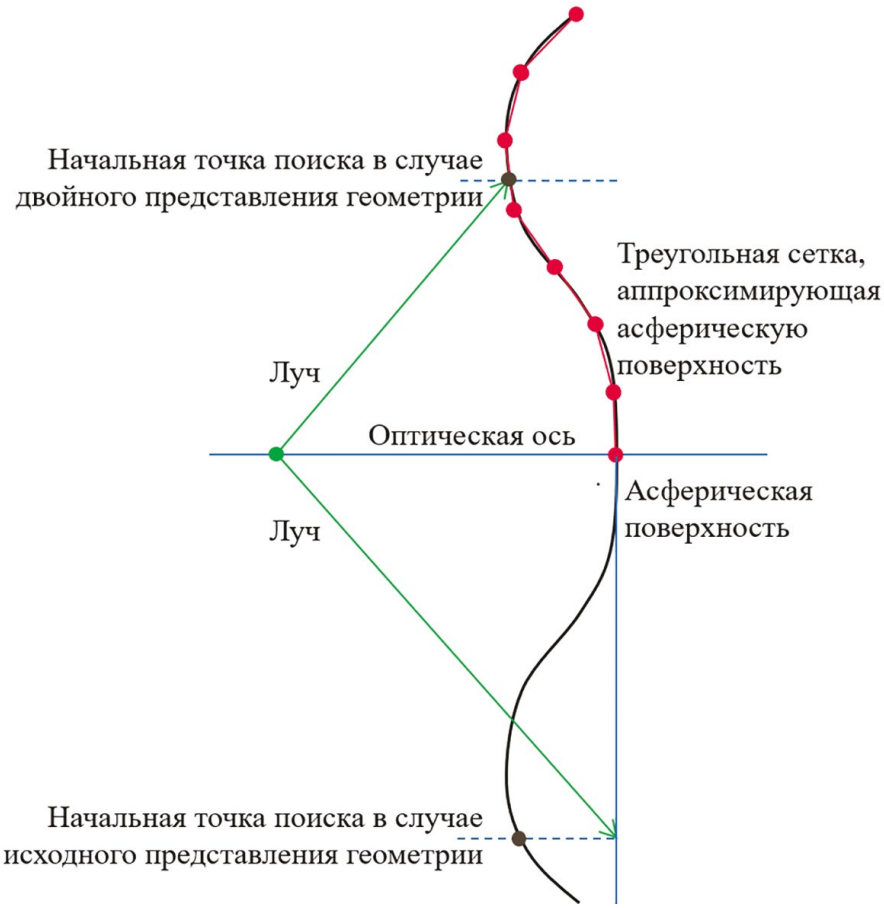


Рис. 7. Определение начальной точки встречи луча с асферической поверхностью, верхняя часть рисунка – случай двойного представления геометрии, нижняя часть рисунка – прямая подгонка по алгоритмам [4] и [1].

изображения и расчете бликов в линзовых системах, когда лучи идут не по расчетным траекториям. Двойное представление геометрии также решает проблему стабильности вычислений точки встречи луча с асферическими поверхностями, заданными полиномами высокой степени. Если луч пытается найти точку встречи с поверхностью в области, находящейся намного выше ее расчетной высоты, то может возникнуть переполнение разрядов вещественных чисел или потеря точности при сложении (вычитании) больших чисел.

Поскольку разработанная программа была ориентирована на синтез изображений, формируемых оптическими системами, то единственным источником формирования освещенности изображения является яркость прямого виденья. Данное ограничение позволило минимизировать программный интерфейс методов работы с данными оптической системы. Для поиска точки встречи луча с поверхностью использовалась библиотека Embree. Для преобразования луча на поверхности задавались коэффициенты отражения (для линзовых поверхностей эти коэффициенты рассчитывались по формулам Френеля) и применялись соответственно законы отражения и преломления. Для оптических сред рассчитывалось поглощение в материале стекла. Лучи испускались источником света (точкой или RGB-D-изображением с заданными световыми характеристиками) и после прохождения через всю оптическую систему их световой поток аккумулировался на приемнике излучения.

Важным моментом ускорения вычислений является организация вычислений в параллельном режиме. Компания Intel предлагает библиотеку oneTBB для организации параллельных вычислений. Важно, что функции Embree оптимизированы на работу в параллельном режиме с функциями oneTBB, и это позволяет достичь практически стопроцентной загрузки всех ядер процессора при трассировке лучей. Алгоритм расчета ФРТ в параллельном режиме представлен на рис. 8.

Данный алгоритм подходит не только для расчета ФРТ, но и для расчета бликов и изображений RGB-D-объектов. В случае расчета бликов меняется алгоритм выбора следующей точки по списку трассировки лучей. Если при расчете обычной ФРТ поверхности идут строго по списку, то при расчете бликов список инвертируется, когда на поверхности выбирается событие отражения. При расчете изображений RGB-D-объектов происходит случайный выбор точки на RGB-изображении, а начало луча переносится на дистанцию, хранящуюся в карте глубин изображения,

Рис. 8. Пример алгоритма расчета ФРТ.

относительно центра входного зрачка. Координата луча на зрачке, которой соответствует угол относительно оси объектива, выбирается случайным образом. Поток, переносимый лучом, в данном случае будет пропорционален потоку, испускаемому соответствующей точкой RGB-D-изображения.

4. ПРИМЕРЫ РАСЧЕТА ФРТ И ИЗОБРАЖЕНИЙ, ФОРМИРУЕМЫХ ЛИНЗОВЫМИ ОБЪЕКТИВАМИ

Разработанный программный модуль распространения лучей в объективе может использоваться не только в задачах прикладной оптики.

Данное решение может найти применение как в системах компьютерной графики для синтеза фотореалистичных изображений трехмерных сцен, формируемых оптической системой, так и в задачах физически корректного виртуального прототипирования линзовых объективов в условиях реальной эксплуатации. Основное преимущество данного решения заключается в возможности внедрения детерминистической модели трассировки лучей в задачи рендеринга, основанного на методах Монте-Карло, который использует не только прямую, но и обратную трассировку лучей. В данном случае модуль трассировки лучей в объективе выступает в виде черного ящика, на вход которого падает луч, а на выходе формируется луч, покидающий объектив. Входом и выходом луча могут служить зрачки объектива или его первая и последняя поверхности, если речь идет о расчете бликов. Используя модель черного ящика [11], модуль трассировки лучей в объективе был встроен в систему компьютерного моделирования Lumicept [12]. Данное решение позволило упростить задание линзовых систем и линзовой камеры в системе реалистичного рендеринга.

Для разработанного программного модуля было проведено тестирование времени расчета ФРТ по разработанным алгоритмам и исходным решениям, описанным в [4] и [1]. Для тестирования использовались объективы, представленные на рис. 2. Тестирования производились на компьютере с восьмиядерным процессором AMD Ryzen 9 и 32 Гб оперативной памяти. В модели двойного представления геометрии поверхности линзы использовались 1.296 треугольника, что соответствовало разрешению треугольной сетки 24×54. Число трассируемых лучей задавалось исходя из регулярной решетки с разрешением 10.000×10.000 на входном зрачке объектива, что в сумме давало 100.000.000 лучей. Поля зрения ω для четырехлинзового объектива составляли 0°, 26° и 37° по вертикали, для пятилинзового объектива поля зрения ω – 0°, 19° и 45° по вертикали. Размеры изображений для соответствующих полей зрения составили 100, 1.000 и 2.000 мкм для четырехлинзового объектива и 5, 15 и 100 мкм для пятилинзового объектива. Разрешение изображения составило 500×500 точек. Длина волны, на которой выполнялся расчет, – 0.5876 мкм. Изображения ФРТ, полученные в результате тестирования, представлены на рис. 9. Время, затраченное на трассировку данного числа лучей для разработанного алгоритма и алгоритма, описанного в работах [4] и [1], представлено в табл. 2. Изображения, полученные с использованием новых и старых алгоритмов, совпадают и не приводятся на рис. 9.

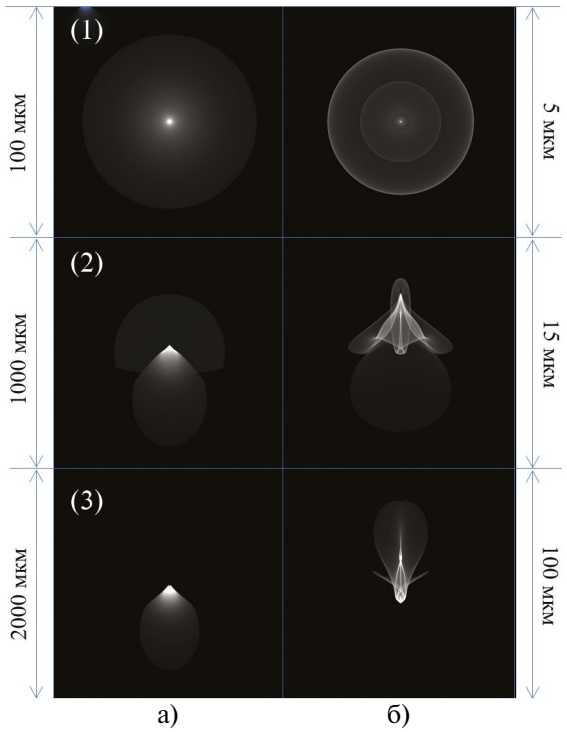


Рис. 9. ФРТ четырехлинзового объектива (а) для трех полей зрения: 0° (1), 26° (2) и 37° (3) и ФРТ пятилинзового объектива (б) для трех полей зрения: 0° (1), 19° (2) и 45° (3).

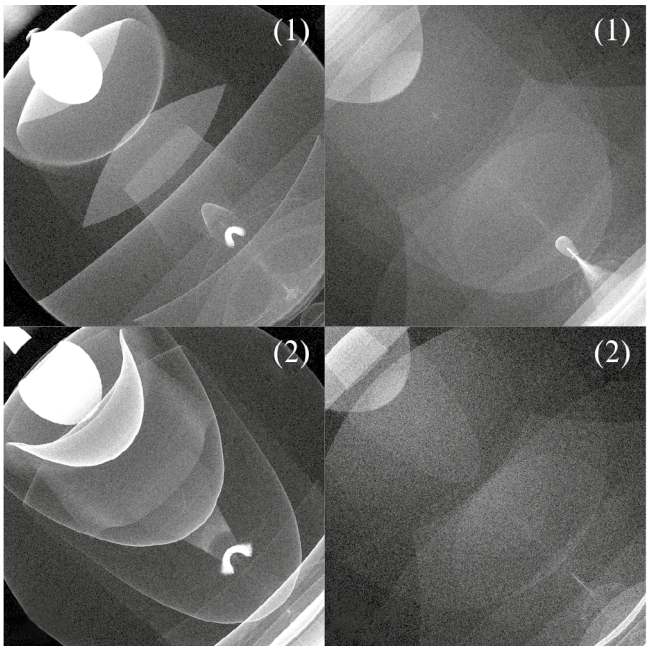
Исследование показало, что для осевых пучков разработанный алгоритм не дает ускорения расчета (скорее появляется небольшое замедление), однако для внеосевых пучков, располагающихся под большими углами к оптической оси объектива, новый алгоритм дает значительное ускорение, которое достигает 2.5 раз. Следует отметить, что при синтезе изображений, формируемых объективом, преобладают внеосевые пучки, поэтому

Таблица 2. Время расчета трех полей зрения для четырех- и пятилинзового объективов

Четырехлинзовый объектив	$\omega = 0°$	$\omega = 26°$	$\omega = 37°$
Двухуровневая модель (время расчета, мс)	4.025	3.788	3.067
Модель по формуле (2) (время расчета, мс)	3.943	4.894	5.161
Пятилинзовый объектив	$\omega = 0°$	$\omega = 19°$	$\omega = 45°$
Двухуровневая модель (время расчета, мс)	7.291	4.910	2.984
Модель по формуле (3) (время расчета, мс)	7.051	7.657	6.923

новый алгоритм может обеспечить значительное ускорение расчета. Также новый алгоритм обеспечивает большую стабильность вычислений трасс лучей, если они сильно отклоняются от расчетных траекторий, например, при расчете бликов.

