

---

---

# СОДЕРЖАНИЕ

---

---

Номер 3, 2021

---

---

## АНАЛИЗ ДАННЫХ

Анализ глубоких нейронных сетей для детекции стенозов коронарных артерий

*В. В. Данилов, О. М. Гергет, К. Ю. Клышников,  
А. Ф. Франжи, Е. А. Овчаренко*

---

3

## КОМПЬЮТЕРНАЯ ГРАФИКА И ВИЗУАЛИЗАЦИЯ

Автоматизация создания кибер-физических музейных экспонатов с использованием системы научной визуализации на кристалле

*К. В. Рябинин, М. А. Колесник*

12

360-видео на основе правильного додекаэдра: технология и методы реализации в системах виртуального окружения

*П. Ю. Тимохин, М. В. Михайлюк, К. Д. Пантелей*

19

Обобщенный вычислительный эксперимент и задачи верификации

*А. К. Алексеев, А. Е. Бондарев, В. А. Галактионов, А. Е. Кувшинников*

30

Метод прогрессивных обратных фотонных карт

*А. Д. Жданов, Д. Д. Жданов*

39

Зависимость шума от числа лучей в двунаправленной стохастической трассировке лучей с фотонными картами

*С. В. Ершов, Е. Д. Бирюков, А. Г. Волобой, В. А. Галактионов*

49

Метод улучшения качества заполнения областей изображений высокого разрешения

*А. В. Москаленко, М. В. Ерофеев, Д. С. Ватолин*

57

Гибридный метод подавления осцилляций Гиббса на изображениях магнитно-резонансной томографии

*М. А. Пенкин, А. С. Крылов, А. В. Хвостиков*

---

---

64

# CONTENTS

---

---

No. 3, 2021

---

---

## DATA ANALYSIS

Analysis of Deep Neural Networks for the Detection of Coronary Artery Stenosis

*V. V. Danilov, O. M. Gerget, K. Yu. Klyshnikov,  
A. F. Frangi, E. A. Ovcharenko*

3

---

## COMPUTER GRAPHICS AND VISUALIZATION

Automated Creation of Cyber-Physical Museum Exhibits Using Scientific Visualization System on a Chip

*K. V. Ryabinin, M. A. Kolesnik*

12

360-Degree Video Based on Regular Dodecahedron: Technology and Methods to Implement in Virtual Environment Systems

*P. Y. Timokhin, M. V. Mikhaylyuk, K. D. Panteley*

19

Generalized Computational Experiment and Verification Problems

*A. K. Alekseev, A. E. Bondarev, V. A. Galaktionov, A. E. Kuvshinnikov*

30

Progressive Backward Photon Mapping

*A. D. Zhdanov, D. D. Zhdanov*

39

Dependence of Noise on the Rays Number in Bidirectional Path Tracing with Photon Maps

*S. V. Ershov, E. D. Birukov, A. G. Voloboy, V. A. Galaktionov*

49

Method for Enhancing High-Resolution Image Inpainting with Two-Stage Approach

*A. V. Moskalenko, M. V. Erofeev, D. S. Vatolin*

57

A Hybrid Method of Suppressing Gibbs Oscillations in Magnetic Resonance Tomography Images

*M. A. Penkin, A. S. Krylov, A. V. Khvostikov*

64

---

---

УДК 004.932

## АНАЛИЗ ГЛУБОКИХ НЕЙРОННЫХ СЕТЕЙ ДЛЯ ДЕТЕКЦИИ СТЕНОЗОВ КОРОНАРНЫХ АРТЕРИЙ

© 2021 г. В. В. Данилов<sup>a,c,\*</sup>, О. М. Гергет<sup>a,\*\*</sup>, К. Ю. Клышников<sup>b,\*\*\*</sup>,  
А. Ф. Франжи<sup>c,\*\*\*\*</sup>, Е. А. Овчаренко<sup>b,\*\*\*\*\*</sup>

<sup>a</sup> Томский политехнический университет,  
634050 Томск, пр. Ленина, д. 30, Россия

<sup>b</sup> Научно-исследовательский институт комплексных проблем сердечно-сосудистых заболеваний,  
650000 Кемерово, Сосновый бульвар, д. 6, Россия

<sup>c</sup> University of Leeds, EC Stoner Building, Woodhouse Lane,  
Leeds, United Kingdom

\*E-mail: viacheslav.v.danilov@gmail.com

\*\*E-mail: gerget@tpu.ru

\*\*\*E-mail: klyshku@kemcardio.ru

\*\*\*\*E-mail: A.Frangi@leeds.ac.uk

\*\*\*\*\*E-mail: ovchea@kemcardio.ru

Поступила в редакцию 18.11.2020 г.

После доработки 25.12.2020 г.

Принята к публикации 12.01.2021 г.

В статье освещается подход на основе технологии машинного обучения, который представляет особый интерес для локализации и определения характеристик как одноочаговых стенозов, так и многоочаговых поражений. В связи со сложностью анализа большого количества данных клиницистом/кардиохирургом, в исследовании большое внимание уделено анализу, обучению и сравнению популярных детекторов для классификации и локализации очагов стеноза на данных коронарной ангиографии. Полный набор данных был собран в НИИ Комплексных проблем сердечно-сосудистых заболеваний на основе исследования коронарографии, среди которых ретроспективно выбраны данные 100 пациентов. Для автоматизированного анализа медицинских данных, в статье подробно рассмотрены 3 модели (SSD MobileNet V1, Faster-RCNN ResNet-50 V1, Faster-RCNN NASNet), которые варьировались по архитектуре, сложности и количеству весов. Приведено сравнение моделей по основным характеристикам эффективности: точность, время обучения и время предсказания. Результаты тестирования показали, что время обучения/предсказания прямо пропорционально сложности модели. Так, наименьшее время предсказания показала модель Faster-RCNN NASNet (среднее время обработки одного изображения составило 880 мс). Что касается точности, то наибольшая точность предсказания была получена моделью Faster-RCNN ResNet-50 V1. Данная модель достигла уровня 0.92 метрики mAP на валидационном наборе данных. С другой стороны, наиболее быстрой оказалась модель SSD MobileNet V1, которая способна выполнять предсказания с частотой предсказания 23 кадра в секунду.

DOI: 10.31857/S0132347421030031

### 1. ВВЕДЕНИЕ

На сегодняшний день ишемическая болезнь сердца (ИБС) является основной причиной смертности в мире [1]. Наличие атеросклеротических обструкций является наиболее частой причиной ИБС и заключается в уменьшении просвета коронарных артерий бляшкой. Клинически “золотым стандартом” для оценки стеноза коронарных артерий является инвазивная ангиография коронарных артерий (коронарография) с рентгенологическим контрастом. В свою очередь, анализ

данных ангиографии играет важную роль в локализации и интерпретации коронарного стеноза. В зависимости от анализа очагов стеноза и оценки метрики SYNTAX формируется тактика лечения пациента: минимально инвазивная чрессосудистая хирургия или чрезмерно инвазивное вмешательство.

Современное состояние исследований, связанных с локализацией, классификацией и интерпретацией такового, что существующие решения поставленных задач [2–8], несмотря на возможность

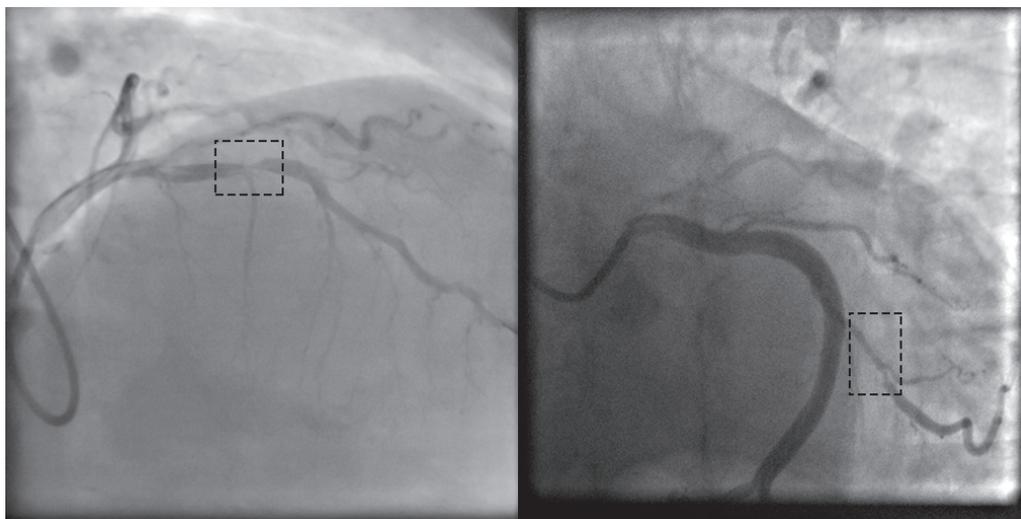


Рис. 1. Пример данных со стенозом для двух пациентов — область интереса обозначена прямоугольником.

получения достаточно высоких результатов по точности локализации (82–95%) и классификации (80–97%), обладают рядом существенных недостатков [9]. Одним из таких недостатков является плохая масштабируемость и не гибкость алгоритмов предварительной обработки, требующей тонкой настройки. Стоит также отметить, что большинство алгоритмов локализации построены по принципу каскадности, который склонен к накоплению ошибки. Исходя из этого, нами предпринята попытка по анализу, обучению и последующему сравнению нейронных сетей, которые способны выполнять оценку стеноза на сырых данных коронарной ангиографии с высокой степенью точности.

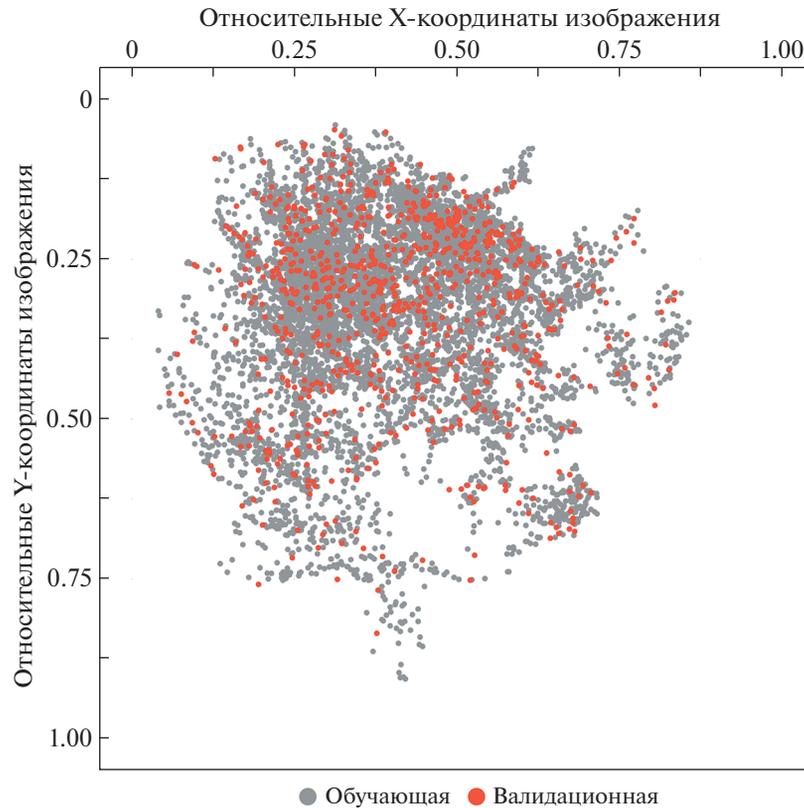
## 2. ИСХОДНЫЕ ДАННЫЕ

Исходные данные представляют собой записи ангиографии коронарных артерий, которые были получены на ангиографических установках “Coroscor” и “Innova” (Siemens) в НИИ Комплексных проблем сердечно-сосудистых заболеваний (г. Кемерово, Россия). Ретроспективно выбраны 100 пациентов, по данным которых было проведено детектирование гемодинамически значимых стенозов. Критерием исключения стало наличие множественных стенозов (двух и более коронарных артерий). В результате каждого ангиографического исследования получен видеоряд прохождения контраста по сосудистой системе сердца. Первичная сортировка данных осуществлялась с участием эксперта-кардиолога, который выбрал информативные изображения, содержащие только этап прохождения контраста и визуализацию

стенозированного участка. Таким образом был выделен набор изображений в оттенках серого (1 канал) с разрешением от  $512 \times 512$  до  $1000 \times 1000$  пикселей. Суммарно выборка составила 8325 изображений, из которых 7492 (90%) изображения использовались в качестве обучающей выборки, а 833 (10%) изображения — в качестве валидационной выборки. Для разметки данных была использована бесплатная версия SaaS (Software as a Service) решения “LabelBox”. Примеры размеченных входных данных для двух пациентов отражены на рис. 1.

Для анализа исходного набора данных нами проведена оценка размера области стеноза (площадь ограничивающего прямоугольника). Аналогично набору данных Common Objects in Context (COCO), размеры прямоугольников были разделены на 3 класса: “small” (площадь региона менее  $32^2$  пикселей), “medium” (площадь региона более  $32^2$  пикселей, но менее  $96^2$  пикселей) и “large” (площадь региона более  $96^2$  пикселей). В результате в исходном наборе данных общее число “small” объектов составило 2509 (30%), “medium” объектов — 5704 (69%) и “large” объектов — 113 (1%). Исходя из данного распределения видна несбалансированность исследуемого набора данных. Потенциально распознавание крупных областей стеноза может быть выполнено с меньшей точностью, чем маленьких и средних областей.

С целью решения задачи локализации в рамках исследования выполнена оценка распределения координат стенозов на изображениях исходного набора данных. Для этого оценивались нормализованные координаты центров ограничивающих пря-



**Рис. 2.** Карта плотности распределения очагов стеноза для обучающей и валидационной выборок. В качестве координат выбраны условные единицы, характеризующие положение точек, где точка (0; 0) соответствует левому верхнему углу изображения (ангиографии), точка (1; 1) – правому нижнему углу.

моугольников областей стеноза. На основе такой оценки получена карта распределения координат центров очагов стеноза, которая отражена ниже на рис. 2. Опираясь на результаты распределения координат, видно, что на карте наблюдаются 2 центра с относительными координатами (0.50; 0.20) и (0.27; 0.27), где располагается наибольшее количество очагов стеноза. Отметим, что координаты центров очагов распределены равномерно, а явные статистические выбросы при этом отсутствуют.

### 3. МЕТОДЫ

#### 3.1. Описание моделей

Для решения задачи локализации стеноза на основе данных ангиографии коронарных артерий использовался подход машинного обучения, который хорошо зарекомендовал себя в сфере компьютерного зрения и обработки изображений. Для решения данной задачи использовались детекторы SSD [10] и Faster-RCNN [11] из репозитория Tensorflow Detection Model Zoo [12] на основе таких моделей как MobileNet [13, 14], ResNet

**Таблица 1.** Краткая характеристика используемых моделей

Модель	Время предсказания, мс	mAP@[0.5:0.95]	Количество весов, млн.	Размер модели, Мб.
SSD MobileNet V1	56	32	4.2	44
Faster-RCNN ResNet-50 V1	89	30	25.6	114
Faster-RCNN NASNet	540	—	88.9	416

**Таблица 2.** Используемые опции при обучении моделей

Модель	Входной размер	Аугментация	Размер батча	Тип LR	LR
SSD MobileNet V1	$640 \times 640 \times 3$	Random horizontal flip Random crop image	4	Cosine decay with warm up	0.04
Faster-RCNN ResNet-50 V1	$600 \times 600 \times 3$	Random horizontal flip	2	Constant LR	0.0003
Faster-RCNN NASNet	$1200 \times 1200 \times 3$	Random horizontal flip	1	Constant LR	0.0003

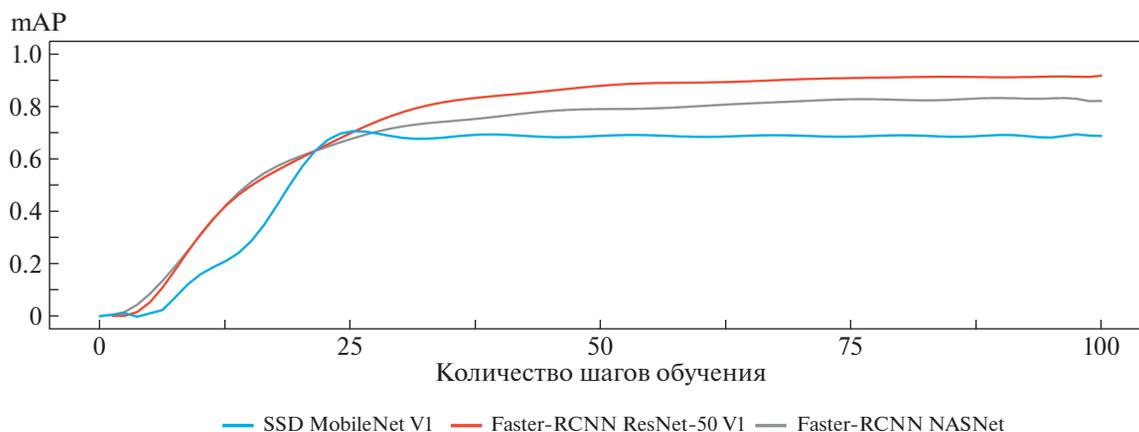
[15, 16] и NASNet [17, 18]. Всего исследовано 3 модели, которые варьировались по архитектуре, сложности и количеству весов. В качестве модели, использующей наименьшее количество весов, был выбран детектор SSD MobileNet V1, который способен обрабатывать данные в реальном формате времени. Наиболее сложной моделью из исследуемых является Faster-RCNN NASNet с более 80 млн весов. Выше в табл. 1 отражена краткая характеристика моделей, которые использовались в данном исследовании. Характеристика, а в особенности метрика mAP, получена на основе обучения нейронных сетей на наборе данных Common Objects in Context (COCO).

### 3.2. Обучение моделей

При обучении нейронных сетей их базовая конфигурация унаследована от конфигурации сетей, обученных на наборе данных COCO. Для однозначного сравнения исследуемых моделей общее число шагов обучения составило 100. Для лучшего восприятия в работе используется сокращенное обозначение количества шагов обучения, т.е. 100 шагов означает 100000 итераций обуче-

ния. В качестве локализационной функции потерь использовался Weighted Smooth L1 loss, а классификационной функции потерь – Weighted Focal Loss [19]. Важным будет отметить, что модель на основе SSD детектора обучалась с использованием техники Cosine decay with warm up. При использовании такой техники скорость обучения LR постепенно снижалась в зависимости от шага обучения. В качестве оборудования для обучения нами были использованы инстансы P2 (Nvidia Tesla K80 12 Gb, 1.87 TFLOPS) и P3 (Nvidia Tesla V100 16 Gb, 7.8 TFLOPS) от Amazon Web Services. Выше в табл. 2 приведены основные характеристики обучения моделей.

В процессе обучения нейронных сетей проведена оценка динамики точности на валидационном наборе данных. В качестве оценки точности выбрана метрика mean Average Precision (mAP) с использованием порогового значения Intersection over Union = 0.5 (mAP@0.5). Ниже на рис. 3 отражена сглаженная динамика изменения метрики mAP на основе валидационной выборки при обучении моделей. Видно, что практически все мо-

**Рис. 3.** Динамика изменения метрики mAP при обучении исследуемых моделей.

**Таблица 3.** Сравнение результатов исследуемых моделей

Модель	Количество весов, млн		Время обучения, часы		Время предсказания, мс		mAP@0.5	
	Абс.	Отн.	Абс.	Отн.	Абс.	Отн.	Абс.	Отн.
SSD MobileNet V1	4.2	1.0×	16	1.0×	43	1.0×	0.70	1.00×
Faster-RCNN ResNet-50 V1	25.6	6.0×	28	1.8×	98	2.3×	0.92	1.33×
Faster-RCNN NASNet	88.9	21.0×	147	9.5×	880	20.4×	0.84	1.22×

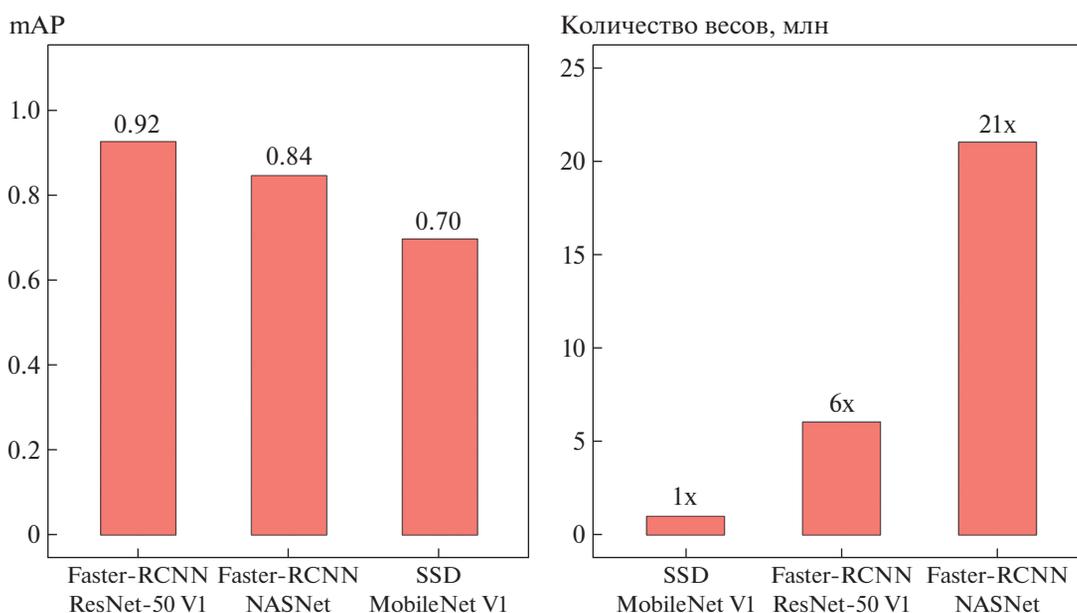
дели сходятся к некоторому значению асимптотической точности mAP.

#### 4. РЕЗУЛЬТАТЫ

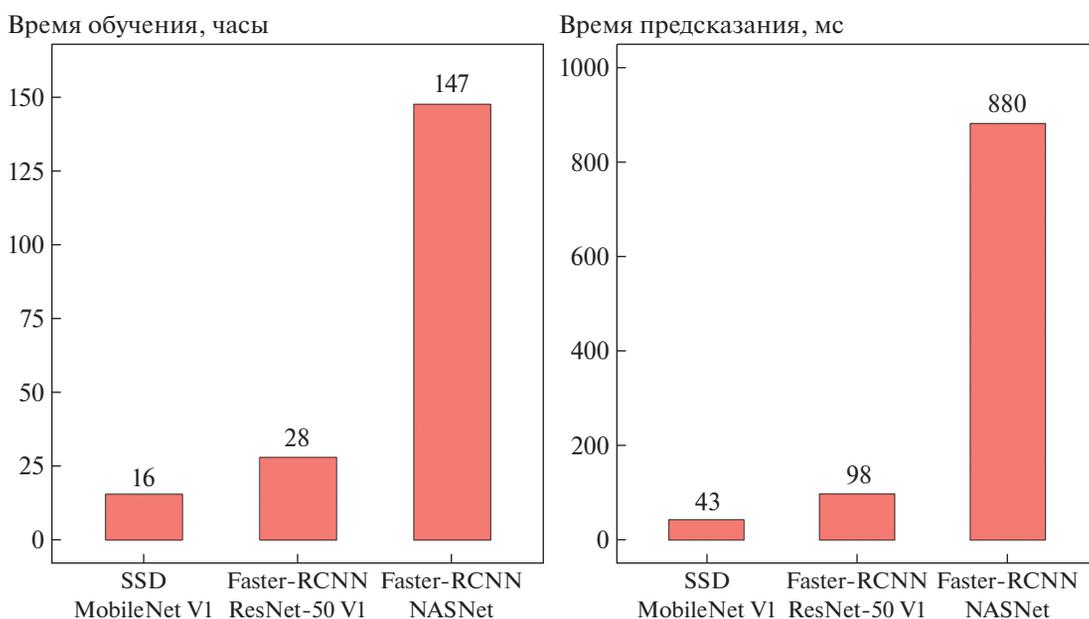
Сравнительный анализ полученных результатов на трех исследованных нейросетевых моделях приведен в табл. 3. Для относительного сравнения используются значения метрик для модели SSD MobileNet V1, а модели Faster-RCNN ResNet-50 V1 и Faster-RCNN NASNet с ней сравниваются. Ниже на рис. 4 и 5 отражены несколько основных метрик эффективности моделей, а именно метрика точности mAP, время обучения и время предсказания. Графики отсортированы таким образом, что на первом месте находятся модели, имеющие наилучшее значение по рассматриваемой метрике.

Для оценки времени предсказания использовался инстанс P3 (Nvidia Tesla V100 16 Gb, 7.8 TFLOPS) от Amazon Web Services. Как показало тестирование, скорость предсказания напрямую зависит от сложности моделей и количества весов. Так, наиболее медленной для предсказания оказалась модель Faster-RCNN NASNet. Среднее время обработки одного изображения для данной модели составляет 880 миллисекунд. С другой стороны, наиболее быстрой из рассматриваемых оказалась модель, базирующаяся на архитектуре MobileNet. Время предсказания одного изображения данной моделью составляет 43 миллисекунды, что говорит о возможности ее использования для предсказания расположения очага стеноза в реальном масштабе времени.

Исходя из полученных результатов видно, что наиболее точной моделью оказалась Faster-



**Рис. 4.** Сравнение метрики mAP и количества весов исследуемых моделей нейросетей, наглядно визуализирующее различия архитектур.



**Рис. 5.** Сравнение времени обучения и времени предсказания исследуемых моделей нейросетей, наглядно визуализирующее различия архитектур.

RCNN ResNet-50 V1. Средняя точность предсказания данной модели на валидационной выборке составила 0.92, а скорость предсказания – 98 миллисекунд/изображение ( $\approx 10$  кадров в секунду). Наиболее быстрая и относительно легковесная модель SSD MobileNet V1 имеет среднюю точность предсказания на валидационной выборке равной 0.70, при этом ее средняя скорость предсказания составляет 43 миллисекунды/изображение ( $\approx 23$  кадра в секунду). Стоит отметить модель Faster-RCNN NASNet, которая обладает более чем 3-х кратным преимуществом по количеству весов по сравнению с моделью Faster-RCNN ResNet-50 V1. Однако точность Faster-RCNN ResNet-50 V1 на 12% выше модели Faster-RCNN NASNet. В связи с этим, в качестве оптимального решения стоит выделить

именно модель Faster-RCNN ResNet-50 V1, которая способна достаточно быстро обрабатывать данные с высокой точностью.

#### 4.1. Тестирование моделей

В данном пункте продемонстрирована работа нейронных сетей на данных двух пациентов, для которых известна эталонная разметка, а их данные не использовались при обучении моделей. Примеры этих изображений с отмеченной областью стеноза отражены на рис. 6. Для тестирования использовались модели, которые давали наилучшие значения функции потерь и метрики mAP. Значения шагов обучения, на которых модели имеют оптимальные веса отражены ниже в табл. 4.

При сравнении полученных результатов на рис. 6 видно, что модели относительно точно выполняют локализацию очагов стеноза. Однако, существует ряд ложных срабатываний, а именно наличие ложноположительных случаев при тестировании модели Faster-RCNN NASNet. В обоих случаях данная модель дополнительно к размеченному эталонному очагу стеноза, с вероятностью более 90%, локализует дополнительные ложные очаги, которые находятся в правой коронарной артерии и передней нисходящей артерии в двух разных сегментах. С другой стороны, модель SSD MobileNet V1 чаще других ошибочно предсказывает отсутствие очага стеноза т.е. данная модель имеет более

**Таблица 4.** Значения шагов обучения, на которых модели имеют оптимальные веса

Модель	Оптимальный шаг обучения
SSD MobileNet V1	24
Faster-RCNN ResNet-50 V1	84
Faster-RCNN NASNet	95

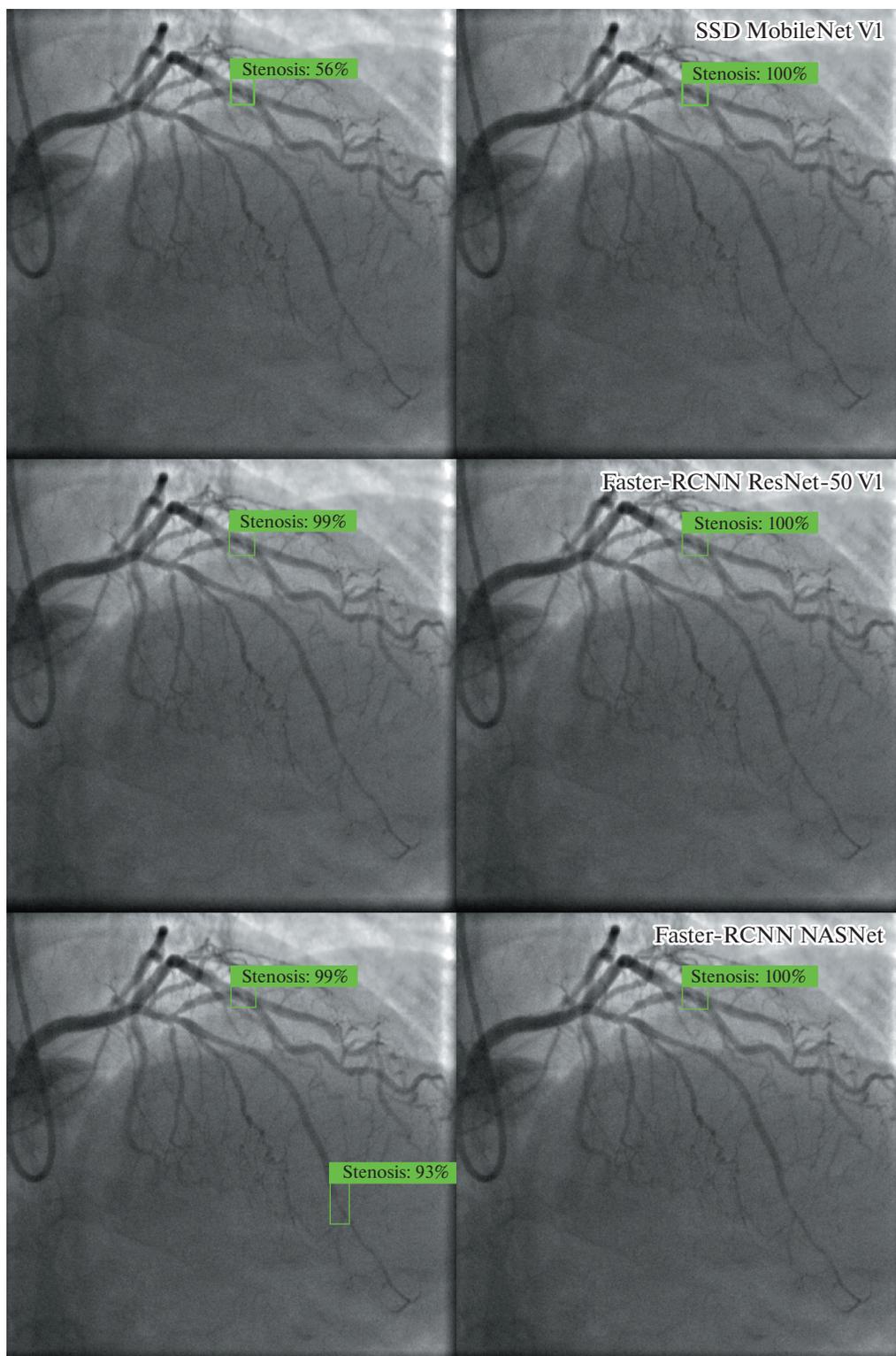


Рис. 6. Пример предсказания новых данных на двух соседних кадрах ангиографии для трех исследованных нейросетей.

высокую ложноотрицательную ошибку. Дополнительно стоит отметить эффективность работы детектора на базе архитектуры ResNet, а именно

Faster-RCNN ResNet-50 V1. Среднее значение индекса Дайса на тестовых данных для этой модели составило 0.85.

## 5. ЗАКЛЮЧЕНИЕ

В рамках рассмотренного исследования с целью выявления одноочаговых стенозов артерий проведено обучение и сравнительный анализ 3 популярных и перспективных детекторов, которые базировались на разных архитектурах нейронных сетей (MobileNet, ResNet-50, NASNet). Так, модель Faster-RCNN ResNet-50 V1 обладает оптимальным соотношением точность/время предсказания. Средняя точность предсказания (метрика mAP@0.5) данной модели на валидационной выборке составила 0.92, а скорость предсказания 10 кадров в секунду. Наиболее быстрая и относительно легковесная модель SSD MobileNet V1 имеет среднюю точность предсказания на валидационной выборке равной 0.70, при этом ее средняя скорость предсказания составляет 23 кадра в секунду. Важно будет отметить, что предложенный в работе подход представляет особый интерес для локализации и определения характеристик многососудистых, многоочаговых поражений. При таком подходе существует возможность не только детектировать очаги стеноза, но и оценивать ряд дополнительных характеристик стенозированного участка, а именно протяженность, диаметр, наличие боковых ветвей и параллельного шунта.