Для двух рассмотренных моделей объективов был произведен расчет бликов, т. е. ФРТ, формирующейся с учетом зеркального переотражения между поверхностями линз. Событие зеркального отражения выбиралось случайным образом в соответствии с коэффициентом отражения, вычисленным по формуле Френеля для границы двух диэлектриков. При выборе события отражения менялся порядок следования поверхностей и луч детерминистически распространялся в обратную сторону. В результате четного числа переотражений формировалась “бликовая” ФРТ. В данном тесте исходные лучи распределялись случайным образом (равномерно) на первой поверхности объектива. Число трассируемых лучей – 1.000.000.000. Поля зрения ω для четырехлинзового объектива составляли 40° и 45° по диагонали, а для пятилинзового объектива поля зрения ω – 48.5° и 50° по диагонали. Размеры изображений составили 2.8 мм для четырехлинзового объектива и 3.6 мм – для пятилинзового объектива. Разрешение изображения составило 500 × 500 точек. Длина волны, на которой выполнялся расчет, – 0.5876 мкм. Время расчета составило 16011 мс для четырехлинзового объектива и 15034 мс – для пятилинзового объектива. Несмотря на небольшой шум изображения, обусловленный недостаточным числом лучей, блики и их характер хорошо видны. Результаты расчета представлены на рис. 10.



а) б)

Рис. 10. ФРТ блика четырехлинзового объектива (а) для двух полей зрения: 48.5° (1) и 50° (2) и ФРТ пятилинзового объектива (б) для двух полей зрения: 48.5° (1), 50° (2).

Кроме того, для рассмотренных моделей объектива были синтезированы изображения RGB-D-объектов, представляющих собой цветную карту глубин, полученную с помощью компьютерной системы Lumicept [12]. На рис. 11 представлены исходные данные для моделирования:

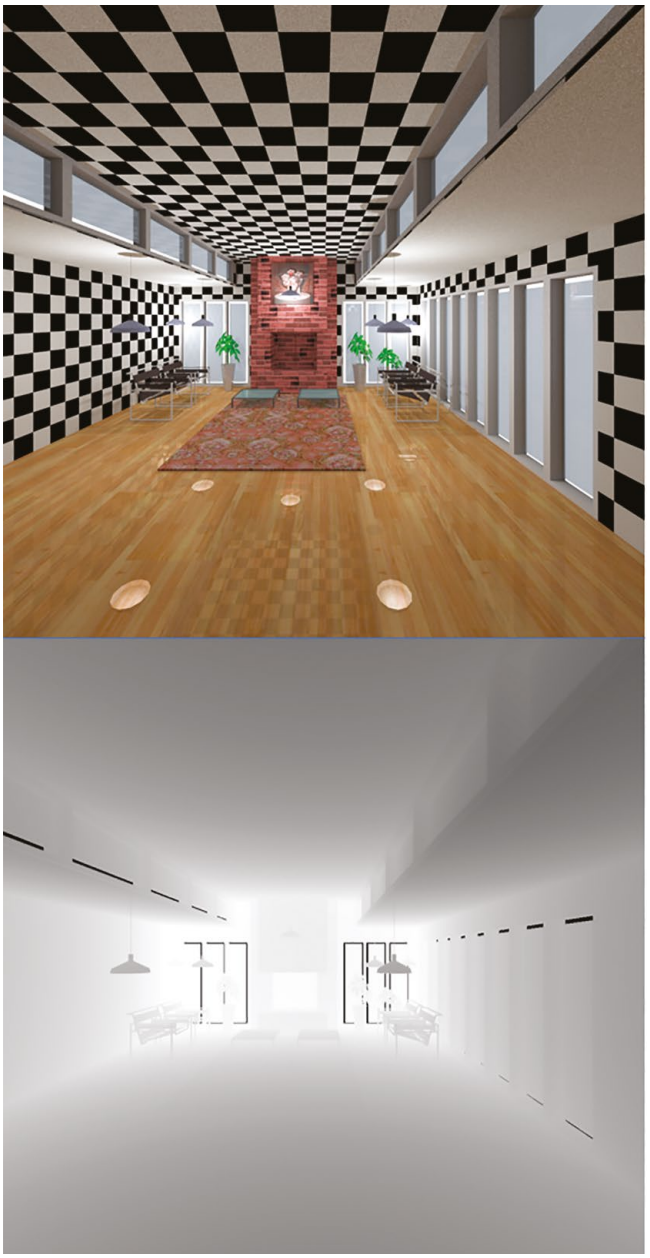


Рис. 11. RGB-изображение (сверху) и карта глубин (снизу).

RGB-изображение (RGB-карта яркостей) и соответствующая карта глубин сцены. Изображения были построены для двойного поля зрения 110° с разрешением 1200 × 1200.

Изображения, формируемые четырех- и пятилинзовым объективами были рассчитаны для разрешения 600 × 600 точек и длины волны 0.5876 мкм. Число трассируемых лучей – 2.000.000.000. Диагональные поля зрения четырех- и пятилинзового объективов составили 37° и 47.5° соответственно. Время расчета составило 79.534 мс для четырехлинзового объектива и 122.134 мс – для пятилинзового объектива.

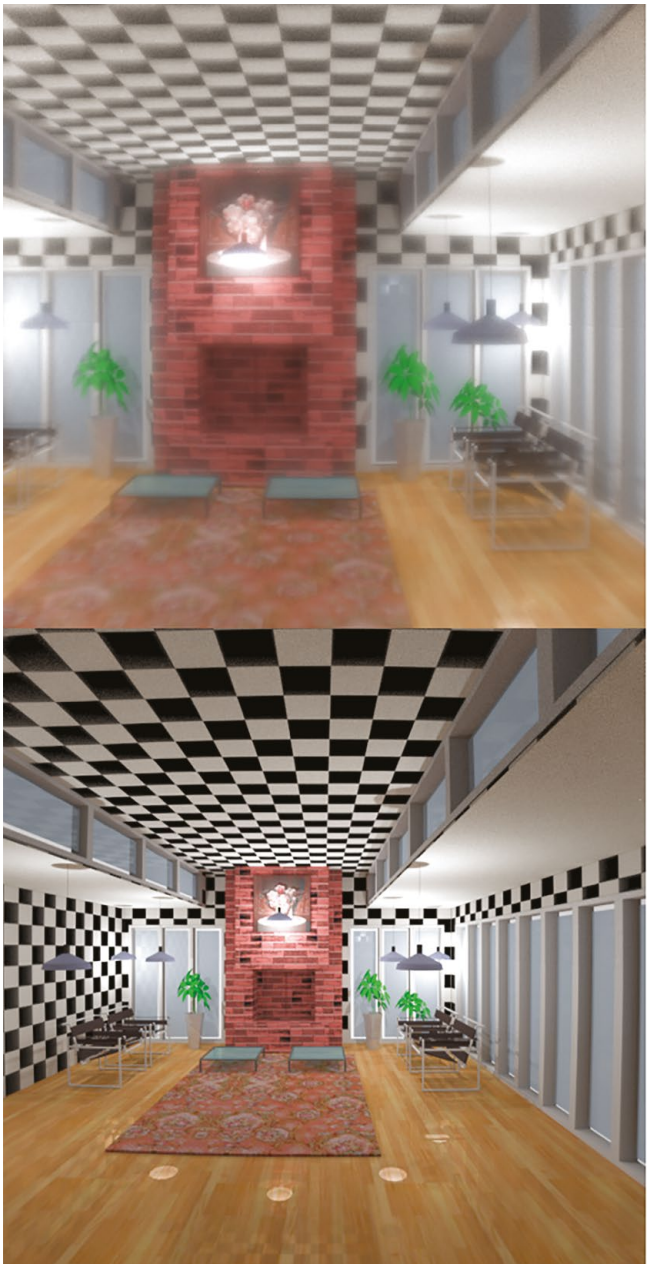


Рис. 12. Изображения, построенные четырехлинзовым (сверху) и пятилинзовым объективом (снизу).

Изображение, построенное для четырехлинзового объектива, представлено на верхней части рис. 12, для пятилинзового объектива – на нижней части рис. 12.

Изображение, формируемое четырехлинзовым объективом, сильно размазано, что соответствует большим аберрациям объектива и согласуется с ФРТ, представленными на рис. 9. На изображении пятилинзового объектива видна небольшая дисторсия и размытие изображения по краю поля зрения. Размытие изображения, видимое как “затекание” светлых квадратов шахматной текстуры на черные квадраты на краю поля зрения, соответствует ФРТ, представленным на рис. 9.

5. ЗАКЛЮЧЕНИЕ

В результате проведенного исследования были продемонстрированы недостатки традиционных методов представления геометрии поверхностей свободной формы в системах компьютерного моделирования, а именно, представления в виде треугольной сетки или аналитического выражения. На основе анализа недостатков был предложен метод двойного представления геометрии свободной формы.

Для метода двойного представления геометрии был разработан алгоритм трассировки лучей в оптических системах, содержащих поверхности свободной формы, включая полиномы высокого порядка или полиномы Якоби. Данный алгоритм был реализован и встроен систему компьютерного моделирования Lumicept и протестирован на ряде сцен для решения задач оптического моделирования.

При тестировании программной реализации алгоритма исследовались скорость процедуры поиска точки встречи луча с поверхностью и объем памяти, необходимый для хранения дополнительной информации о поверхности. Результаты тестирования показали высокую эффективность предложенного подхода при решении таких задач, как расчет функции рассеивания точки и расчет рассеянного света в оптической системе. Кроме того, была подтверждена стабильность вычислений точки встречи луча с поверхность для лучей, чьи траектории сильно отличаются от расчетных, например, при анализе рассеянного света.

6. ИСТОЧНИК ФИНАНСИРОВАНИЯ

Работа была выполнена при финансовой поддержке Российского научного фонда, грант № 22-11-00145.

СПИСОК ЛИТЕРАТУРЫ

1. *Forbes G.W.* Shape specification for axially symmetric optical surfaces // Opt. Express. 2007. № 15. P. 5218–5226.
2. *Forbes G.W.* Robust, efficient computational methods for axially symmetric optical aspheres // Opt. Express. 2010. № 18. P. 19700–19712.
3. *Thompson K.P., Fournier F., Rolland J.P., Forbes G.W.* The Forbes Polynomial: A more predictable surface for fabricators // International Optical Design Conference and Optical Fabrication and Testing. OSA Technical Digest (CD) (Optica Publishing Group, 2010), paper OTuA6.
4. *Feder D.* Optical Calculations with Automatic Computing Machinery. J. Opt. Soc. Am. 1951. № 41. P. 630.
5. *Allen W., Snyder J.* Ray Tracing through Uncentered and Aspheric Surfaces // J. Opt. Soc. Am. 1952. № 42. P. 243.
6. *Spencer M.G.H., Murty V.R.K.* General Ray-Tracing Procedure // J. Opt. Soc. Am. 1962. № 52. P. 672–678.
7. *Pharr M., Jakob W., Humphreys G.* Physically Based Rendering: From Theory to Implementation. M.: Morgan Kaufmann, 2016. 1266 p.
8. *Karhu K.* Displacement Mapping. Tik-111.500 Seminar on computer graphics. Telecommunications Software and Multimedia Laboratory. Spring 2002: Rendering high-quality 3D graphics.
9. GitHub – embree/embree: Embree ray tracing kernels repository. https://github.com/embree/embree
10. Intel® oneAPI Threading Building Blocks. https://www.intel.com/content/www/us/en/developer/tools/oneapi/onetbb.html
11. *Волобой А.Г., Галактионов В.А., Жданов Д.Д.* Технология оптических элементов в компьютерном моделировании оптико-электронных приборов // Информационные технологии в проектировании и производстве. 2006. № 3. С. 46–56.
12. Integra® Lumicep. https://integra.jp/en/products/lumicept

DUAL REPRESENTATION OF GEOMETRY FOR RAY TRACING ACCELERATION IN OPTICAL SYSTEMS WITH FREEFORM SURFACES

© 2024 D. D. Zhdanov[a], I. S. Potemin[a], A. D. Zhdanov[a]

[a]St. Petersburg National Research University of Information Technologies, Mechanics, and Optics, Kronverkskii pr. 49, St. Petersburg, 197101 Russia

This paper explores the possibility of using dual geometry representation to improve the speed of ray tracing and ensure the robustness of light propagation simulations in complex optical systems containing free-form surfaces defined by high-order polynomials (up to order 34) or Jacobi polynomials. An analysis was carried out of traditional methods of representing this geometry both in the form of a triangular mesh and in the form of an analytical expression. The analysis demonstrated the disadvantages of traditional approaches, which consist in the insufficient accuracy of calculating the coordinates of the meeting point of the ray with a triangular mesh, as well as the instability of the results of searching for the hit point of tangent rays with the analytically defined surface when using existing calculation methods. As a result, it was proposed to use a dual representation of the geometry in the form of a rough approximation of the surface by a triangular mesh, which is subsequently used as an initial approximation to find the point where the ray hits the surface specified by the analytical expression. This solution made it possible to significantly speed up the convergence of analytical methods and increase their stability. Moreover, using the Intel® Embree library to quickly find the point of intersection of a ray with a coarse triangular mesh and a vector calculation model to refine the coordinates of the point of intersection of the ray with the geometry represented analytically allowed authors to develop and implement a ray tracing algorithm in an optical system containing surfaces with dual geometry representation. Experiments conducted using the developed and implemented algorithm show the significant acceleration of ray tracing while maintaining computational accuracy and high stability of results. The results were demonstrated by calculating the point and flare spread function for two lenses with free-form surfaces defined by Jacobi polynomials. In addition, for these two lenses, the image formed by an RGB-D object simulating a real scene was calculated.