## 6. ФИНАНСИРОВАНИЕ

Работа по сбору исходных данных, предварительной обработке и созданию подхода для локализации очагов стеноза, основанного на машинном обучении, выполнена за счет гранта Российского научного фонда проект № 18-75-10061 “Исследование и реализация концепции роботизированного малоинвазивного протезирования клапана аорты”. Обучение разработанных моделей с помощью сервиса Amazon Web Services финансировалось в рамках государственного задания “Наука” № FFSWW-2020-0014 “Разработка научных основ технологии роботизированной мультипараметрической томографии на основе методов обработки больших данных и машинного обучения для исследования перспективных композиционных материалов”. Выбор основных метрик эффективности моделей и их анализ выполнен при поддержке гранта РФФИ 19-07-00351 “Методы и интеллектуальные технологии научного обоснования стратегических решений по цифровой трансформации”.

## СПИСОК ЛИТЕРАТУРЫ

1. GBD 2017 Causes of Death Collaborators. Global, regional, and national age-sex-specific mortality for 282 causes of death in 195 countries and territories, 1980–2017: a systematic analysis for the Global Burden of

- Disease Study 2017. *Lancet* (London, England). 2018. V. 392. № 10159. P. 1736–1788.
2. Toledano M., Lindenbaum M., Lessick J., Dragu R., Ghersin E., Engel A. Learning to Detect Coronary Artery Stenosis from Multi-Detector CT imaging. Haifa, 2010. 1–23 p.
3. Kang D., Dey D., Slomka P.J., Arsanjani R., Nakazato R., Ko H., Berman D.S., Li D., Kuo C.-C.J. Structured learning algorithm for detection of nonobstructive and obstructive coronary plaque lesions from computed tomography angiography // *J. Med. Imaging. SPIE-Intl. Soc. Optical Eng.* 2015. V. 2. № 1. P. 014003.
4. Kang D., Slomka P.J., Nakazato R., Arsanjani R., Cheng V.Y., Min J.K., Li D., Berman D.S., Jay Kuo C.C., Dey D. Automated knowledge-based detection of nonobstructive and obstructive arterial lesions from coronary CT angiography // *Med. Phys.* John Wiley and Sons Ltd. 2013. V. 40. № 4. P. 041912.
5. Chi Y., Huang W., Zhou J., Toe K.K., Zhang J.-M., Wong P., Lim S.T., Tan R.S., Zhong L. Stenosis detection and quantification on cardiac CTCA using panoramic MIP of coronary arteries / 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). IEEE., 2017. P. 4191–4194.
6. Antczak K., Liberadzki Ł. Stenosis Detection with Deep Convolutional Neural Networks. *MATEC Web Conf.* / Ed. Mastorakis N., Mladenov V., Bulucea A. 2018. V. 210. P. 04001.
7. Zreik M., Van Hamersvelt R.W., Wolterink J.M., Leiner T., Viergever M.A., Išgum I. A Recurrent CNN for Automatic Detection and Classification of Coronary Artery Plaque and Stenosis in Coronary CT Angiography // *IEEE Trans. Med. Imaging.* Institute of Electrical and Electronics Engineers Inc. 2019. V. 38. № 7. P. 1588–1598.
8. de Vos B.D., Wolterink J.M., Leiner T., de Jong P.A., Lessmann N., Išgum I. Direct Automatic Coronary Calcium Scoring in Cardiac and Chest CT // *IEEE Trans. Med. Imaging.* NLM (Medline). 2019. V. 38. № 9. P. 2127–2138.
9. Danilov V.V., Gerget O.M., Skirnevskii I.P., Manakov R.A., Kolpashchikov D.Y. Segmentation Based on Propagation of Dynamically Changing Superpixels // *Program. Comput. Softw.* 2020. V. 46. № 3. P. 191–202.
10. Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C.-Y., Berg A.C. SSD: Single Shot MultiBox Detector / *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics).* Springer Verlag, 2016. V. 9905 LNCS. P. 21–37.
11. Ren S., He K., Girshick R., Sun J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks // *IEEE Trans. Pattern Anal. Mach. Intell.* IEEE Computer Society. 2017. V. 39. № 6. P. 1137–1149.
12. Huang J., Rathod V., Sun C., Zhu M., Korattikara A., Fathi A., Fischer I., Wojna Z., Song Y., Guadarrama S., Murphy K. Speed/accuracy trade-offs for modern convolutional object detectors / *Proceedings – 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017.* Institute of Electrical and Electronics Engineers Inc., 2017. V. 2017-January. P. 3296–3305.

13. *Sandler M., Howard A., Zhu M., Zhmoginov A., Chen L.-C.* MobileNetV2: Inverted Residuals and Linear Bottlenecks / Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. IEEE Computer Society. 2018. P. 4510–4520.
14. *Howard A.G., Zhu M., Chen B., Kalenichenko D., Wang W., Weyand T., Andreetto M., Adam H.* MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. 2017.
15. *He K., Zhang X., Ren S., Sun J.* Deep residual learning for image recognition / Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society. 2016. V. 2016-Decem. P. 770–778.
16. *He K., Zhang X., Ren S., Sun J.* Identity Mappings in Deep Residual Networks / Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). Springer Verlag, 2016. V. 9908 LNCS. P. 630–645.
17. *Zoph B., Le Q.V.* Neural Architecture Search with Reinforcement Learning / 5th Int. Conf. Learn. Represent. ICLR 2017 – Conf. Track Proc. International Conference on Learning Representations, ICLR., 2016.
18. *Zoph B., Vasudevan V., Shlens J., Le Q. V.* Learning Transferable Architectures for Scalable Image Recognition / 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE., 2018. P. 8697–8710.
19. *Lin T.-Y., Goyal P., Girshick R., He K., Dollar P.* Focal Loss for Dense Object Detection / 2017 IEEE International Conference on Computer Vision (ICCV). IEEE, 2017. V. 42. № 2. P. 2999–3007.

## АВТОМАТИЗАЦИЯ СОЗДАНИЯ КИБЕР-ФИЗИЧЕСКИХ МУЗЕЙНЫХ ЭКСПОНАТОВ С ИСПОЛЬЗОВАНИЕМ СИСТЕМЫ НАУЧНОЙ ВИЗУАЛИЗАЦИИ НА КРИСТАЛЛЕ

© 2021 г. К. В. Рябинин<sup>a,\*</sup>, М. А. Колесник<sup>b,\*\*</sup>

<sup>a</sup> ФГБОУ ВО «Пермский государственный национальный исследовательский университет»,  
614990 Пермь, ул. Букирева, 15, Россия

<sup>b</sup> ГКБУК «Пермский краеведческий музей»,  
614000 Пермь, ул. Монастырская, 11, Россия

\*E-mail: kostya.ryabinin@gmail.com

\*\*E-mail: kolesnik.ma@outlook.com

Поступила в редакцию 10.10.2020 г.

После доработки 20.10.2020 г.

Принята к публикации 12.01.2021 г.

Статья посвящена вопросам автоматизации процесса создания автономных модулей научной визуализации на базе систем на кристалле с настраиваемым осязаемым пользовательским интерфейсом. Такие модули могут быть использованы в роли интерактивных экспонатов в рамках концепции так называемых умных музеев. Ключевой идеей автоматизации является генерация итогового программного обеспечения средствами онтологически управляемой платформы SciVi. В рамках этой платформы путем расширения управляющих онтологий организована поддержка генерации кода для систем на кристалле Raspberry Pi и Orange Pi. Алгоритм работы генерируемого программного обеспечения описывается в платформе SciVi высокоуровневым образом при помощи диаграмм потоков данных. При этом научная визуализация имеет аппаратную поддержку через графический API OpenGL ES, а поддержка осязаемого пользовательского интерфейса обеспечивается подключением специализированных библиотек и средств операционной системы для взаимодействия с внешними периферийными устройствами. Эффективность предложенных методов и средств подтверждена на практике при разработке нескольких кибер-физических экспонатов для выставки «Превращения» в Детском музейном центре, филиале Пермского краеведческого музея (г. Пермь).

DOI: 10.31857/S0132347421030092

### 1. ВВЕДЕНИЕ

Концепция т. н. умных музеев (англ. Smart Museum) [1] – закономерное следствие цифровизации институтов культуры. Суть этой концепции состоит во внедрении программируемой микроэлектроники в повседневные музейные практики. В рамках концепции умного музея параллельно решаются следующие задачи:

1. Мониторинг активности посетителей.
2. Организация навигации в музейном пространстве.
3. Организация интерактивных экспонатов.

Технологической базой для решения этих задач зачастую выступает Интернет вещей (англ. Internet of Things, IoT) [1]. Ключевой проблемой, при этом, оказывается высокий порог вхождения при освоении соответствующих программных инструментов и аппаратных средств ввиду их недостаточной высокоуровневости. Это заставляет музейных сотрудников обращаться к сторонним

ИТ-специалистам для развертывания специфических программно-аппаратных решений в заданных условиях. Актуальными оказываются вопросы автоматизации процессов создания таких решений и разработка высокоуровневого настраиваемого инструментария, который, в идеале, могли бы освоить люди без глубоких специальных знаний в области программирования и электроники.

Мониторинг активности посетителей и предоставление им электронных средств навигации актуальны в первую очередь для музеев, располагающих большими площадями, в то время как потребность в интерактивных экспонатах не зависит ни от размера, ни от тематики музея. В рамках решаемых в ходе данного исследования практических задач первоочередным было именно обеспечение интерактивности.

Важным требованием, предъявляемым к интерактивности экспонатов в рамках концепции

умного музея, является обеспечение естественности интерфейса и его привязка к музейному предмету. Взаимодействуя с экспонатом, посетитель должен получать некоторый новый опыт, не теряя при этом нити повествования той истории, которую рассказывает музейная экспозиция. С практической точки зрения это означает, что традиционные кнопочные интерфейсы далеко не всегда оказываются адекватны поставленной задаче, ведь они зачастую ассоциируются в сознании пользователя с традиционным человеко-машинным общением, ставшим для многих обыденностью. Это, в свою очередь, нарушает “погружение” посетителя в экспозицию, выводя его из нужного контекста, и ограничивает новый опыт, получаемый от физического посещения музея.

Проблему контекстуальной привязки, обеспечения естественности и нового опыта для посетителя может решить внедрение так называемых осязаемых пользовательских интерфейсов (англ. Tangible User Interface, TUI), переводящих интерактивные музейные экспонаты в плоскость кибер-физических систем (англ. Cyber-Physical System, CPS).

По аналогии с обобщенной кибер-физической системой, кибер-физический музейный экспонат состоит из специфического аппаратного пользовательского интерфейса (в общем случае обеспечивающего и прямую, и обратную связь с посетителем) и виртуального объекта, состояние которого управляется физическим интерфейсом (и влияет на состояние этого интерфейса при условии наличия обратной связи). При этом сам музейный предмет (экспонируемый на выставке артефакт, имеющий научное или культурное значение) может быть как физическим (фактически, частью осязаемого интерфейса), так и виртуальным (оцифрованной копией реального прототипа или оригинальной цифровой моделью какого-либо процесса/явления). В случае полностью виртуального артефакта, для его демонстрации посетителю адекватным является использование методов и средств научной визуализации. Научная визуализация, сочетая в себе когнитивную ясность и визуальную эстетику [2], позволяет точно передавать именно те черты экспоната, которые оказываются наиболее значимыми в контексте конкретной выставки.

Данная статья посвящена разработке программного инструментария, автоматизирующего построение и развертывание кибер-физических музейных экспонатов, основанных на технологиях Интернета вещей и научной визуализации. Практическим приложением этой работы стало создание нескольких музейных предметов для выставки “Превращения” в Детском музейном центре, филиале Пермского краеведческого музея, что позволило провести как теоретическое

исследование, так и практическую проверку предложенного подхода.

## 2. ТРЕБОВАНИЯ К КИБЕР-ФИЗИЧЕСКИМ МУЗЕЙНЫМ ЭКСПОНАТАМ

Эффективное использование кибер-физических систем в музейных выставках имеет определенную специфику, которая может быть выражена в следующих требованиях, предъявляемых к программно-аппаратному составу этих систем:

1. Быстрая и простая сборка (кибер-физические экспонаты должны состоять из интероперабельных модулей, не требующих заводских условий для сборки).

2. Быстрое и простое развертывание (для обеспечения мобильности выставок, кибер-физические экспонаты должны легко устанавливаться и демонтироваться).

3. Быстрое и простое включение/выключение (в идеале должно быть достаточно включения/выключения электроэнергии для запуска/остановки кибер-физических экспонатов).

4. Автономность (кибер-физические экспонаты не должны требовать постоянного обслуживания).

5. Интуитивный интерфейс (принцип человеко-машинного взаимодействия с кибер-физическими экспонатами должен быть понятен даже неподготовленному посетителю).

6. Низкая цена (в условиях зачастую ограниченного бюджета, выделяемого на формирование выставок, кибер-физические экспонаты должны состоять из максимально дешевых компонентов).

7. Заменяемые части (в случае поломки, должна быть возможность быстро найти и заменить вышедшие из строя части кибер-физических экспонатов).

8. Реконфигурируемость (при изменении выставки, должна быть возможность переиспользования модулей старых кибер-физических экспонатов для создания новых).

Для одновременного удовлетворения всех указанных требований мы предлагаем в качестве аппаратной основы кибер-физических экспонатов использовать либо микроконтроллерные системы, либо системы на кристалле (англ. System on Chip, SoC). Выбор конкретного аппаратного обеспечения зависит от специфики демонстрации кибер-физического экспоната.

## 3. МУЗЕЙНАЯ ПРАКТИКА ПРИМЕНЕНИЯ КИБЕР-ФИЗИЧЕСКИХ ЭКСПОНАТОВ

Микроконтроллерные системы целесообразно использовать тогда, когда в цифровой инфраструктуре музея для данного экспоната уже име-

ются средства визуализации, такие, например, как интерактивный киоск, или соединенный с монитором компьютер. Кроме того, микроконтроллерная система может использовать для визуализации мобильные устройства посетителей. Эти случаи были в подробностях рассмотрены в наших предыдущих работах [3, 4].

Если же в наличии нет подходящей рабочей станции, пригодной для визуализации виртуальной составляющей экспоната, наиболее экономически выгодным вариантом является построение кибер-физического экспоната на основе SoC (например, Raspberry Pi или Orange Pi). Выбор аппаратуры в пользу SoC определяет в данном случае их невысокая цена, сочетающаяся, при этом, с производительностью, достаточной для обеспечения качественной научной визуализации. Дополнительными важными достоинствами SoC являются их компактность и низкое энергопотребление, что позволяет осуществлять их монтаж даже в условиях очень ограниченного пространства.

В данной работе мы рассматриваем случай построения кибер-физических экспонатов с использованием SoC. В настоящее время SoC активно используются как в научных, так и в культурологических проектах. Например, популярная библиотека научной визуализации VTK успешно портирована на Raspberry Pi [5]. Существуют примеры использования Raspberry Pi как основы приборов для научных исследований [6], а также в качестве центрального компонента интерактивных музейных экспонатов с осязаемым интерфейсом, ориентированных на широкую аудиторию посетителей, включающую людей с ограниченными возможностями зрения [7]. Кроме того, Raspberry Pi широко применяется в робототехнике, в частности, для создания автономных роботов-экскурсоводов [8].

Однако, во всех вышеописанных примерах, использование SoC сопряжено с необходимостью низкоуровневого программирования управляющих модулей и взаимодействий, что делает процесс разработки конечных продуктов трудоемким и осуществимым лишь с участием высококвалифицированных профессионалов в области информационных технологий. Таким образом, вопрос создания высокоуровневого инструментария для работы с SoC остается весьма актуальным, особенно в контексте внедрения таких систем в музейную практику.

#### 4. АВТОМАТИЗИРОВАННАЯ ГЕНЕРАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ СИСТЕМ НА КРИСТАЛЛЕ

В качестве основы высокоуровневого инструментария для создания кибер-физических музейных экспонатов из SoC предлагается использо-

вать платформу визуального проектирования программно-аппаратных систем научной визуализации SciVi (<https://scivi.tools/>) [9]. Данная платформа может служить генератором программного обеспечения для интерактивного рендеринга, в том числе поддерживающего взаимодействие с аппаратными человеко-машинными интерфейсами [10]. Ключевой особенностью SciVi является ее гибкость в отношении адаптации к разнообразным сценариям работы. Эта гибкость достигается использованием онтологий как основы функционирования платформы: поведение SciVi полностью декларируется лежащей в ее основе онтологической базой знаний (БЗ), описывающей поддерживаемые форматы и типы данных, алгоритмы их обработки и визуализации, способы интерактивного взаимодействия с пользователем, особенности программно-аппаратного окружения, протоколы взаимодействия с другим программным обеспечением и т.п.

Одной из базовых функций SciVi является генерация программного обеспечения для интерактивной визуализации научных данных. Именно эта функция и может быть использована при создании кибер-физических музейных экспонатов. В рамках наших предыдущих исследований были разработаны методы и средства для генерации программного обеспечения кибер-физических музейных экспонатов, функционирующих на базе микроконтроллеров, и использующих для визуализации внешнюю аппаратуру (киоски, оснащенные компьютером и монитором, или мобильные устройства посетителей, подключенные к внутримузейной беспроводной компьютерной сети) [3, 4]. В данной работе мы обобщаем этот подход до поддержки SoC, что позволяет сделать кибер-физические экспонаты полностью автономными, не зависящими ни от внешнего оборудования, ни от цифровой инфраструктуры музея.

##### *4.1. Высокоуровневая настройка посредством диаграмм потоков данных*

Настройка платформы SciVi на решение конкретной задачи интерактивной визуализации осуществляется посредством диаграмм потоков данных (англ. Data Flow Diagram, DFD). В этом контексте DFD выступают визуальным языком программирования, позволяющим наглядно, при помощи высокоуровневых операторов (блоков DFD), связанных по данным, описать требуемые алгоритмы отображения виртуального музейного экспоната и интерактивного взаимодействия с ним. При этом пользователю предоставляется соответствующий графический интерфейс, обеспечивающий возможность строить DFD путем добавления операторов из заранее заготовленной палитры. Подобный способ записи алгоритмов хорошо зарекомендовал себя на практике в раз-

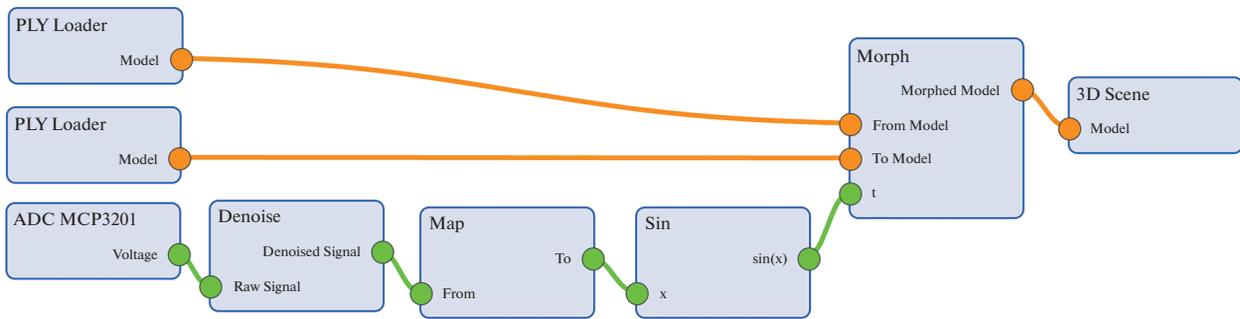


Рис. 1. DFD в среде платформы SciVi, описывающая работу кибер-физического музейного экспоната.

личных системах обработки и визуализации данных, например, KNIME [11] или Blender 3D [12].

Ключевым отличием SciVi от других систем, использующих DFD, является подход к формированию палитры операторов, доступных пользователю: в SciVi эта палитра формируется автоматически на основе онтологической БЗ о поддерживаемых аппаратных элементах, для которых платформа может генерировать прошивку, и программных методах обработки/визуализации данных. Таким образом, палитра операторов DFD может быть пополнена или изменена без модификации программного кода SciVi, лишь путем изменения БЗ.

DFD, описывающая работу одного из кибер-физических экспонатов, созданных в рамках данного исследования, представлена на рис. 1.

Этот экспонат посвящен наглядной демонстрации экогеографического правила Аллена [13] о том, что теплокровные животные в полярных широтах имеют более округлую форму тела, чем представители тех же семейств, живущие ближе к экватору. Это связано с необходимостью минимизировать площадь поверхности тела для сокращения потерь тепла в более холодном климате. Корпус экспоната оформлен в виде схематичной карты Земли, на которой при помощи рычага посетитель может выбрать интересующую широту. Положение рычага детектируется соединенным с ним потенциометром, напряжение на котором измеряется при помощи аналогово-цифрового преобразователя (АЦП) MCP3201, соединенного с SoC через шину SPI.

За захват данных с АЦП на DFD отвечает оператор “ADC MCP3201”. Полученные данные подвергаются фильтрации с целью подавления возможных помех (фильтр “Denoise”, использующий метод скользящего среднего), затем отображаются на диапазон  $[0; \pi]$  (оператор “Map”) и преобразуются функцией синуса (оператор “Sin”) в плавную кривую  $[0; 1]$ ,  $[1; 0]$ . Кривая определяет изменение параметра  $t$  у оператора “Morph”, который осуществляет по этому параметру линейную интерполяцию формы двух 3D-моделей, загружае-

мых операторами “PLY Loader” из файлов формата PLY. Модели были созданы специально для данного экспоната в программе Blender 3D, состоят примерно из 10 000 полигонов каждая, имеют одинаковую топологию и представляют собой “полярную” (песец) и “экваториальную” (корсак) формы “среднестатистического” млекопитающего, мелкого представителя семейства псовых (лат. canidae). Результаты рендеринга этих моделей средствами платформы SciVi (с применением псевдоглобального освещения, имитируемого методом matcap [14]) представлены на рис. 2.

В соответствии с приведенной DFD, когда рычаг в составе экспоната указывает на северный или южный полюс на карте,  $t = 0$  и отображается “полярная” форма животного. На экваторе, в свою очередь,  $t = 1$ , и отображается “экваториальная форма”. В промежуточных широтах отображаемая форма принимает вид, определяемый линейной интерполяцией граничных форм. Следует отметить, что результат линейной интерполяции форм при  $t = 0.5$  (средняя широтная полоса) внешне напоминает европейскую лисицу. Это подтверждает корректность использованной математической модели с учетом принятого уровня художественной стилизации.

#### 4.2. Управление посредством онтологий

Каждому оператору DFD соответствует фрагмент онтологии из БЗ SciVi. Пример такого фрагмента, описывающего оператор загрузки 3D-модели “PLY Loader”, приведен на рис. 3. Онтологии создаются и редактируются при помощи высокоуровневого графического редактора ONTOLIS [15].

“PLY Loader” в терминах системы SciVi является источником данных (“DataSource”), имеет настройку (“Setting”) “PLY File” типа “файл” (“File”) и выходной сокет (“Output”) “Model” типа “3D-модель” (“3Dmodel”). Для генерации кода важна реализация этого оператора, выраженная на онтологии вершиной “PLY Parser”. Эта реализация выполнена на языке C++, предназначена для исполнения на стороне клиента (“ClientSideWorker” –

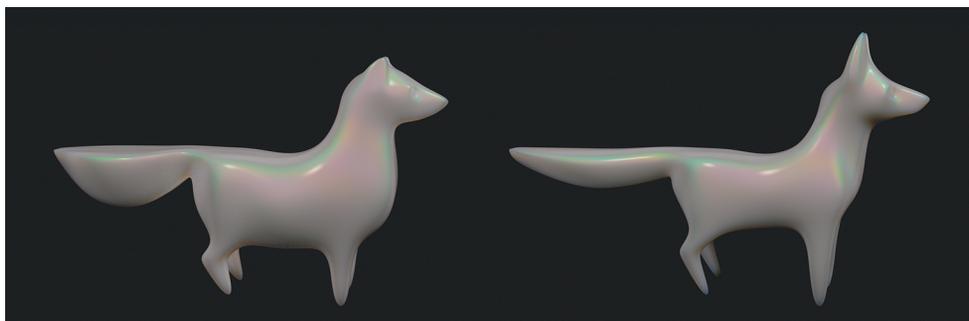


Рис. 2. Результат визуализации 3D-моделей “полярной” (слева) и “экваториальной” (справа) формы млекопитающего для демонстрации правила Аллена.

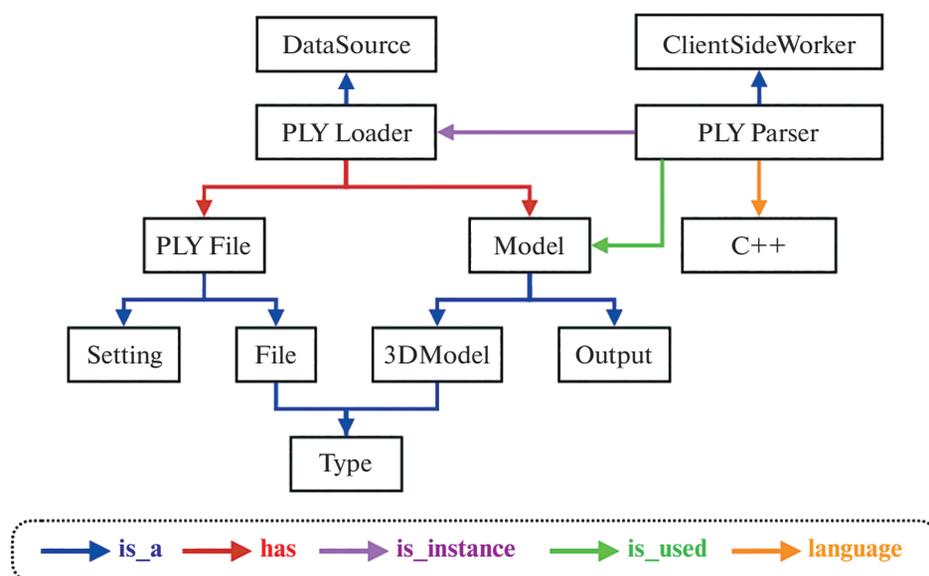


Рис. 3. Фрагмент онтологии из БЗ SciVi, описывающий оператор загрузки 3D-модели.

в данном случае это означает локальное исполнение без привлечения сетевых ресурсов) и используется для означивания выходной модели (“Model”). Ссылка на код реализации оператора содержится в атрибуте “path” вершины “PLY Parser” (сервисные атрибуты вершин на приведенной иллюстрации не обозначены в целях экономии места).

Вместе с реализацией для некоторых операторов хранятся также сведения об особых параметрах компиляции, компоновки и настройки операционной системы (ОС). Например, для оператора “ADC MCP3201” помимо исходного кода онтологией описывается еще и ряд настроек ОС, связанных со включением шины SPI, по которой к SoC подключается соответствующий внешний модуль АЦП.

Генератор кода в составе SciVi автоматически обходит онтологию операторов в соответствии с заданной DFD и собирает итоговую программу для целевой SoC из найденных реализаций, а так-

же необходимые скрипты для компиляции и установки. Таким образом, онтология операторов фактически выступает семантическим индексом для репозитория реализаций этих операторов.

#### 4.3. Стек используемых технологий

Стек основных технических и программных средств, на основе которых функционирует генерируемое программное обеспечение (ПО), представлен на рис. 4.

“SciVi Renderer” представляет собой основное приложение научной визуализации, отвечающее за отображение виртуальной части кибер-физического экспоната. Исходный код этого приложения автоматически генерируется платформой SciVi на языке C++. Оно использует графический API OpenGL ES (доступный на большинстве SoC через библиотеку Mesa) для рендеринга и различные библиотеки доступа к периферийному оборудованию

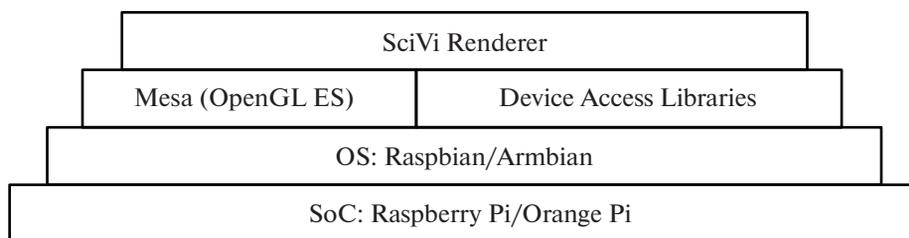


Рис. 4. Стек основных программно-аппаратных средств в составе кибер-физического музейного экспоната.

(“Device Access Libraries”) для взаимодействия с аппаратной частью TUI. Целевыми операционными системами (ОС), под управлением которых затем будет работать сгенерированное ПО, выступают Raspbian (для SoC Raspberry Pi) и Armbian (для SoC Orange Pi). При необходимости список поддерживаемых SoC и ОС легко можно расширить, дополнив БЗ SciVi соответствующими описаниями. В контексте создания кибер-физических музейных экспонатов предпочтение отдается Orange Pi, так как эта модель одноплатных компьютеров, с одной стороны, обладает достаточно высокой производительностью, но с другой — отличается низкой ценой.

В качестве экрана для отображения виртуальной составляющей экспоната может быть использован любой доступный монитор или проектор. В нашем случае был использован 7-дюймовый HDMI-совместимый мини-монитор разрешением  $1024 \times 600$  пикселей. Производительность рендеринга сгенерированным ПО требуемых 3D-моделей на Orange Pi PC2 для данного разрешения составляет 16 кадров в секунду (англ. Frames per Second, FPS), что достаточно для восприятия плавных движений.

Компиляция и развертывание сгенерированного ПО осуществляется классической последовательностью команд `configure`, `make`, `sudo make install`. Установка осуществляется с правами суперпользователя, так как в общем случае включает в себя ряд настроек ОС, необходимых для корректного функционирования целевого ПО. К таким настройкам относится, например, автоматический вход в ОС и запуск необходимого ПО при старте SoC. Таким образом, после однократной процедуры развертывания, SoC становится автономно функционирующим кибер-физическим экспонатом, не требующим дополнительной настройки.

#### 4.4. Соответствие заявленным требованиям

Кибер-физические экспонаты на основе SoC, программная часть которых генерируется платформой SciVi, соответствуют всем требованиям, отмеченным в разделе 1:

1. Простота и скорость сборки обусловлена модульностью SoC и их периферийных устройств, а также модульным характером DFD, при помощи которых осуществляется программная настройка.

2. Простота и скорость развертывания обеспечена автоматической генерацией платформой SciVi всего необходимого ПО и скриптов для его установки.

3. Простота включения обеспечивается автозапуском всего необходимого ПО, что позволяет включать SoC лишь подачей электропитания, без необходимости совершения каких-либо ручных настроек на старте. Простота выключения обеспечивается установкой режима “только чтение” для основной файловой системы, что позволяет отключать SoC путем простого выключения электропитания без риска повреждения контента карты памяти. Обе настройки записаны в автоматически генерируемый скрипт установки.

4. Автономность обеспечивается тем, что сгенерированное ПО включает в себя все необходимые настройки ОС SoC и периферийных устройств.

5. Интуитивность интерфейса — вопрос, скорее относящийся к дизайну экспоната. Автоматический контроль этого параметра никак не учтен в платформе SciVi, однако совместимость SciVi с TUI открывает широкие возможности для обеспечения интуитивности интерфейса и привязки его к контексту конкретной музейной выставки.

6. Низкая стоимость обеспечивается тем, что SoC и совместимые с ними периферийные устройства (сенсоры, актуаторы) достаточно дешевы, а платформа SciVi является OpenSource-проектом и может быть использована бесплатно. Так, например, стоимость одного кибер-физического экспоната для выставки “Превращения” составила около \$130.

7. Элементная база SoC может быть подобрана практически в любом магазине радиоэлектроники.

8. Реконфигурируемость обеспечивается возможностью описывать алгоритм работы при помощи DFD и автоматически генерировать код итогового ПО в среде платформы SciVi.

## 5. ЗАКЛЮЧЕНИЕ

Проведенное исследование было направлено на разработку методов и средств автоматизации создания кибер-физических музейных экспонатов. Ключевая идея состоит в использовании платформы SciVi, позволяющей по высокоуровневым описаниям автоматически генерировать ПО систем научной визуализации на кристалле, то есть ПО, обеспечивающее научно-точный рендеринг данных на аппаратном обеспечении SoC. Предложенные методы и средства прошли успешную практическую проверку при разработке выставки “Превращения” в Детском музейном центре, филиале Пермского краеведческого музея (г. Пермь). Ввиду малых размеров SoC и мониторов итоговые размеры экспонатов оказались небольшими, что помогло оптимально использовать маленькие экспозиционные площади.

Производительность рендеринга сгенерированного ПО на целевой SoC (Orange Pi PC2 под управлением ОС Armbian) на реальных данных составляет 16 FPS, что, в целом, удовлетворяет требованиям, предъявляемым к интерактивному музейному экспонату, однако тем не менее является сравнительно низкой. Для решения этой проблемы в дальнейшем планируется изучить вопрос возможной оптимизации алгоритмов рендеринга с учетом аппаратной специфики SoC. Кроме того, планируется организовать поддержку кросс-компиляции ПО для SoC на стороне платформы SciVi так, чтобы для развертывания программной части кибер-физического экспоната достаточно было только установить подготовленный в SciVi бинарный пакет, не производя компиляцию и компоновку программ на стороне целевого устройства.

## 6. БЛАГОДАРНОСТИ

Выражаем благодарность С.Л. Островскому и У.Ш. Сайрановой за предоставленную нам возможность участия в создании выставки “Превращения”, А.С. Кошелеву за помощь в создании корпусов кибер-физических экспонатов.

## СПИСОК ЛИТЕРАТУРЫ

1. *Chianese A., Piccialli F.* Designing a Smart Museum: when Cultural Heritage Joins IoT // Third International Conference on Technologies and Applications for Smart Cities (I-TASC'14). IEEE. 2014. 7 p. <https://doi.org/10.1109/NGMAST.2014.21>.
2. *Manakov D.V.* Visual Analytics and Data Abstraction Models // Proceedings of 28th International Conference on Computer Graphics and Vision “GraphiCon 2018”. Tomsk, 2018. P. 146–150.
3. *Ryabinin K.V., Kolesnik M.A.* Adaptive Scientific Visualization Tools for a Smart Paleontological Museum // Programming and Computer Software. Pleiades Publishing, Ltd. 2019. V. 45. № 4. P. 180–186. <https://doi.org/10.1134/S0361768819040066>
4. *Ryabinin K.V., Kolesnik M.A., Akhtamzyan A.I., Sudarikova E.V.* Cyber-Physical Museum Exhibits Based on Additive Technologies, Tangible Interfaces and Scientific Visualization // Scientific Visualization. National Research Nuclear University “MEPhI”. 2019. Q. 3. V. 11. № 4. P. 27–42. <https://doi.org/10.26583/sv.11.4.03>
5. *Ibanez L.* Raspberry Pi likes VTK [Электронный ресурс]. Kitware, 2012. <https://blog.kitware.com/raspberry-pi-likes-vtk/>.
6. *Buscher N., Ojeda A., Francoeur M., Hulyalkar S., Claros C., Tang T., Terry A., Gupta A., Fakhraei L., Ramanathan D.S.* Open-Source Raspberry Pi-Based Operant Box for Translational Behavioral Testing in Rodents // Journal of Neuroscience Methods. Elsevier, 2020. V. 342. <https://doi.org/10.1016/j.jneumeth.2020.108761>
7. *Rossetti V., Furfari F., Leporini B., Pelagatti S., Quarta A.* Enabling Access to Cultural Heritage for the Visually Impaired: an Interactive 3D Model of a Cultural Site // Procedia Computer Science. Elsevier, 2018. V. 130. P. 383–391. <https://doi.org/10.1016/j.procs.2018.04.057>
8. *Diallo A.D., Gobe S., Durairajah V.* Autonomous Tour Guide Robot Using Embedded System Control // Procedia Computer Science. Elsevier, 2015. V. 76. P. 126–133. <https://doi.org/10.1016/j.procs.2015.12.302>
9. *Ryabinin K.V., Chuprina S.I.* Using Scientific Visualization Systems to Automate Monitoring of Data Generated by Lightweight Programmable Electronic Devices // Programming and Computer Software. Pleiades Publishing, 2018. V. 44. № 4. P. 278–285. <https://doi.org/10.1134/S0361768818040102>
10. *Ryabinin K., Chuprina S., Belousov K.* Ontology-Driven Automation of IoT-Based Human-Machine Interfaces Development // Lecture Notes in Computer Science. Springer, 2019. V. 11540. P. 110–124. [https://doi.org/10.1007/978-3-030-22750-0\\_9](https://doi.org/10.1007/978-3-030-22750-0_9)
11. KNIME [Электронный ресурс]. <https://www.knime.com>.
12. Blender 3D [Электронный ресурс]. <https://www.blender.org>.
13. *Allen J.A.* The Influence of Physical Conditions in the Genesis of Species // Radical Review. 1877. V. 1. P. 108–140.
14. *Barros C.* MatCap – Render & Art Pipeline Optimization for Mobile Devices [Электронный ресурс]. 2019. <https://medium.com/playkids-tech-blog/matcap-render-art-pipeline-optimization-for-mobile-devices-4e1a520b9f1a>.
15. *Chuprina S., Nasraoui O.* Using Ontology-Based Adaptable Scientific Visualization and Cognitive Graphics Tools to Transform Traditional Information Systems into Intelligent Systems // Scientific Visualization. National Research Nuclear University “MEPhI”. 2016. Q. 1. V. 8. № 1. P. 23–44.

## 360-ВИДЕО НА ОСНОВЕ ПРАВИЛЬНОГО ДОДЕКАЭДРА: ТЕХНОЛОГИЯ И МЕТОДЫ РЕАЛИЗАЦИИ В СИСТЕМАХ ВИРТУАЛЬНОГО ОКРУЖЕНИЯ

© 2021 г. П. Ю. Тимохин<sup>a,\*</sup>, М. В. Михайлюк<sup>a,\*\*</sup>, К. Д. Пантелей<sup>a,\*\*\*</sup>

<sup>a</sup> ФГУ «ФНЦ Научно-исследовательский институт системных исследований РАН»,  
117218 Москва, Нахимовский пр., д. 36, к. 1, Россия

\*E-mail: [webpismo@yahoo.de](mailto:webpismo@yahoo.de)

\*\*E-mail: [mix@niisi.ras.ru](mailto:mix@niisi.ras.ru)

\*\*\*E-mail: [kpanteley@mail.ru](mailto:kpanteley@mail.ru)

Поступила в редакцию 23.12.2020 г.

После доработки 25.12.2020 г.

Принята к публикации 14.01.2021 г.

В статье предлагаются новые технология и методы реализации панорамного видео с обзором 360 градусов, основанные на проекции виртуального окружения на правильный додекаэдр. Идея состоит в построении виртуальной панорамы, наблюдаемой зрителем, из прямоугольных снимков виртуального пространства, имитирующих внутреннюю поверхность додекаэдра. Разработан метод вычисления параметров проекции и ориентации 12 камер додекаэдра, основанный на геометрии “золотых прямоугольников”, метод и алгоритмы синтеза кадра 360-видео, основанные на оригинальной схеме упаковки пентагонов, а также метод и алгоритм визуализации прямоугольных снимков, обеспечивающий синтез непрерывной виртуальной панорамы. Предложенные решения реализованы в программном комплексе и апробированы на примере задачи визуализации полета по орбите МКС над земной поверхностью. Результаты исследования могут быть применены в системах виртуального окружения, видеосимуляторах, научной визуализации, виртуальных лабораториях, образовательных приложениях, видеоинструкциях и др.

DOI: 10.31857/S0132347421030109

### 1. ВВЕДЕНИЕ

Во многих современных научных исследованиях и программах подготовки прикладных специалистов активно востребованы системы виртуального окружения [1]. Особенно это актуально для областей, где цена ошибки крайне высока: космос, авиация, медицина, нефтегазовая и атомная промышленность и др. [1–5]. Современные системы виртуального окружения являются достаточно сложными (нередко стационарными) программно-аппаратными комплексами, для запуска и настройки которых требуется участие VR-специалиста. Вместе с этим часто возникают ситуации, когда необходимо оперативно визуализировать уже пройденный процесс работы в виртуальном окружении, например, для анализа ошибок, допущенных при обучении в симуляторах, обмена результатами между удаленными исследователями, создания образовательных приложений и инструкций, и др. Одним из эффективных подходов является исследование и разработка новых технологий и форматов реализации видеозаписи виртуального окружения, при просмотре которой

зритель испытывал бы эффект погружения. Примерами являются технологии, основанные на различных форматах стерео, например, технология отложенного синтеза 4К-стереороликов в сложных динамических виртуальных сценах [6] или технология создания стереоанимаций на основе автостереоскопических мониторов [7, 8]. Благодаря бурному прогрессу экшн-камер (GoPro) и видеохостинга YouTube, в последнее время стало активно развиваться направление 360-видео [9], при котором съемка пространства выполняется с охватом 360 градусов по горизонтали и вертикали. Каждый кадр такого видео содержит изображение сферической панорамы, окружающей зрителя, что дает возможность пользователю произвольно менять направление взгляда в процессе воспроизведения видео и ощущать эффект присутствия в визуализируемом пространстве. Как известно, поверхность сферы невозможно отобразить на плоскости без искажений, поэтому изображение сферической панорамы обычно формируют из нескольких частей, которые при воспроизведении сшиваются между собой. Однако, в виртуальном пространстве визуальные и гео-

метрические свойства объектов часто моделируются с помощью различных имитационных техник (для оптимизации расхода вычислительного ресурса), которые при крутом повороте ракурса снимающей виртуальной камеры могут давать отображения, проблематично сшиваемые между собой. В виду этого возникает задача разработки технологий и методов реализации 360-видео в системах виртуального окружения, направленных на обеспечение непрерывности синтезируемой 360-видеопанорамы виртуального пространства. Для решения этой задачи в данной работе предлагается новая технология, основанная на отображении сферической виртуальной панорамы на поверхность правильного додекаэдра, а также методы создания и воспроизведения такого 360-видео. Предложенное решение реализовано на языке C++ с использованием графической библиотеки OpenGL.

## 2. СОВРЕМЕННЫЕ ПОДХОДЫ К РЕАЛИЗАЦИИ 360-ВИДЕО

Анализ современных научных публикаций и докладов разработчиков ведущих IT-компаний (YouTube, Facebook, GoPro) показывает высокую востребованность новых технологий и методов реализации 360-видео. Можно выделить два основных направления, по которым ведутся исследования и разработки в данной области.

*Первое направление* включает в себя технологии и методы, в которых сферическая панорама разворачивается на плоскость с помощью одной или нескольких картографических проекций [10, 11]. Так, в популярном видеохостинге YouTube базовым форматом для добавления 360-видео является эквидистантная цилиндрическая проекция [12]. В ней масштаб отображаемой панорамы является истинным только вдоль экватора, а при приближении к полюсам он возрастает до бесконечного. Чтобы обеспечить более однородный масштаб, в исследовании [13] предлагается отображать сферическую панораму на горизонтальные полосы, длины которых уменьшены в соответствии с близостью к полюсам (при этом сами области полюсов записываются в виде двух отдельных (“крышек”).

Ко *второму направлению* относятся решения, в которых сферическая панорама отображается на некоторый выпуклый многогранник, который разворачивается на плоскость. Широкое распространение получила кубическая проекция (cubemap projection) [14–17], при которой выполняется перспективное отображение панорамы на шесть граней куба. Хотя такой подход не дает вырожденные полюсы, как в эквидистантной проекции, детализация получаемого отображения все же имеет существенный разброс от центра до границ граней куба (до трех раз). Если не скомпенсиро-

вать такой разброс повышением разрешения всего отображения, то при проигрывании 360-видео будут наблюдаться артефакты “блочности”, что приведет к потере эффекта присутствия [14]. Для повышения качества получаемого 360-видео были также разработаны различные усовершенствования кубической проекции: равноугольная кубическая проекция [15], смещение центра съемки из центра куба [16], ориентирование граней куба на объекты переднего плана [17] и др. Несмотря на большую популярность кубической проекции, были также предложены и принципиально отличные способы отображения сферической панорамы. Так, в исследовании [18] панорама проецируется на ромбододекаэдр, а в публикации [19] разработчики из компании Facebook предложили проецировать панораму на грани тридцати пирамид видимости, равномерно охватывающих сферу. Несмотря на определенные преимущества перед кубической проекцией (уменьшение разброса детализации, сокращение объема видео-файла), данные подходы не получили развития в виду сложности реализации.

В ранних публикациях, посвященных разработке устройства для панорамного телевидения [20, 21], отмечается уникальное свойство правильного додекаэдра, делающее его лучшим выбором для съемки сферической панорамы. Правильный додекаэдр является единственным Платоновым телом, которое не только по форме близко к сфере, но и грани (пентагоны) которого близки к окружности. Благодаря этому свойству, разброс детализации отображения внутри пентагонов получается более сбалансированным, чем у куба или икосаэдра, а переход смежных ракурсов друг в друга – плавным (двухгранный угол около 116 градусов), что очень важно для непрерывной стыковки частей виртуальной панорамы [22]. В данной работе предлагаются новые технологии и методы реализации 360-видео в системах виртуального окружения, которые основаны на правильном додекаэдре и позволяют осуществлять синтез эффективного по площади кадра из отображений всех пентагонов, а также выполнять реконструкцию непрерывной панорамы из такого кадра.

## 3. ТЕХНОЛОГИЯ РЕАЛИЗАЦИИ 360-ВИДЕО НА ОСНОВЕ ПРАВИЛЬНОГО ДОДЕКАЭДРА

Рассмотрим задачу реализации 360-видео для наблюдателя, выполняющего исследование виртуального пространства, при котором он может изменить свое положение и направление взгляда. Обозначим через  $P_v$ ,  $\mathbf{v}$  и  $\mathbf{u}$  позицию, направление взгляда и вектор “вверх” наблюдателя в некоторый момент времени в мировой системе координат (World Coordinate System, WCS [23]). Расположим вокруг наблюдателя воображаемый правиль-

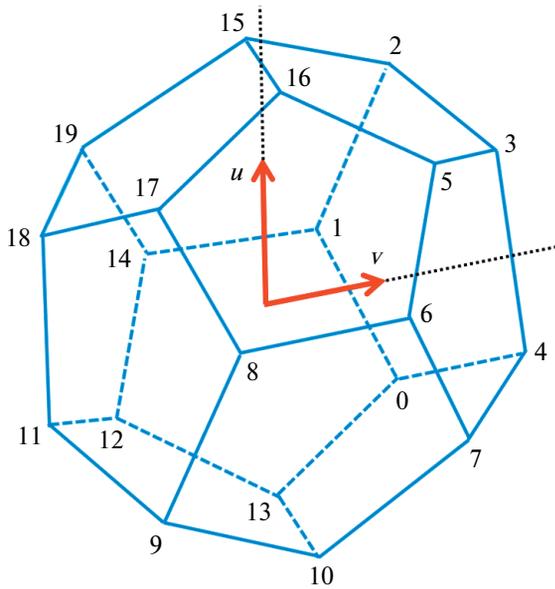


Рис. 1. Воображаемый додекаэдр.

ный додекаэдр, центр которого совпадает позицией  $P_V$ , а ориентация жестко связана с векторами  $\mathbf{v}$  и  $\mathbf{u}$  (см. рис. 1). В предлагаемой технологии реализации 360-видео мы вводим *камеры додекаэдра* – 12 одинаковых виртуальных камер  $C_0, \dots, C_{11}$  с симметричными пирамидами видимости, которые расположены в позиции  $P_V$  наблюдателя и направлены в центры граней (пентагонов) воображаемого додекаэдра.

Предлагаемая технология включает в себя три этапа. На *первом этапе* выполняется вычисление параметров проекции и ориентации камер додекаэдра, при которых каждый пентагон будет вписан в область вывода направленной в него камеры. На *втором этапе* для каждого момента времени синтеза кадра 360-видео выполняется съемка виртуального пространства через камеры додекаэдра и упаковка полученных “снимков” в видеокادر оригинального формата. Первый и второй этапы реализуются на стороне создателя 360-видео (в системе виртуального окружения). На *третьем этапе* для каждого кадра 360-видео из упакованных “снимков” выполняется реконструкция внутренней поверхности додекаэдра, имитирующей видимую панораму виртуального пространства. Данный этап выполняется на стороне зрителя 360-видео (в плеере). Рассмотрим более подробно методы реализации этих этапов.

### 3.1. Метод вычисления параметров проекции и ориентации камер додекаэдра

**Параметры проекции.** Как отмечалось выше, камеры додекаэдра имеют одинаковые пирамиды видимости, симметричные по вертикали и гори-

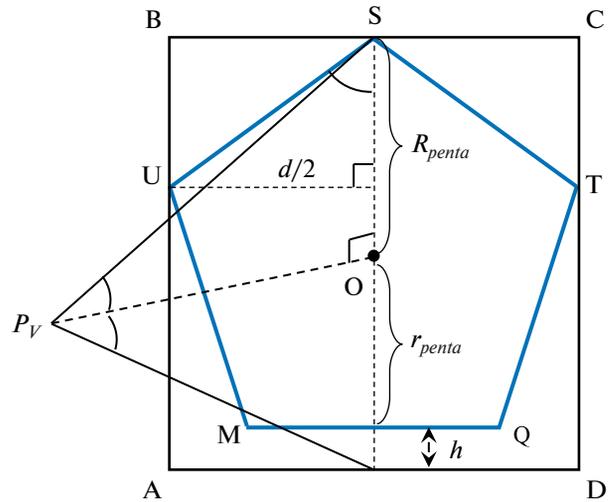


Рис. 2. Пентагон, вписанный в область вывода камеры додекаэдра.

зонтали. На рис. 2 показано расположение пентагона в области вывода  $C_i$ -й камеры додекаэдра, при котором их центры совпадают. Нетрудно заметить, что пентагон можно также вписать в область вывода и без расстояния  $h$ , если использовать камеру с наклонной пирамидой видимости (*oblique frustum*). Однако, в этом случае проекция виртуального пространства на нижней половине пентагона будет иметь больший масштаб, чем на верхней, что нарушает непрерывность стыковки снимков, получаемых от соседних камер додекаэдра.

Чтобы задать симметричную пирамиду видимости для каждой  $C_i$ -й камеры додекаэдра, вычислим угол  $\gamma$  вертикального раствора и отношение *aspect* ширины к высоте области вывода.

Из рис. 2 видно, что размеры области вывода  $C_i$ -й камеры должны быть пропорциональны размерам прямоугольника  $ABCD$ . Его центр  $O$  совпадает с центром пентагона  $MUSTQ$ ,  $AB = CD = 2R$ ,  $BC = AD = d$ , а  $AD$  отстоит от  $MQ$  на расстоянии  $h = R - r$ , где  $R, r$  – радиусы описанной и вписанной в пентагон окружностей, а  $d$  – диагональ пентагона [24]:

$$\begin{aligned} R &= (MQ/2) \operatorname{cosec}(\pi/5), \\ r &= (MQ/2) \operatorname{ctg}(\pi/5), \quad d = 2MQ \sin(3\pi/10). \end{aligned} \quad (1)$$

Тогда величину *aspect* области вывода камеры  $C_i$  можно выразить как

$$\operatorname{aspect} = \frac{BC}{AB} = \frac{d}{2R} = \frac{2 \sin(3\pi/10)}{\operatorname{cosec}(\pi/5)} \approx 0.951. \quad (2)$$

Угол  $\gamma$  найдем из прямоугольного  $\Delta P_V OS$ . Его катет  $P_V O$  является радиусом сферы, вписанной в додекаэдр, и находится как  $P_V O = (MQ/2) \operatorname{ctg}(\pi/5) \operatorname{tg}(\theta/2)$ , где  $\theta = 2 \arcsin$

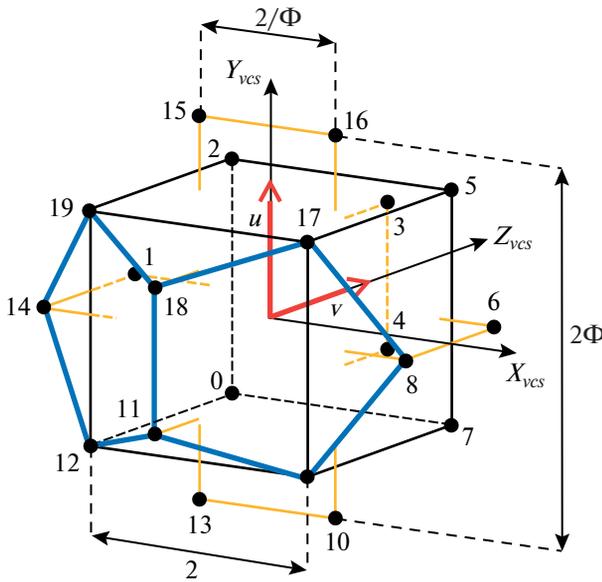


Рис. 3. Связь системы координат наблюдателя с воображаемым додекаэдром.

$(\cos(\pi/3)/\sin(\pi/5)) \approx 116.565^\circ$  – двухгранный угол правильного додекаэдра [24]. Из  $\Delta P_i OS$ :

$$\begin{aligned} \operatorname{tg}(\gamma/2) &= \frac{R}{P_i O} = \frac{\operatorname{cosec}(\pi/5)}{\operatorname{ctg}(\pi/5) \operatorname{tg}(\theta/2)} = \\ &= \frac{1}{\cos(\pi/5) \operatorname{tg}(\theta/2)} = \sec(\pi/5) \operatorname{ctg}(\theta/2), \end{aligned}$$

откуда

$$\gamma = 2 \operatorname{arctg}(\sec(\pi/5) \operatorname{ctg}(\theta/2)) \approx 74.8^\circ. \quad (3)$$

**Параметры ориентации.** В данной работе ориентация каждой  $C_i$ -й камеры задается аналогично ориентации наблюдателя с помощью вектора  $\mathbf{v}_i$  взгляда и вектора  $\mathbf{u}_i$  “вверх” в системе WCS. Чтобы определить эти векторы, пронумеруем вершины додекаэдра, как показано на рис. 1, и запишем через них последовательность  $H$  пентагонов: 0 – {4, 7, 10, 13, 0}, 1 – {0, 1, 2, 3, 4}, 2 – {4, 3, 5, 6, 7}, 3 – {7, 6, 8, 9, 10}, 4 – {10, 9, 11, 12, 13}, 5 – {13, 12, 14,

1, 0}, 6 – {12, 11, 18, 19, 14}, 7 – {1, 14, 19, 15, 2}, 8 – {3, 2, 15, 16, 5}, 9 – {6, 5, 16, 17, 8}, 10 – {9, 8, 17, 18, 11}, 11 – {18, 17, 16, 15, 19}. Вектор  $\mathbf{v}_i$  направлен в центр  $i$ -го пентагона, а вектор  $\mathbf{u}_i$  – из этого центра в некоторую, например, вторую, вершину  $i$ -го пентагона (порядковый номер этой вершины фиксируется для всех пентагонов).

Рассмотрим вычисление координат векторов  $\mathbf{v}_i$  и  $\mathbf{u}_i$ . Для этого вначале определим связь левой системы координат наблюдателя (View Coordinate System, VCS) [23] с рассматриваемым додекаэдром. Начало системы VCS находится в центре додекаэдра, ось  $X_{vcs}$  проходит через середину ребра {6, 8}, ось  $Y_{vcs}$  – через середину ребра {15, 16} и совпадает с вектором  $\mathbf{u}$  “вверх” наблюдателя, а ось  $Z_{vcs}$  – через середину ребра {3, 4} и совпадает с вектором  $\mathbf{v}$  взгляда наблюдателя (см. рис. 3).

Обозначим через  $P_0, \dots, P_{19}$  координаты вершин додекаэдра в системе VCS и найдем их значения. Для этого будем использовать “золотые прямоугольники” [25] – три взаимно-перпендикулярных прямоугольника (см. рис. 3), стороны которых связаны “золотым отношением”  $\Phi$  (числом Фидия) [26]:

$$\Phi = (1 + \sqrt{5})/2 = 2 \sin(3\pi/10) = 2 \cos(\pi/5), \quad (4)$$

при котором большая сторона относится к меньшей так же, как их сумма относится к большей из них. Восемь вершин додекаэдра будут вершинами вписанного куба размера 2, а остальные двенадцать – вершинами “золотых прямоугольников”. Выпишем на основе рис. 3 табл. 1 координат  $P_0, \dots, P_{19}$ .

Обозначим через  $K_i$  координаты центра  $i$ -го пентагона, а через  $\mathbf{v}_{vcs,i}$  и  $\mathbf{u}_{vcs,i}$  – координаты вектора взгляда и вектора “вверх” камеры  $C_i$  в системе VCS. Запишем их выражения через координаты  $P_0, \dots, P_{19}$ :

$$K_i = \frac{1}{5} \sum_{j=0}^4 P_{n_{i,j}}, \quad \mathbf{v}_{vcs,i} = K_i, \quad \mathbf{u}_{vcs,i} = P_{n_{i,1}} - K_i, \quad (5)$$

Таблица 1. Координаты вершин додекаэдра в системе VCS

Координаты	Значения			
$P_0, P_2, P_5, P_7$	(-1 -1 1)	(-1 1 1)	(1 1 1)	(1 -1 1)
$P_9, P_{12}, P_{17}, P_{19}$	(1 -1 -1)	(-1 -1 -1)	(1 1 -1)	(-1 1 -1)
$P_1, P_6, P_8, P_{14}$	(-Φ 0 1/Φ)	(Φ 0 1/Φ)	(Φ 0 -1/Φ)	(-Φ 0 -1/Φ)
$P_3, P_4, P_{11}, P_{18}$	(0 1/Φ Φ)	(0 -1/Φ Φ)	(0 -1/Φ -Φ)	(0, 1/Φ -Φ)
$P_{10}, P_{13}, P_{15}, P_{16}$	(1/Φ -Φ 0)	(-1/Φ -Φ 0)	(-1/Φ Φ 0)	(1/Φ Φ 0)

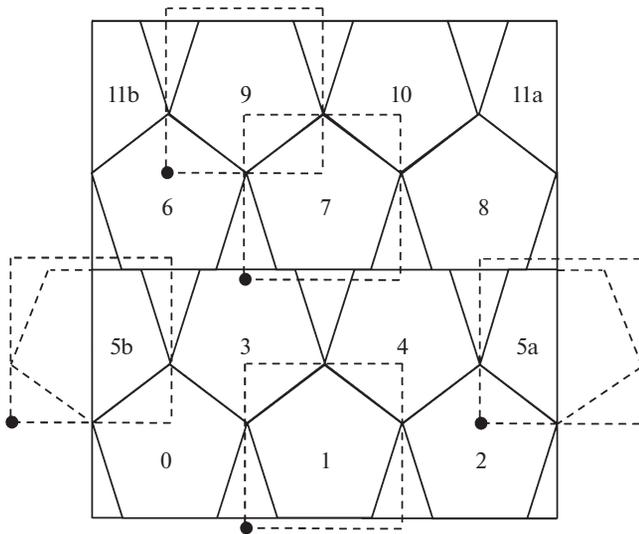


Рис. 4. Схема упаковки пентагонов в кадр 360-видео.

где  $n_{i,j}$  – номер  $j$ -й вершины  $i$ -го пентагона из последовательности  $H$ . Обозначим через  $M_v$  верхнюю левую  $3 \times 3$ -подматрицу матрицы преобразования координат из системы WCS в систему VCS [23]. Тогда координаты искоемых векторов  $\mathbf{v}_i$  и  $\mathbf{u}_i$  в системе WCS можно вычислить как

$$\mathbf{v}_i = M_v^{-1} \mathbf{v}_{vcs,i}, \quad \mathbf{u}_i = M_v^{-1} \mathbf{u}_{vcs,i}. \quad (6)$$

Полученные в результате описанного этапа параметры проекции ( $aspect$  и  $\gamma$ ) и параметры ориентации ( $\mathbf{v}_i$  и  $\mathbf{u}_i$ ) мы устанавливаем для каждой виртуальной камеры  $C_i$  с помощью операторов `gluPerspective` и `gluLookAt` библиотеки OpenGL [27].

### 3.2. Метод синтеза кадра 360-видео

Синтез кадра 360-видео в данной работе предлагается выполнять на основе разработанной оригинальной схемы упаковки, состоящей из *прямых* и *перевернутых* пентагонов (см. рис. 4) – “снимков” виртуального пространства, полученных с помощью камер додекаэдра. Отметим, что рендеринг виртуального пространства мы выполняем в `rgba`-текстуру, исходно заполненную непрозрачным белым цветом, а перевернутые “снимки” получаем путем инвертирования знака вектора “вверх” (из формулы (6)) соответствующей камеры. На рис. 4 номерами обозначен порядок обхода камер додекаэдра, а пунктиром и черными кружками – ряд показательных случаев расположения их областей вывода. Как можно видеть, в предложенной схеме присутствуют разбитые пополам перевернутые пентагоны (5 и 11). Это достигается путем сдвига левого нижнего угла области вывода таким образом, чтобы нужная половина попадала в кадр (текстуру), а другая ее часть отсекалась границами кадра. На примере пентагонов 7 и 9 видно, что содержащие их области вывода будут перекрывать друг друга. Чтобы перевернутые “снимки” не затирали прямые, мы используем специальные корректирующие маски (см. рис. 5), состоящие из треугольников непрозрачного белого цвета (областей кадра, не несущих информации для пользователя) и треугольников непрозрачного черного цвета, представляющих собой места пересечения прямых и перевернутых “снимков”. Синтез прямого “снимка” в текстуре кадра реализует следующий алгоритм:

1. Рендерим треугольники 1–4 (см. рис. 5a) непрозрачного белого цвета.

2. Если номер снимка из отрезка [6, 8], то:

блокируем запись в `rgb`-каналы (с помощью оператора `glColorMask(0, 0, 0, 1)`),

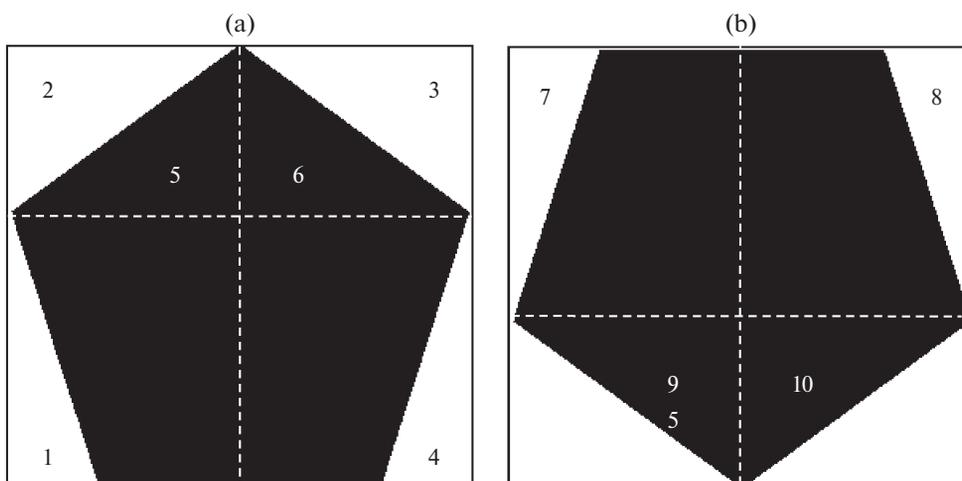


Рис. 5. Корректирующие маски для прямого (a) и перевернутого (b) пентагона.

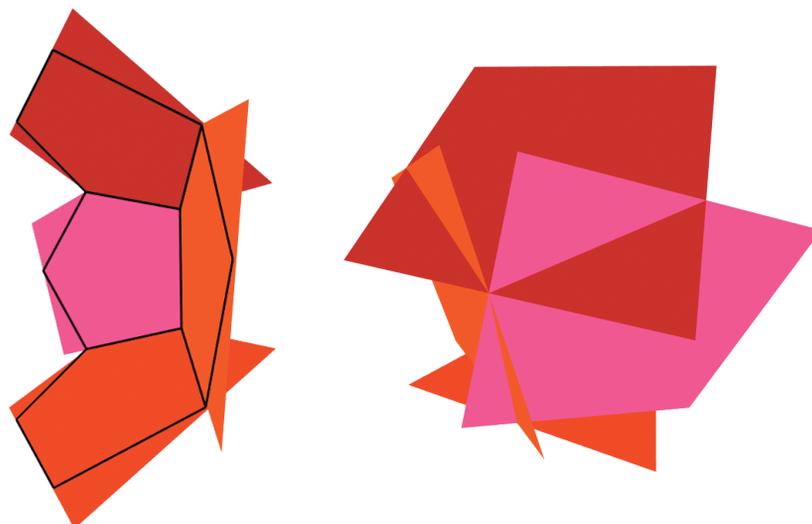


Рис. 6. Принцип реконструкции виртуальной панорамы.

рендерим в нижнюю часть области вывода прямоугольную полосу непрозрачного черного цвета, которая по высоте соответствует расстоянию  $h$  (см. рис. 2)

разрешаем запись в rgba-каналы.

3. Задаем режим альфа-смешивания `glBlendFunc(GL_ONE_MINUS_DST_ALPHA, GL_ONE)` [27].

4. Рендерим виртуальное пространство через соответствующую камеру додекаэдра.

5. Отключаем альфа-смешивание.

6. Блокируем запись в rgb-каналы.

7. Рендерим треугольники 5 и 6 (см. рис. 5a) непрозрачного черного цвета.

8. Разрешаем запись в rgba-каналы.

Конец алгоритма.

Синтез перевернутого “снимка” в текстуре кадра реализует следующий алгоритм

1. Рендерим треугольники 7, 8 непрозрачного белого цвета и треугольники 9, 10 непрозрачного черного цвета (см. рис. 5b).

2. Задаем режим альфа-смешивания `glBlendFunc(GL_ONE_MINUS_DST_ALPHA, GL_ONE)`.

3. Рендерим виртуальное пространство через соответствующую камеру додекаэдра.

4. Отключаем альфа-смешивание.

5. Если номер снимка из отрезка  $[3, 5]$ , то:

Разрешаем запись в rgba-каналы.

Рендерим в верхнюю часть области вывода прямоугольную полосу непрозрачного черного цвета, которая по высоте соответствует расстоянию  $h$  (см. рис. 2)

Конец алгоритма.

Отметим, что в приведенных алгоритмах рендеринг корректирующих треугольников и полосок выполняется в ортографической проекции. В результате выполнения для каждой камеры додекаэдра соответствующего ей алгоритма получим в rgba-текстуре упакованный кадр 360-видео.