Keyword: ray tracing, rendering, triangle mesh, free-form surfaces

REFERENCES

1. *Forbes G.W.* Shape specification for axially symmetric optical surfaces // Opt. Express. 2007. № 15. P. 5218–5226.
2. *Forbes G.W.* Robust, efficient computational methods for axially symmetric optical aspheres // Opt. Express 2010. № 18. P. 19700–19712.
3. *Thompson K.P., Fournier F., Rolland J.P., Forbes G.W.* The Forbes Polynomial: A more predictable surface for fabricators // International Optical Design Conference and Optical Fabrication and Testing. OSA Technical Digest (CD) (Optica Publishing Group, 2010), paper OTuA6.
4. *Feder D.* Optical Calculations with Automatic Computing Machinery // J. Opt. Soc. Am. 1951. № 41. P. 630.

5. *Allen W., Snyder J.* Ray Tracing through Uncentered and Aspheric Surfaces // J. Opt. Soc. Am. 1952. № 42. P. 243.
6. *Spencer M.G.H., Murty V.R.K.*. General Ray-Tracing Procedure // J. Opt. Soc. Am. 1962. № 52. P. 672–678.
7. *Pharr M., Jakob W., Humphreys G.* Physically Based Rendering: From Theory to Implementation. M.: Morgan Kaufmann, 2016. 1266 p.
8. *Karhu K.* Displacement Mapping. Tik-111.500 Seminar on computer graphics. Telecommunications Software and Multimedia Laboratory. Spring 2002: Rendering high-quality 3D graphics.
9. GitHub – embree/embree: Embree ray tracing kernels repository. https://github.com/embree/embree
10. Intel® oneAPI Threading Building Blocks. https://www.intel.com/content/www/us/en/developer/tools/oneapi/onetbb.html
11. *Voloboi A.G., Galaktionov V.A. and Zhdanov D.D.* Technology of optical elements in computer modeling of optical-electronic devices // Inf. Tekhnol. Proekt. Proizvod. 2006. № 3. P. 46–56.
12. Integra® Lumicept. https://integra.jp/en/products/lumicept

ПРОГРАММИРОВАНИЕ, 2024, № 3, с. 67–74

АНАЛИЗ ДАННЫХ

УДК 519.6+004.891.3+616-018

НЕЙРОСЕТЕВОЙ МЕТОД ДЕТЕКТИРОВАНИЯ РАЗМЫТИЯ НА ГИСТОЛОГИЧЕСКИХ ИЗОБРАЖЕНИЯХ

© 2024 г. Г. С. Назаренко[a,*], А. С. Крылов[a,]**

[a] *Московский государственный университет имени М.В. Ломоносова*
119991 Москва, ГСП-1, Ленинские горы, д. 1
**E-mail: s02190303@gse.cs.msu.ru*
***E-mail: kryl@cs.msu.ru*

Поступила в редакцию 15.01.2024 г.
После доработки 19.01.2024 г.
Принята к публикации 24.01.2024 г.

В работе рассматривается задача обнаружения размытых областей на полнослайдовых гистологических изображениях высокого разрешения. Предлагаемый метод основан на использовании нейронного оператора Фурье, обучаемого на результатах двух одновременно использованых подходов: обнаружения размытия с помощью многомасштабного анализа коэффициентов дискретного косинусного преобразования и оценки степени резкости границ объектов на изображении. Эффективность алгоритма подтверждена на изображениях из наборов данных PATH-DT-MSU [1] и FocusPath [2].

Ключевые слова: гистология, глубокое обучение, область размытия, нейронный оператор Фурье

DOI: 10.31857/S0132347424030075, **EDN:** QAGUUM

1. ВВЕДЕНИЕ

Современная гистопатологическая диагностика включает в себя извлечение образцов тканей и проведение гистотехнических процедур для подготовки оцифрованного предметного стекла, называемого полнослайдовым изображением (WSI), для дальнейшего исследования. Неправильное обращение с образцом ткани может привести к появлению артефактов (областей, не имеющих диагностического значения, поскольку внешний вид ткани изменен или поврежден) в полученном WSI [26], что может влиять на системы вычислительной патологии и на поставленный диагноз, если их не исключить или не обработать. Недавние работы по WSI с использованием методов глубокого обучения для диагностического прогнозирования в основном основывались на ручном выборе областей интереса (ROI) без артефактов, имеющих диагностическую значимость [27]. Однако ручной выбор трудоемок и требует много времени. И, поскольку артефакты, включенные во время обучения или вывода, могут ухудшить результаты прогнозирования, то необходимо разрабатывать алгоритмы и программы по автоматическому обнаружению артефактов.

Одним из наиболее часто встречающихся является артефакт размытия изображений, возникающий при смещении фокуса (т.е. профиля сканирования) во время сканирования предметных стекол [28, 29]. Два основных подхода оценки уровня размытия областей гистологических изображений основаны на использовании классических математических методов или методов глубокого обучения. Используя подход, основанный на DCNN, в работе [30] были обучены семь современных архитектур на наборе данных FocusPath[1] [25] и было проведено сравнение результатов с классическими методами. DeepFocus [31] использовал аналогичный подход с CNN для анализа размытых областей в WSI с различными красителями. Другие системы оценки качества фокуса, такие как ConvFocus [32] и FocusLiteNN [33], использовали модели на основе CNN для количественной оценки и локализации размытых областей в WSI. В работе [34] предложен метод, основанный на глубоком ядерном обучении, проанализированный, наряду с оригинальными наборами, также на наборах данных FocusPath [25] и TCGA [33].

При анализе результатов на конкретных наборах данных несомненное преимущество имеют методы глубокого обучения. В то же время хорошо известна проблема плохой устойчивости

[1] https://sites.google.com/view/focuspathuoft

результатов, возникающая при использовании нейросетевых моделей на отличающихся наборах данных, например, при использовании данных, полученных другими сканерами. В связи с этим несомненный интерес представляют нейросетевые методы, основанные, в том числе, на использовании классических моделей – “нейросетевые методы, основанные на физике” (PINN) [35]. В настоящее время по близкой к PINN тематике в задаче размытия гистологических изображений нам известна только статья [36]. В данной работе нами рассмотрен метод PINN, основанный на использовании нейросетевого оператора Фурье (FNO) и базирующийся на классических методах оценки размытия [37, 38] для задачи бинарной классификации уровня размытия гистологических изображений. Структура статьи включает в себя введение, раздел описывающий моделируемый классический метод, краткое описание нейросетевого оператора Фурье и параметры оператора, выбранные для моделирования, результаты работы алгоритма и заключение.

2. ИСПОЛЬЗУЕМЫЙ КЛАССИЧЕСКИЙ АЛГОРИТМ

Моделируемый классический метод поиска размытых фрагментов на гистологических изображениях состоит из следующих этапов:

1. Построение карты размытия с помощью применения высокочастотного многомасштабного преобразования [37].

2. Оценка степени резкости изображения на основе анализа границ объектов [38].

3. Объединение полученных результатов и детекция размытых областей.

2.1. Высокочастотное многомасштабное преобразование

Дискретное косинусное преобразование (ДКП) – это преобразование, которое преобразует сигнал (или в данном случае изображение) из пространственного представления в частотное. Широко известно, что размытие приводит к уменьшению амплитуд высокочастотных коэффициентов ДКП изображения. На рис. 1 приведен график отсортированных по возрастанию абсолютных значений амплитуд высокочастотных коэффициентов ДКП для резких и размытых участков гистологических изображений, демонстрирующий заметную разницу в их значениях. Все последующие шаги описываемого подхода основаны на этом различии.

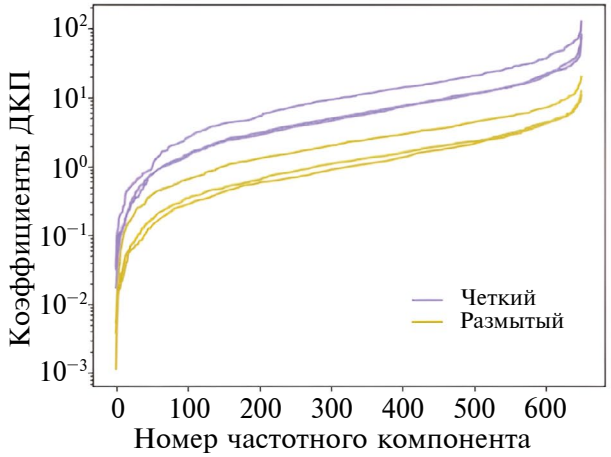
Обозначим через I входное изображение в градациях серого размером $N_1 \times N_2$ пикселов. Через $P_{i,j}^M$ обозначим фрагмент (патч) изображения I размером $2M + 1 \times 2M + 1$ с центром в пикселе (i, j), а через $K_{i,j}^M$ – результат применения к $P_{i,j}^M$ ДКП:

$$K_{i,j}^M(u,v) = \frac{c(u)c(v)}{2M+1}\sum_{s=0}^{2M}\sum_{t=0}^{2M} P_{i,j}^M(s,t)S^M(s,u,t,v);$$

$$S^M(s,u,t,v) = \cos\left(\frac{\pi(2s+1)u}{2(2M+1)}\right)\cos\left(\frac{\pi(2t+1)v}{2(2M+1)}\right); \quad (1)$$

$$c(p) = \begin{cases} 1, p = 0 \\ \sqrt{2}, p \neq 0 \end{cases}, \text{ где } p - \text{переменная.}$$

В предложенном преобразовании вычисленные коэффициенты ДКП для каждого патча делятся на три группы, а именно, на соответ-



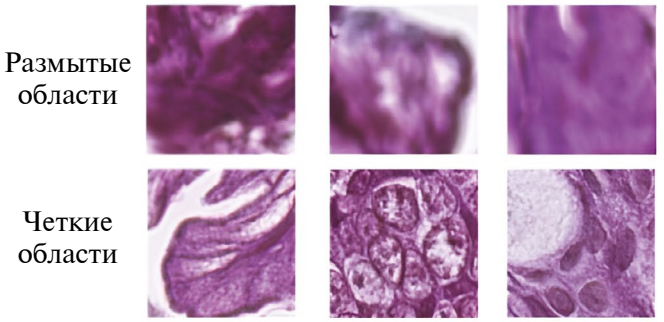


Рис. 1. График отсортированных по возрастанию абсолютных значений коэффициентов ДКП для размытых и четких фрагментов гистологических изображений.

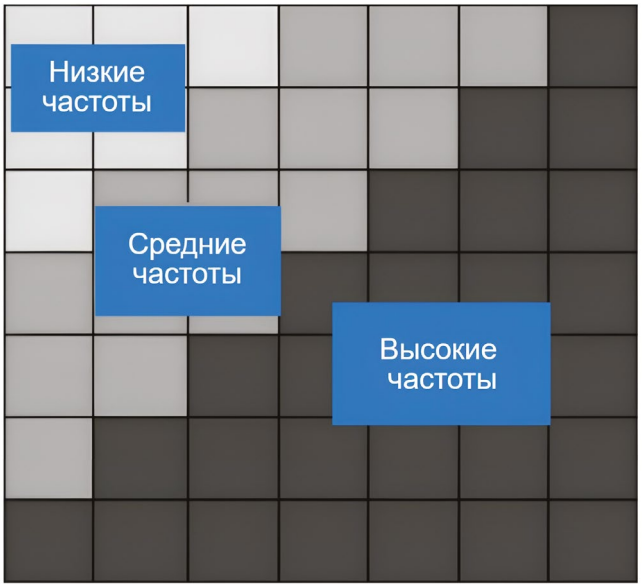


Рис. 2. Иллюстрация коэффициентов ДКП для блока при их разделении на группы соответствия различным частотам.

ствующие низким, средним и высоким частотам [39]. Рис. 2 иллюстрирует эти три группы коэффициентов на примере патча размером 7×7 пикселов.