### 3.3. Метод реконструкции виртуальной панорамы

Реконструкция виртуальной панорамы в данной работе выполняется в отдельной виртуальной сцене (*сцене плеера*). Оси мировой системы WCS этой сцены совпадают с осями системы VCS из раздела 3.1, за исключением оси  $Z_{wcs} = -Z_{vcs}$ . В начало системы WCS поместим зрителя, вокруг которого будем визуализировать 12 прямоугольников, имитирующих внутреннюю поверхность додекаэдра, как показано на рис. 6. Из рисунка видно, что на данном этапе добавка  $h$ , введенная в разделе 3.1, будет не видима, поэтому ее целесообразно не использовать. Все прямоугольники пронумерованы аналогично пентагонам воображаемого додекаэдра и жестко связаны с системой WCS — так же, как и додекаэдр с системой VCS: ось  $X_{wcs}$  проходит через середину ребра, образуемого пересечением 3-го и 9-го прямоугольника и т.д. Чтобы построить непрерывную виртуальную панораму, выполним следующие шаги.

**Шаг 1. Вычисление координат вершин прямоугольников в системе WCS.** Обозначим через  $A'_i, B'_i, C'_i, D'_i$  — координаты вершин  $i$ -го прямоугольника в системе WCS, а через  $M'_i, U'_i, S'_i, T'_i, Q'_i$  и  $O'_i$  — координаты вершин и центра  $i$ -го пентагона, вписанного в  $A'_i B'_i C'_i D'_i$ , как показано на рис. 7. Координаты вершин пентагона можно легко найти из табл. 1, ин-

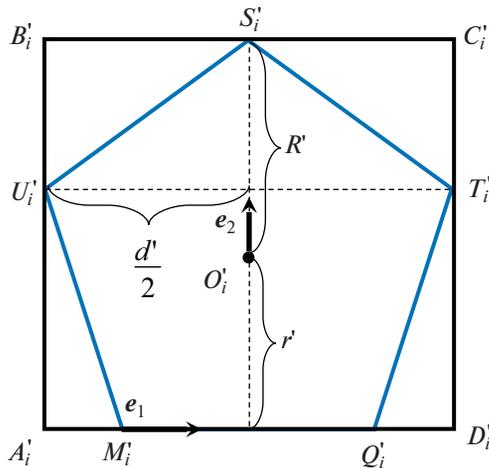


Рис. 7. Координаты вершин прямоугольника, имитирующего пентагон внутренней поверхности додекаэдра.

вертировав знак координаты  $z$ , а координаты центра – с помощью первой формулы из (5). Введем орты  $\mathbf{e}_1 = \mathbf{M}'_i\mathbf{Q}'_i/|\mathbf{M}'_i\mathbf{Q}'_i|$  и  $\mathbf{e}_2 = \mathbf{O}'_i\mathbf{S}'_i/|\mathbf{O}'_i\mathbf{S}'_i|$ . Тогда искомые координаты вершин прямоугольника  $A'_iB'_iC'_iD'_i$  можно вычислить как

$$\begin{aligned} B'_i &= S'_i - \frac{d'}{2}\mathbf{e}_1, & C'_i &= S'_i + \frac{d'}{2}\mathbf{e}_1, \\ A'_i &= B'_i - (R' + r')\mathbf{e}_2, & D'_i &= C'_i - (R' + r')\mathbf{e}_2, \end{aligned} \quad (7)$$

где  $d'$ ,  $R'$  и  $r'$  – диагональ, радиусы описанной и вписанной окружности пентагона додекаэдра из рис. 3, вычисленные аналогично  $d$ ,  $R$  и  $r$  из (1).

Таблица 2. Текстурные координаты вершин прямоугольников

Номера прямоугольников	Текстурные координаты			
	$\{s_0, t_0\}$	$\{s_1, t_1\}$	$\{s_2, t_2\}$	$\{s_3, t_3\}$
0, 1, 2	(0, 1)	(1, 1)	(1, 0)	(0, 0)
3, 4, 5, 11	(1, 0)	(0, 0)	(0, 1)	(1, 1)
6, 7, 8	(1, 1)	(0, 1)	(0, 0)	(1, 0)
9, 10	(0, 0)	(1, 0)	(1, 1)	(0, 1)

**Шаг 2. Задание текстурных координат вершин прямоугольников.** Обозначим через  $\{s_0, t_0\}$ ,  $\{s_1, t_1\}$ ,  $\{s_2, t_2\}$  и  $\{s_3, t_3\}$  текстурные координаты вершин  $A'_i$ ,  $B'_i$ ,  $C'_i$ ,  $D'_i$ . Значения этих текстурных координат мы задаем с помощью табл. 2, учитывающей, прямой или перевернутый был “снимок”  $C_i$ -й камеры.

**Шаг 3. Визуализация текстурированных прямоугольников.** Создадим массив  $T$  из 12 текстурированных объектов, в которые будут считываться текстуры  $w_i \times h_i$  пикселей из кадра 360-видео, где  $w_i$ ,  $h_i$  – ширина и высота области вывода  $C_i$ -й камеры без добавки  $h$ . Обозначим через  $V_{coords}$  двумерный массив из 14 пар (10 целых пентагонов и 4 половины) координат  $(x, y)$  (в пикселах) левых нижних углов областей вывода камер додекаэдра, вычисленных на этапе синтеза кадра 360-видео. Визуализацию текстурированных прямоугольников реализует следующий алгоритм

1. Инициализируем  $indx = 0$  – индекс пары координат в массиве  $V_{coords}$ .
2. Цикл по  $i$  от 0 до 11, где  $i$  – номер прямоугольника.
 

Зададим ширину  $frmW$  кадра видео 360, число  $V_{coords}[indx][1]$  пропускаемых строк и число  $V_{coords}[indx][0]$  пропускаемых пикселей в строке.

Если  $i > 5$ , то  $indx = (i + 1)$ , в противном случае  $indx = i$ .

`glPixelStorei (GL_UNPACK_ROW_LENGTH, frmW).`

`glPixelStorei (GL_UNPACK_SKIP_ROWS, V_{coords}[indx][1]).`

`glPixelStorei (GL_UNPACK_SKIP_PIXELS, V_{coords}[indx][0]).`

Загрузим текстуру в  $T[i]$ -й текстурированный объект (с помощью оператора `glTexSubImage2D`).

Если  $(i == 5) \parallel (i == 11)$ , // номера пентагонов, складывающихся из двух половин, то:

`glPixelStorei (GL_UNPACK_SKIP_ROWS, V_{coords}[indx+1][1]).`

`glPixelStorei (GL_UNPACK_SKIP_PIXELS, V_{coords}[indx+1][0] + w_i/2).`

Загрузим текстуру в  $T[i]$ -й текстурированный объект.

Конец Если.

Отообразим  $i$ -й прямоугольник с наложенным  $T[i]$ -м текстурированным объектом.

Конец цикла.

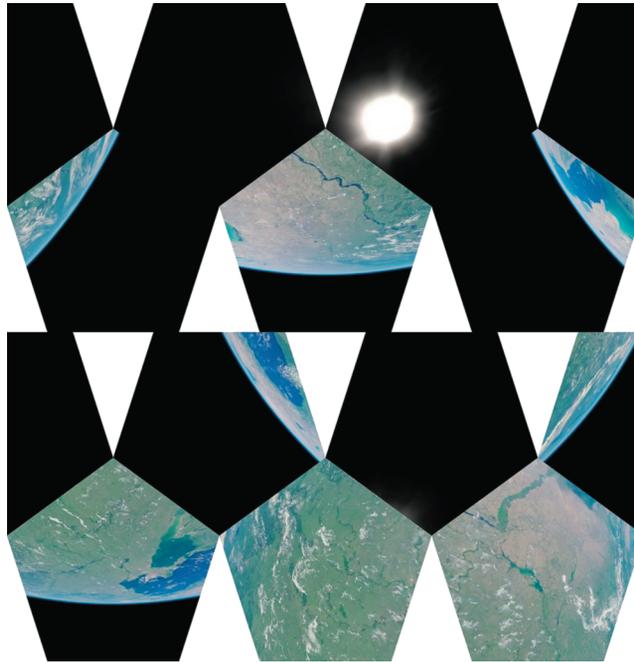


Рис. 8. Пример синтезированного кадра 360-видео с упаковкой пентагонов.

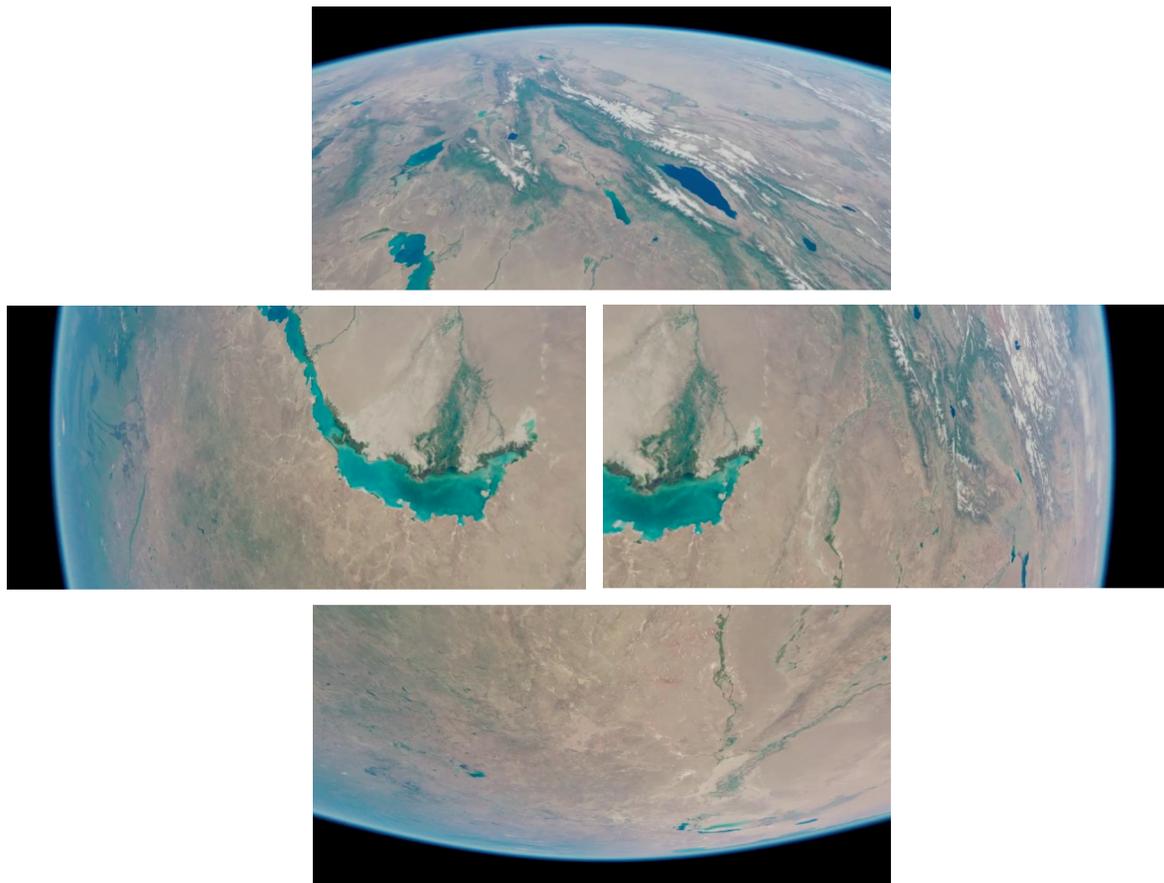


Рис. 9. Примеры полученных виртуальных панорам Земли

После выполнения трех шагов для каждого из 12 прямоугольников, имитирующих внутреннюю поверхность додекаэдра, перед зрителем в виртуальной сцене плеера будет сформирована непрерывная виртуальная панорама.

#### 4. РЕЗУЛЬТАТЫ

Предложенные технология и методы были реализованы в программном комплексе, осуществляющем синтеза 360-видео на основе правильного додекаэдра в системах виртуального окружения. Созданный комплекс написан на языке C++ с применением графической библиотеки OpenGL и набора библиотек FFmpeg по кодированию/декодированию цифровых видеопоследовательностей. В состав предложенного программного решения входят модуль захвата 360-кадров, встраиваемый в систему визуализации, а также проигрыватель созданных 360-видео.

Разработанное решение было протестировано в подсистеме комплекса VirSim [1, 28], осуществляющей моделирование и визуализацию полета над Землей по орбите Международной космической станции (МКС). Было создано 360-видео полета вдоль участка трассы 3-го суточного витка МКС, проходящего через озеро Балхаш. Запись 360-видео выполнялась с частотой 25 кадров/с на основе стандарта H.264 сжатия видео и контейнера MP4. Размеры кадра 360-видео составили  $3000 \times 3077$  пикселей. На рисунке 8 показан пример синтезированного кадра с реализованной схемой упаковки пентагонов. При воспроизведении 360-видео зритель может менять направление взгляда на 360 градусов по горизонтали и вертикали, а также увеличивать/уменьшать изображение. На рис. 9 показаны полученные виртуальные панорамы Земли для различных направлений взгляда.

Было выполнено сравнение качества проекций виртуального окружения, получаемых на основе правильного додекаэдра и куба. Мы оценивали изменение числа пикселей, которое может занимать проекция одного и того же отрезка на текстуре грани (додекаэдра и куба). Наибольшее число  $r_{max}$  пикселей такой отрезок будет занимать в области вершины грани, а наименьшее  $r_{min}$  – в центре грани. При этом у додекаэдра отношение  $r_{min}/r_{max}$  составляет около 0.6, а у куба – только около 0.3 (в идеале, у сферы такое отношение равно 1). Таким образом использование проекции на правильный додекаэдр позволяет уменьшить до 2 раз потери детализации по сравнению с кубической проекцией. Также было проведено количественное сравнение уменьшения площади не-

информативных участков кадра с нашим предыдущим исследованием [22]:

$$\sigma_1 = 1 - \frac{S_{penta}}{S_{quad}} =$$

$$= 1 - \frac{10}{8\text{tg}(36)\sin(54)(\text{cosec}(36) + \text{ctg}(36))} \approx 0.31, \quad (8)$$

$$\sigma_2 = \frac{S_{emptiness}}{S_{frame}} =$$

$$= \frac{\text{tg}(36)}{\text{cosec}(36) + \text{ctg}(36) + 2\sin(72)} \approx 0.15, \quad (9)$$

где  $\sigma_1$ ,  $\sigma_2$  – доли неинформативных областей кадра 360-видео для исследования [22] и нашей текущей реализации, соответственно;  $S_{penta}$  – площадь пентагона;  $S_{quad}$  – площадь области вывода, содержащей пентагон (без добавки  $h$ );  $S_{emptiness}$  – площадь всех пустот между пентагонами для кадра из рисунка 4, и  $S_{frame}$  – площадь кадра из рисунка 4.

Оценка показала, что за счет реализованной схемы упаковки пентагонов удалось уменьшить почти в два раза площадь неинформативных участков, что позволяет создавать более детализированные 360-видео при том же расходе битрейта видеопотока.

#### 5. ЗАКЛЮЧЕНИЕ

В данной работе предложены решения, которые относятся к направлению исследования и разработки новых технологий и методов синтеза 360-видео в системах виртуального окружения, основанных на проекции виртуального пространства на грани выпуклых многогранников. Широко распространенная в настоящее время кубическая проекция обладает рядом недостатков, ограничивающих ее применение в системах виртуального окружения. В данном исследовании предлагается новая технология, основанная на проекции виртуального пространства на грани правильного додекаэдра. Технология включает в себя этап вычисления параметров проекции и ориентации камер додекаэдра, этап синтеза кадра 360-видео и этап реконструкции виртуальной панорамы. Для реализации этих этапов предложены методы и алгоритмы, основанные на оригинальной схеме упаковки “снимков” виртуального пространства. Созданные технология и методы были реализованы в программном комплексе и апробированы в подсистеме комплекса VirSim, осуществляющей моделирование и визуализацию

полета над Землей по орбите МКС. Апробация подтвердила эффективность и адекватность полученных результатов поставленной задаче. Полученные решения могут быть применены при развитии существующих и создании новых перспективных систем виртуального окружения, в видеотренажерах, системах научной визуализации, виртуальных лабораториях, образовательных приложениях и др.

Публикация выполнена в рамках государственного задания по проведению фундаментальных научных исследований (ГП 14) по теме (проекту) “34.9. Системы виртуального окружения: технологии, методы и алгоритмы математического моделирования и визуализации” (0580-2021-0012).

### СПИСОК ЛИТЕРАТУРЫ

1. Михайлюк М.В., Мальцев А.В., Тимохин П.Ю., Страшнов Е.В., Крючков Б.И., Усов В.М. Система виртуального окружения Virsim для имитационно-тренажерных комплексов подготовки космонавтов // Пилотируемые полеты в космос, 2020. № 4 (37). С. 72–95.
2. Barladian B.K., Shapiro L.Z., Mallachiev K.A., Khoroshilov A.V., Solodov Yu.A., Voloboy A.G., Galaktionov V.A., Koverninskii I.V. Visualization Component for the Aircraft Real-Time Operating System JetOS // Programming and Computer Software. 2020. V. 46. № 3. P. 167–175.
3. Lubimov M., Vidiger D., Shergin D. Limitless dynamic landscapes of 1 mm per pixel density, zooming included // SIGGRAPH 2020. 2020. Article № 6. <https://doi.org/10.1145/3407662.3407753>
4. Piedra J.A., Ojeda-Castelo J.J., Quero-Valenzuela F., Piedra-Fdez I. Virtual Environment for the Training of the Hands in Minimally Invasive Thoracic Surgery // IEEE 2016 8th International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES), 2016. P. 1–4.
5. Страшнов Е.В., Мироненко И.Н., Финагин Л.А. Моделирование режимов полета квадрокоптера в системах виртуального окружения // Информационные технологии и вычислительные системы, 2020. № 1. С. 85–94.
6. Timokhin P.Yu., Mikhaylyuk M.V., Vozhegov E.M., Panteley K.D. Technology and Methods for Deferred Synthesis of 4K Stereo Clips for Complex Dynamic Virtual Scenes // Programming and Computer Software. 2021. V. 47. № 1. P. 67–75.
7. Andreev S., Bondarev A., Bondarenko A., Galaktionov V., Bondareva N. Constructing Stereo Images of Error Surfaces in Problems of Numerical Methods Verification // Proceedings of the 30th International Conference on Computer Graphics and Vision, 2020. V. 2744.
8. Andreev S.V., Bondarev A.E., Galaktionov V.A., Bondareva N.A. The problems of stereo animations construction on modern stereo devices // Scientific Visualization. 2018. № 10 (4). P. 40–52.
9. Corbillon X., Simon G., Devlic A., Chakareski J. Viewport-Adaptive Navigable 360-Degree Video Delivery // IEEE 2017 International Conference on Communications (ICC), 2017. P. 1–7.
10. Jiang H., He G., Yu W., Wang Z., Li Y. Scalable Video Coding Based on the User’s View for Real-Time Virtual Reality Applications. In: Advances in Multimedia Information Processing – PCM 2017, LNCS 10736. Ed. by B. Zeng Springer, 2018. P. 766–775.
11. Yu M., Lakshman H., Girod B. A Framework to Evaluate Omnidirectional Video Coding Schemes // IEEE 2015 International Symposium on Mixed and Augmented Reality, 2015. P. 31–36.
12. Ray B., Jung J., Larabi M.-C. A Low-Complexity Video Encoder for Equirectangular Projected 360 Video Content // IEEE 2018 International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018. P. 1723–1727.
13. Li J., Wen Z., Li S., Zhao Y., Guo B., Wen J. Novel tile segmentation scheme for omnidirectional video // IEEE 2016 International Conference on Image Processing (ICIP), 2016. P. 370–374.
14. Тимохин П.Ю., Михайлюк М.В., Вожегов Е.М. Эффективные методы и алгоритмы синтеза видео 360 градусов на основе кубической проекции виртуального окружения // Труды ИСП РАН. 2020. Т. 32. № 4. С. 73–88.
15. Brown C. Bringing pixels front and center in VR video // Google AR and VR. March 14, 2017. <https://www.blog.google/products/google-ar-vr/bringing-pixels-front-and-center-vr-video/>.
16. Kuzuyakov E., Liu S., Pio D. Optimizing 360 Video for Oculus. Facebook F8 developers conference, 2016. <https://developers.facebook.com/videos/f8-2016/optimizing-360-video-for-oculus/>.
17. Chen Z., Wang X., Zhou Y., Zou L., Jiang J. Content-Aware Cubemap Projection for Panoramic Image via Deep Q-Learning // MultiMedia Modeling. Lecture Notes in Computer Science. 2020. V. 11962. P. 304–315.
18. Fu C.-W., Wan L., Wong T.-T., Leung C.-S. The Rhombic Dodecahedron Map: An Efficient Scheme for Encoding Panoramic Video // IEEE Transactions on Multimedia. 2009. V. 11. № 4. P. 634–644.
19. Kuzuyakov E., Pio D. Next-generation video encoding techniques for 360 video and VR. <https://engineering.fb.com/2016/01/21/virtual-reality/next-generation-video-encoding-techniques-for-360-video-and-vr/>.
20. McCutchen D. A dodecahedral approach to immersive imaging and display // Computer Graphics. 1997. P. 35–37.
21. McCutchen D. Immersive dodecahedral video viewing system. US Patent, No. 5703604. Dec. 30, 1997.
22. Timokhin P., Mikhaylyuk M., Panteley K. A technology to synthesize 360-degree video based on regular dodecahedron in virtual environment systems // Proceed-

- ings of the 30th International Conference on Computer Graphics and Vision. 2020. V. 2744. <https://doi.org/10.51130/graphicon-2020-2-3-1>.
23. Хулл Ф. OpenGL. Программирование компьютерной графики. 2-е издание. Питер, 2002.
  24. Zwillinger D. CRC Standard Mathematical Tables and Formulas. 33rd ed. CRC Press, 2018.
  25. Vasilieva V.N. Golden Section and Golden Rectangles When Building Icosahedron, Dodecahedron and Archimedean Solids Based On Them // Geometry & Graphics. 2019. V. 7. № 2. P. 47–55.
  26. Akhtaruzzaman M., Shafie A.A. Geometrical Substantiation of Phi, the Golden Ratio and the Baroque of Nature, Architecture, Design and Engineering // International Journal of Arts. 2011. V. 1. № 1. P. 1–22.
  27. Segal M., Akeley K. The OpenGL Graphics System: A Specification. Version 4.6, Core Profile. The Khronos Group Inc., 2006–2018. <https://www.khronos.org/registry/OpenGL/specs/gl/glspec46.core.pdf>.
  28. Михайлюк М.В., Тимохин П.Ю., Мальцев А.В. Метод тесселяции на GPU рельефа Земли для космических видеотренажеров // Программирование. 2017. № 4. С. 39–47.

## ОБОБЩЕННЫЙ ВЫЧИСЛИТЕЛЬНЫЙ ЭКСПЕРИМЕНТ И ЗАДАЧИ ВЕРИФИКАЦИИ

© 2021 г. А. К. Алексеев<sup>а,\*</sup>, А. Е. Бондарев<sup>а,\*\*</sup>,  
В. А. Галактионов<sup>а,\*\*\*</sup>, А. Е. Кувшинников<sup>а,\*\*\*\*</sup>

<sup>а</sup> *Институт прикладной математики им. М.В. Келдыша РАН,  
Миусская пл., 4, Москва, 125047, Россия*

*\*E-mail: aleksey.k.alekseev@gmail.com*

*\*\*E-mail: bond@keldysh.ru*

*\*\*\*E-mail: vlgal@gin.keldysh.ru*

*\*\*\*\*E-mail: kuvsh90@yandex.ru*

Поступила в редакцию 25.12.2020 г.

После доработки 14.01.2021 г.

Принята к публикации 18.01.2021 г.

Данная работа рассматривает построение обобщенного вычислительного эксперимента для решения задач верификации. Проблема сравнительной оценки точности численных методов в настоящее время приобретает особую актуальность ввиду введения федеральных стандартов и распространению программных пакетов, включающих большое количество разнообразных солверов. Обобщенный вычислительный эксперимент позволяет получить численное решение для класса задач, определяемых диапазонами изменения определяющих параметров. Анализ результатов, представленных в виде многомерных массивов, где количество измерений определяется размерностью пространства определяющих параметров, требует применения инструментов научной визуализации и визуальной аналитики. Обсуждаются подходы к применению обобщенного вычислительного эксперимента при наличии эталонного решения и в его отсутствие. Приведен пример построения поверхностей ошибок при сравнении решателей программного пакета OpenFOAM. В качестве основной используется классическая задача невязкой косой ударной волны. Рассмотрены вариации основных параметров задачи — числа Маха и угла атаки. Также рассматривается пример задачи обтекания конуса под углом атаки с изменяющимся числом Маха, углом конуса и углом атаки. Вводится понятие индекса ошибки как интегральная характеристика отклонений от точного решения для каждого решателя в рассматриваемом классе задач.

DOI: 10.31857/S013234742103002X

### 1. ВВЕДЕНИЕ

На протяжении всей истории развития вычислительной математики и математического моделирования проблемы верификации численных методов занимали особое место. При создании или модификации численного метода авторам необходимо было показать эффективность своих разработок и оценить их точность. Этим исследованиям посвящено огромное количество работ. В качестве примера можно указать на классические работы [1–19], рассматривающие различные аспекты оценки точности и верификации. Оценка точности численного метода была и по сей день является обязательной частью исследований в области математического моделирования физических процессов.

Как правило, сравнение численных результатов проводится с некоторым референтным (эталонным) решением, в роли которого используют

точное решение, если таковое имеется, или имеющиеся экспериментальные данные. Отдельной проблемой является оценка точности численных методов при отсутствии эталонного решения.

В практике физического эксперимента результаты, не имеющие полосы разброса ошибок, не считаются полными и, как правило, не публикуются. Напротив, в практике вычислительного эксперимента (особенно в задачах CFD) результаты, содержащие информацию о погрешности расчета, в настоящее время являются скорее исключением. Нельзя сказать, что методов оценки погрешности аппроксимации вообще нет, проблема заключается в другом. Довольно давно (1986 г.) в статье [1] было перечислено тринадцать способов оценки ошибки. Сейчас их точно существенно больше, но такое количество методов неявно намекает на отсутствие одного точного, надежного и экономичного метода.

Об актуальности проблем верификации численных методов и расчетов на их основе также свидетельствует наличие федеральных стандартов, как зарубежных [20, 21], так и недавно появившихся российских [22].

Следует отметить, что в настоящее время актуальность задач верификации неуклонно возрастает в связи с широким использованием открытых и коммерческих пакетов для решения различных задач математического моделирования. Как правило, такие пакеты предоставляют пользователю определенный набор численных методов, представленных в виде солверов, интегрированных в программный комплекс. В этом случае пользователь сталкивается с проблемой выбора солвера. И здесь обнаруживается ряд трудностей. Не все солверы проходят всестороннее тестирование перед внедрением в программный комплекс. Коммерческие пакеты не предоставляют полной открытой информации о реализованных численных методах и их свойствах. Различные группы разработчиков могут добавлять солверы в пакеты с открытым исходным кодом, но зачастую не могут обеспечить полное тестирование. Поэтому исследования в области верификации и сравнительной оценки точности численных методов становятся все более необходимыми.

Исторически верификация в задачах вычислительной аэрогазодинамики состояла из двух основных этапов. На первом этапе предполагалось адекватное воспроизведение качественной картины течения, содержащей неоднородности, отрывные потоки, вихри и т.д. Второй этап предполагал обеспечение точности расчета количественных характеристик. Здесь необходимо было полагаться на некоторое эталонное решение — экспериментальное, точное или полученное в результате расчетов другими численными методами.

Подобная верификация обычно проводилась для одной отдельной задачи. По умолчанию предполагалось, что при небольшом изменении определяющих параметров задачи (скорость, вязкость, масштабы времени, теплофизические характеристики среды, геометрические параметры) рассматриваемый численный метод будет применим и обеспечит аналогичную точность.

На современном этапе исследователям необходимы более полные оценки точности численных методов. Например, при оценке точности не для отдельной задачи, а для класса задач. Под классом задач понимается базовая задача, рассматриваемая в диапазонах изменения набора ключевых параметров. Такими параметрами в вычислительной аэродинамике могут служить характеристические числа, определяющие скорость потока, вязкость, теплофизические свойства среды, геометрические параметры и т.д. Возможность получения решения для класса задач обеспечивается путем построения

обобщенного вычислительного эксперимента. Кроме того, обобщенный вычислительный эксперимент может быть очень полезен при оценке точности при отсутствии эталонного решения. В этом случае можно оценить точность, используя ансамбль решений, полученных различными численными методами. Вариация солвера рассматривается как параметр, а параметрическая задача решается с помощью обобщенного вычислительного эксперимента.

При построении обобщенного вычислительного эксперимента его неотъемлемой составляющей являются инструменты научной визуализации и визуальной аналитики. Без этих инструментов невозможно провести полноценный анализ результатов, представляющих собой многомерные объемы данных. В тех случаях, когда размерность пространства определяющих параметров не превышает 3, достаточно разработанных инструментов научной визуализации. Когда размерность пространства определяющих параметров больше 3, необходимо применять комплекс инструментов визуальной аналитики. Анализ результатов обобщенного вычислительного эксперимента порождает ряд специфических задач визуализации. Классификация подобных задач, методы и инструменты их решения подробно описаны в работе [31].

Необходимо заметить, что визуальное представление результатов обобщенного вычислительного эксперимента в ряде практических случаев приводит к созданию емких и выразительных образов, позволяющих полноценный анализ физического явления в классе задач. Это в полной мере относится и к задачам сравнительной оценки точности численных методов. Ниже будет приведен пример построения поверхностей ошибок для класса задач о косом скачке уплотнения, позволяющий эффективно оценить и сравнить точность численных методов, реализованных в 4-х солверах открытого программного пакета OpenFOAM.

Концепция, основные методы и подходы к построению обобщенного вычислительного эксперимента, а также ряд программных средств для его реализации разработаны в Институте прикладной математики им. М.В. Келдыша РАН. Основные аспекты построения обобщенного вычислительного эксперимента и примеры его реализации подробно описаны в [28–35, 37–45].

## 2. ОБОБЩЕННЫЙ ВЫЧИСЛИТЕЛЬНЫЙ ЭКСПЕРИМЕНТ

Возникновение концепции обобщенного вычислительного эксперимента связано с развитием высокопроизводительных вычислительных кластеров и параллельных технологий. В задачах вы-

числительной аэродинамики параллельные технологии обычно предоставляют возможность быстрого расчета на детальных сетках. Однако параллельные технологии предоставляют нам еще одну важную возможность. Это возможность одновременно рассчитывать на разных узлах одну и ту же задачу с разными входными данными. Как правило, такой расчет выполняется в режиме многозадачности.

Это открывает возможность проведения обобщенного вычислительного эксперимента. Ключевые параметры рассматриваемой задачи разбиваются на определенные диапазоны с определенным шагом, образуя сеточное разбиение многомерного бокса в многомерном пространстве ключевых параметров. Основная задача решается с использованием параллельных технологий в каждой точке раздела сетки. Полученные результаты представляют собой многомерные объемы данных. Обработка, анализ и визуальное представление этих данных осуществляется с помощью методов визуальной аналитики и научной визуализации. Эта вычислительная технология является наиболее общим описанием обобщенного вычислительного эксперимента.

Очевидно, что такую концепцию можно применить к широкому кругу задач. В этот диапазон входят параметрические исследования, задачи оптимизации. Обобщенный вычислительный эксперимент является эффективным инструментом для решения обратных задач.

Большое количество различных приложений обобщенного вычислительного эксперимента подробно описано в [28–35, 37–45]. Концепция обобщенного вычислительного эксперимента применялась к широкому кругу как модельных, так и практических задач.

Эти задачи включают анализ взаимодействия вязкого сверхзвукового потока со струйной преградой, течения в следе за телом, задачи взаимодействия струй, задачу обтекания конуса под углом атаки, задачу косых ударных волн и многие другие. Подход к построению обобщенного вычислительного эксперимента был применен к задаче поиска оптимальной с точки зрения силовых нагрузок трехмерной формы узла лопастей в силовой установке [44].

Также этот подход был применен к задачам верификации численных методов. Подробный сравнительный анализ ряда солверов открытого программного пакета OpenFOAM [36] проведен в [32–35, 41, 42]. В качестве базовых задач использовались задачи, имеющие эталонное решение (точное решение или экспериментальные данные). К таким задачам относятся задача о сверхзвуковом невязком обтекании конуса под углом атаки и задача об образовании наклонной ударной волны. В обоих случаях рассматривался класс задач,

образованный вариациями основных параметров рассматриваемой задачи.

### 3. ОЦЕНКА ТОЧНОСТИ НА АНСАМБЛЕ РЕШЕНИЙ

Отдельную сложную задачу представляет собой оценка точности численных методов при отсутствии эталонного решения.

Современные стандарты в области аэрогазодинамики [20–22] требуют верификацию программного обеспечения (ПО) и численных решений, что подразумевает оценку погрешности аппроксимации. Существует огромное количество методов, позволяющих тем или иным способом оценить погрешность аппроксимации [1–19, 23–27]. Часть из них являются интрузивными [13–15, 19], то есть требуют модификации кода программы. Например, учитывают погрешности усечения (дифференциального приближения [14]) или перенос погрешности аппроксимации [15]. Другие методы являются неинтрузивными, но зачастую требуют расчета многосеточных асимптотик, что приводит к огромным требованиям по памяти и быстродействию [24, 25].

В соответствии с современными стандартами основными методиками верификации являются сравнение с эталонным по точности (или аналитическим) решением и анализ асимптотик решения, возникающих при последовательном измельчении расчетной сетки.

К сожалению, количество известных аналитических решений для задач аэрогазодинамики очень невелико, в связи с этим верификация конкретного решения возможна только в очень редких случаях и лишь частично, так как практические задачи содержат, как правило, много различных структур, которые между собой взаимодействуют в неравномерном поле течения. В связи с этим сравнение практических расчетов с имеющимися точными аналитическими решениями (такими, как скачки уплотнения, пограничные слои и т.д.) будет содержать большие погрешности, превышающие погрешность аппроксимации. Верификация ПО также не сводится к верификации элементарных решений (хотя и включает ее). Например, точный расчет уединенного скачка и пограничного слоя никак не гарантирует точного расчета взаимодействия скачков в тройной точке или взаимодействия скачка и пограничного слоя.

В отсутствие эталонного решения оценка погрешности аппроксимации конкретного решения представляет нетривиальную и интересную задачу, которую мы вкратце обсудим.

В качестве альтернативы использованию эталонного решения современными стандартами предполагается использовать сходимость решения на последовательности измельчающихся се-

ток. В части стандартов и большинстве практических приложений решение останавливают по “слипанию” (когда некоторые ценные функционалы не изменяются при измельчении сетки). Согласно [3] этот подход восходит к работам С. Runge. Фактически, при таком подходе решение на самой мелкой сетке воспринимается как точное, что дает оценку погрешности расчета. Экстраполяция Ричардсона [23–25] использует несколько последовательных решений и априорную информацию о порядке сходимости (предполагается, что он соответствует порядку аппроксимации). Для линейных задач и достаточно гладких решений (описываемых, например, эллиптическими уравнениями) этот подход вполне реализуем. К сожалению, для течений с разрывами порядок сходимости зависит от элемента течения (например, скачка уплотнения) и является локальными по пространству [11, 12]. Даже в конечно-элементных задачах для эллиптических уравнений сходимость на разрывах может отличаться от номинального порядка аппроксимации [11]. В такой ситуации рекомендуется использовать обобщенную экстраполяцию Ричардсона (generalized Richardson extrapolation) [24, 25], в которой определяются, как асимптотический диапазон сходимости, так и локальный порядок сходимости. Это подразумевает использование минимум четырех последовательных сеток (встречаются работы, где из-за медленного выхода на асимптотический режим использовалось семь последовательных сеток), что делает этот подход практически нереализуемым в связи с огромными требованиями к вычислительным ресурсам. Дополнительную трудность составляют неустойчивости, возникающие при оценке локального порядка сходимости.

Обобщенный вычислительный эксперимент предоставляет возможности для оценки погрешности аппроксимации за счет перехода от единственного решения к ансамблю численных решений, выполненных различными численными алгоритмами на одной сетке [37–39]. Не следует сводить данный подход к *hp* методу, используемому в конечно-элементном анализе, так как, например, порядок сходимости решения на скачке уплотнения практически одинаков для схем разного формального порядка аппроксимации. Анализ расстояний между численными решениями, принадлежащими анализируемому ансамблю, позволяет частично упорядочить решения по их точности (по крайней мере, разбить на подгруппы “точных” и “неточных” решений). Это позволяет определить норму погрешности аппроксимации и гиперсферу в пространстве сеточных функций, содержащую проекцию точного решения. Расширение способов апостериорной оценки погрешности потенциально способно дать и более существенную информацию о точном решении, в частности связанную с его существованием [26, 27].

В работах [37–39] приведено большое количество вычислительных экспериментов, подтверждающих возможности апостериорной оценки погрешности аппроксимации на ансамбле численных решений, полученных методами различной внутренней структуры. Нельзя исключить, что переход от последовательности решений, полученных на измельчающихся сетках единственным алгоритмом, к ансамблю решений, полученных на одной сетке разными алгоритмами, соответствует некоторому альтернативному пониманию численного решения. В рамках этого понимания вместо последовательности численных решений, сходящихся к точному, мы имеем дело со сферами, построенными вокруг некоторого приближенного решения и содержащими проекцию точного. В некоторой мере этот подход напоминает подход Syngge [6]. Однако в целом данная тематика далеко выходит за рамки данной статьи и нуждается в существенной дополнительной проработке.

Потребность в наборе солверов, реализующих различные алгоритмы, является той платой, которая требуется за использование данного подхода. К счастью, в последнее время достаточно большое число алгоритмов находится в распоряжении вычислителей, например открытый программный комплекс OpenFOAM [36], что заметно облегчает указанное затруднение. В качестве дополнительного бонуса можно указать на естественные параллельные свойства подхода, основанного на ансамбле решений.

#### 4. СРАВНИТЕЛЬНАЯ ОЦЕНКА ТОЧНОСТИ С ИСПОЛЬЗОВАНИЕМ ЭТАЛОННОГО РЕШЕНИЯ

В этом разделе представлены два примера построения обобщенного вычислительного эксперимента для сравнительной оценки точности численных методов. В качестве примеров мы используем численные результаты, подробно описанные в работах авторов [32, 35, 41, 42]. В этих работах были рассмотрены два класса задач вычислительной газовой динамики.

Первый класс задач описывает обтекание конуса под углом атаки. Второй класс описывает падение невязкого сверхзвукового потока газа на плоскую пластину под углом атаки. Обе эти проблемы достаточно хорошо известны. Первая задача имеет табличное решение [51], используемое в качестве эталонного. Вторая проблема имеет точное решение.

Рассмотрим первый класс задач. Решается задача об обтекании конуса сверхзвуковым потоком газа под углом атаки. Переменными параметрами являются угол атаки  $\alpha = 0^\circ, 5^\circ, 10^\circ$ , число Маха  $M = 3, 5$  и угол полураствора конуса  $\beta = 10^\circ, 15^\circ, 20^\circ$ . Схема потока представлена на рис. 1.

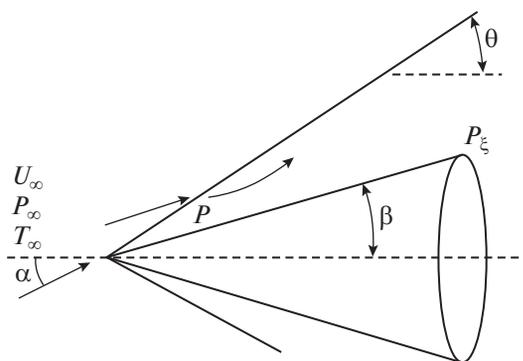


Рис. 1. Схема потока.

Условия входящего потока на входе обозначаются индексом “ $\infty$ ”, а на выходе — индексом  $\xi$ , так как решение является автомодельным и зависит от безразмерной переменной.

Для расчета используется система уравнений Эйлера, дополненная уравнением состояния идеального газа.

Из программного пакета OpenFOAM были выбраны три солвера: rhoCentralFoam (rCF), sonicFoam (sF) и pisoCentralFoam (pCF). Солвер rhoCentralFoam основан центрально-противопотоковой схеме, которая является комбинацией центрально-разностной и противопотоковой схем [46, 47]. Солвер sonicFoam основан на алгоритме PISO (Pressure Implicit with Splitting of Operator) [48]. Здесь для расчета давления используются два разностных уравнения для поправки поля давления, полученных из дискретных аналогов уравнений моментов и неразрывности. Такой подход связан с тем, что скорректированные первой поправкой скорости, могут не удовлетворять уравнению неразрывности, поэтому вводится второй корректор, который позволяет вычислить скорости и давления, удовлетворяющие линейризованным уравнениям количества движения и неразрывности. Солвер pisoCentralFoam представляет собой комбинацию центрально-противопотоковой схемы с алгоритмом PISO [49]. Этот солвер не входит в стандартный набор солверов программного пакета OpenFOAM. Он создан независимой командой разработчиков Института системного программирования им. В.П. Иванникова РАН. Все расчеты проводились с использованием OpenFOAM версии 2.3.0.

Мы решали задачу с помощью каждого из солверов для всего набора варьируемых параметров. Таким образом был получен набор численных решений. Точное решение было получено путем интерполяции табличного решения из [36]. Затем для каждого солвера была найдена погрешность решения в нормах L1 и L2. Поскольку различные солверы реализуют разные численные методы, ошибки заметно отличались друг от друга. На-

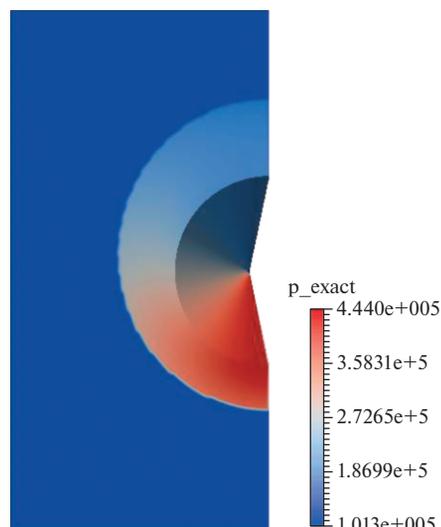


Рис. 2. Поле давления для стационарного течения.

чальные и граничные условия, а также настройки решателей задавались аналогично [38, 40].

На рис. 2 показано стационарное решение для поля давления, полученное интерполяцией табличного решения из [51], угол полураствора конуса  $\beta = 20^\circ$ , угол атаки  $\alpha = 10^\circ$ , число Маха  $M = 3$ .

На рис. 3 показана поверхность ошибки (отклонения от точного решения) в норме L2 для этой задачи при вариации выбора солвера и угла полураствора для фиксированного числа Маха 3 и фиксированного угла атаки  $5^\circ$ . Видно, что отклонение от точного решения увеличивается с ростом угла полураствора. Также видно, что солверы rhoCentralFoam и pisoCentralFoam примерно одинаковы по точности, тогда как точность солвера sonicFoam намного ниже.

На рис. 4 показана поверхность ошибки для этой же задачи с вариацией солвера и угла атаки при числе Маха 5 и угле полураствора конуса  $15^\circ$ .

Таким образом, анализируя рис. 3 и 4, можно увидеть, что для всех солверов ошибка увеличивается с увеличением угла атаки и угла полураствора конуса.

Итак, в результате у нас есть оценка точности для всех трех солверов, участвующих в этом исследовании. Это результат построенного обобщенного численного эксперимента для рассматриваемого класса задач.

Рассмотрим другой класс задач. В данном случае мы рассматриваем известную задачу о формировании наклонной ударной волны.

Постановка задачи описывается следующим образом. Сверхзвуковой поток газа падает на пластину под углом. Отражаясь от пластины, поток образует наклонную ударную волну. Эта задача имеет точное решение. В задаче варьировались в

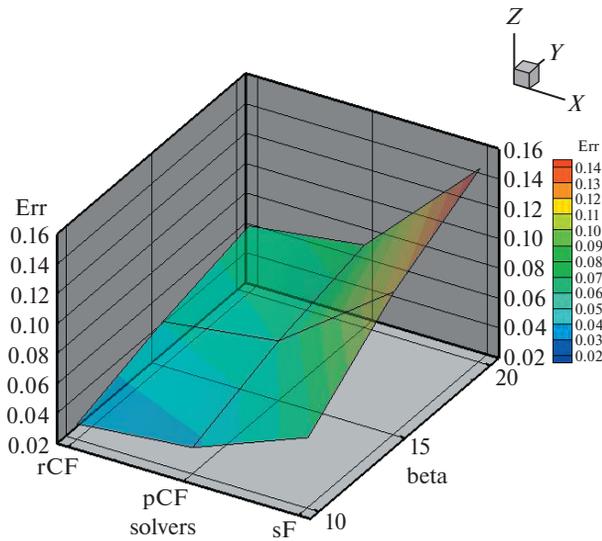


Рис. 3. Изображение поверхности отклонения от точного решения для числа Маха 3 и угла атаки  $5^\circ$  с вариацией солвера и угла полураствора конуса.

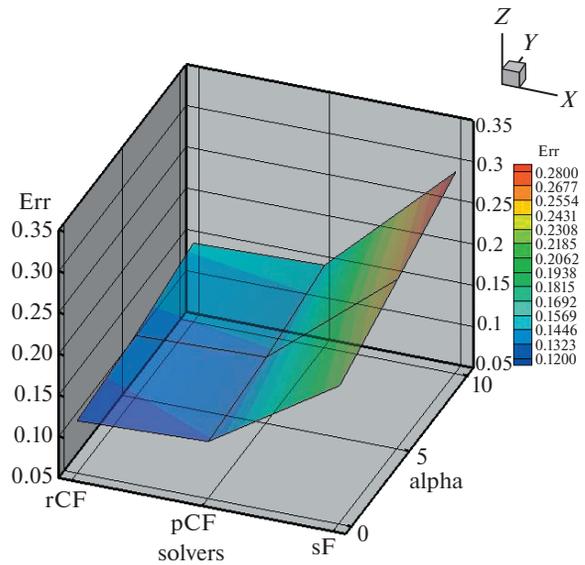


Рис. 4. Изображение поверхности отклонения от точного решения для числа Маха 5 и полуугла конуса  $15^\circ$  при вариации солвера и угла атаки.

определенных диапазонах число Маха  $M$  и угол падения сверхзвукового потока  $\beta$ , аналогично [41, 42]. На рис. 5 показаны поверхности ошибок для этой задачи, построенные для 4 солверов: rhoCentralFoam (rCF), sonicFoam (sF), pisoCentralFoam (pCF), QGDFoam (QGDF). Здесь мы включаем в рассмотрение новый солвер QGDFoam (QGDF). Этот солвер основан на системе квазигидродинамических уравнений и также создан независимыми разработчиками [50].

Проведение аналогичных расчетов для нескольких численных методов, реализованных в солверах открытого программного пакета OpenFOAM, позволяет построить несколько таких поверхностей на одном рисунке. Это открывает возможность для глубокого и четкого сравнительного анализа точности исследуемых численных методов. Построение обобщенного вычислительного эксперимента предполагает создание вычислительной техники от решения прямой задачи до визуального анализа результатов. Это пример построения яркого и выразительного образа для решения в классе задач. Подобный визуальный образ позволяет получить мгновенное и полноценное представление о точности сравниваемых численных методов не для одной, отдельно взятой задачи, а для целого класса задач, определяемых диапазонами вариации определяющих параметров, в данном случае — числа Маха и угла атаки.

Эта методика позволяет провести подробное визуальное сравнение отклонений от точного решения. Видно, что в нашем случае все поверхности ошибок меняются одинаково. Погрешность увеличивается с ростом основных параметров. Наилучшую точность в этом классе задач обеспе-

чивают решатели rCF и pCF, для которых поверхности ошибок практически идентичны. Таким образом, построение обобщенного вычислительного эксперимента позволяет провести полноценную сравнительную оценку точности четырех солверов программного комплекса OpenFOAM в классе задач. Класс задач в данном конкретном случае определяется основной задачей (косая ударная волна) и диапазонами изменения ключевых параметров задачи — числа Маха и угла атаки.

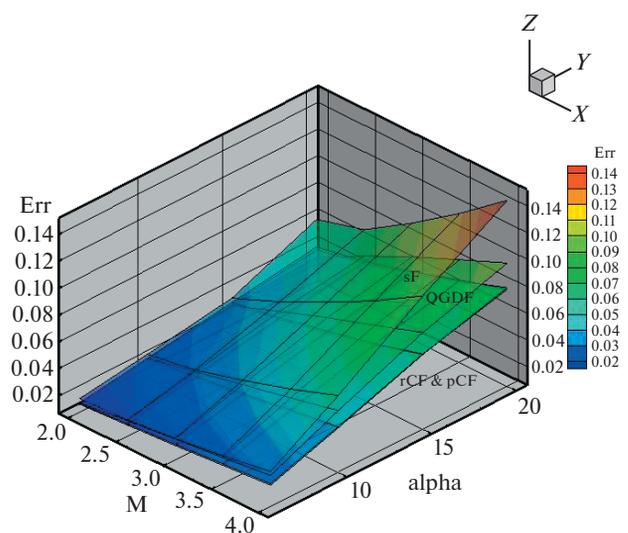


Рис. 5. Поверхности погрешности при вариации числа Маха и угла атаки для задачи о наклонной ударной волне [41].

**Таблица 1.** Значения индекса погрешности для задачи обтекания конуса под углом атаки

Solver	rCF	pCF	sF
Error Index	0.089811	0.095587	0.182798

**Таблица 2.** Значения индекса погрешности для задачи образования косоугольного скачка уплотнения

Solver	rCF	pCF	QGDF	sF
Error Index	0.037734	0.038751	0.0453406	0.058216

Изображение поверхностей ошибок, представленное на рис. 5, дает достаточное представление о сравнительной точности солверов OpenFOAM в классе задач. Однако для более полной оценки мы вводим интегральную характеристику для каждой поверхности. Мы будем называть эту характеристику индексом погрешности (EI – Error Index). Индекс погрешности определяется следующим образом. Пусть  $i = 1, M$  и  $j = 1, N$  – точки сеточного разбиения по ключевым параметрам, а  $A_{ij}$  – отклонение от точного решения в каждой точке сеточного разбиения области пространства параметров. Тогда индекс погрешности определяется как:

$$EI = \sum_{ij} A_{ij} / (M \times N)$$

Сначала мы вычисляем индекс погрешности для задачи обтекания конуса при угле атаки.

Далее рассчитываем значения индекса погрешности для задачи образования косоугольного скачка уплотнения.

Таблицы 1 и 2 показывают, что значения индекса погрешности EI полностью соответствуют взаимному расположению численных результатов, представленных на рис. 3–5. Следовательно, рассчитанный индекс погрешности может служить характеристикой точности численных методов в выбранный класс задач.

## 5. ЗАКЛЮЧЕНИЕ

Рассмотрено применение обобщенного вычислительного эксперимента к задачам сравнительной оценки точности численных методов. Обобщенный вычислительный эксперимент позволяет одновременно рассчитывать одну и ту же задачу с разными входными данными на основе параллельных технологий в режиме многозадачности. Полученные многомерные результаты исследуются с помощью инструментов визуального анализа. Обобщенный вычислительный эксперимент является вычислительной технологией, основанной на синтезе решения задач математического моделирования, параллельных вычислений и анализа

многомерных данных с помощью методов научной визуализации и визуальной аналитики.

Приведены два примера построения обобщенного вычислительного эксперимента для следующих классов задач – обтекание конуса под углом атаки и формирование косой ударной волны. Для обоих случаев класс задач формируется на основе базовой задачи и вариаций определяющих параметров задачи в определенных диапазонах. Для обоих классов задач проведена сравнительная оценка точности решателей программного пакета OpenFOAM. Приведен пример построения поверхностей ошибок. Вводится понятие индекса погрешности численного метода для класса задач. Построение обобщенного вычислительного эксперимента позволило реализовать визуальное представление для класса задач, обеспечивающее эффективное проведение сравнительной оценки точности численных методов.

Построение обобщенного вычислительного эксперимента может служить эффективным инструментом для решения задач верификации.

## СПИСОК ЛИТЕРАТУРЫ

1. *Skeel R.* Thirteen ways to estimate global error // *Numerische Mathematik*. 1986. V. 48. P. 1–20.
2. *Roy C.J., Oberkampf W.L.* A comprehensive framework for verification, validation, and uncertainty quantification in scientific computing // *Computer Methods in Applied Mechanics and Engineering*. 2011. V. 200. № 25–28. P. 2131–2144.
3. *Repin S.* A posteriori estimates for partial differential equations. Walter de Gruyter., 2008. V. 4.
4. *Repin S.* A unified approach to a posteriori error estimation based on duality error majorants // *Mathematics and Computers in Simulation*. 1999. V. 50. № 1–4. P. 305–321.
5. *Repin S., Frolov M.* A posteriori error estimates for approximate solutions of elliptic boundary value problems // *Computational Mathematics and Mathematical Physics*. 2002. V. 42. № 12. P. 1704–1716.
6. *Syngde J.L.* The Hypercircle in Mathematical Physics. London: CUP., 1957.
7. *Oden J., Prudhomme S.* Goal-oriented error estimation and adaptivity for the finite element method // *Computers and Mathematics with Appl.* 2001. V. 41. P. 735–756.
8. *Prudhomme S., Oden J.* On goal-oriented error estimation for elliptic problems: Application to the control of pointwise errors // *Computer Methods in Applied Mechanics and Engineering*. 1999. V. 176. P. 313–331.
9. *Ainsworth M., Oden J.* A posteriori error estimation in finite element analysis. N.Y.: Wiley–Interscience, 2000.
10. *Carpenter M., Casper J.* Accuracy of shock capturing in two spatial dimensions // *AIAA Journal*. 1999. V. 37. № 9. P. 1072–1079. <https://doi.org/10.2514/2.835>

11. *Babuska I., Osborn J.* Can a finite element method perform arbitrarily badly? // *Mathematics of Computation of the American Mathematical Society.* 2000. V. 69. № 230. P. 443–462.
12. *Godunov S.K., Manuzina Yu.D., Nazareva M.A.* Experimental analysis of convergence of the numerical solution to a generalized solution in fluid dynamics // *Computational Mathematics and Mathematical Physics.* 2011. V. 51. P. 88–95.
13. *Linss T., Kopteva N.* A Posteriori Error Estimation for a Defect-Correction Method Applied to Convection-Diffusion Problems // *Int. J. of Numerical Analysis and Modeling.* 2009. V. 1. № 1. P. 1–16.
14. *Shokin Yu.I.* Method of differential approximation. Springer-Verlag, 1983.
15. *Banks J., Hittinger J., Woodward C.* Numerical error estimation for nonlinear hyperbolic PDEs via nonlinear error transport // *Computer Methods in Applied Mechanics and Engineering.* 2012. V. 213. P. 1–15. <https://doi.org/10.1016/j.cma.2011.11.021>
16. *Rausser F., Marotzke J., Korn P.* Ensemble-type numerical uncertainty quantification from single model integrations // *Journal Comp. Physics.* 2015. V. 292. P. 30–42. <https://doi.org/10.1016/j.jcp.2015.02.043>
17. *Johnson C.* On computability and error control in CFD // *International J. for Numerical Methods in Fluids.* 1995. V. 20. P. 777–788. <https://doi.org/10.1002/flid.1650200806>
18. *Babuska I., Rheinboldt W.* A posteriori error estimates for the finite element method // *International Journal for Numerical Methods in Engineering.* 1978. V. 12. P. 1597–1615. <https://doi.org/10.1002/nme.1620121010>
19. *Roy Ch., Raju A.* Estimation of discretization errors using the method of nearby problems // *AIAA Journal.* 2007. V. 45. № 6. P. 1232–1243. <https://doi.org/10.2514/1.24282>
20. *Guide for the Verification and Validation of Computational Fluid Dynamics Simulations,* American Institute of Aeronautics and Astronautics, AIAA-G-077-1998, Reston, VA, 1998.
21. *Standard for Verification and Validation in Computational Fluid Dynamics and Heat Transfer,* ASME V&V 20-2009, 2009.
22. *Federal standard P 57700.12–2018.* Numerical simulation of supersonic flows for an inviscid gas. Software verification. National standard of the Russian Federation for numerical modeling of physical processes, 2018.
23. *Richardson L.F.* The Approximate Arithmetical Solution by Finite Differences of Physical Problems Involving Differential Equations with an Application to the Stresses in a Masonry Dam // *Transactions of the Royal Society of London, Series A.* 2010. V. 1908. P. 307–357.
24. *Roy Ch.J.* Grid Convergence Error Analysis for Mixed – Order Numerical Schemes // *AIAA Journal.* 2003. V. 41. № 4. P. 595–604.
25. *Phillips Tyrone S., Roy Christopher J.* Richardson Extrapolation-based Discretization Uncertainty Estimation for Computational Fluid Dynamics // *ASME Journal of Fluids Engineering.* 2014. V. 136. № 12. 121401.
26. *Ortner C.* A Posteriori Existence in Numerical Computations // *SIAM Journal on Numerical Analysis.* 2009. V. 47. № 4. P. 2550–2577.
27. *Chernyshenko S.I., Constantin P., Robinson J.C., Titi E.S.* A posteriori regularity of the three-dimensional Navier-Stokes equations from numerical computations // *J. of Mathematical Physics.* 2007. V. 48. 065204.
28. *Bondarev A.* Analysis of space-time flow structures by optimization and visualization methods // *Transactions on Computational Science XIX, Lecture Notes in Computer Science.* 2013. V. 7870. P. 158–168.
29. *Bondarev A., Galaktionov V.* Parametric optimizing analysis of unsteady structures and visualization of multidimensional data // *International Journal of Modeling, Simulation and Scientific Computing* 04 (Supp. 01). 2013.
30. *Bondarev A.* On the construction of the generalized numerical experiment in fluid dynamics // *Mathematica Montisnigri XLII.* 2018. P. 52–64.
31. *Bondarev A.* On visualization problems in a generalized computational experiment // *Scientific Visualization.* 2019. V. 11. № 2. P. 156–162. <https://doi.org/10.26583/sv.11.2.12>
32. *Bondarev A., Kuvshinnikov A.* Analysis of the accuracy of OpenFOAM solvers for the problem of supersonic flow around a cone // *Lecture Notes in Computer Science.* 2018. V. 10862. P. 221–230. [https://doi.org/10.1007/978-3-319-93713-7\\_18](https://doi.org/10.1007/978-3-319-93713-7_18)
33. *Bondarev A.* On the estimation of the accuracy of numerical solutions in CFD problems // *LNCS.* 2019. V. 11540. P. 325–333. [https://doi.org/10.1007/978-3-030-22750-0\\_26](https://doi.org/10.1007/978-3-030-22750-0_26)
34. *Bondarev A., Galaktionov V.* Generalized computational experiment and visual analysis of multidimensional data // *Scientific Visualization.* 2019. V. 11. № 4. P. 102–114. <https://doi.org/10.26583/sv.11.4.09>
35. *Alekseev A., Bondarev A., Galaktionov V., Kuvshinnikov A.* On the construction of a generalized computational experiment in verification problems // *Matematica Montisnigri XLVIII.* 2020. P. 19–31. <https://doi.org/10.20948/mathmontis-2020-48-3>
36. *OpenFOAM Foundation.* <http://www.openfoam.org>. last accessed 2021/01/08.
37. *Alekseev A., Bondarev A.* On exact solution enclosure on ensemble of numerical simulations // *Mathematica Montisnigri XXXVIII.* 2017. P. 63–77.
38. *Alekseev A., Bondarev A., Kuvshinnikov A.* Verification on the ensemble of independent numerical solutions // *LNCS.* 2019. V. 11540. P. 315–324. [https://doi.org/10.1007/978-3-030-22750-0\\_26](https://doi.org/10.1007/978-3-030-22750-0_26)
39. *Alekseev A., Bondarev A.* Estimation of the distance between true and numerical solutions // *Computational Mathematics and Mathematical Physics.* 2019. V. 59. № 6. P. 857–863. <https://doi.org/10.1134/S0965542519060034>
40. *Alekseev A., Bondarev A., Kuvshinnikov A.* On uncertainty quantification via the ensemble of independent numerical solutions // *Journal of Computational Sci-*

- ence. 2020. V. 42. 101114.  
<https://doi.org/10.1016/j.jocs.2020.101114>
41. *Alekseev A., Bondarev A., Kuvshinnikov A.* Comparative analysis of the accuracy of OpenFOAM solvers for the oblique shock wave problem // *Mathematica Montisnigri XLV*. 2019. P. 95–105.  
<https://doi.org/10.20948/mathmontis-2019-45-8>
42. *Bondarev A., Kuvshinnikov A.* Parametric study of the accuracy of OpenFOAM solvers for the oblique shock wave problem // *IEEE Proceedings of the 2019 Ivanikov ISPRAS Open Conference* 2019. P. 108–112.  
<https://doi.org/10.1109/ISPRAS47671.2019.00023>
43. *Bondarev A.E., Galaktionov V.A.* Multidimensional data analysis and visualization for time-dependent CFD problems // *Programming and Computer Software*. 2015. V. 41. № 5. P. 247–252.  
<https://doi.org/10.1134/S0361768815050023>
44. *Andreev S.V., Bondarev A.E., Bondarenko A.V., Vizilter Yu.V., Galaktionov V.A., Gudkov A.V., Zheltov S.Yu., Zhukov V.T., Ilovaiskaya E.B., Knyaz V.A., Manukovskii K.V., Novikova N.D., Ososkov M.V., Silaev N.Zh., Feodoritova O.B.* A Computational Technology for Constructing the Optimal Shape of a Power Plant Blade Assembly Taking into Account Structural Constraints // *Programming and Computer Software*. 2017. V. 43. № 6. P. 345–352.  
<https://doi.org/10.1134/S0361768817060020>
45. *Bondarev A.E., Galaktionov V.A., Kuvshinnikov A.E.* Parallel Solutions of Parametric Problems in Gas Dynamics Using DVM/DVMH Technology // *Programming and Computer Software*. 2020. V. 46. № 3. P. 176–182.  
<https://doi.org/10.1134/S0361768820030032>
46. *Kurganov A., Tadmor E.* New high-resolution central schemes for nonlinear conservation laws and convection-diffusion equations // *Journal of Computational Physics*. 2000. V. 1. P. 241–282.  
<https://doi.org/10.1006/jcph.2000.6459>
47. *Greenshields C., Wellerr H., Gasparini L., Reese J.* Implementation of semi-discrete, non-staggered central schemes in a colocated, polyhedral, finite volume framework, for high-speed viscous flows // *International Journal for Numerical Methods in Fluids*. 2010. V. 63. № 1. P. 1–21.  
<https://doi.org/10.1002/flid.2069>
48. *Issa R.* Solution of the implicit discretized fluid flow equations by operator splitting // *Journal of Computational Physics*. 1986. V. 66. № 1. P. 40–65.  
[https://doi.org/10.1016/0021-9991\(86\)90099-9](https://doi.org/10.1016/0021-9991(86)90099-9)
49. *Kraposhin M., Bovtrikova A., Strijhak S.* Adaptation of Kurganov-Tadmor numerical scheme for applying in combination with the PISO method in numerical simulation of flows in a wide range of Mach numbers // *Procedia Computer Science*. 2015. V. 66. P. 43–52.  
<https://doi.org/10.1016/j.procs.2015.11.007>
50. *Kraposhin M., Smirnova E., Elizarova T., Istomina M.* Development of a new OpenFOAM solver using regularized gas dynamic equations // *Computers & Fluids*. 2018. V. 166. P. 163–175.  
<https://doi.org/10.1016/j.compfluid.2018.02.010>
51. *Бабенко К.И., Воскресенский Г.П., Любимов А.Н., Русанов В.В.* Пространственное обтекание гладких тел идеальным газом. М.: Наука, 1964, 505 с.

## МЕТОД ПРОГРЕССИВНЫХ ОБРАТНЫХ ФОТОННЫХ КАРТ

© 2021 г. А. Д. Жданов<sup>a,\*</sup>, Д. Д. Жданов<sup>a,\*\*</sup><sup>a</sup> Санкт-Петербургский национальный исследовательский университет  
информационных технологий, механики и оптики, 197101  
Санкт-Петербург, Кронверкский пр., д. 49, Россия

\*E-mail: adzhdanov@itmo.ru

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

Поступила в редакцию 19.12.2020 г.

После доработки 25.12.2020 г.

Принята к публикации 14.01.2021 г.

Данная работа посвящена исследованию методов фотонных карт для решения проблемы реалистичного рендеринга. В отличие от традиционных методов рендеринга основой для расчета яркости вторичного и каустического освещений являются обратные фотонные карты или карты наблюдения. Представлены основные преимущества метода обратных фотонных карт, которые заключаются, во-первых, в естественном распределении фотонов в областях, формирующих яркость изображения, а во-вторых, в уменьшении числа фотонов, формируемых на трассе одного луча. Рассмотрена основная алгоритмическая сложность метода обратных фотонных карт, заключающаяся в необходимости синхронизации данных при расчете и накоплении яркости вторичного и каустического освещений. Для решения данной проблемы авторы предлагают использовать промежуточные прямые фотонные карты вторичного и каустического освещения, распределенные по вычислительным потокам, выполняющими рендеринг соответствующих участков изображения. На основе проведенных исследований вводится метод прогрессивных обратных фотонных карт и описывается алгоритм реалистичного рендеринга, основанный на методе прогрессивных обратных фотонных карт. Разработанный алгоритм не требует дополнительной синхронизации при накоплении яркости в точках изображения, что позволяет эффективно реализовать его не только с использованием ресурсов центрального процессора, но и на графическом процессоре. Представлены результаты качественного и количественного сравнения результатов рендеринга методами прогрессивных прямых и обратных фотонных карт.

DOI: 10.31857/S0132347421030110

## 1. ВВЕДЕНИЕ

Реалистичный рендеринг является составной частью систем современной реалистичной визуализации, виртуального прототипирования и виртуальной реальности. Кроме того, он используется при решении широкого круга прикладных задач, включающих в себя формирование реалистичных изображений, моделирование оптических эффектов, виртуальное прототипирование сложных оптических систем и т.д. [1].