Обозначим через $H_{i,j}^{M}$ вектор, состоящий из модулей высокочастотных коэффициентов ДКП $H_{i,j}^{M}$ патча:

$$H_{i,j}^{M} = \left\{ \left| K_{i,j}^{M}(u,v) \right| : u + v \geq 2M,\ 0 \leq u,v \leq 2M \right\}. \quad (2)$$

Рассмотрение только одного разрешения (размера патча) может оказаться недостаточным для определения размытия на изображении. Проблема выбора количества разрешений обсуждалась в [40, 41]. В этих работах было высказано предположение о том, что человеческая зрительная система при определении различных объектов на изображении достаточно сложным образом использует информацию с различных разрешений. В связи с этим было предложено решать поставленную задачу сразу для нескольких размеров патчей, а затем агрегировать полученные результаты следующим образом:

$$L_{i,j} = \operatorname{sort}\left(\bigcup_{r=0}^{m} H_{i,j}^{M_r} \right), \quad (3)$$

где $H_{i,j}^{M_r}$ патч размером $2M_r + 1 \times 2M_r$ ($M_r = 2^r$), а $\bigcup_{r=0}^{m} H_{i,j}^{M_r}$ означает объединение модулей высокочастотных коэффициентов ДКП, взятых со всех $m + 1$ разрешений. Таким образом, $L_{i,j}$ является вектором размерности $\sum_{r=0}^{m} \frac{(2M_r+1)^2 + 2M_r + 1}{2}$, но исходя из наблюдений достаточно использовать лишь первые $\sum_{r=0}^{m}(2M_r + 1)$ [37] компонентов для оптимизации времени работы метода.

Пусть $L_{i,j;t}$ – t-й элемент вектора $L_{i,j}$, а L_t – t-й слой, образованный путем взятия $L_{i,j;t}$ элемента во всех векторах $L_{i,j}$ каждого пиксела (i, j) изображения I. Размерность L_t совпадает с размерностью исходного изображения.

Результатом предложенного высокочастотного многомасштабного преобразования являются слои L_t, которые далее min-max нормализуются [37]. По мере увеличения номера слоя, слои начинают состоять из больших по абсолютному значению высокочастотных коэффициентов, тем самым все больше структур и границ объектов исходного изображения появляется на слоях.

2.2. Построение карты размытия

К входному изображению в градациях серого I применяется Гауссов фильтр с параметром $\sigma = 0.5$ для подавления высокочастотного шума. Обозначим отфильтрованное изображение I за I_g. Далее для I_g вычисляется карта величин градиентов G следующим образом:

$$G = \sqrt{\left(I_g h_x\right)^2 + \left(I_g h_y\right)^2}, \quad (4)$$

где

$$h_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \text{ и } h_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

Для карты градиентов G вычисляются слои L_t по формулам (1)–(3). В результате искомая карта размытия изображения рассчитывается как

$$D_{i,j} = \max\left(\left\{ \tilde{L}_t(i,j) : t = 1, \ldots, \sum_{r=1}^{m} M_r \right\} \right); \quad (5)$$

$$0 \leq i < N_1,\ 0 \leq j < N_2.$$

Окончательная карта размытия D сглаживается с помощью фильтра [42]. На рис. 3 приведен пример карты размытия. Чем темнее область на карте размытия, тем сильнее размыта эта область на исходном изображении.

3. ОЦЕНКА РЕЗКОСТИ ИЗОБРАЖЕНИЯ

Границей объекта на изображении (edge) называется область неоднородности или область перехода между двумя различными по интенсивности областями изображения.

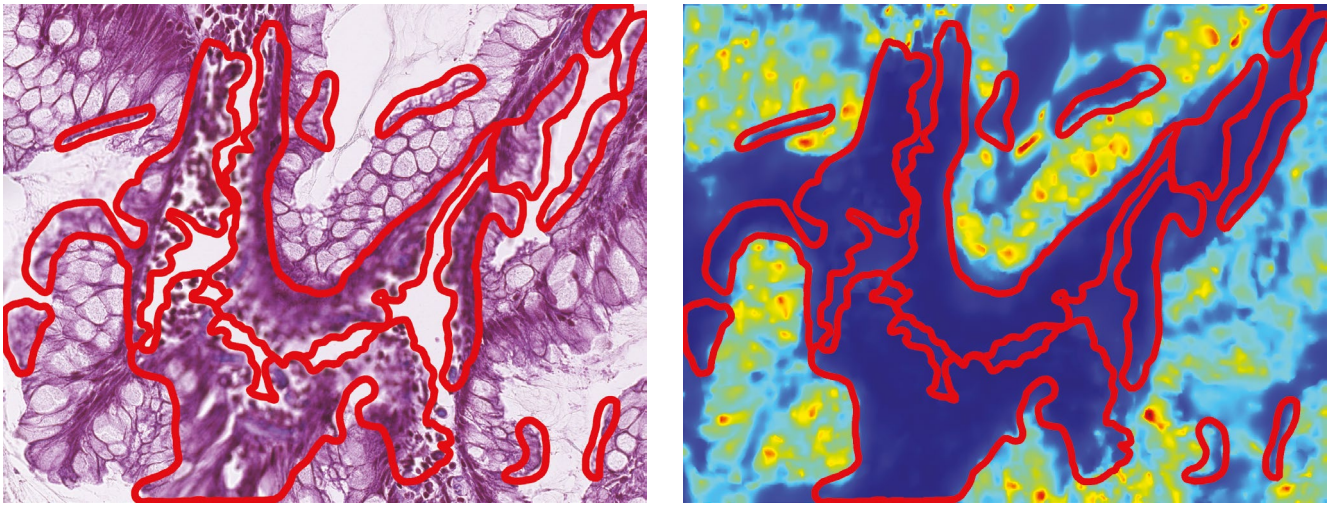


Рис. 3. Визуализация карты размытия для гистологического изображения с отмеченными красным областями размытия.

Оценка резкости изображения S_I вычисляется на основании оценки размытия вдоль границ объектов на изображении. В оригинальной статье [38] рассматривались лишь горизонтальные и вертикальные границы изображения, однако в данной работе было принято решение рассмотреть, в том числе и диагональные границы.

Более подробно об оценке можно прочитать в оригинальной статье [38], а о ее модификации – в работе [44].

4. ОБЩИЙ АЛГОРИТМ КЛАССИЧЕСКОГО МЕТОДА

Дано входное изображение I. Для инвариантности к средней интенсивности изображений это изображение нормируется по параметру *value* цветового пространства HSV способом, описанным в [45]. Далее это изображение переводится в изображение в градациях серого, и для него строится карта размытия способом, описанным в разделе 2.2. Детекция размытия осуществляется с помощью скользящего окна размером 64×64 пиксела с шагом в 32 пиксела как по вертикальной, так и по горизонтальной оси. Тем самым задача детекции размытия переходит в задачу классификации патчей на два класса – резкие и размытые. Патч считается размытым при одновременном выполнении трех условий.

1. Патч принадлежит темной зоне на карте размытия. Если же среднее значение патча (обозначим за V) на карте размытия D превышает 80, то патч считается четким (карта размытия рассматривается в градациях серого).

2. Патч не принадлежит фону. Патчи, средняя интенсивность которых превышает 170, классифицируются как фон.

3. Для всех остальных патчей анализируется пара значений (V, S_I). Патчи из обучающей выборки были сгруппированы по значению V, для каждой из групп путем максимизации метрики F_1 было определено пороговое значение $S_I(V)$, ниже которого патч классифицируется как размытый. Полученный график $S_I(V)$ приведен на рис. 4.

Таким образом, для каждого изображения все его патчи делятся на резкие и размытые.

В пункте 2 был приведен порог интенсивности фоновых патчей равный 170, он был получен следующим образом: на базе набора данных PATH-DT-MSU[2] WSS2 был создан набор фоновых патчей размера 64×64 с предварительной нормализацией параметра *value* в цветовом пространстве HSV [45] и конвертацией патчей в изображения в градациях серого. Для каждого

[2] https://imaging.cs.msu.ru/en/research/histology/path-dt-msu

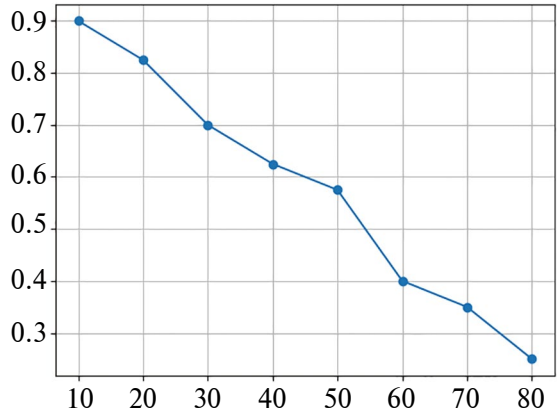


Рис. 4. График пороговых значений $S_I(V)$.

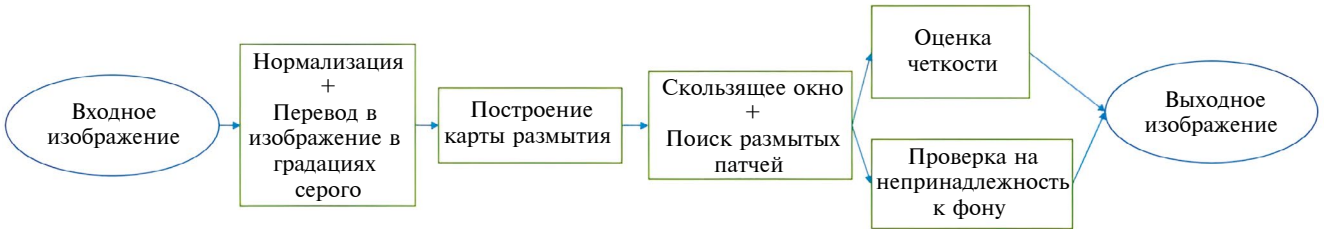


Рис. 5. Общий алгоритм работы классического метода.

такого патча считалась средняя интенсивность, а затем для массива полученных значений был высчитан 0.1 квантиль, оказавшийся равным 170. Тем самым патчи, интенсивность которых была больше 170, исключались из рассмотрения.

Общая схема работы предложенного метода приведена на рис. 5.

5. МЕТОД НА ОСНОВЕ НЕЙРОННОГО ОПЕРАТОРА ФУРЬЕ

Fourier Neural Operator (FNO) – это метод, основанный на данных, для решения задач, описываемых классическими моделями, в том числе уравнениями в частных производных, способный сопоставлять параметры классических моделей с его решением. Его ключевыми особенностями по сравнению с классическими методами решения являются быстродействие и инвариантность к дискретизации сетки. В данной работе рассматривается применение этого метода к задаче аппроксимации классического метода обнаружения размытых регионов на гистологических изображениях.

5.1. FNO

Архитектура сети FNO представлена на рис. 6. Схема преобразований входного изображения, используемая в данной работе:

1. Входное изображение подается в блок P, в котором с помощью линейных локальных операторов число каналов входа приводится к числу каналов внутреннего представления сети *hidden_channels* = 32.

2. Далее выход блока P подается на вход нескольким слоям Фурье. Внутри этих слоев размерность входов и выходов сохраняется.

3. Выход пункта 2 подается в блок Q, в котором также с помощью линейных локальных операторов число каналов приводится к числу каналов выходного представления *out_channels* = 3.

Внутри Фурье слоя:

1. С помощью быстрого преобразования Фурье (F) для входа вычисляются коэффициенты Фурье, а также параллельно к входу применяется локальное линейное преобразование W.

2. Производится линейное преобразование R над низкочастотными коэффициентами Фурье и исключение из рассмотрения части высокочастотных коэффициентов.

3. Обратное преобразование Фурье.

4. Поэлементное суммирование двух преобразованных входов.

5. Применение нелинейной функции активации к выходу из слоя.