Проблема реалистичного рендеринга сводится к расчету значения яркости наблюдаемой поверхности сцены в каждой точке изображения. В 1985 г. James Kajiya предложил [2] метод расчета видимой яркости в виде решения уравнения рендеринга, проиллюстрированного на рис. 1:

$$L(\vec{p}, \vec{v}_r) = \tau(t) \frac{n_r^2}{n_0^2} \times \left( L_0(\vec{p}, \vec{v}_r) + \int \int_{2\pi} L_i(\vec{p}, \vec{v}_i) f(\vec{p}, \vec{v}_i, \vec{v}_r) \cos(\vec{n}, \vec{v}_i) d\omega \right) \quad (1)$$

При разработке алгоритмов реалистичного рендеринга за основу берутся физически-корректные законы распространения и преобразования светового излучения. Поскольку уравнение рендеринга имеет аналитическое решение только для ограниченного количества сочетаний свойств оптических материалов и геометрических форм, был разработан ряд численных методов для решения этого уравнения. Наиболее подходящие методы основаны на стохастической трассировке лучей с использованием методов Монте-Карло. Это методы прямой, обратной и двунаправлен-

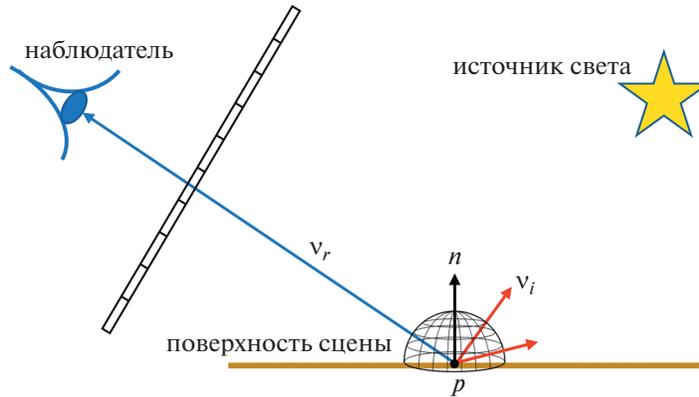


Рис. 1. Расчет значения яркости точки поверхности сцены с помощью уравнения рендеринга.

ной трассировки лучей. Эти методы используют различные решения для расчета компонент яркости прямого виденья ( $L_0(\vec{p}, \vec{v}_r)$ ) и яркостей компонент прямого, каустического и вторичного освещения ( $\iint_{2\pi} L_i(\vec{p}, \vec{v}_i) f(\vec{p}, \vec{v}_i, \vec{v}_r) \cos(\vec{n}, \vec{v}_i) d\omega$ ), создаваемо-

го поверхностью с двунаправленной функцией рассеивания (ДФР)  $f(\vec{p}, \vec{v}_i, \vec{v}_r)$ . Прямое и каустическое освещение представляет собой первый член этой бесконечной рекурсии и разница между ними заключается в том, что каустическое освещение осуществляется через поверхности, не имеющие рассеивающих свойств, или градиентные среды. При этом освещенность поверхности масштабируется на зеркальный коэффициент пропускания от источника света до наблюдаемой поверхности  $\tau(t_o)$ . При наблюдении яркость источника света просто масштабируется на зеркальный коэффициент пропускания  $\tau(t_r)$  до поверхности наблюдения ( $\tau(t_r)L_0(\vec{p}, \vec{v}_r)$ ) – в случае непосредственного наблюдения источника света, или

$(\tau(t_r) \iint_{2\pi} L_i(\vec{p}, \vec{v}_i) f(\vec{p}, \vec{v}_i, \vec{v}_r) \cos(\vec{n}, \vec{v}_i) d\omega)$  – в случае яркости прямого, каустического или вторичного освещения.

Основными методами расчета видимой яркости являются методы прямой, обратной и двунаправленной трассировки лучей [3–5]. Эти методы позволяют получить несмещенное уравнение рендеринга. Однако вычислительная эффективность методов, основанных на исключительно методах прямой или обратной трассировки лучей, может быть крайне низка для сцен, имеющих сложное вторичное и каустическое освещение. Классические методы двунаправленной трассировки лучей могут обеспечить эффективный расчет несмещенного значения яркости вторичного освеще-

ния сцен. Однако для методов двунаправленной трассировки лучей неразрешимой задачей является расчет яркости каустического освещения, особенно в сценах, содержащих источники света малого размера. Поэтому наиболее универсальными методами расчета физически корректного значения яркости вторичного и каустического освещения являются методы, основанные на использовании фотонных карт [6].

В 1996 году Henrik Wann Jensen представил алгоритм расчета глобальной освещенности с помощью фотонных карт. Алгоритм рендеринга с использованием фотонных карт [7] состоит из следующих трех основных шагов: на первом шаге из источников света испускаются лучи, переносящие световой поток  $\Delta\Phi(\vec{v}_i)$ , распространяющиеся по сцене и формирующие распределение фотонов на поверхностях сцены; на втором шаге на основе полученного распределения фотонов формируются фотонные карты, обеспечивающие быстрый доступ к фотонам в заданной области сцены; на последнем шаге наблюдатель считывает распределение яркости с фотонов, наблюдаемых из камеры и покрываемых сферой интегрирования радиуса  $r$ , зависящего от длины пути от камеры до точки сбора яркости. Яркость наблюдения складывается из яркостей, создаваемых всеми фотонами, наблюдаемыми на трассе обратного луча:

$$L_{(id,c)}(\vec{p}, \vec{v}_r) \approx \frac{1}{\pi r^2} \sum_{i=1}^K f(\vec{p}, \vec{v}_i, \vec{v}_r) \Delta\Phi(\vec{v}_i) \quad (2)$$

Данный метод является смещенным, и для снижения ошибки дискретизации, вызванной конечным радиусом сферы интегрирования  $r$ , яркость прямого освещения вычисляется методом многократной выборки по значимости [8] между значениями яркости, вычисленными методами выбора случайных точек на источниках света (находящихся на расстоянии  $d_i$  до точки наблюдения и имеющих веса  $w_{i,1}$ ), и случайным выбором на-

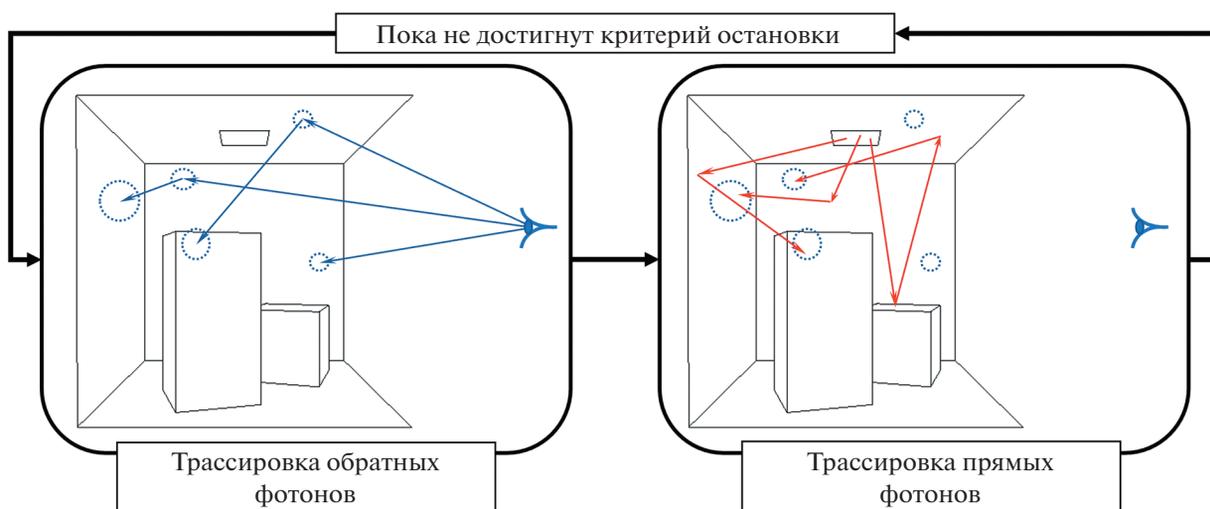


Рис. 2. Метод прогрессивных обратных фотонных карт.

правлений рассеивания на поверхности (имеющих веса  $w_{i,2}$ ):

$$L_d(\vec{p}, \vec{v}_r) = \frac{1}{N} \sum_{i=1}^N \left( w_{i,1} \cdot f(\vec{p}, \vec{v}_i, \vec{v}_r) \cdot \frac{I(\vec{v}_i) \cos(\vec{n}_i, \vec{v}_i)}{d_i^2} + w_{i,2} \cdot L(\vec{p}, \vec{v}_i) \right) \quad (3)$$

Основными недостатками данного метода являются его смещенность и необходимость сохранения истории трасс лучей, т.е. состояния фотонов, на всех диффузных поверхностях сцены. Для “замкнутых светлых” сцен это может привести к значительному увеличению памяти, требуемой для хранения фотонных карт, и замедлению рендеринга, связанного с обработкой большого объема данных.

Метод прямых фотонных карт и метод двунаправленной трассировки лучей получили дальнейшее расширение в методе соединения и слияния вершин [9] (Vertex Connection and Merging), который призван объединить метод фотонных карт и двунаправленной трассировки лучей. В этом методе прямые фотонные карты используются для формирования путей для учета каустической составляющей освещения. Данный метод частично снимает проблему смещенности, однако проблема большого объема фотонных карт сохраняется.

Настоящее исследование посвящено анализу методов расчета глобального освещения и реалистичного рендеринга с использованием фотонных карт. В рамках проведенного исследования для вычисления физически корректных значений яркости вторичного и каустического освещений авторы предлагают использовать обратные фотонные карты [10, 11], формируемые стохастическими лучами, испускаемыми из камеры в сцену.

В данной работе рассматриваются реализация метода прогрессивных обратных фотонных карт и методов распараллеливания вычислений для эффективного расчета яркостей вторичного и каустического освещений.

## 2. ОБРАТНЫЕ ФОТОННЫЕ КАРТЫ

Метод прогрессивных обратных фотонных карт практически зеркален методу прогрессивных прямых фотонных карт. Единственное отличие – это то, что вместо карты освещения строится карта наблюдения. Однако это отличие приводит к существенным изменениям алгоритмов трассировки лучей и накопления яркостей вторичного и каустического освещений. На рис. 2 представлена основная идея метода рендеринга с использованием прогрессивных обратных фотонных карт.

С точки зрения расчета яркости вторичного и каустического освещения принципиальная разница между сферами интегрирования, построенными на трассах прямых лучей, и сферами интегрирования, построенными на трассах обратных лучей, отсутствует. Значение радиуса сферы интегрирования – это область, определяющая ошибку дискретизации, для которой поток, испущенный фотоном, преобразуется в освещенность. Радиус этой области выбирается пропорционально размеру пикселя, видимому на расстоянии от наблюдателя до этой области (точки расчета яркости). Коэффициент пропорциональности определяется в зависимости от компоненты яркости, для которой производится расчет. Так для расчета яркости прямого освещения данный коэффициент не превышает 1, для расчета яркости каустического освещения данный коэффициент варьируется от 2 до 1, а для расчета яркости вторичного освеще-

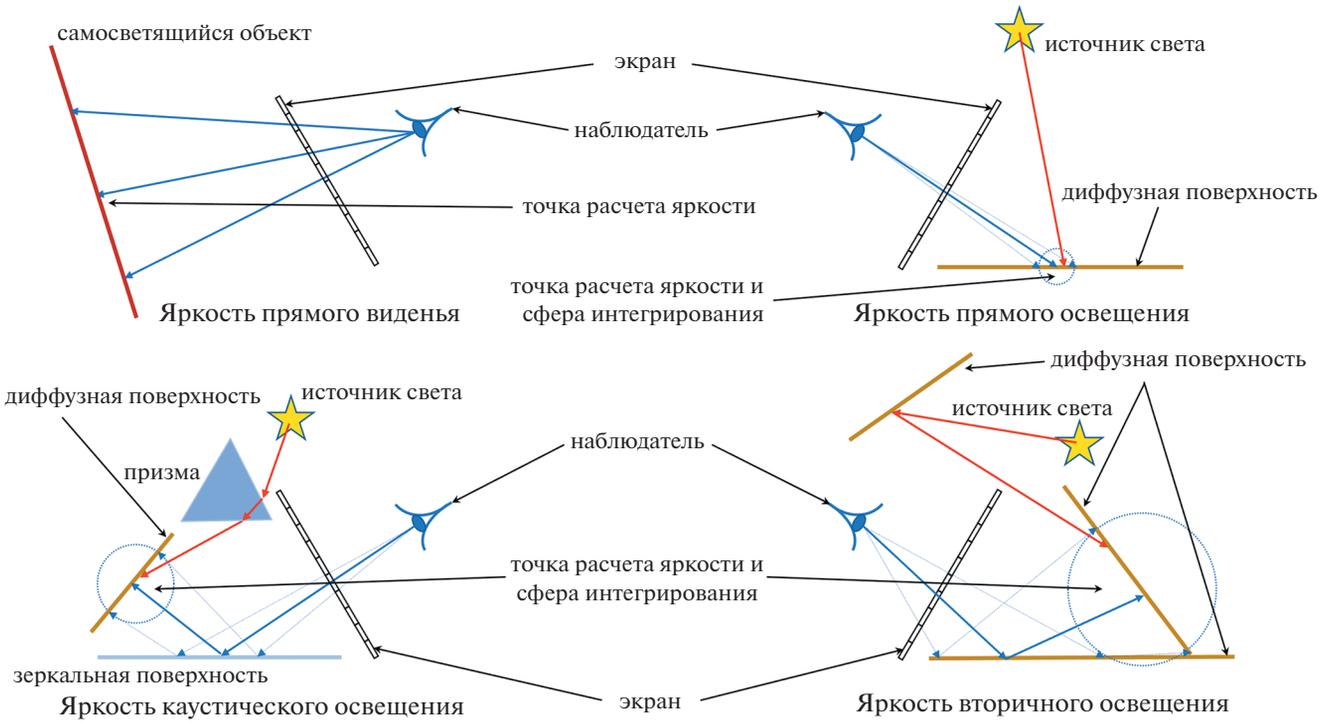


Рис. 3. Выбор радиуса сферы интегрирования значения яркости в зависимости от компоненты освещения.

ния этот коэффициент может достигать 10. Кроме того, данный коэффициент пропорциональности может варьироваться в процессе вычислений (как правило, он уменьшается в процессе выполнения итераций или устанавливается в зависимости от локальной плотности распределения фотонов). Выбор значения радиуса сферы интегрирования в зависимости от рассчитываемой компоненты освещения проиллюстрирован на рис. 3. Поскольку для прямых фотонных карт радиус сферы интегрирования заранее не известен, фотонные карты строятся для множества точек (координат световых фотонов на поверхностях сцены), которые потом пересекаются сферой интегрирования, формируемой на обратном луче. В случае формирования обратных фотонных карт радиус сферы интегрирования каждого из обратных фотонов известен заранее, что позволяет строить обратную фотонную карту для множества сфер, а не координат точек. Замена точек на сферы интегрирования в фотонных картах позволяет значительно ускорить процесс нахождения фотонов, покрываемых сферой интегрирования.

При этом является очевидным, что в общем случае для расчета значения яркости компоненты прямого видения нет необходимости в формировании фотонной карты. Однако, в ряде случаев фотонная карта для расчета яркости компоненты прямого освещения является эффективной и формируется только в случае, если сцена содержит источники света, которые не имеют методов

расчета яркости и интенсивности излучения, или число источников света велико. Примером источника света, не имеющего методов расчета яркости и интенсивности, может служить источник света, заданный набором лучей, который формируется в результате измерений реальных источников света или в результате моделирования сложных осветительных систем [12].

При использовании обратных фотонных карт и методов обратной стохастической трассировки лучей также необходимо учесть специфику моделирования процесса рассеивания обратных лучей на диффузных поверхностях сцены. Если поверхность имеет двунаправленную функцию рассеивания  $f(\vec{p}, \vec{v}_i, \vec{v}_r)$ , то в соответствии с принципом обратимости Геймгольца [13] должно выполняться следующее тождество:

$$f(\vec{p}, \vec{v}_i, \vec{v}_r) = f(\vec{p}, \vec{v}_r, \vec{v}_i) \tag{4}$$

Такое тождество справедливо только для ограниченного ряда моделей оптических свойств поверхности, как, например, Ламберта, Фонга или Блинн-Фонга. Для большинства моделей оптических свойств поверхностей (например, ДФР, определяемая функцией Гаусса или полученная в результате измерений оптических свойств поверхности с использованием гониофотометров [14] или результатов моделирования [15]), используемых в сценах, данное тождество не выполняется. Это означает, что результат, полученный методом

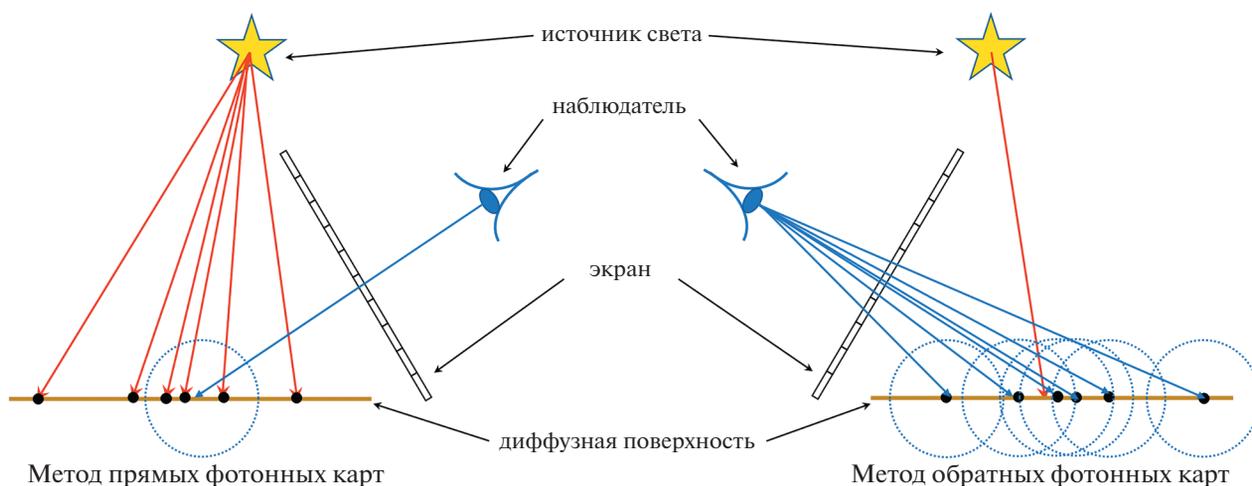


Рис. 4. Отличие метода расчета и накопления яркости при использовании прямых и обратных фотонных карт.

прямой трассировки лучей, будет отличаться от результата, в котором использовался метод обратной трассировки лучей. Поскольку за основу принимается результат, полученный методом прямой трассировки лучей, что соответствует измерениям ДФР, полученным на гониофотометре, то результат, полученный методом обратной трассировки лучей, должен быть приведен в соответствие с результатом прямой трассировки лучей.

Для обеспечения обратимости по Геймгольцу метода обратной трассировки лучей при рассеивании обратного луча на диффузной поверхности сцены следует использовать выборку по значимости направления рассеивания в соответствии с ДФР для прямого хода луча с последующей компенсацией “энергии” обратного луча на величину ДФР для прямого хода выбранного направления луча.

Другим принципиальным различием между методами прямых и обратных фотонных карт в расчете яркости вторичного и каустического освещения является метод сбора освещенности с фотонных карт, который проиллюстрирован на рис. 4.

Видно, что при использовании прямой фотонной карты формируется глобальное освещение и луч от камеры собирает яркость глобального освещения в направлении наблюдения. При использовании обратной фотонной карты формируется область видения, при попадании светового луча в которую формируется локальная освещенность. Эта локальная освещенность пересчитывается в яркость пикселей изображения, сферы интегрирования которых покрывают данный световой фотон. Таким образом формируется локальное освещение и локальная яркость ряда пикселей изображения (от первого до  $K$ -го):

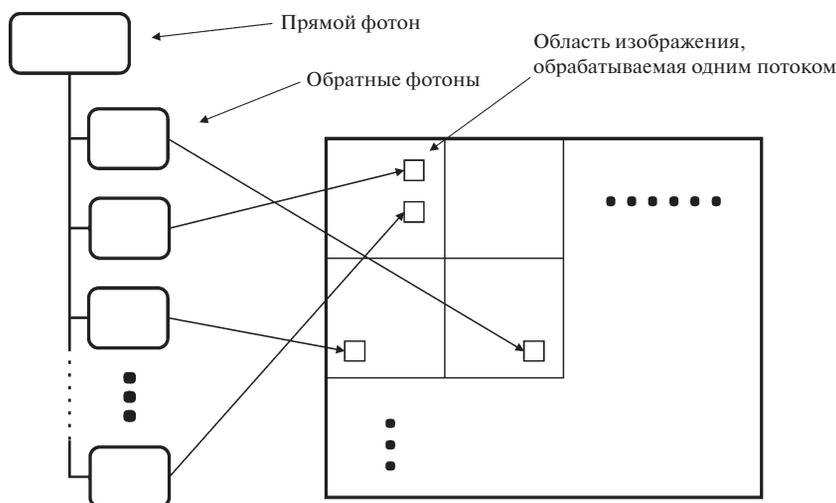
$$\begin{aligned}
 L_{1(id,c)}(\vec{p}_1, \vec{v}_{1,r}) &= \frac{1}{\pi r_1^2} f(\vec{p}_1, \vec{v}_i, \vec{v}_{1,r}) \Delta\Phi(\vec{v}_i) \\
 L_{2(id,c)}(\vec{p}_2, \vec{v}_{2,r}) &= \frac{1}{\pi r_2^2} f(\vec{p}_2, \vec{v}_i, \vec{v}_{2,r}) \Delta\Phi(\vec{v}_i) \\
 &\dots \\
 L_{K(id,c)}(\vec{p}_K, \vec{v}_{K,r}) &= \frac{1}{\pi r_K^2} f(\vec{p}_K, \vec{v}_i, \vec{v}_{K,r}) \Delta\Phi(\vec{v}_i)
 \end{aligned} \tag{5}$$

Поскольку трассируется большое количество прямых лучей, то некоторый  $j$ -й пиксель изображения, которому соответствует сфера интегрирования с центром в точке  $\vec{p}_j$ , направлением наблюдения  $\vec{v}_{j,r}$  и радиусом  $r_j$ , соберет яркость от всех фотонов, попавших в эту сферу интегрирования. Тогда формула яркости глобального освещения примет следующий вид:

$$L_{j(id,c)}(\vec{p}_j, \vec{v}_{j,r}) = \frac{1}{\pi r_j^2} \sum_{i=1}^K f(\vec{p}_j, \vec{v}_i, \vec{v}_{j,r}) \Delta\Phi_i(\vec{v}_i) \tag{6}$$

Это выражение полностью соответствует формуле (2), используемой при вычислении значения вторичной и каустической компонент яркости в методе прямых фотонных карт. Таким образом, с точки зрения физической корректности вычисления яркости глобального освещения в некоторой точке сцены методы прямых и обратных фотонных карт дают одинаковую оценку данной яркости.

С точки зрения программной реализации различие заключается в том, какие лучи взаимодействуют с фотонной картой, и в том, что фотон, сохраненный в обратной карте, должен дополнительно содержать координаты пикселя изображения, в котором будет накапливаться яркость. Кроме того, проблемой обратных фотонных карт является



**Рис. 5.** Накопление списков обратных фотонов в дополнительные прямые фотонные карты и последующая их выборка при многопоточном рендеринге.

синхронизация, необходимая для записи данных в общую область памяти изображения. В методе прямых фотонных карт необходимость в данной синхронизации отсутствует, поскольку лучи, испускаемые камерой, могут быть распределены по вычислительным потокам, что исключает возможность обращения к одной точке изображения из разных вычислительных потоков. Для того, чтобы избежать необходимости синхронизации в методе обратных фотонных карт, трассировка световых лучей и расчет яркости вторичного и каустического освещения были разделены на две части. В первой части происходит трассировка световых лучей и поиск пересечения этих лучей с обратными фотонными картами, по итогам чего формируется дополнительная карта прямых фотонов, содержащая списки всех сфер интегрирования (точек изображения), которые были пересечены прямым лучом. При этом каждый вычислительный поток формирует собственную дополнительную прямую фотонную карту. После завершения трассировки порции световых лучей производится обработка всех дополнительных фотонных карт и вычисляется яркость каустического и вторичного освещения для соответствующих точек изображения. Расчет данных компонент яркости выполняется группой параллельных потоков, каждый из которых обрабатывает отделенную часть изображения и вычисляет яркость точек этой части изображения. Поскольку дополнительная карта прямых фотонов содержит ссылки на обратные фотоны и, соответственно, индексы пикселей экрана, с которых они были испущены, то каждый поток работает только с точками изображения (и соответственно фотонами), соответствующими обрабатываемой им области экрана, и вычисляет для них яркости каустического и вторичного освеще-

ний, что полностью исключает необходимость синхронизации данных от различных потоков. Данный подход проиллюстрирован на рис. 5.

### 3. РЕНДЕРИНГ МЕТОДОМ ПРОГРЕССИВНЫХ ОБРАТНЫХ ФОТОННЫХ КАРТ

На основе перечисленных идей был предложен алгоритм рендеринга методом двунаправленной трассировки лучей с использованием прогрессивных обратных фотонных карт. На рис. 6 представлена блок-схема алгоритма рендеринга методом трассировки лучей на основе прогрессивных обратных фотонных карт. Алгоритм состоит из повторяющихся фаз, каждая из которых состоит из следующих пяти шагов:

1. На первом шаге камера испускает случайные обратные лучи, направленные от наблюдателя на точки изображения, выбранные с плотностью вероятности, пропорциональной распределению оценки ошибки яркости по изображению. В пределах пикселя также происходит случайный выбор координат луча, что обеспечивает равномерное усреднение яркости в точке изображения. Кроме того, распределение ошибки по изображению предварительно подвергается Гауссовой фильтрации, что позволяет избежать резких скачков количества лучей, выпущенных с соседних точек изображения. Лучи трассируются в сцене до заданной глубины диффузных событий (заданного порядкового номера диффузного события на трассе луча, как правило, первого или второго). Глубина диффузной трассы обратного луча может выбираться адаптивно для каждой точки изображения на основании оценки вариации яркости в точке диффузного события и зависит от

таких параметров, как расстояние до источников света и форма ДФР в точках сбора яркости вторичного и каустического освещения. В процессе обратной трассировки лучей производится расчет яркости прямого освещения с использованием метода многократной выборки по значимости в соответствии с формулой (3). В итоге обратной трассировки лучей сохраняются яркости прямого виденья и прямого освещения, а также данные лучей в финальных точках их трасс на поверхностях с диффузными свойствами (для оценки яркости вторичного освещения) и данные лучей во всех точках их трасс на поверхностях с диффузными свойствами (для оценки яркости каустического освещения и прямого освещения от источников, если оно не было вычислено ранее на этом шаге).

2. На втором шаге формируется обратная фотонная карта для сфер интегрирования с центрами в точках диффузных событий и радиусами, зависящими от длины пути от камеры до каждой конкретной точки, которая имеет диффузные свойства. Для фотонной карты строятся ускоряющие структуры в виде kd-деревьев. Данная обратная фотонная карта будет в дальнейшем использована для расчета яркости каустической и вторичной компонент освещения.

3. На третьем шаге генерируются случайные прямые световые лучи от источников света, которые распространяются в сцене и взаимодействуют с обратными фотонными картами. В процессе прямой трассировки лучей для точек трасс прямых лучей, попадающих на диффузные поверхности, производится анализ их попадания в сферы интегрирования, хранящиеся в обратной фотонной карте. В случае попадания индексы обратных фотонов накапливаются в соответствующих дополнительных прямых фотонных картах.

4. На четвертом шаге происходит обработка дополнительных прямых фотонных карт и в соответствии с формулой (6) вычисляется и накапливается яркость каустического и вторичного освещения. Полученная яркость сохраняется в точках изображения, связанных со сферами интегрирования, в которые попал световой фотон. Третий и четвертый шаги повторяются несколько раз до тех пор, пока не будет выпущено требуемое число прямых лучей.

5. На пятом шаге яркость, накопленная на текущей фазе вычислений, суммируется с полной накопленной яркостью и формируется полное изображение. На основании оценки точности полного изображения и времени вычислений принимается решение о необходимости продолжить или остановить вычисления. По окончании расчета производится нормировка накопленных значений яркостей на соответствующее число прямых



Рис. 6. Блок-схема алгоритма рендеринга методом прогрессивных обратных фотонных карт.

и обратных лучей, накопленных каждой отдельной точкой изображения.

Прерывание процесса рендеринга до полного завершения фазы вычислений (например, по сигналу на прерывание от пользователя) возможно только на третьем и четвертом шагах вычислений, когда идет трассировка прямых лучей и вычисляются яркости вторичного и каустического освещения. Несмотря на то, что это представляет некоторое неудобство, поскольку не позволяет

мгновенно остановить процесс рендеринга, данная задержка обеспечивает “однородность” изображения, не допуская наличия частей изображения с невычисленными компонентами прямого, вторичного и каустического освещения.

#### 4. ОЦЕНКА ТОЧНОСТИ РЕЗУЛЬТАТОВ МОДЕЛИРОВАНИЯ

Стохастическая природа вычисления яркости изображения является причиной возникновения ошибки ее вычисления. В ряде случаев возможность оценки ошибки вычисления значений яркости точек изображения позволяет ускорить сходимость процесса рендеринга за счет, например, формирования “оптимальной” плотности распределения вероятности лучей, испускаемых из камеры в точки изображения. Кроме того, оценка ошибки значения яркости изображения может быть использована для фильтрации изображения. Поскольку процесс вычисления величины яркости изображения носит случайный характер, то для оценки ее ошибки в точке изображения может быть использована формула стандартной ошибки среднего. Это формула может быть применена как для оценки полной ошибки значения яркости точки изображения, так и для оценки ошибки отдельных компонент яркости (яркостей первичного, вторичного и каустических освещений). Компоненты ошибки по отдельным составляющим яркости удобно использовать в процессе фильтрации изображений, например, когда необходимо выполнить фильтрацию только одной (диффузной) компоненты яркости изображения. Таким образом, оценка ошибки среднего при расчете значения яркости в  $i$ -й точке изображения методом прогрессивных обратных фотонных карт имеет следующий вид:

$$SEM_i^2 = \frac{1}{N} \left( \frac{\sum_{j=1}^N L_{i,j}^2}{N} - \left( \frac{\sum_{j=1}^N L_{i,j}}{N} \right)^2 \right), \quad (7)$$

где  $N$  – количество фаз расчета яркости,  $L_{i,j}$  – яркость  $i$ -й точки изображения, вычисленной в процессе  $j$ -й фазы рендеринга. Видно, что для оценки погрешности в точке изображения достаточно дополнительно сохранить только накопленный квадрат яркостей точки изображения  $L_{i,j}^2$  по фазам расчета и, при необходимости, накопленные квадраты всех компонент яркости точки изображения.

Средняя точность по всему изображению с разрешением  $w \times h$  будет вычисляться по формуле:

$$\delta = \sqrt{\frac{\sum_{i=1}^{w \cdot h} SEM_i^2}{\sum_{i=1}^{w \cdot h} L_i^2}} \quad (8)$$

Данная величина может информировать пользователя о достигнутой точности рендеринга и служить одним из критериев для завершения процесса рендеринга.

#### 5. РЕАЛИЗАЦИЯ И РЕЗУЛЬТАТЫ МОДЕЛИРОВАНИЯ

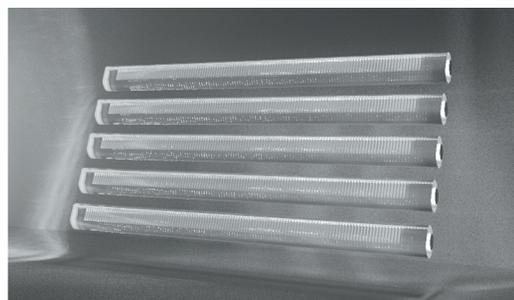
Для демонстрации преимуществ метода прогрессивных обратных фотонных карт по сравнению с методом прогрессивных прямых фотонных карт оба метода были реализованы на базе общего набора программных реализаций моделей геометрии сцены, источников света и материалов. Было проведено сравнение среднего числа прямых и обратных фотонов, формируемых каждым из методов. Сравнение производилось на ряде стандартных сцен, используемых в задачах компьютерной графики:



Сцена Cornell Box



Сцена Room2



Сцена Light Guides

Рис. 7. Изображения тестовых сцен.

**Таблица 1.** Сравнительная таблица числа прямых фотонов, формируемых одним прямым лучом, с числом обратных фотонов, формируемых одним обратным лучом для тестовых сцен

Сцена	Прямых фотонов на прямой луч	Обратных фотонов на обратный луч	Уменьшение размера фотонной карты	Ускорение рендеринга
Cornell Box	3.45	1.59	2.17	1.6
Room 2	7.6	1.93	3.94	1.8
Light guides (вариант 1)	7.56	3.09	2.45	2.2
Light guides (вариант 2)	5.6	3.09	1.81	2.1

- Cornell Box – стандартная тестовая сцена компьютерной графики;
- Room2 – интерьерная сцена;
- Light Guides – сцена с пятью светопроводящими трубками, расположенными в виртуальном прототипе Judge II Viewing Booth. Для данной сцены были рассмотрены два варианта расположения источников света: в оптическом контакте с торцами световодов (вариант 1) и с минимальным воздушным зазором с торцами световодов (вариант 2).