В блоке P находится два сверточных слоя с ядрами размера 1×1. В Фурье-слоях в качестве линейного преобразования над низкочастотными коэффициентами Фурье R используется преобразование из [46], а в качестве линейного локального преобразования W – свертка с ядром

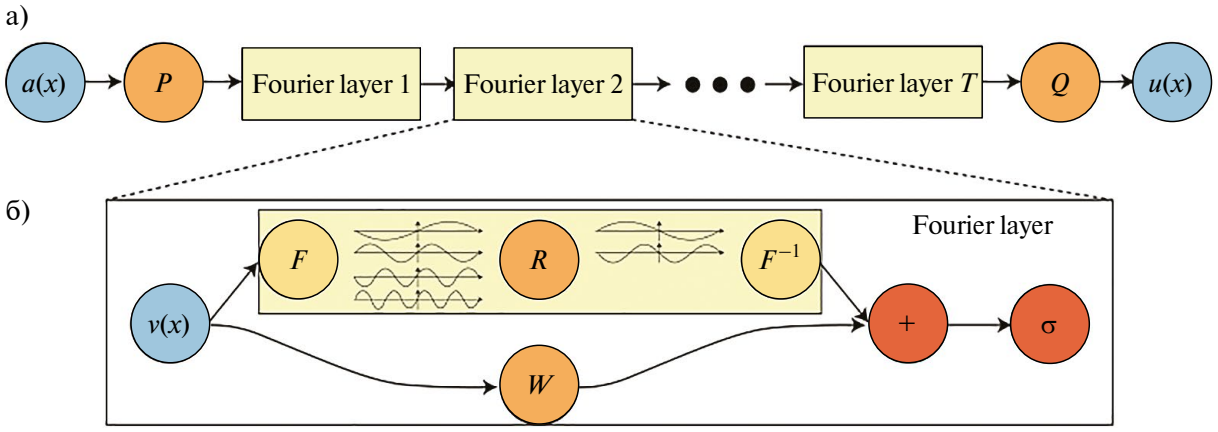


Рис. 6. Архитектура нейронного оператора Фурье.

размера 1×1. Нелинейной функцией активации на выходе из Фурье-слоя является сигмоида. Всего в архитектуре данной работы восемь Фурье-слоев. В блоке Q также находятся два сверточных слоя с ядрами размерами 1×1 .

5.2. Общий алгоритм и данные нейросетевого метода

Для обучения нейронной сети было использовано $\approx 4.300.000$ патчей размерами 64×64. Из них $\approx 4.000.000$ резких и ≈ 300.000 размытых.

Среди четких патчей:

- $\approx 3.600.000$ получены из набора данных PATH-DT-MSU WSS2 классическим методом из раздела 4;
- ≈ 400.000 получены из тренировочной выборки набора данных FocusPath. Патчи были выбраны из изображений с уровнем размытия $[-2, 2]$.

Среди размытых патчей:

- ≈ 10.000 получены из набора данных PATH-DT-MSU WSS2 классическим методом из раздела 4;
- ≈ 180.000 получены путем применения синтетического размытия к четким патчам из набора данных PATH-DT-MSU WSS2;
- ≈ 110.000 получены из тренировочной выборки набора данных FocusPath. Патчи были выбраны из изображений с уровнем размытия либо ≥ 3, либо ≤ 3.

В качестве классификатора используется нейронная сеть на базе архитектуры FNO с добавлением к ней classification head, включающую в себя average pooling слой и два полносвязных слоя с функцией активации GeLU между ними.

6. ТЕСТИРОВАНИЕ НЕЙРОСЕТЕВОГО МЕТОДА

Тестирование метода проводилось на двух тестовых выборках из наборов данных PATH-DT-MSU WSS1 и FocusPath. Выборка из WSS1 содержит 100 изображений размерами $\approx 1500 \times 1500$, к которым было применено синтетическое Гауссовское размытие. Выборка из FocusPath содержит 192 размеченных изображения размерами 1024×1024.

В табл. 1 и 2 продемонстрировано сравнение классического и предложенного нейросетевого метода на этих двух тестовых выборках. Для определения порогов классического метода использовался набор данных PATH-DT-MSU WSS2, изображения и размытия которого похожи на присутствующие в наборе WSS1. Соответственно

Таблица 1. Значения метрик методов на наборе данных WSS1

Метод	Precision	Recall	Accuracy	F1
Нейросетевой	0.811	**0.909**	0.965	0.857
Классический	**0.899**	0.897	**0.979**	**0.897**

Таблица 2. Значения метрик методов на наборе данных FocusPath

Метод	Precision	Recall	Accuracy	F1
Нейросетевой	**0.863**	**0.884**	**0.876**	**0.873**
Классический	0.280	0.018	0.297	0.033

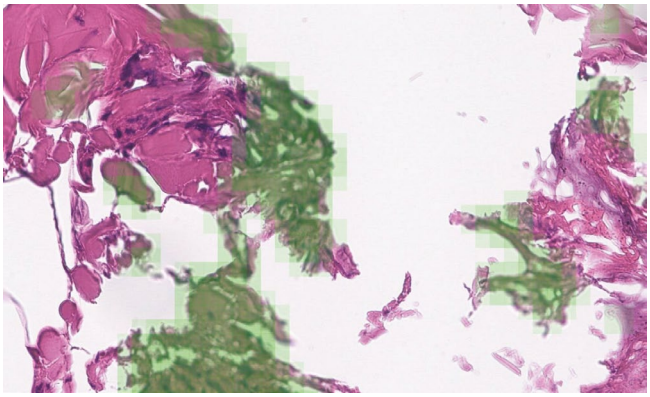


Рис. 7. Визуализация детекций размытия.

нейросетевой метод, обученный, в том числе, на наборе данных, собранных классическим методом из WSS2, показывает на WSS1 немного более низкие результаты по сравнению с классическим методом. Однако классический метод [37] не способен строить карты размытия для целиком размытых изображений, отмечая все области карты размытия для таких изображений как четкие. Набор данных FocusPath состоит из целиком размытых и целиком четких изображений. В то же время нейросетевой метод лишен такого недостатка и демонстрирует удовлетворительные результаты на тестовой выборке из FocusPath. На рис. 7 зеленым отмечены результаты детекции реального размытия на изображении из набора данных WSS1.

7. ЗАКЛЮЧЕНИЕ

Предложен нейросетевой метод обнаружения размытых фрагментов на гистологических изображениях. Алгоритм FNO был обучен на результатах детектирования размытых областей классическим алгоритмом и дополнительных данных. Он продемонстрировал близкие по качеству работы результаты на изображениях набора PATH-DT-MSU, на части которого обучался

классический алгоритм. Однако, метод FNO позволил достаточно хорошо детектировать размытие и для набора FocusPath.

8. БЛАГОДАРНОСТИ

Исследования выполнены при финансовой поддержке Российского научного фонда в рамках научного проекта № 22-41-02002.

СПИСОК ЛИТЕРАТУРЫ

1. *Khvostikov, A., Krylov A., Mikhailov I., Malkov P.* Visualization and Analysis of Whole Slide Histological Images // Lecture Notes in Computer Science. 2023. V. 13644. P. 403–413.
2. *Hosseini M., Zhang Y., Plataniotis K.* Encoding visual sensitivity by maxpol convolution filters for image sharpness assessment // IEEE Transactions on Image Processing. 2019. V. 28. P. 4510–4525.
3. *Taqi S.A., Sami S.A., Sami L.B., Zaki S.A.* A review of artifacts in histopathology // Journal of oral and maxillofacial pathology: JOMFP. 2018. V. 22. P. 279–287.
4. *Priego-Torres B.M., Sanchez-Morillo D., Fernandez-Granero M.A., Garcia-Rojo M.* Automatic segmentation of whole-slide H&E stained breast histopathology images using a deep convolutional neural network architecture // Expert Systems With Applications. 2020. V. 151. P. 113387.
5. *Kanwal N., Perez-Bueno F., Schmidt A., Engan K., Molina R.* The devil is in the details: Whole slide image acquisition and processing for artifacts detection, color variation, and data augmentation: A review // IEEE Access. 2022. V. 10. P. 58821.
6. *Janowczyk A. et al.* HistoQC: an open-source quality control tool for digital pathology slides // JCO clinic. cancer informatics. 2019. V. 3. P. 1–7.
7. *Albuquerque T., Moreira A., Cardoso J.* Deep ordinal focus assessment for whole slide images // Proceedings of the IEEE/CVF ICCV. 2021. P. 657–663.
8. *Senaras C., Niazi M., Lozanski G., Gurcan M.* DeepFocus: detection of out-of-focus regions in whole slide digital images using deep learning // PloS one. 2018. V. 13. P. e0205387.
9. *Kohlberger T., Liu Y., Moran M. et al.* Whole-slide image focus quality: Automatic assessment and impact on ai cancer detection // Journal of pathology informatics. 2019. V. 10. P. 39.
10. *Wang Z., Hosseini M., Miles A., Plataniotis K., Wang Z.* Focuslitenn: High efficiency focus quality assessment for digital pathology. MICCAI. Springer International Publishing, 2020. P. 403–413.
11. *Kanwal N. et al.* Are you sure it's an artifact? Artifact detection and uncertainty quantification in histological images // Computerized Medical Imaging and Graphics. 2024. V. 112. P. 102321.
12. *Faroughi S., Pawar N., Fernandes C. et al.* Physics-guided, physics-informed, and physics-encoded neural networks in scientific computing // arXiv preprint. arXiv:2211.07377, 2022.
13. *Li Q., Liu X., Han K., Guo C., Jiang J., Ji X., Wu X.* Learning to autofocus in whole slide imaging via physics-guided deep cascade networks // Optics Express. 2022. V. 30. P. 14319–14340.
14. *Alireza Golestaneh S., Karam L.* Spatially-varying blur detection based on multiscale fused and sorted transform coefficients of gradient magnitudes // Proceedings of ICPR. 2017. P. 5800–5809.
15. *Kumar J., Chen F., Doermann D.* Sharpness estimation for document and scene images // Proceedings of ICPR. 2012. P. 3292–3295.
16. *Langelaar G.C., Setyawan I., Lagendijk R.L.* Watermarking digital image and video data. A state- of-the-art overview // IEEE Signal processing magazine. 2000. V. 17. No. 5. P. 20–46.
17. *Shi J., Xu L., Jia J.* Discriminative blur detection features // Proceedings of ICPR. 2014. P. 2965–2972.
18. *Yan Q., Xu L., Shi J., Jia J.* Hierarchical saliency detection // Proceedings of ICPR. 2013. P. 1155–1162.
19. *Gastal E., Oliveira M.* Domain transform for edge-aware image and video processing // ACM SIGGRAPH. 2011. P. 1–12.
20. *Ferzli R., Karam L.J.* A no-reference objective image sharpness metric based on the notion of just noticeable blur (JNB) // IEEE transactions on image processing. 2009. V. 18. No. 4. P. 717–728.
21. *Назаренко Г., Насонов А., Крылов А.* Метод поиска областей размытия на гистологических изображениях. Proceedings of the 33nd International Conference on Computer Graphics and Vision. М.: ИПМ им. М.В. Келдыша РАН, 2023. С. 598–608. https://doi.org/10.20948/graphicon-2023-620-632
22. *Yuki Mochizuki* Normalize image brightness. https://cvtech.cc/std/
23. *Li Z., Kovachki N., Azizzadenesheli K., Liu B., Bhattacharya K., Stuart A., Anandkumar A.* Fourier neural operator for parametric partial differential equations // arXiv preprint. arXiv:2010.08895, 2020.

NEURAL NETWORK METHOD FOR DETECTING BLUR IN HISTOLOGICAL IMAGES

© 2024 G. S. Nazarenko[a], A. S. Krylov[a],

[a] *Lomonosov Moscow State University, GSP-1, Moscow, 119991 Russia*

In this paper we consider the problem of detecting blurred regions in high-resolution full-slide histologic images. The proposed method is based on the use of a Fourier neural operator trained on the results of two simultaneously used approaches: blur detection using multiscale analysis of the discrete cosine transform coefficients and estimation of the degree of sharpness of objects edges in the image. The efficiency of the algorithm is confirmed on images from the datasets PATH-DT-MSU [1] and FocusPath [2].