Изображения данных сцен представлены на рис. 7. В табл. 1 приведены результаты тестирования методов, основанных на прогрессивных прямых и обратных фотонных картах. Из результатов видно, что за счет формирования обратных фотонных карт удается значительно уменьшить размер фотонной карты. Кроме того, в отличие от прямых фотонов обратные фотоны создаются исключительно в области, участвующей в построении изображения, что одновременно с уменьшением объема фотонной карты позволяет повысить скорость рендеринга.

На рис. 8 и 9 представлены изображения, полученные в результате рендеринга одной и той же

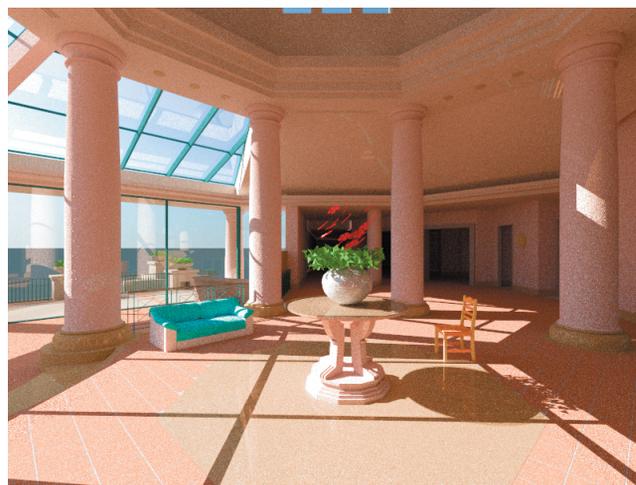
интерьерной сцены в течение одного часа с использованием метода прогрессивных прямых и обратных фотонных карт соответственно. Данная сцена освещена каустическим и вторичным освещением. Видно, что использование прогрессивных обратных фотонных карт позволяет заметно снизить шум синтезированного изображения.

## 6. ЗАКЛЮЧЕНИЕ

Был предложен оригинальный метод прогрессивных обратных фотонных карт, основанный на двунаправленной стохастической трассировке лучей. Основными преимуществами предложенного метода являются меньший объем данных, необходимый для хранения информации о фотонах, возможность формирования радиуса сферы интегрирования в соответствии с типом накапливаемой яркости и формирования фотонной карты в областях возможного сбора яркости вторичного и каустического освещения. Несмотря на технические проблемы, вызванные возможностью организации параллельных вычислений, предложенный метод показал высокую эффективность, и для сцен со сложным вторичным и каустическим освещением



**Рис. 8.** Изображение, полученное в результате рендеринга методом прогрессивных прямых фотонных карт за один час вычислений.



**Рис. 9.** Изображение, полученное в результате рендеринга методом прогрессивных обратных фотонных карт за один час вычислений.

ем качество изображения, полученного предложенным методом, в большинстве случаев выше качества изображения, полученного методом с использованием прямых фотонных карт.

Метод прогрессивных обратных фотонных карт был оптимизирован для работы на центральном процессоре, однако алгоритм параллельных вычислений, предложенный для расчета компонент яркости вторичного и каустического освещений, может быть эффективно реализован на GPU.

## 7. БЛАГОДАРНОСТИ

Работа выполнена при финансовой поддержке Российского научного фонда (проект № 18-79-10190).

## СПИСОК ЛИТЕРАТУРЫ

1. Волобой А.Г., Галактионов В.А., Жданов А.Д., Гарбуль А.А., Соколов В.Г., Потемин И.С., Жданов Д.Д. Методы фотореалистичной визуализации изображений, формируемых оптическими системами дополненной реальности // Программирование. 2018. № 4. С. 11–25.
2. Kajiya J.T. The rendering equation // Proceedings of the 13th annual conference on Computer graphics and interactive techniques. 1986. P. 143–150.
3. Kay T.L., Kajiya J.T. Ray tracing complex scenes // ACM SIGGRAPH computer graphics. 1986. V. 20. № 4. P. 269–278.
4. Chattopadhyay S., Fujimoto A. Bi-directional ray tracing // Computer Graphics 1987. Tokyo: Springer, 1987. P. 335–343.
5. Lafortune E.P., Willems Y. Bi-directional path tracing // Compugraphics' 93. 1993. P. 145–153.
6. Jensen H.W. Global illumination using photon maps // Eurographics workshop on Rendering techniques. Vienna: Springer, 1996. P. 21–30.
7. Jensen H.W., Christensen P. High quality rendering using ray tracing and photon mapping // ACM SIGGRAPH 2007 courses. 2007. 116 p.
8. Veach E., Guibas L.J. Optimally combining sampling techniques for Monte Carlo rendering // Proceedings of the 22nd annual conference on Computer graphics and interactive techniques. 1995. P. 419–428.
9. Georgiev I., Krivánek J., Davidovic T., Slusallek P. Light transport simulation with vertex connection and merging // ACM Trans. Graph. 2012. V. 31. № 6. P. 192:1–192:10.
10. Havran V., Herzog R., Seidel H.P. Fast final gathering via reverse photon mapping // Computer Graphics Forum. 2005. V. 24. № 3. P. 323–332.
11. Hachisuka T., Ogaki S., Jensen H.V. Progressive photon mapping // ACM SIGGRAPH Asia 2008 papers. 2008. P. 1–8.
12. Жданов Д.Д., Алейников А.А., Жданов А.Д., Дерябин Н.Б. Построение модели непрерывного источника света, заданного фиксированным набором лучей // Программирование. 2016. № 6. С. 6–14.
13. Snyder W.C. Reciprocity of the bidirectional reflectance distribution function (BRDF) in measurements and models of structured surfaces // IEEE Transactions on Geoscience and Remote Sensing. 1998. V. 36. № 2. P. 685–691.
14. Волобой А.Г., Галактионов В.А., Ершов С.В., Летунов А.А., Потемин И.С. Аппаратно-программный комплекс для измерения светорассеивающих свойств поверхностей // Информационные технологии и вычислительные системы. 2006. № 4. С. 24–39.
15. Sokolov V.G., Zhdanov D.D., Potemin I.S., Bogdanov N.N., Zhdanov A.D., Denisov E.U. Reconstruction of BSDF based on optimization of microrelief normal distribution // GraphiCon 2017 – 27th International Conference on Computer Graphics and Vision. 2017. P. 37–41.

ЗАВИСИМОСТЬ ШУМА ОТ ЧИСЛА ЛУЧЕЙ В ДВУНАПРАВЛЕННОЙ  
СТОХАСТИЧЕСКОЙ ТРАССИРОВКЕ ЛУЧЕЙ С ФОТОННЫМИ КАРТАМИ© 2021 г. С. В. Ершов<sup>а</sup>, Е. Д. Бирюков<sup>а</sup>, А. Г. Волобой<sup>а</sup>, В. А. Галактионов<sup>а,\*</sup><sup>а</sup>Институт прикладной математики им. М.В. Келдыша РАН,  
125047 Москва, Миусская пл., д. 4, Россия

\*E-mail: voloboy@gin.keldysh.ru

Поступила в редакцию 19.12.2020 г.

После доработки 25.12.2020 г.

Принята к публикации 14.01.2021 г.

Классическая трассировка лучей методом Монте-Карло – это мощный метод, позволяющий моделировать практически все эффекты в лучевой оптике, но он может быть недопустимо медленным для многих случаев, таких как, например, вычисление изображений, видимых объективом или камерой с точечным отверстием. Поэтому часто используются его различные модификации, в частности, двунаправленная стохастическая трассировка лучей с фотонными картами. Недостатком всех стохастических методов является нежелательный шум. Уровень шума, то есть дисперсия яркости пикселей, рассчитанной за одну итерацию метода, зависит от различных параметров, таких как количество лучей от источника света и от камеры, способ слияния их траекторий, радиус интегрирующей сферы и т.д. Выбор оптимальных параметров позволит получить минимальный уровень шума при данном времени расчета. Данной проблеме и посвящена эта статья. Показано, что дисперсия яркости пикселя представляет собой сумму трех функций, масштабируемых обратным числом лучей из источника и из камеры, причем сами эти функции не зависят от количества лучей. Поэтому, зная их, можно предсказать шум для любого количества лучей и, таким образом, найти оптимальный вариант. Вычисление этих функций на основе полученных в трассировке лучей данных является нетривиальной задачей. В статье приведен практический метод их расчета и продемонстрировано, что по результатам всего одного пробного расчета можно предсказать дисперсию для произвольного числа лучей. Таким образом, становится возможным минимизация шума благодаря выбору оптимального числа лучей.

DOI: 10.31857/S0132347421030043

## 1. ВВЕДЕНИЕ

В настоящее время моделирование распространения света широко используется в реалистичной компьютерной графике, проектировании новых материалов и оптических систем [1]. Оно интенсивно применяется в архитектурных, автомобильных и авиационных конструкторских задачах. Если задача позволяет пренебречь волновыми эффектами, то хорошим выбором является группа методов стохастической трассировки лучей. Эта область главным образом включает моделирование переноса излучения методом Метрополиса [2] и стохастическую трассировку лучей [3]. При расчете изображения классическая прямая трассировка лучей от источника света неэффективна и поэтому заменяется двунаправленными модификациями метода [4–6]. Из них рассмотрим так называемую двунаправленную стохастическую трассировку лучей с фотонными картами (BDPM – Bidirectional Photon Mapping) [5, 7]. Слабой стороной всех стохастических методов

является то, что их результаты зашумлены. Поэтому задача снижения шума всегда актуальна, ей посвящено большое число работ, в частности, можно выделить работы [8–10].

Уровень шума в методе BDPM в основном зависит от способа случайного рассеяния прямого и обратного лучей, от выбора вершины для их слияния (или, другими словами, в какой вершине траектории камеры использовать фотонные карты для оценки освещенности) и, наконец, от количества прямых и обратных лучей, протрассированных за одну итерацию. Большинство публикаций посвящено первым двум средствам, например, [9–12], в то время как меньшее внимание было уделено количеству лучей. Между тем это важный фактор, и часто бывает так, что, например, количество прямых лучей уже избыточно, поэтому дальнейшее его увеличение не уменьшает шум, а только увеличивает время выполнения. В других случаях бывает так, что количество прямых лучей

**Таблица 1.** Среднее значение RMS по части изображения (отмеченной красным квадратиком на рис. 2) после 1000 секунд вычисления для различного числа лучей  $N_F$ ,  $N_B$

$N_F \backslash N_B$	5	10	25	100
100	0.135	0.135	0.135	0.135
1000	0.101	0.101	0.101	0.101
3000	0.099	0.099	0.099	0.099
10000	0.101	0.101	0.101	0.101
30000	0.110	0.110	0.109	0.109
100000	0.136	0.136	0.135	0.135

действительно критично, в то время как мы трассируем избыточное количество обратных лучей.

Подобное положение иллюстрируется табл. 1, где показана зависимость дисперсии (RMS) после фиксированного времени расчета от числа прямых лучей (из источника света)  $N_F$  и обратных (от камеры через пиксель)  $N_B$ . Описание сцены приведено в разделе *Результаты*. В приведенном примере зависимость от числа обратных лучей  $N_B$  (приведенных по столбцам) практически отсутствует, а вот по числу лучей из источника имеется минимум (шума) при  $N_F \approx 3000$ . Здесь мы видим, что общее утверждение “чем больше лучей, тем меньше шум”, очевидно, ошибочно.

Обычно трудно предсказать, какая доля прямых и обратных лучей является оптимальной, в то время как ее хороший выбор может ускорить расчеты в несколько раз.

В этой статье мы рассмотрим последнюю проблему. В работе [8] был выведен общий закон, определяющий шум в VDPM. Он утверждает, что дисперсия яркости пикселя представляет собой сумму трех составляющих, масштабированных обратным числом лучей. Эти 3 компоненты не зависят от количества лучей и, таким образом, однажды вычисленные они могут быть позже использованы для предсказания того, как уровень шума будет зависеть от количества лучей, протрассированных за одну итерацию. Таким образом, предполагается прогнозировать оптимальное количество лучей. Другими словами, если мы знаем эти 3 компоненты, которые обычно зависят от пикселя, то можем предсказать уровень шума для любого количества лучей.

Хотя их математическое определение тривиально, эти три значения не так легко вычислить численно в процессе трассировки лучей. В данной работе описан метод их эффективного расчета и продемонстрировано их использование для выбора оптимального числа лучей.

## 2. ШУМ В ДВУНАПРАВЛЕННОЙ ТРАССИРОВКЕ ЛУЧЕЙ С ФОТОННЫМИ КАРТАМИ

Обычно вычисления в VDPM идут итерациями. На каждой итерации мы трассируем  $N_F$  световых путей, и  $N_B(p)$  путей из камеры для каждого пикселя  $p$ . Затем проверяем каждую пару, и если световой путь проходит достаточно близко с узлом пути из камеры, *соединяем* их, получая полный путь (от камеры до источника). Затем мы вычисляем вклад этого *объединенного* пути в яркость пикселя и увеличиваем яркость пикселя на это значение. Накопленная сумма, деленная на произведение  $N_F N_B(p) N_I$  (где  $N_I$  – число итераций), сходится к математическому ожиданию яркости. Это среднее значение не зависит от  $N_F$  и  $N_B(p)$ , а вот его дисперсия (шум) может весьма сильно меняться. Скажем, если световых лучей уже слишком много, а траекторий лучей из камеры мало, то трассировка этих избыточных световых лучей только тратит время без улучшения качества изображения, поэтому выгодно уменьшить их количество.

Оптимальное количество лучей (общее для каждой сцены для световых путей и индивидуальное для каждого пикселя для путей из камеры) – это то, которое приводит к наименьшему шуму после фиксированного времени расчета. Для решения этой задачи минимизации нам нужно знать, как дисперсия значения, полученного за одну итерацию, зависит от этого числа лучей. Эта зависимость описывается несложной алгебраической формулой [8], однако ее члены не так легко найти численно.

Здесь и ниже мы всегда будем говорить о значениях для одного, заданного пикселя  $p$  и поэтому отбросим эту нотацию, написав  $N_B$  вместо  $N_B(p)$ .

Как показано в [8], дисперсия яркости в данном пикселе, *рассчитанной за одну итерацию*, равна

$$V = \frac{1}{N_F N_B} (\langle\langle C^2 \rangle\rangle - \langle\langle C \rangle\rangle^2) + \frac{1 - N_F^{-1}}{N_B} (\langle\langle C^2 \rangle\rangle_F - \langle\langle C \rangle\rangle^2) + \frac{1 - N_B^{-1}}{N_F} (\langle\langle C^2 \rangle\rangle_B - \langle\langle C \rangle\rangle^2),$$

где  $C(X^{(F)}, X^{(B)})$  – вклад в яркость пикселя от “слияния” пути из источника  $X^{(F)}$  и пути из камеры  $X^{(B)}$ , его среднее  $\langle\langle C \rangle\rangle$  очевидно совпадает с предельной (точной) яркостью пикселя  $L$ , а  $\langle\cdot\rangle_F$  и  $\langle\cdot\rangle_B$  обозначают усреднение по ансамблю путей из источника и из камеры, соответственно.

Заметим, что сама эта функциональная форма не зависит от того, используется ли Multiple Im-

portance Sampling [13] или нет, и даже от того, используем мы слияние вершин (Vertex Merging) или соединение вершин дополнительным сегментом траектории (Vertex Connection) [9]. Хотя величины  $\langle\langle C^2 \rangle\rangle$ ,  $\langle\langle C \rangle_F^2 \rangle_B$  и  $\langle\langle C \rangle_B^2 \rangle_F$  будут меняться в зависимости от конкретной стратегии вычислений. При этом в любом случае они не зависят от числа лучей, хотя могут зависеть от радиуса интегрирующей сферы (т.е. “радиуса слияния вершин”).

### 3. ВЫЧИСЛЕНИЕ КОЭФФИЦИЕНТОВ ФОРМУЛЫ ШУМА

Вычисление  $\langle\langle C^2 \rangle\rangle$  тривиально (точно как при расчете выборочной дисперсии): добавляя  $C$  к накопленной яркости пикселя, мы заодно добавляем  $C^2$  к накопленному в этом пикселе  $\langle\langle C^2 \rangle\rangle$ . То есть, на каждой итерации мы вычисляем

$$C \equiv \frac{1}{N_B N_F} \sum_{j=1}^{N_B} \sum_{i=1}^{N_F} C^2(X_i^{(F)}, X_j^{(B)})$$

и среднее от этой величины по итерациям  $C_{\text{avg}}$  сойдется к тому, что нам и надо:  $C_{\text{avg}} \rightarrow \langle\langle C^2 \rangle\rangle$ . Однако с  $\langle\langle C \rangle_F^2 \rangle_B$  и  $\langle\langle C \rangle_B^2 \rangle_F$  такой метод, конечно же, невозможен.

В самом деле, допустим, что  $N_F$  так велико, что для каждого луча из камеры сумма  $\frac{1}{N_F} \sum_{i=1}^{N_F} C(X_i^{(F)}, X_j^{(B)})$  дает хорошее приближение к  $\langle C \rangle_F(X_j^{(B)})$ . В то же время, как правило, число лучей через один пиксель  $N_B$  невелико, и потому усреднение  $\langle C \rangle_F^2(X_j^{(B)})$  по этому малому числу лучей недостаточно. Однако тут мы можем добавить усреднение по итерациям, так как в каждой из них у нас свой случайный и независимый набор лучей из камеры, и так получить уже хорошую оценку для  $\langle\langle C \rangle_F^2 \rangle_B$ .

Это, разумеется, лишь мысленный эксперимент, и на практике подобный метод работать не может, так как обычно количество лучей из источника тоже не столь велико, чтобы иметь достаточно точную оценку  $\langle C \rangle_F(X_j^{(B)})$  в пределах одной итерации. Использовать же тут усреднение по итерациям крайне сложно, так как эта величина должна вычисляться для каждой трассы из камеры  $X_j^{(B)}$ , а она встречается лишь однажды. Конечно, на самом деле в окрестность этого пути мы будем попадать регулярно, так что, если на каждой итерации вычислять и запоминать сеточную функцию  $X_j^{(B)}$  (на достаточно частой сетке), то все и получит-

ся. Но это весьма неэкономично, так как размерность пространства, равная максимальной длине трассы из камеры, достаточно велика.

Решение, как ни странно, весьма просто. Пусть для удобства число лучей из камеры и из источника не зависит от итерации. Чтобы вычислить  $\langle\langle C \rangle_F^2 \rangle_B$ , разобьем все множество  $N_F$  лучей из источника на две половины (не обязательно равные), содержащие  $N_{F_1}$  и  $N_{F_2}$  лучей соответственно, и будем на каждой итерации вычислять

$$B \equiv \frac{1}{N_B} \sum_{j=1}^{N_B} \left( \left( \frac{1}{N_{F_1}} \sum_{i \in \text{half}_1} C(X_i^{(F)}, X_j^{(B)}) \right) \times \left( \frac{1}{N_{F_2}} \sum_{i \in \text{half}_2} C(X_i^{(F)}, X_j^{(B)}) \right) \right)$$

Среднее от этой величины по итерациям  $B_{\text{avg}}$  есть не что иное, как среднее по ансамблю  $N_F$  лучей из источника и  $N_B$  лучей из камеры. Эти два усреднения независимы и потому перестановочны. Усредним сначала  $B$  по ансамблю лучей из источника:  $B_{\text{avg}} = \langle\langle B \rangle_F \rangle_B$ . Поскольку лучи первой и второй половины этого ансамбля очевидно независимы, то среднее от произведения двух внутренних сумм есть произведение средних по этим половинкам:

$$\langle B \rangle_F \equiv \frac{1}{N_B} \sum_{j=1}^{N_B} (\langle C \rangle_{F_1}(X_j^{(B)}) \langle C \rangle_{F_2}(X_j^{(B)})),$$

а поскольку статистические характеристики лучей первой и второй половины, конечно же, одинаковы, то средние по половинкам ансамбля и по всему ансамблю одинаковы тоже

$$\langle B \rangle_F \equiv \frac{1}{N_B} \sum_{j=1}^{N_B} \langle C \rangle_F^2(X_j^{(B)})$$

Оставшееся усреднение по ансамблю лучей из камеры дает

$$B_{\text{avg}} = \langle\langle B \rangle_F \rangle_B = \langle\langle C \rangle_F^2 \rangle_B$$

Таким образом, второй коэффициент формулы шума может быть вычислен со сколь угодно высокой точностью по последовательности итераций, даже если в каждой из них лучей мало.

Вычисление  $\langle\langle C \rangle_B^2 \rangle_F$  организуется совершенно аналогично, разбивая на две части теперь уже набор лучей из камеры (через данный пиксель!) и вычисляя на каждой итерации:

$$F \equiv \frac{1}{N_F} \sum_{i=1}^{N_F} \left( \left( \frac{1}{N_{B_1}} \sum_{i \in \text{half}_1} C(X_i^{(F)}, X_j^{(B)}) \right) \times \left( \frac{1}{N_{B_2}} \sum_{i \in \text{half}_2} C(X_i^{(F)}, X_j^{(B)}) \right) \right)$$

Ее среднее по итерациям  $F_{\text{avg}}$  сходится к  $\langle\langle C \rangle_B^2\rangle_F$ .

Заметим, что разбиение набора лучей из камеры или из источника, может быть и на неравные части. Но смысла в этом нет, так как суммы по этим двум частям входят в наши формулы симметрично, и потому нет резона “улучшать” одну за счет другой. На практике проще использовать разбиение на четные и нечетные лучи.

Зная теперь квадратичные средние и предельную яркость пикселя  $L$ , мы можем вычислить дисперсию *вклада одной итерации* в эту яркость как

$$V = \frac{1}{N_F N_B} (C_{\text{avg}} - L^2) + \frac{1 - N_F^{-1}}{N_B} (B_{\text{avg}} - L^2) + \frac{1 - N_B^{-1}}{N_F} (F_{\text{avg}} - L^2), \quad (1)$$

а после  $N_I$  итераций дисперсия будет  $\frac{V}{N_I}$ .

Отметим, что  $C_{\text{avg}}, B_{\text{avg}}, F_{\text{avg}}$ , как и  $L$ , зависят от пикселя, но *не зависят от числа лучей*. Правда, как мы уже говорили,  $C_{\text{avg}}, B_{\text{avg}}, F_{\text{avg}}$  могут зависеть от радиуса интегрирующей сферы. Это очень важное свойство, так как оно позволяет предсказать шум для *произвольного числа лучей*. Таким образом, можно провести *один* предварительный расчет для как-либо выбранного, пусть и весьма неудачного, числа лучей, найти значения  $C_{\text{avg}}, B_{\text{avg}}, F_{\text{avg}}, L$  и затем вычислить *оптимальное* число лучей (в общем случае при этом  $N_B$  будет в каждом пикселе свое), которое обеспечивает минимально возможный шум.

#### 4. ОСОБЕННОСТИ УЧЕТА ПРЯМОГО ОСВЕЩЕНИЯ

В методе ВДРМ прямое освещение может учитываться двумя разными способами. Во-первых, можно все лучи из источника – как первичные (не испытавшие ни одного рассеяния), так и вторичные (рассеянные), обрабатывать единообразно. То есть прямое освещение тоже берется из фотонных карт. Во-вторых, можно вычислять его непосредственно: каждую вершину пути из камеры мы соединяем с источником, вычисляем его яркость вдоль данного сегмента и складываем с

оценкой яркости из фотонных карт [14] (формула (2) при  $w_0 = 0, w_1 = 1$ ). Как правило, второй способ выгоднее, так как в нем компонента яркости, связанная с прямым освещением, не содержит шума фотонных карт.

Разумеется, в обоих случаях шум также описывается формулами из раздела 3. В самом деле, если все компоненты яркости берутся из фотонных карт, как в первом способе, то и *ненулевых* членов  $C(X_i^{(F)}, X_j^{(B)})$  окажется много меньше предельного значения  $N_F N_B N_p$  (где  $N_p$  – число пикселей изображения), поскольку вероятность “пересечения” трасс мала. Поэтому вычисление сумм в  $C, B, F$  будет достаточно эффективно.

Иное дело второй способ, в котором *каждый* путь из камеры приносит какой-то вклад из-за прямого освещения. Теперь уже *все*  $C(X_i^{(F)}, X_j^{(B)})$  оказываются отличны от нуля, даже если  $X_i^{(F)}$  и  $X_j^{(B)}$  не пересекаются. Вычисление сумм в  $C, B, F$  становится слишком дорогим. Поэтому для второго способа выгодно его несколько преобразовать, что позволит упростить вычисления.

Для борьбы с этим эффектом необходимо в суммах обособить вклад от прямого освещения, ибо он не зависит от луча из источника, а потому может быть вынесен за суммирование:

$$C(X_i^{(F)}, X_j^{(B)}) = C^{(I)}(X_i^{(F)}, X_j^{(B)}) + C^{(0)}(X_j^{(B)}),$$

где  $C^{(I)}(X_i^{(F)}, X_j^{(B)})$  – вклад от пересечения камерного и *вторичного* лучей из источника, а  $C^{(0)}(X_j^{(B)})$  – вклад от прямого освещения, получаемого соединением узла луча от камеры с источником света.

Теперь  $C$  запишем так:

$$\begin{aligned} C &= \frac{1}{N_B N_F} \sum_{j=1}^{N_B} \sum_{i=1}^{N_F} (C^{(I)}(X_i^{(F)}, X_j^{(B)}) + C^{(0)}(X_j^{(B)}))^2 = \\ &= \frac{1}{N_B N_F} \sum_{j=1}^{N_B} \sum_{i=1}^{N_F} ((C^{(I)}(X_i^{(F)}, X_j^{(B)}))^2 + \\ &\quad + 2C^{(I)}(X_i^{(F)}, X_j^{(B)})C^{(0)}(X_j^{(B)}) \\ &\quad + \frac{1}{N_B} \sum_{j=1}^{N_B} (C^{(0)}(X_j^{(B)}))^2 \end{aligned}$$

Суммы теперь содержат столько же (ненулевых) членов, как и в отсутствии прямого освещения (т.е. при  $C^{(0)} = 0$ ).

Затем, выражение для  $B$  преобразуем:



Рис. 1. Виртуальная фотография модельной сцены Cornell Box.

$$B = \frac{1}{N_B} \sum_{j=1}^{N_B} \left( \left( \frac{\sum_{i \in \text{half}_1} C^{(I)}(X_i^{(F)}, X_j^{(B)})}{N_{F_1}} + C^{(0)}(X_j^{(B)}) \right) \times \left( \frac{\sum_{i \in \text{half}_2} C^{(I)}(X_i^{(F)}, X_j^{(B)})}{N_{F_2}} + C^{(0)}(X_j^{(B)}) \right) \right) = \frac{1}{N_B} \sum_{j=1}^{N_B} B_1(X_j^{(B)}) B_2(X_j^{(B)}),$$

где

$$B_k(X_j^{(B)}) \equiv C^{(0)}(X_j^{(B)}) + \frac{1}{N_{F_k}} \sum_{i \in \text{half}_k} C^{(I)}(X_i^{(F)}, X_j^{(B)})$$

Аналогично,

$$F = \frac{1}{N_F} \sum_{i=1}^{N_F} ((F_1^{(I)}(X_i^{(F)}) + F_1^{(0)})(F_2^{(I)}(X_i^{(F)}) + F_2^{(0)})) = \frac{1}{N_F} \sum_{i=1}^{N_F} ((F_1^{(I)}(X_i^{(F)}) F_2^{(I)}(X_i^{(F)}) + F_1^{(I)}(X_i^{(F)}) F_2^{(0)} + F_2^{(I)}(X_i^{(F)}) F_1^{(0)})) + F_1^{(0)} F_2^{(0)},$$

где

$$F_k^{(I)}(X_i^{(F)}) \equiv \frac{1}{N_{B_k}} \sum_{j \in \text{half}_k} C^{(I)}(X_i^{(F)}, X_j^{(B)}) F_k^{(0)} \equiv \frac{1}{N_{B_k}} \sum_{j \in \text{half}_k} C^{(0)}(X_j^{(B)})$$

Теперь все суммы содержат столько же ненулевых членов, как и в отсутствии прямого освещения (т.е. при  $C^{(0)} = 0$ ), и потому вычисления достаточно экономичны. Затрачиваемое время примерно то же, что и при вычислении самой яркости изображения.

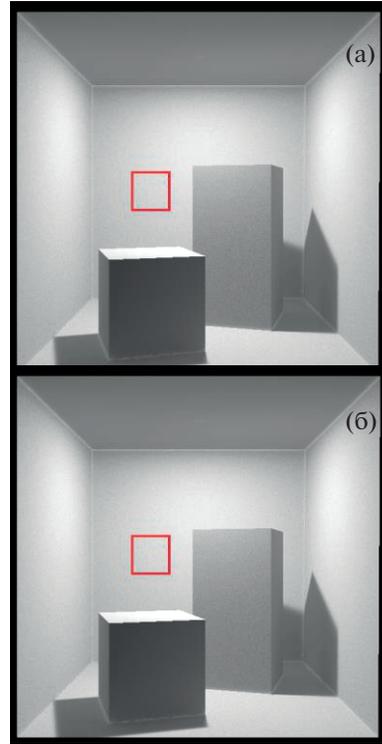


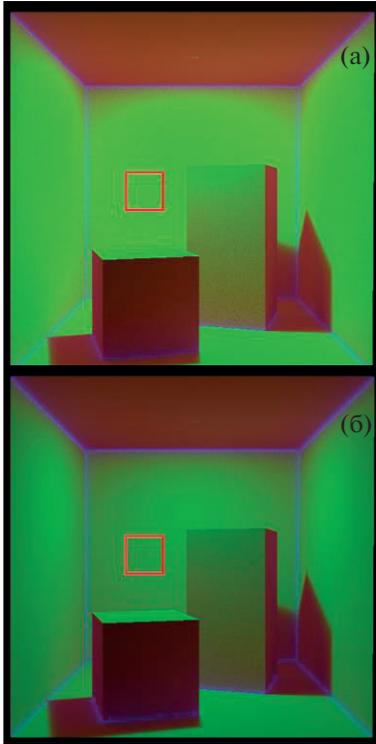
Рис. 2. Шум как  $\sqrt{V}$  для  $N_B = 25$ ,  $N_F = 1000$ . Сверху (рис. 2а): эталонное значение, найденное как выборочная дисперсия по последовательности итераций. Снизу (рис. 2б):  $V$  вычислено по формуле (1). Усредненное по обозначенной красным квадратиком области RMS равно 1.2767 на верхнем изображении и 1.2769 на нижнем.

## 5. РЕЗУЛЬТАТЫ

В качестве примера использовалась известная тестовая сцена Cornell Box. Изотропный точечный источник света, расположен немного ниже центра потолка. Все поверхности – ламбертовские с альбедо 0.5. Метод BDPM использовался без Multiple Importance Sampling, после второго диффузного рассеяния луч из камеры принудительно обрывался. Радиус интегрирующей сферы –  $1/120$  размера сцены. Прямое освещение бралось не из фотонных карт, а вычислялось непосредственным соединением точки поверхности с источником света.

Изображение, полученное виртуальной камерой, показано на рис. 1.

Три члена формулы шума были вычислены, как описано в разделах 3 и 4, и полная дисперсия вклада в яркость за одну итерацию была вычислена по формуле (1). Для сравнения она также была вычислена как выборочная дисперсия по серии итераций, что возможно, так как вклады от различных итераций независимы.



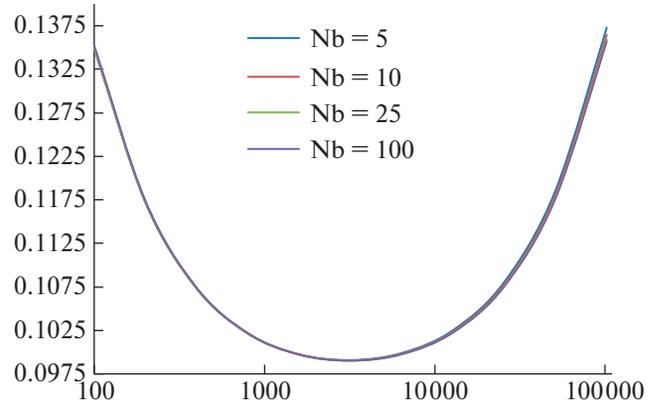
**Рис. 3.** Вклады в полный шум  $\sqrt{\frac{1}{N_F N_B} (C_{\text{avg}} - L^2)}$ ,  $\sqrt{\frac{1 - N_F^{-1}}{N_B} (B_{\text{avg}} - L^2)}$  и  $\sqrt{\frac{1 - N_B^{-1}}{N_F} (F_{\text{avg}} - L^2)}$  записаны в цветные каналы R, G и B изображения. **Сверху (Рис. 3а):**  $N_B = 25$ , **снизу (Рис. 3б):**  $N_B = 100$ ; в обоих случаях  $N_F = 1000$ . Средние значения по области, ограниченной красным квадратиком: (0.3812065, 1.21883, 0.01907) **сверху** и (0.19602, 0.610129, 0.0204105) **снизу**, так что первые две компоненты уменьшились как раз в  $\sqrt{4}$  раз. Полный RMS, т.е.  $\sqrt{V}$ , усредненный по красному квадратику, 1.27689 **сверху** и 0.639361 **снизу**.

Амплитуда шума, т.е.  $\sqrt{V}$ , вычисленная этими двумя способами, показана на рис. 2. Видно, что результаты практически неразличимы.