Keywords: histology, deep learning, blur region, Fourier neural operator

REFERENCES

1. *Khvostikov A., Krylov A., Mikhailov I., Malkov P.* Visualization and Analysis of Whole Slide Histological Images // Lecture Notes in Computer Science. 2023. V. 13644. P. 403–413.
2. *Hosseini M., Zhang Y., Plataniotis K.* Encoding visual sensitivity by maxpol convolution filters for image sharpness assessment // IEEE Transactions on Image Processing. 2019. V. 28. P. 4510–4525.
3. *Taqi S.A., Sami S.A., Sami L.B., Zaki S.A.* A review of artifacts in histopathology // Journal of oral and maxillofacial pathology: JOMFP. 2018. V. 22. P. 279–287.
4. *Priego-Torres B.M., Sanchez-Morillo D., Fernandez-Granero M.A., Garcia-Rojo M.* Automatic segmentation of whole-slide H&E stained breast histopathology images using a deep convolutional neural network architecture // Expert Systems With Applications. 2020. V. 151. P. 113387.
5. *Kanwal N., Perez-Bueno F., Schmidt A., Engan K., Molina R.* The devil is in the details: Whole slide image acquisition and processing for artifacts detection, color variation, and data augmentation: A review // IEEE Access. 2022. V. 10. P. 58821.
6. *Janowczyk A. et al.* HistoQC: an open-source quality control tool for digital pathology slides // JCO clinic. cancer informatics. 2019. V. 3. P. 1–7.
7. *Albuquerque T., Moreira A., Cardoso J.* Deep ordinal focus assessment for whole slide images // Proceedings of the IEEE/CVF ICCV. 2021. P. 657–663.
8. *Senaras C., Niazi M., Lozanski G., Gurcan M.* DeepFocus: detection of out-of-focus regions in whole slide digital images using deep learning // PloS one. 2018. V. 13. P. e0205387.
9. *Kohlberger T., Liu Y., Moran M. et al.* Whole-slide image focus quality: Automatic assessment and impact on ai cancer detection // Journal of pathology informatics. 2019. V. 10. P. 39.
10. *Wang Z., Hosseini M., Miles A., Plataniotis K., Wang Z.* Focuslitenn: High efficiency focus quality assessment for digital pathology. MICCAI. Springer International Publishing, 2020. P. 403–413.
11. *Kanwal N. et al.* Are you sure it's an artifact? Artifact detection and uncertainty quantification in histological images // Computerized Medical Imaging and Graphics. 2024. V. 112. P. 102321.
12. *Faroughi S., Pawar N., Fernandes C. et al.* Physics-guided, physics-informed, and physics-encoded neural networks in scientific computing // arXiv preprint. arXiv:2211.07377, 2022.
13. *Li Q., Liu X., Han K., Guo C., Jiang J., Ji X., Wu X.* Learning to autofocus in whole slide imaging via physics-guided deep cascade networks // Optics Express. 2022. V. 30. P. 14319–14340.
14. *Alireza Golestaneh S., Karam L.* Spatially-varying blur detection based on multiscale fused and sorted transform coefficients of gradient magnitudes // Proceedings of ICPR. 2017. P. 5800–5809.
15. *Kumar J., Chen F., Doermann D.* Sharpness estimation for document and scene images // Proceedings of ICPR. 2012. P. 3292–3295.
16. *Langelaar G.C., Setyawan I., Lagendijk R.L.* Watermarking digital image and video data. A state of-the-art overview // IEEE Signal processing magazine. 2000. V. 17. No. 5. P. 20–46.
17. *Shi J., Xu L., Jia J.* Discriminative blur detection features // Proceedings of ICPR. 2014. P. 2965–2972.
18. *Yan Q., Xu L., Shi J., Jia J.* Hierarchical saliency detection // Proceedings of ICPR. 2013. P. 1155–1162.
19. *Gastal E., Oliveira M.* Domain transform for edge-aware image and video processing // ACM SIGGRAPH. 2011. P. 1–12.
20. *Ferzli R., Karam L.J.* A no-reference objective image sharpness metric based on the notion of just noticeable blur (JNB) // IEEE transactions on image processing. 2009. V. 18. No. 4. P. 717–728.
21. *Nazarenko G., Nasonov A., Krylov A.* A procedure for finding blur areas in histological images, Proc. // 33nd Int. Conf. on Computer Graphics and Vision. M.: Keldysh Institute of Applied Mathematics, RAS, 2023.
22. *Yuki Mochizuki* Normalize image brightness. https://cvtech.cc/std/
23. *Li Z., Kovachki N., Azizzadenesheli K., Liu B., Bhattacharya K., Stuart A., Anandkumar A.* Fourier neural operator for parametric partial differential equations // arXiv preprint. arXiv:2010.08895, 2020.

ПРОГРАММИРОВАНИЕ, 2024, № 3, с. 75–82

АНАЛИЗ ДАННЫХ

УДК 519.6+004.891.3+616-018

СОВМЕСТНОЕ ПОВЫШЕНИЕ РАЗРЕШЕНИЯ И КЛАССИФИКАЦИЯ ФРАГМЕНТОВ ТКАНИ ДЛЯ ПОЛНОСЛАЙДОВЫХ ГИСТОЛОГИЧЕСКИХ ИЗОБРАЖЕНИЙ

© 2024 г. Чж. Сун[a],*, А. В. Хвостиков[a],, А. С. Крылов[a],***, А. Сетхи[b],****, И. А. Михайлов[a],*****, П. Г. Мальков[a],********

[a] *Московский государственный университет имени М.В. Ломоносова*
119991 Россия, Москва, ГСП-1, Ленинские горы, д. 1
[b] *Индийский технологический институт Бомбея*
400076 Индия, Мумбаи, Повай, район IIT
**E-mail: sbitb0009@gse.cs.msu.ru*
***E-mail: khvostikov@cs.msu.ru*
****E-mail: kryl@cs.msu.ru*
*****E-mail: asethi@ee.iitb.ac.in*
******E-mail: imihailov@mc.msu.ru*
*******E-mail: pmalkov@mc.msu.ru*

Поступила в редакцию 15.01.2024 г.
После доработки 19.01.2024 г.
Принята к публикации 23.01.2024 г.

Сегментация полнослайдовых гистологических изображений через классификацию типов ткани небольших фрагментов – чрезвычайно актуальная задача в цифровой патологии, необходимая для разработки методов автоматического анализа полнослайдовых гистологических изображений. Крайне большой размер подобных изображений также делает актуальной задачу повышения разрешения изображений, что позволяет хранить изображения в пониженном разрешении и повышать его при необходимости. Разметка таких изображений специалистами-гистологами сложна и трудоемка, поэтому важно максимально эффективно использовать имеющиеся данные, как размеченные, так и неразмеченные. В данной работе предлагается новый нейросетевой метод одновременного решения задач повышения разрешения гистологических изображений с оптического увеличения ×20 до ×40 и классификации фрагментов изображений по типам ткани на ×20 увеличении. Использование единого кодировщика, а также предложенной схемы обучения вспомогательной и основной нейросетевых моделей позволяет достичь лучших результатов на обоих задачах по сравнению с имеющимися подходами. Для обучения и тестирования метода использовался впервые представленный в данной статье набор данных PATH-DT-MSU WSS2v2. На тестовой выборке было достигнуто значение точности 0.971 и сбалансированной точности 0.916 в задаче классификации на 5 типов ткани, для задачи повышения разрешения были достигнуты значения $PSNR = 32.26$ и $SSTM = 0.89$. Исходный код предложенного метода доступен по ссылке: https:/github.com/Kukty/WSI_SR_CL.

Ключевые слова: гистология, глубокое обучение, сверточные нейронные сети, повышение разрешения, определение типов ткани, полнослайдовые изображения

DOI: 10.31857/S0132347424030086, **EDN:** QAEJQD

1. ВВЕДЕНИЕ

Пример полнослайдового гистологического изображения из набора PATH-DT-MSU WSS2v2 с выполненной полигональной разметкой на 5 классов ткани, включая фон.

Современная цифровая патология позволяет получать большие объемы гистологических изображений в очень высоком разрешении при оптическом увеличении ×40. Такие изображения называются полнослайдовыми и позволяют полностью визуализировать исследуемыей препарат. Пример полнослайдового изображения с разметкой приведен на рис. 1. На эффективную обработку таких объемных и сложных по структуре изображений на данный момент способны только методы глубокого обучения [1, 2]. Подобные методы могут упростить и автоматизировать многие процессы, важные для специалистов-гистологов,

например, позволяют проводить грубую сегментацию образца с разделением на несколько типов ткани путем классификации отдельных небольших фрагментов (патчей) [3, 4]. Также большой объем полнослайдовых гистологических изображений, полученных с увеличением ×40, делает актуальной и задачу повышения разрешения (например, с ×20 до ×40), позволяя хранить в 4 раза меньше информации и повышать разрешение изображений при необходимости [5, 6].

Большинство существующих методов автоматической обработки и анализа гистологических изображений ориентированы только на решение одной фиксированной задачи цифровой патологии, например, повышения разрешения [7] или классификации фрагментов ткани [4]. Однако гораздо более эффективным представляется построение моделей, решающих сразу несколько задач, максимально эффективно использующих имеющиеся данные. Данное направление только развивается и пока не получило широкого освещения. Однако существуют подходы, объединяющие в себе несколько моделей. Так, в работе [8] предложена нейросетевая модель, выполняющая повышение разрешения фрагментов изображений с последующей классификацией результатов. При этом подход является избыточным, так как информация, используемая нейросетью для классификации, сгенерирована этой же нейросетью по изображению более низкого разрешения. Данная работа вдохновлена подходами [9] и [10], которые показывают хорошие результаты на задачах повышения разрешения и сегментации.

Основной вклад данной работы заключается в следующем:

• предложен нейросетевой метод одновременного решения задачи повышения разрешения (с увеличения ×20 на ×40) и классификации фрагментов (на ×20 увеличении) полнослайдовых гистологических изображений, демонстрирующий лучший результат на используемом наборе изображений по сравнению с существующими методами;

• предложена двухэтапная схема обучения, включающая в себя обучение вспомогательной и основной нейросетевых моделей;

• представлена обновленная версия полигональной разметки коллекции полнослайдовых изображений PATH-DT-MSU WSS2v2;

• программно реализованы (в виде библиотеки) pytorch-совместимые загрузчики данных, позволяющие извлекать патчи из полигональной разметки с соблюдением баланса классов;

• предложенный метод одновременного решения двух задач из области цифровой патологии более эффективен с точки зрения обучения и позволяет провести обучение быстрее по сравнению с обучением двух отдельных моделей.

2. ПОСТАНОВКА ЗАДАЧИ

Необходимо построить и обучить нейросетевую модель, работающую с фрагментами полнослайдовых гистологических изображений, имеющую один вход (патч $224 \times 224 \times 3$ на увеличении ×20) и два выхода (изображение $448 \times 448 \times 3$ на увеличении ×40 и вектор вероятностей длины N, соответствующий N типам ткани). Первый выход соответствует увеличенному в 2 раза входному патчу, второй выход – предсказанию типа ткани входного патча. Таким образом, эта нейросетевая модель должна одновременно решать задачи повышения разрешения и классификации фрагментов полнослайдовых изображений.

3. ИСПОЛЬЗУЕМЫЙ НАБОР ДАННЫХ

В данной работе используется набор полнослайдовых гистологических изображений PATH-DT-MSU WSS2v2[1] с полигональной разметкой участков по типу ткани. Данный набор изображений был собран и подготовлен лабораторией математических методов обработки изображений факультета вычислительной математики и кибернетики МГУ имени М.В. Ломоносова и кафедрой патологии Медицинского научно-образовательного центра МГУ имени М.В. Ломоносова.

Набор данных PATH-DT-MSU постоянно обновляется и расширяется. В текущей работе впервые представлен поднабор WSS2v2, который предоставляет более подробную и уточненную разметку поднабора WSS1v1, представленного ранее [11]. Используемый набор данных состоит из 10 полнослайдовых изображений препаратов при раке желудка, полученных с оптическим увеличением ×40, разрешение изображений составляет около 110000×90000 пикселов. Полигональная разметка имеет 5 классов, включающих в себя классы 4 типов ткани (AT, LP, MM, TUM) и фона (BG). Площадь размеченных областей обучающей выборки составляет 1560, 7098, 533, 895, 1303 млн пикселов для классов AT, BG, LP, MM, TUM соответственно на увеличении ×40. Площадь размеченных областей тестовой выборки составляет 1086, 8032, 318, 743, 1199 млн пикселов для классов AT, BG, LP, MM, TUM соответственно на увеличении ×40. Пример полнослайдового изображения с визуализацией разметки из набора PATH-DT-MSU WSS2v2 приведен на рис. 1.

[1] https://imaging.cs.msu.ru/en/research/histology/path-dt-msu

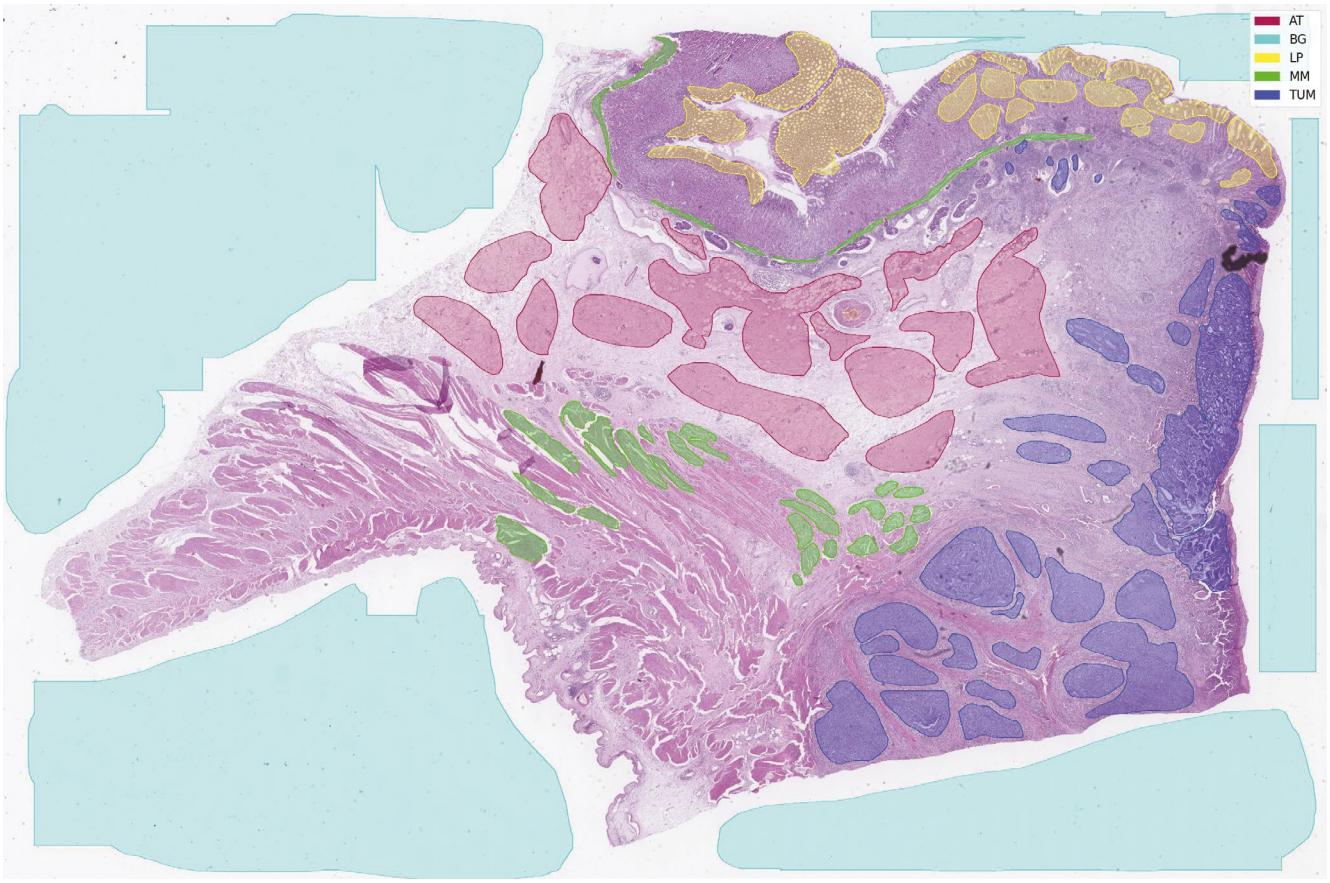


Рис. 1. Пример полнослайдового гистологического изображения из набора PATH-DT-MSU WSS2v2 с выполненной полигональной разметкой на 5 классов ткани, включая фон.

4. ПРЕДЛАГАЕМЫЙ МЕТОД

Предлагаемый в данной работе метод, одновременно решающий задачи повышения разрешения и классификации фрагментов полнослайдовых гистологических изображений, включает в себя этапы подготовки данных, построения и обучения вспомогательной и основной нейросетевой модели. Данные этапы описаны ниже.

Предлагаемый подход одновременного решения двух задач можно рассматривать как метод с частичным привлечением учителя (англ. semi-supervised learning), так как вспомогательная нейросетевая модель обучается на большом количестве неразмеченных данных для задачи повышения разрешения, затем из нее формируется основная модель, которая обучается на относительно небольшом количестве размеченных данных для задачи классификации.

Подготовка данных

Особенностью используемого в работе набора данных PATH-DT-MSU WSS2v1 является полигональная разметка частей изображения, соответствующих разным типам ткани. Такая разметка хорошо интерпретируема с медицинской точки зрения, удобна для самого процесса разметки, но не подходит для обучения сверточный нейронных сетей, которые обычно работают с небольшими прямоугольными или квадратными фрагментами (патчами). Для решения этой проблемы написана библиотека psimage[2] для извлечения патчей из полнослайдовых изображений. Извлечение может проходить как полностью случайно (для задачи повышения разрешения разметка не требуется), так и в соответствии с размеченными полигонами. Во втором случае патчи могут извлекаться как по равномерной сетке (на этапе тестирования), так и случайно (на этапе обучения).

При обучении модели в задаче повышения разрешения патчи из полнослайдового изображения извлекаются случайно, без привязки к разметке, поэтому в обучающую выборку могут попадать фоновые однородные патчи, обучение на которых может привести к ухудшению качества работы. Для предотвращения этого недостатка

[2] https://github.com/xubiker/psimage

в данной работе используется энтропия Шеннона [12] для фильтрации фоновых патчей:

$$H(X) = -\sum_{i=1}^{n} p_i \log_2 p_i, \qquad (1)$$

где p_i – яркость i-го пиксела патча. Значения с низкой энтропией (< 0.2) не попадают в обучающую выборку при решении задачи повышения разрешения.

Архитектура нейронных сетей

В данной работе используются две нейронных сети: вспомогательная (рис. 2) и основная (рис. 3). Обучение происходит последовательно.

Вспомогательная модель представляет собой генеративно-состязательную сеть [13] с UNet-подобным дискриминатором [14]. Данная модель обучается для задачи повышения разрешения на фрагментах изображений без разметки.

Основная модель имеет один вход и два выхода (повышение разрешения, классификация) и обучается и для задачи классификации, и для задачи повышения разрешения, но уже на размеченных фрагментах.

Обе эти модели имеют общий кодировщик, играющий роль экстрактора признаков. Данный кодировщик состоит из двух сверточных слоев и 15 RRDB блоков [15], каждый из которых состоит из трех RDB (Residual-Dense-Block) [16] блоков. Каждый RDB блок состоит из пяти сверточных слоев с активационной функцией Leaky-RELU.

Ветка увеличения размерности основной модели состоит из декодировщика, включающего в себя 4 сверточных слоя. Данный декодировщик совпадает с декодировщиком, используемым в вспомогательной модели, и реализует операцию пиксельного перемешивания (англ. pix shuffle) [17].

Классифицирующая ветка основной модели состоит из двух остаточных блоков [18] и полносвязного слоя.

Обучение вспомогательной модели

Полная функция потерь на первом этапе может быть выражена как взвешенная сумма трех функций потерь:

$$L = \lambda_1 L_{L1} + \lambda_2 L_{per} + \lambda_3 L_{gan}, \qquad (2)$$

где L_{L1} – попиксельная функция потерь; L_{per} – функция потерь восприятия [19]; L_{gan} – генеративно-состязательная функция потерь [20], позволяющая сохранять большее число деталей на изображении. В данной работе значения весов были выбраны как $\lambda_1 = 0.2$, $\lambda_2 = 1.0$ и $\lambda_3 = 1.0$.

Обучение основной модели

Кодировщик вспомогательной сети, обученный на предыдущем этапе, используется в качестве экстрактора признаков и в основной предлагаемой нейросетевой модели (рис. 3). На этом этапе обучения происходит тонкая настройка данного экстрактора признаков и его адаптация для решения задачи классификации, при этом

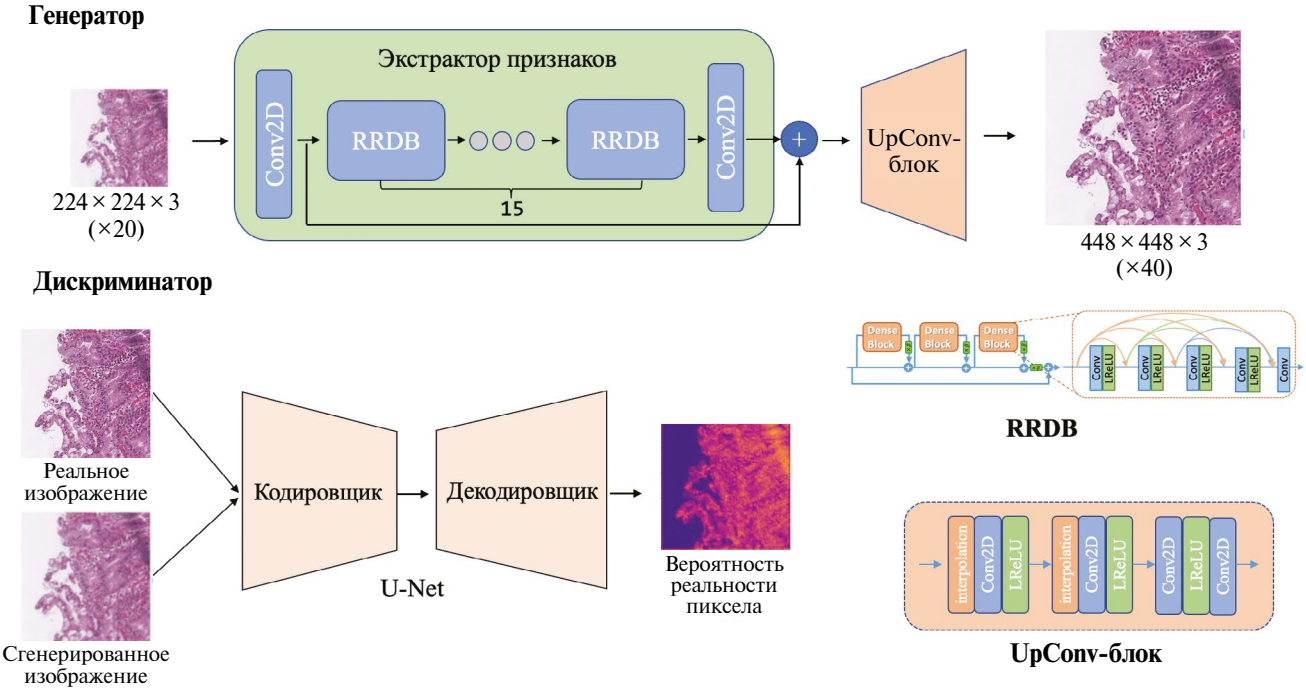


Рис. 2. Архитектура вспомогательной нейронной сети для повышения разрешения.

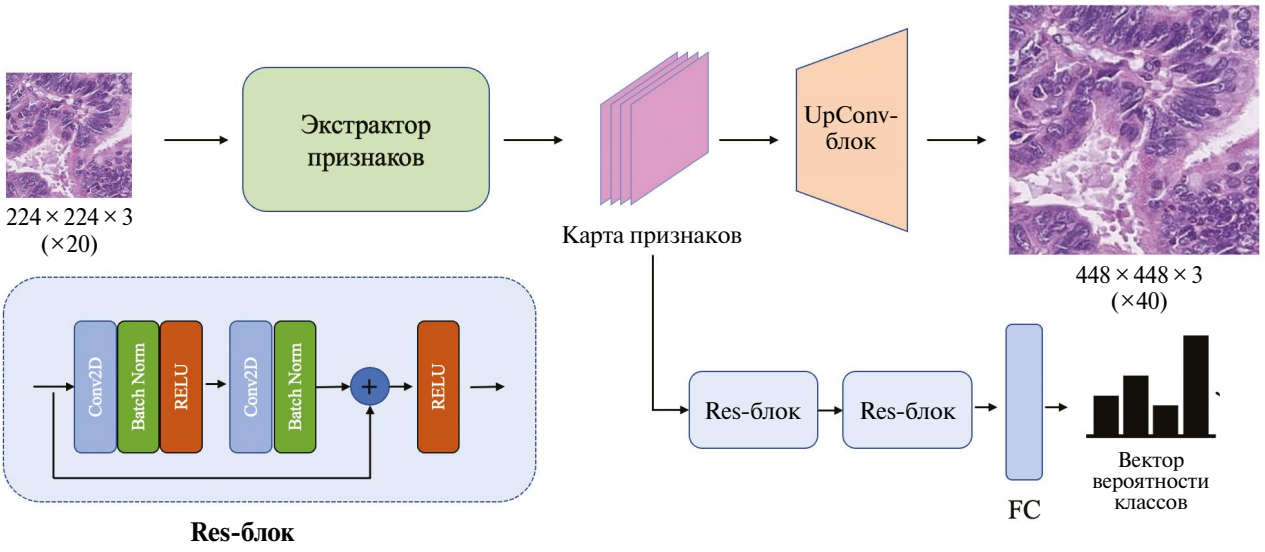


Рис. 3. Архитектура основной нейронной сети для повышения разрешения и классификации.

сохраняя его эффективность для решения задачи повышения разрешения. В качестве функции потерь используется сумма функции $L1$, используемой для выхода с повышением разрешения, и кросс-энтропии, используемой для выхода с предсказанием вектора вероятности классов:

$$L = L_{L1} + L_{CE}. \quad (3)$$

Стоит отметить, что в работе используется дополнительная регуляризация со сглаживанием меток (англ. label smoothing) [21], повышающая устойчивость модели в задаче классификации.

Детали реализации

Во всех экспериментах нейросетевые модели обучались на обучающей выборке набора изображений PATH-DT-MSU WSS2v2 и тестировались на соответствующей тестовой выборке набора. В качестве аугментаций использовались вертикальные и горизонтальные отражения, Гауссово размытие, поворот. Более сложные аугментации, такие как изменения цвета, яркости, тона и т. д., не использовались, так как они негативно влияют на результаты повышения разрешения [22]. Размер пакета был выбран равным 32. Обучение проводилось с помощью оптимизатора ADAM с разогревом. Начальная скорость обучения выбиралась 1×10^{-3} и убывала по косинусному правилу. При обучении основной модели в полносвязном слое классификатора использовался Dropout с отключением 25% нейронов. Каждая из моделей (вспомогательная и основная) обучались по 50 эпох.

Предлагаемый в работе метод программно реализован на языке Python 3. При этом использовались фреймворк pytorch и разработанная авторами библиотека psimage[3]. Также были реализованы загрузчики данных, совместимые с pytorch, позволяющие извлекать патчи из полигональных разметок, проводя при этом балансировку данных по классам.

5. РЕЗУЛЬТАТЫ

В табл. 1 приведены результаты предложенного метода для задачи классификации фрагментов гистологических изображений на 5 классов по типам ткани. Модели, с которыми проводилось сравнение, были предобучены на наборе изображений ImageNet, после чего дообучены на патчах, полученных из набора PATH-DT-MSU WSS2v2. Видно, что признаки, полученные общим кодировщиком во время обучения вспомогательной модели, информативны и для задачи классификации и позволяют в итоге получить лучшие результаты как по общей, так и по сбалансированной точности.

В табл. 2 приведены результаты предложенного метода для задачи повышения разрешения фрагментов гистологических изображений для метрик отношения пикового сигнала к шуму (PSNR) и индекса структурного сходства (SSIM)[26]. Видно, что дообучение кодировщика в основной модели позволяет улучшить эффективность в задаче повышения разрешения за счет использования дополнительной информации о классах, которой не было у вспомогательной модели.

Матрица ошибок для тестовой выборки PATH-DT-MSU WSS2v2 представлена в табл. 3.

[3] https://github.com/xubiker/psimage

Таблица 1. Результаты различных нейросетевых моделей, предварительно обученных на ImageNet, для задачи классификации патчей на 5 классов для тестовой выборки набора изображений PATH-DT-MSU WSS2v2 при ×20 увеличении

Точность / Метод	AT	BG	LP	MM	TUM	Общая	Сбалансированная
ResNet50 [18]	0.947	1.000	0.871	0.929	0.810	0.969	0.911
DenseNet121 [23]	0.942	1.000	0.833	0.941	0.810	0.968	0.905
Efficient Net [24]	0.930	1.000	0.913	0.942	0.738	0.963	0.904
Mobile Net [25]	0.961	1.000	0.807	0.907	0.789	0.966	0.893
SR+CLS (предложенный)	0.947	1.000	0.873	0.911	0.849	**0.971**	**0.916**
ResNet50 (на патчах ×40)	0.962	1.000	0.846	0.949	0.827	0.973	0.917

Таблица 2. Сравнение моделей повышения разрешения в задаче увеличения с ×20 до ×40 для тестовой выборки набора изображений PATH-DT-MSU WSS2v2 (аннотации не использовались)

Метрика / Метод	3em]PSNR	3em]SSIM
SRGAN [13]	27.2	0.78
ESRGAN [27]	31.82	0.86
SR+CLS (предложенный)	**32.66**	**0.89**

Таблица 3. Матрица ошибок классификации патчей предлагаемого метода (на тестовой выборке набора изображений PATH-DT-MSU WSS2v2). GT – истинные значения, Pred – предсказанные

Pred / GT	AT	BG	LP	MM	TUM
AT	4631	110	10	91	48
BG	7	38571	0	1	2
LP	0	0	1210	1	175
MM	91	0	6	2763	173
TUM	320	6	342	102	4334

6. ЗАКЛЮЧЕНИЕ

В данной работе предложен новый нейросетевой метод одновременного решения задач классификации и повышения разрешения фрагментов полнослайдовых гистологических изображений. Метод включает в себя последовательное обучение вспомогательной и основной нейросетевых моделей. Благодаря общему кодировщику, участвующего при решении обеих задач и использующего при обучении максимально возможное количество информации, продемонстрированы лучшие результаты предложенного метода как для задачи классификации, так и для задачи повышения разрешения. Достигнутые значения сбалансированной точности классификации 0.916 и $PSNR = 32.36$, $SSIM = 0.89$ для задачи повышения разрешения на тестовой выборке набора изображений PATH-DT-MSU WSS2v2 позволяют говорить о применимости предложенного метода на практике и использовании в реальных системах обработки и анализа гистологических изображений.

7. ИСТОЧНИК ФИНАНСИРОВАНИЯ

Исследования выполнены при финансовой поддержке Российского научного фонда в рамках научного проекта № 22-41-02002.

СПИСОК ЛИТЕРАТУРЫ

1. *Hu W. et al.* A state-of-the-art survey of artificial neural networks for whole-slide image analysis: from popular convolutional neural networks to potential visual transformers // Computers in Biology and Medicine. 2023. V. 161. P. 107034.
2. *Rodriguez J.P.M. et al.* Artificial intelligence as a tool for diagnosis in digital pathology whole slide images: A systematic review // Journal of Pathology Informatics. 2022. P. 100138.
3. *Wulczyn E. et al.* Deep learning-based survival prediction for multiple cancer types using histopathology images // PloS one. 2020. V. 15. № 6. P. e0233678.
4. *Khvostikov A., Krylov A., Mikhailov I., Malkov P.* Visualization and Analysis of Whole Slide Histological Images // Lecture Notes in Computer Science. 2023. V. 13644. P. 403–413.
5. *Juhong A. et al.* Super-resolution and segmentation deep learning for breast cancer histopathology image analysis // Biomedical Optics Express. 2023. V. 14. № 1. P. 18–36.
6. *Afshari M., Yasir S., Keeney G.L., Jimenez R.E., Garcia J.J., Tizhoosh H.R.* Single patch super-resolution of histopathology whole slide images: a comparative study // Journal of Medical Imaging. 2023. V. 10. № 1. P. 017501–017501.
7. *Chen Z., Wang J., Jia C., Ye X.* Pathological image super-resolution using mix-attention generative adversarial network // International Journal of Machine Learning and Cybernetics. 2023. P. 1–11.

8. *Xie L., Li C., Wang Z., Zhang X., Chen B., Shen Q., Wu Z.* SHISRCNet: Super-resolution And Classification Network For Low-resolution Breast Cancer Histopathology Image // Medical Image Computing and Computer Assisted Intervention – MICCAI 2023. 2023. P. 23–32.
9. *Salgueiro L., Marcello J., Vilaplana V.* SEG-ESRGAN: A Multi-Task Network for Super-Resolution and Semantic Segmentation of Remote Sensing Images // Remote Sensing. 2022. V. 14. № 22. P. 5862.
10. *Wang L., Li D., Zhu Y., Tian L., Shan Y.* Dual super-resolution learning for semantic segmentation // Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020. P. 3774–3783.
11. *Khvostikov A., Krylov A., Mikhailov I., Malkov P., Danilova N.* Tissue type recognition in whole slide histological images // CEUR Workshop Proceedings. 2021. V. 3027. P. 50.
12. *Larkin K.G.* Reflections on shannon information: In search of a natural information-entropy for images // arXiv preprint arXiv:1609.01117, 2016.
13. *Ledig C. et al.* Photo-realistic single image super-resolution using a generative adversarial network // Proceedings of the IEEE conference on computer vision and pattern recognition. 2017. P. 4681–4690.
14. *Schonfeld E., Schiele B., Khoreva A.* A U-Net based discriminator for generative adversarial networks // Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020. P. 8207–8216.
15. *Wang X., Xie L., Dong C., Shan Y.* Real-ESRGAN: Training real-world blind super-resolution with pure synthetic data // Proceedings of the IEEE/CVF international conference on computer vision. 2021. P. 1905–1914.
16. *Zhang Y., Tian Y., Kong Y., Zhong B., Fu Y.* Residual dense network for image super-resolution // Proceedings of the IEEE conference on computer vision and pattern recognition. 2018. P. 2472–2481.
17. *Shi W. et al.* Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network // Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. P. 1874–1883.
18. *He K., Zhang X., Ren S., Sun J.* Deep residual learning for image recognition // Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. P. 770–778.
19. *Schreiber S., Geldenhuys J., De Villiers H.* Texture synthesis using convolutional neural networks with long-range consistency and spectral constraints // 2016 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech). IEEE. 2016. P. 1–6.
20. *Krichen M.* Generative adversarial networks // 14th International Conference on Computing Communication and Networking Technologies (ICCCNT). IEEE. 2023. P. 1–7.
21. *Müller R., Kornblith S., Hinton G.E.* When does label smoothing help? // Advances in neural information processing systems. 2019. P. 4696–4705.
22. *Yoo J., Ahn N., Sohn K.A.* Rethinking data augmentation for image super-resolution: A comprehensive analysis and a new strategy // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020. P. 8375–8384.
23. *Huang G. et al.* Densely connected convolutional networks // Proceedings of the IEEE conference on computer vision and pattern recognition. 2017. P. 4700–4708.
24. *Tan M., Le Q.* Efficientnetv2: Smaller models and faster training // International conference on machine learning. PMLR. 2021. P. 10096–10106.
25. *Howard A.G. et al.* Mobilenets: Efficient convolutional neural networks for mobile vision applications // arXiv preprint arXiv:1704.04861, 2017.
26. *Chen Z., Guo X., Woo P.Y., Yuan Y.* Super-resolution enhanced medical image diagnosis with sample affinity interaction // IEEE Transactions on Medical Imaging. 2021. V. 40. № 5. P. 1377–1389.
27. *Wang X. et al.* ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks // Computer Vision–ECCV 2018 Workshops: Munich, Germany, September 8–14, 2018. Proceedings. Part V 15. Springer International Publishing. 2019. P. 63–79.

JOINT SUPER-RESOLUTION AND TISSUE PATCH CLASSIFICATION FOR WHOLE SLIDE HISTOLOGICAL IMAGES

© 2024 Zh. Sun[a], A. Khvostikov[a], A. Krylov[a], A. Sethi[b], I. Mikhailov[a], P. Malkov[a]

[a] *Lomonosov Moscow State University*
119991 Russia, Moscow, Leninskie Gory, 1
[b] *Indian Institute of Technology Bombay*
400076 India, Mumbai, Powai

Segmentation of wholeslide histological images through the classification of tissue types of small fragments is an extremely relevant task in digital pathology, necessary for the development of methods for automatic analysis of wholeslide histological images. The extremely large resolution of such images also makes the task of increasing image resolution relevant, which allows storing images at a reduced resolution and increasing it if necessary. Annotating whole slide images by histologists is complex and time-consuming, so it is important to make the most efficient use of the available data, both labeled and unlabeled. In this paper we propose a novel neural network method to simultaneously solve the problems of super-resolution of histological images from 20× optical magnification to 40x and classifying image fragments into tissue types at 20× magnification. The use of a single encoder as well as the proposed neural network training scheme allows to achieve better results on both tasks compared to existing approaches. The PATH-DT-MSU WSS2v2 dataset presented for the first time in this paper was used for training and testing the method. On the test sample, an accuracy value of 0.971 and a balanced accuracy value of 0.916 were achieved in the classification task on 5 tissue types, for the super-resolution task, values of $PSNR = 32.26$ and $SSIM = 0.89$ were achieved. The source code of the proposed method is available at: https://github.com/Kukty/WSI_SR_CL.

Keywords: histology, deep learning, convolutional neural networks, super-resolution, tissue type classification, wholeslide images