Для иллюстрации, который из трех членов формулы (1) дает какой вклад в полную дисперсию, все они показаны на рис. 3 в виде (R, G, B) компонент цвета изображения. А именно, вели-

чины  $\sqrt{\frac{1}{N_F N_B} (C_{\text{avg}} - L^2)}$ ,  $\sqrt{\frac{1 - N_F^{-1}}{N_B} (B_{\text{avg}} - L^2)}$  и  $\sqrt{\frac{1 - N_B^{-1}}{N_F} (F_{\text{avg}} - L^2)}$  записаны в R, G и B каналы изображения, соответственно.

Видно, что картинка на рис. 3 в основном зеленая, т.е. доминирует вторая компонента. Таким образом, в нашем примере для снижения шума



**Рис. 4.** Средний RMS по части изображения (красный квадратик на рис. 2) после 1000 с вычисления для различных  $N_B$  и  $N_F$  (по горизонтальной оси).

необходимо главным образом уменьшить этот доминирующий, второй член, т.е. увеличить  $N_B$ , так как  $C_{\text{avg}}$ ,  $B_{\text{avg}}$ ,  $F_{\text{avg}}$  и  $L$  от числа лучей не зависят. Во время как увеличение числа лучей из источника  $N_F$  почти не дает эффекта и лишь замедляет расчет. Результаты для увеличенного втрое  $N_B$  показаны на рис. 3б.

На практике значение имеет не столько дисперсия вклада одной итерации  $V$ , сколько дисперсия яркости после заданного времени расчета  $T$ . В течение этого времени будет сделано  $N_I = \frac{T}{\tau}$  итераций и дисперсия полученного изображения составит  $V_T = \frac{\tau V}{T}$ . В то время как сама  $V$  монотонно убывает с ростом числа лучей, время одной итерации, напротив, монотонно возрастает. Эти два “противодействующих фактора” и обеспечивают наличие оптимума по числу лучей, как продемонстрировано на рис. 4 и в табл. 2. Они показывают поведение среднего по красному квадратику RMS (т.е.  $\sqrt{V_T}$ ) для нашей испытательной сцены в зависимости от числа лучей на 1 итерацию при  $T = 1000$  с. Заметим, что число лучей из камеры было взято одинаковым во всех пикселях, а не только внутри красного квадрата.

Из рис. 4 хорошо видно, что зависимость от числа лучей из камеры (представленных графиками разного цвета) в данном примере очень слаба. Зависимость же от числа лучей из источника (отложенных по горизонтальной оси) имеет минимум в окрестности  $N_F = 3100$ . Точное значение минимума может слегка меняться в зависимости от числа лучей из камеры. Минимум этот, однако, весьма плоский, то есть RMS почти не меняется в широких пределах изменения  $N_F$ , и заметный его

**Таблица 2.** Статистика трассировки тестовой сцены в течение 1000 с при 25 лучах из камеры через пиксель в зависимости от числа лучей из источника

$N_F$	1000	3000	10000	30000	100000	300000
$\sqrt{V_T}$	0.1	0.098	0.1	0.11	0.13	0.18
$\sqrt{V}$	1.277	1.234	1.224	1.222	1.21	1.202
$N_i$	160	158	149	130	89	47
$\tau$	6.25	6.33	6.7	7.7	11.25	21.47
$\tau_{\text{BMCRT}}$	6	6	6	6	6	6
$\tau_{\text{FMCRT}}$	$\approx 0$	$\approx 0$	$\approx 0$	$\approx 0$	0.05	0.17
$\tau_{\text{merge}}$	0.25	0.33	0.7	1.7	5.2	15.3

рост начинается только при отклонении от оптимума на порядки.

Необходимо пояснить, что плавная кривая по многим точкам на рис. 4 построена, вычисляя RMS за одну итерацию по *точной* формуле (1), а время на одну итерацию по аппроксимации  $\tau = \alpha N_F + \beta N_B + \gamma N_B N_F$  коэффициенты которой получены подгонкой по расчетам из табл. 1. Аппроксимация эта работает очень хорошо.

Более детальная информация показана в табл. 2 для случая  $N_B = 25$ . Время одной итерации  $\tau$  складывается из времени, потраченного на трассировку лучей из камеры  $\tau_{\text{BMCRT}}$ , трассировку лучей из источника  $\tau_{\text{FMCRT}}$ , и времени слияния этих трасс  $\tau_{\text{merge}}$ . Помимо итогового шума  $\sqrt{V_T}$  табл. 2 показывает дисперсию по одной итерации  $\sqrt{V}$ , число итераций, сделанных за период  $T$ , время на одну итерацию  $\tau$  и ее составляющие.

В табл. 2 хорошо видно, что дисперсия по одной итерации уменьшается с увеличением числа лучей от источника света (что ожидаемо), в то время как итоговый шум возрастает. Также анализ времен составляющих позволяет оценить их вклад в общее время трассировки и понять, почему увеличение числа лучей ведет в конечном счете к ухудшению качества итогового изображения.

## 6. ЗАКЛЮЧЕНИЕ

Предложен практичный метод вычисления входящих в формулу шума для ВДРМ коэффициентов  $\langle\langle C^2 \rangle\rangle$ ,  $\langle\langle C \rangle_F^2\rangle_B$ ,  $\langle\langle C \rangle_B^2\rangle_F$  на основе полученных в трассировке лучей величин. Также рассмотрен практический способ вычисления коэффициентов для случая непосредственного вычисления прямого освещения.

Само итоговое значение шума может быть вычислено непосредственно по выборочной дисперсии ряда значений, накопленных в последовательных итерациях. Однако это дает только его значение для данного расчета и не позволяет предсказать, как он изменится для другого числа лучей. Между тем, это легко вычислить, зная коэффициенты  $\langle\langle C^2 \rangle\rangle$ ,  $\langle\langle C \rangle_F^2\rangle_B$ ,  $\langle\langle C \rangle_B^2\rangle_F$ . Таким образом, по результатам всего одного расчета можно предсказать дисперсию (вклада одной итерации) для произвольного числа лучей.

Если время, затраченное на трассировку лучей из камеры и из источника, известно вместе со временем, затраченным на их “слияние”, как это приводится в табл. 2, то можно также вычислить оптимальное количество лучей или, другими словами, такое число лучей, которое дает наименьший шум в течение того же времени расчета.

## СПИСОК ЛИТЕРАТУРЫ

1. Zhdanov D., Galaktionov V., Voloboy A., Zhdanov A., Garbul A., Potemin I. and Sokolov V. Photorealistic rendering of images formed by augmented reality optical systems // Programming and Computer Software. 2018. V. 44. № 4. P. 213–224. <https://doi.org/10.1134/S0361768818040126>
2. Sik M., Krivanek J. Survey of Markov Chain Monte Carlo methods in light transport simulation // IEEE Transactions on Visualization and Computer Graphics. 2018. V. 26. № 4. P. 1821–1840.
3. Pharr M., Humphreys G. Physically Based Rendering. Second Edition: From Theory To Implementation. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2010.
4. Dodik N. Implementing probabilistic connections for bidirectional path tracing in the Mitsuba Renderer, Sep. 2017. <https://www.cg.tuwien.ac.at/research/publications/2017/dodik-2017-pcbpt/>
5. Jensen H.W., Christensen P. High quality rendering using ray tracing and photon mapping // ACM SIGGRAPH 2007 Courses. Ser. SIGGRAPH '07. 2007. NY, USA: ACM. <http://doi.acm.org/10.1145/1281500.1281593>
6. Veach E. A dissertation: Robust Monte-Carlo methods for light transport simulation. 1997. [http://graphics.stanford.edu/papers/veach\\_thesis/thesis.pdf](http://graphics.stanford.edu/papers/veach_thesis/thesis.pdf)
7. Vorba J. Bidirectional photon mapping / Proceedings of CESC 2011: The 15th Central European Seminar on Computer Graphics. Prague: Charles University, 2011. P. 25–32. <https://cgg.mff.cuni.cz/~jaroslav/papers/2011-bdpm/vorba2011-bdpm.pdf>
8. Ershov S.V., Zhdanov D.D., Voloboy A.G. Estimation of noise in calculation of scattering medium luminance by MCRT // Mathematica Montisnigri. 2019. V. XLV. P. 60–73.
9. Georgiev I., Krivanek J., Davidovic T., Slusallek P. Light transport simulation with vertex connection and merg-

- ing // ACM Trans. Graph. 2012. V. 31. № 6. P. 192:1–192:10. <http://doi.acm.org/10.1145/2366145.2366211>
10. *Hachisuka T., Pantaleoni J., Jensen H.W.* A path space extension for robust light transport simulation // ACM Trans. Graph. 2012. V. 31. P. 191:1–191:10.
  11. *Ershov S.V., Voloboy A.G.* Calculation of MIS weights for bidirectional path tracing with photon maps in presence of direct illumination // *Mathematica Montisnigri*. 2020. V. XLVIII. P. 86–102.
  12. *Sbert M., Havran V., Szirmay-Kalos L.* Multiple importance sampling revisited: breaking the bounds // *EURASIP Journal on Advances in Signal Processing*. 2018. V. 15. P. 1–15.
  13. *Hachisuka T., Jensen H.W.* Stochastic progressive photon mapping // *ACM SIGGRAPH Asia 2009 Papers*. 2009. P. 141:1–141:8. <http://doi.acm.org/10.1145/1661412.1618487>
  14. *Ершов С.В., Бирюков Е.Д., Волобой А.Г.* Эффективное вычисление оптимальных весов множественной выборки по значимости в двунаправленной трассировке лучей с фотонными картами // *Препринты ИПМ им. М.В. Келдыша*. 2020. № 107. 22 с. <https://doi.org/10.20948/prepr-2020-107>

## МЕТОД УЛУЧШЕНИЯ КАЧЕСТВА ЗАПОЛНЕНИЯ ОБЛАСТЕЙ ИЗОБРАЖЕНИЙ ВЫСОКОГО РАЗРЕШЕНИЯ

© 2021 г. А. В. Москаленко<sup>а,\*</sup>, М. В. Ерофеев<sup>а,\*\*</sup>, Д. С. Ватолин<sup>а,\*\*\*</sup>

<sup>а</sup> *Московский государственный университет им. М.В. Ломоносова,  
Факультет вычислительной математики и кибернетики,  
119991 Москва, ГСП-1, Ленинские горы, д. 1, Россия*

*\*E-mail: andrey.moskalenko@graphics.cs.msu.ru*

*\*\*E-mail: merofeev@graphics.cs.msu.ru*

*\*\*\*E-mail: dmitriy@graphics.cs.msu.ru*

Поступила в редакцию 10.10.2020 г.

После доработки 20.10.2020 г.

Принята к публикации 12.01.2021 г.

В работе рассматривается задача заполнения областей изображений. В последние годы эта область стремительно развивалась, новые нейросетевые методы показывают впечатляющие результаты, однако большинство нейросетевых подходов сильно зависят от разрешения, на котором их обучали. Незначительное увеличение разрешения приводит к серьезным артефактам и неудовлетворительному результату заполнения, из-за чего подобные методы не применимы в средствах интерактивной обработки изображений. В этой статье мы представляем метод, позволяющий решить проблему заполнения областей изображений разного разрешения. Мы также описываем способ более качественного восстановления текстурных фрагментов в заполняемой области. Для этого мы предлагаем использовать информацию из соседних пикселей путем сдвига исходного изображения в четырех направлениях. Предлагаемый подход применим к уже существующим методам без необходимости их переобучения.

DOI: 10.31857/S0132347421030079

### 1. ВВЕДЕНИЕ

Задача заполнения областей изображения нацелена на восстановление некоторой поврежденной или неизвестной области. На входе алгоритм получает поврежденное изображение, а также маску области, где требуется восстановление. На выходе алгоритм выдает восстановленное изображение, заполняя неизвестную область наиболее реалистичным способом.

В последние годы развитие нейросетевых подходов существенно способствовало появлению различных методов решения этой задачи. Однако нейросетевые подходы сильно привязаны к разрешению, на котором их обучали, из-за недостатка рецептивного поля. Большинство моделей имеет размер входа не превышающий 512 пикселей. В результате они не могут обрабатывать изображения произвольной формы, например, в интерактивных инструментах обработки изображений. Когда разрешение входного изображения повышается, у большинства подходов начинают проявляться значительные артефакты. Пример приведен на рис. 1.

В этой статье предлагается метод заполнения областей, не зависящий от разрешения. Он ис-

пользует coarse-to-fine подход, восстанавливая структуру изображения на низком разрешении и текстуру на высоком. Также, для улучшения качества текстурного заполнения, мы предлагаем использовать сдвиги исходного изображения, тем самым искусственно увеличивая рецептивное поле на величину сдвига. Наш подход теоретически применим к любому существующему методу без необходимости переобучения.

### 2. СВЯЗАННЫЕ ИССЛЕДОВАНИЯ

Решение поставленной задачи возможно классическими подходами, ориентированными на подбор наиболее подходящего блока для неизвестной области [1–4] из исходного изображения и итеративном заполнении. Такие методы хорошо справляются с заполнением текстурных фрагментов, но плохо работают на сложных структурных объектах.

Развитие нейронных сетей привело к появлению различных глубоких методов [5–8]. Такие алгоритмы используют знания, полученные во время обучения из некоторого множества приме-

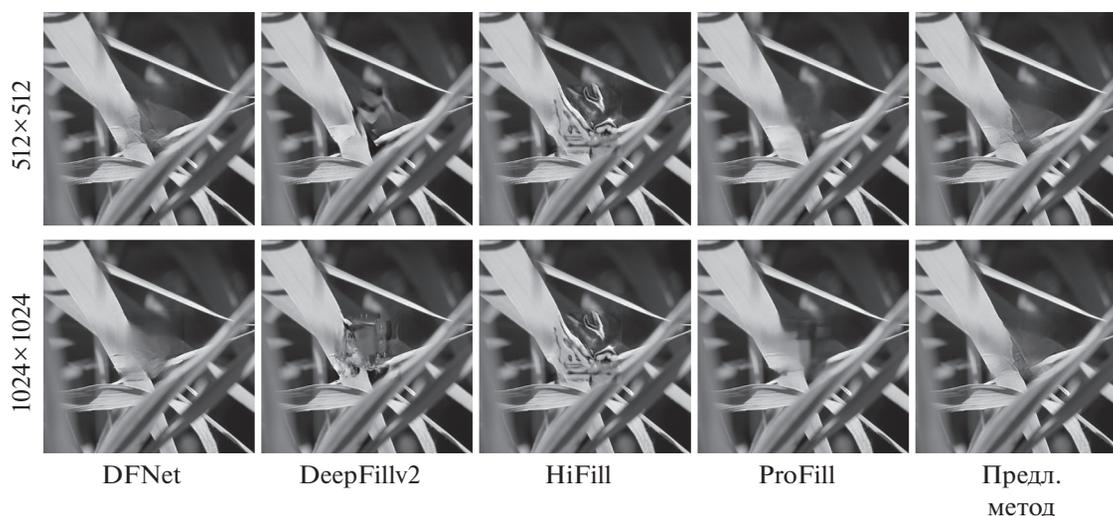


Рис. 1. Пример работы методов при изменении разрешения (маскирован центральный квадрат).

ров. Они лучше классических алгоритмов восстанавливают структурные особенности изображения.

Сложность к обучению нейросетевых подходов добавляет то, что предлагаемые заполнения области могут быть совершенно различными и при этом все быть достаточно реалистичными. Имея исходное изображение становится трудным задать подходящую функцию потерь, наилучшим образом отражающую зрительное восприятие. Как показано в [9], для этой задачи нейросетевые признаки становятся лучшим способом оценки качества заполнения.

Появление генеративно-сопоставительных сетей (GAN) [10] легло в основу создания генеративных методов заполнения изображений [5, 6, 8, 11, 12], которые используют сопоставительную функцию потерь как один из компонентов.

### 2.1. Модуль контекстного внимания и маскированные свертки

В [11], исследователи предложили заменить классические свертки на маскированные, являющиеся продолжением идей [5]. Также они предложили модуль контекстного внимания, являющегося дифференцируемым аналогом алгоритма выбора наиболее подходящего блока для заполнения неизвестной области. Метод реализован как генеративно-сопоставительная нейросеть. Вместо использования вычислительно сложного модуля контекстного внимания в этой работе мы предлагаем использовать обычные сдвиги исходного изображения как средство заполнения текстуры.

### 2.2. Глубокая сеть с модулем слияния

Авторы [7] разработали блок слияния, позволяющий нейросети производить альфа-смешивание каждого пикселя в соответствии с предсказанной альфа-картой признаков, полученных на разных разрешениях. Они реализовали архитектуру U-Net [13], используя высокоуровневые нейросетевые признаки из VGG16 [14] в качестве функции потерь. Мы используем предобученную DFNet на первой стадии нашего метода, тем самым восстанавливая структуру неизвестной области на низком разрешении. Мы не применяли блоки слияния в нашей уточняющей сети на втором этапе восстановления текстур, поскольку он изменяет даже немаскированную область.

### 2.3. Восстановление областей высокого разрешения

В [12] исследователи предложили подход, модифицирующий маскированные свертки, которые требуют значительных вычислительных ресурсов, уменьшив число параметров в нейросети. Также они предложили разделить изображение на высокие и низкие частоты путем вычитания из изображения его версию с использованием фильтра блюра и использования модифицированного блока контекстного внимания для финальной агрегации. Авторы обучали сеть восстановления на небольших, но полноразмерных изображениях. Цель же предлагаемой сети лишь в восстановлении текстурной составляющей, поэтому мы обучали её на небольших случайных фрагментах исходного изображения, используя полноразмерное изображение только при тестировании.

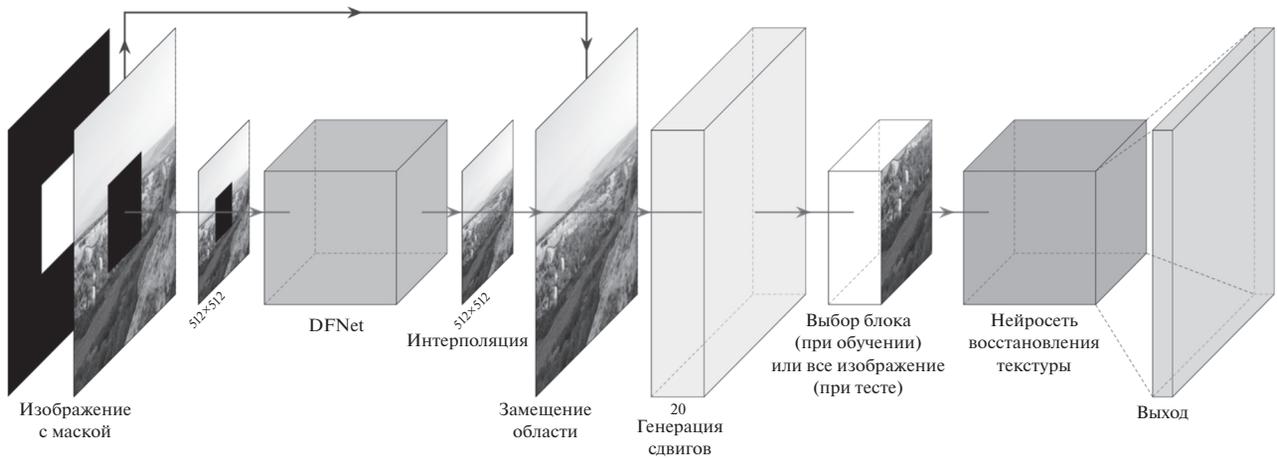


Рис. 2. Общая схема функционирования.

### 3. ПРЕДЛАГАЕМЫЙ МЕТОД

#### 3.1. Подготовка данных

Для обучения мы взяли все изображения из набора DIV2K [15]. Из каждого изображения мы вырезали случайный наибольший квадрат. Всего в обучающий набор данных вошло 7.218 изображений, в валидационный – 1.650. К каждому изображению мы применили случайную маску из [5]. Для тестирования мы использовали естественные изображения с квадратной маской в центре.

#### 3.2. Общая схема алгоритма

##### Первая стадия

На первой стадии наш алгоритм восстанавливает структуру изображения на низком разрешении. Сначала мы уменьшаем исходное изображение и маску до  $512 \times 512$ , затем применяем предобученную DFNet [7] для получения грубого результата заполнения на низком разрешении. Затем мы производим интерполяцию изображения до начального разрешения и выполняем замещение известной области из исходного изображения. Наконец, мы генерируем сдвиги исходного изображения в четырех направлениях: влево, вправо, вниз, вверх. В экспериментах использовались сдвиги на 20% от ширины изображения в пикселях. Отметим, что во время этого процесса мы также пересчитываем маски, помечая все пиксели в областях открытия как невалидные. Таким образом, после первой стадии получается 20-канальное изображение: пять RGB изображений (основное и четыре сдвига) и пять масок.

##### Вторая стадия

Вторая стадия нацелена на восстановление текстуры. Для предотвращения привязанности сети восстановления к структуре и ускорения процесса

обучения, мы вырезаем случайный  $512 \times 512$  квадрат из исходного тензора глубины 20 такой, что-маскированная область составляет не менее 10%, но не более 90% от размера этого квадрата.

Отметим, что во время тестирования мы пропускаем стадию выбора патча и применяем сеть для изображения в полном разрешении. Выход сети восстановления является финальным результатом заполнения. Иллюстрация с общей схемой приведена на рис. 2.

#### 3.3. Архитектура сети восстановления текстуры

Предложенная архитектура построена на базе сети U-Net [13] (подробная иллюстрация архитектуры доступна на рис. 3). Отметим, что после каждого слоя, кроме последнего, мы использовали нормализацию из [16]. На иллюстрации отражены размеры сверток кодировщика, размеры всех сверток декодировщика  $3 \times 3$ .

#### 3.4. Функция потерь

Мы обучали нашу сеть в не генеративно-сопоставительной манере. Аналогично методу предложенному в [7] в качестве функции потерь мы использовали линейную комбинацию:

$$\mathcal{L} = 0.1 \cdot \mathcal{L}_v + 6.0 \cdot \mathcal{L}_1 + 0.1 \cdot \mathcal{L}_p + 240.0 \cdot \mathcal{L}_s, \quad (3.1)$$

где для исходного изображения  $\mathbf{I}$  и предсказанного  $\hat{\mathbf{I}}$ :  $\mathcal{L}_v$  – total variation расстояние внутри маскированной области,  $\mathcal{L}_1$  – расстояние, рассчитываемое как

$$\mathcal{L}_1 = \frac{1}{CHW} \|\mathbf{I} - \hat{\mathbf{I}}\|_1, \quad (3.2)$$

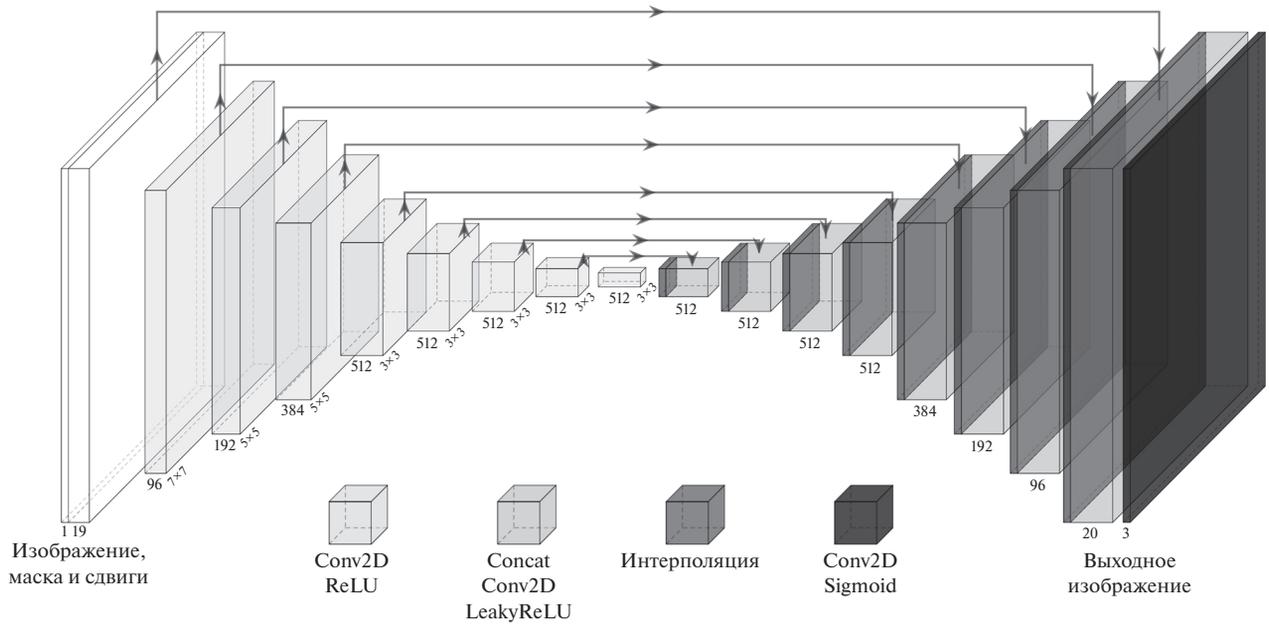


Рис. 3. Архитектура сети восстановления текстуры.

где  $C$ ,  $W$ ,  $H$  это число каналов, ширина и высота изображения соответственно.  $L_p$ ,  $L_s$  – Perceptual и Style функции потерь [17]:

$$\mathcal{L}_p = \sum_{j \in J} \|\psi_j(\mathbf{I}) - \psi_j(\hat{\mathbf{I}})\|_1 \quad (3.3)$$

$$\mathcal{L}_s = \sum_{j \in J} \|G_j(\mathbf{I}) - G_j(\hat{\mathbf{I}})\|_1, \quad (3.4)$$

где  $J$  определяет набор индексов в нейросети VGG16,  $\psi_j$  определяет  $j$ -й признаковый слой. И  $G_j$  определяет матрицу Грамма  $j$ -го признакового слоя.

#### 4. ЭКСПЕРИМЕНТЫ

Нейросеть обучалась с использованием оптимизатора Adam [18] со стандартными настройками. Обучение заняло два дня на двух видеоускорителях Nvidia Tesla P100 с размером батча в 18 изображений. Отметим, что модификации в процессе обучения подвергались только веса, относящиеся ко второй стадии нейросети – для оптимизации мы посчитали вывод сети из первой стадии отдельно.

##### 4.1. Сравнения

Для сравнений использовался набор из 34 естественных изображений из [9] с полным разрешением в  $2048 \times 2048$ . Для уменьшения разрешения использовалась интерполяция ближайшего соседа. На рис. 1, 6 представлены примеры работы предлагаемого метода в сравнении с аналогами.

Иллюстрации приведены для разрешения  $1024 \times 1024$ , так как метод обрабатывает почти идентично для всех разрешений.

##### Экспертная оценка

Участникам показывали по два изображения и предлагали выбрать то, которое обладало более высоким визуальным качеством. Мы также добавили два проверочных вопроса для исключения недобросовестных зрителей. Из-за ограничений реализации метода [8] мы провели два отдельных сравнения на разрешениях  $2048 \times 2048$  и  $1024 \times 1024$  соответственно. В первом сравнении приняло участие 150 зрителей, было получено 3.750 голоса. Второе сравнение прошло 248 участников, было получено 6200 голосов. Затем полученные парные оценки были переведены в численные с помощью модели Bradley-Terry [19]. Результаты субъективных сравнений представлены на рис. 4

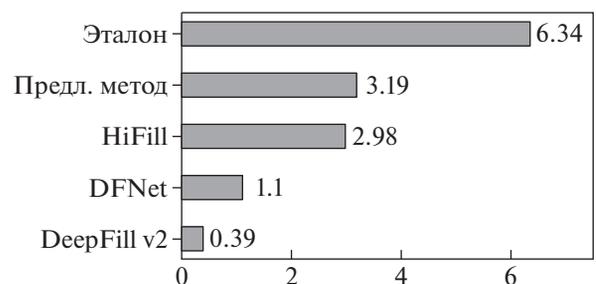


Рис. 4. Результаты субъективного сравнения (разрешение  $2048 \times 2048$ ).

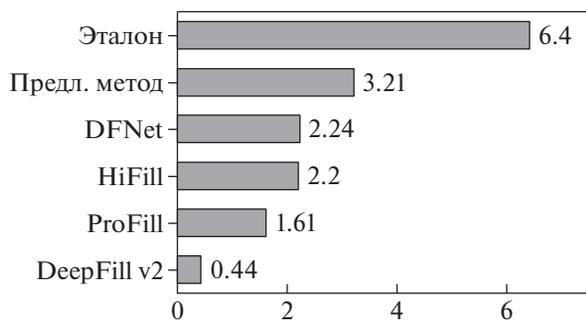


Рис. 5. Результаты субъективного сравнения (разрешение  $1024 \times 1024$ ).

и 5. Предложенный алгоритм показал наиболее высокое визуальное качество на обоих разрешениях в сравнении с аналогами.

**Объективное сравнение**

Из-за сложности получения экспертных оценок мы провели объективные сравнения для других разрешений, хотя это может не быть подтверждено зрительскими оценками. В объективное сравнение мы также добавили результаты метода

Adobe Photoshop 2020, являющегося коммерческим продуктом, реализующим классический метод заполнения. В качестве метрик были выбраны расстояние L1, SSIM [20] и PSNR. Для уменьшения разрешения мы использовали интерполяцию ближайшего соседа. Результаты объективного сравнения представлены в табл. 1. Предложенный алгоритм показал лучшие значения метрик качества, чем аналоги.

**5. ЗАКЛЮЧЕНИЕ**

В этой работе представлен метод заполнения областей изображений различного разрешения. Мы провели субъективное и объективное сравнения с существующими нейросетевыми подходами, показав, что предложенный метод способен показать более удовлетворительное качество заполнения. Также наш подход применим к уже существующим методам без необходимости их переобучения. Используя [21], предлагаемый метод был встроен в среду интерактивной обработки изображений GIMP. Плагин, код модели, а также полученные в сравнениях изображения доступны в репозитории [https://github.com/a-mos/High\\_Resolution\\_Image\\_Inpainting](https://github.com/a-mos/High_Resolution_Image_Inpainting).

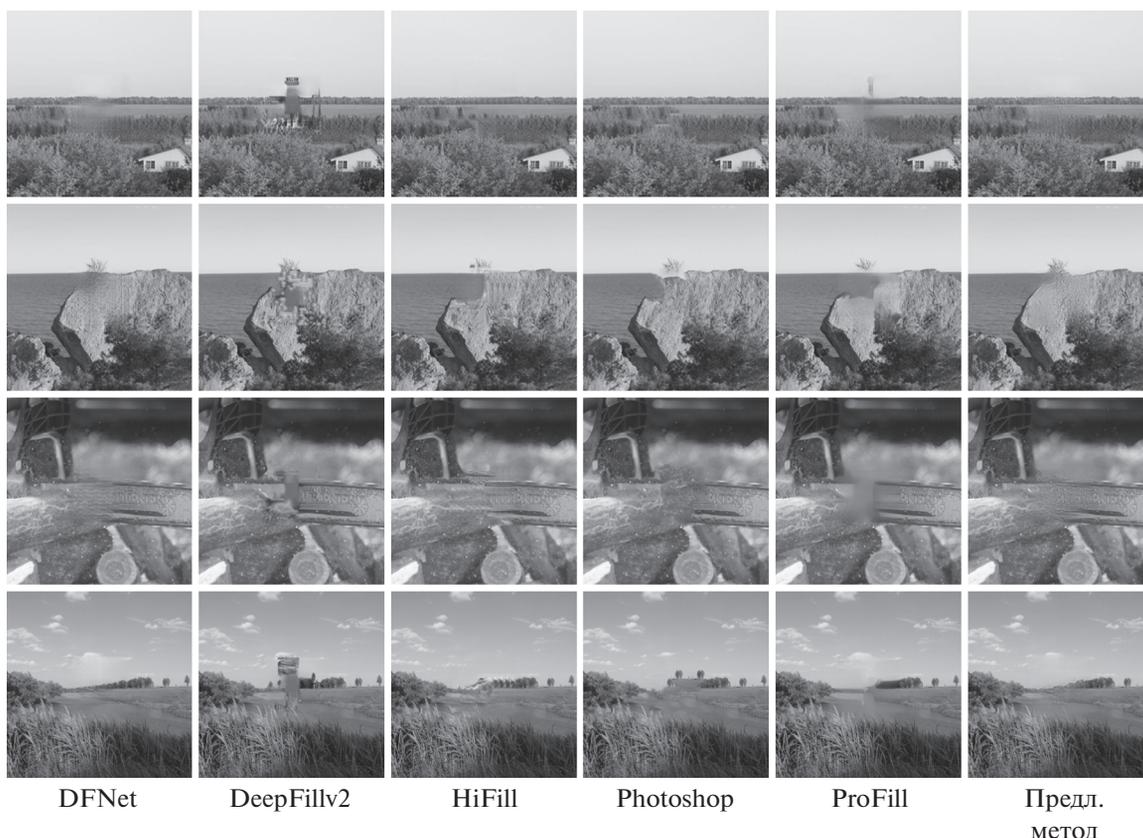


Рис. 6. Примеры работы протестированных методов ( $1024 \times 1024$ ).

Таблица 1. Результаты объективного сравнения

Модель Метрика	L1	PSNR	SSIM		L1	PSNR	SSIM
DeepFill v2	4.981	22.389	0.938		5.397	21.956	0.944
DFNet	3.592	24.910	0.946		4.132	23.836	0.946
HiFill	4.422	23.667	0.935		4.373	23.738	0.943
Photoshop 2020	4.115	23.789	0.944		13.320	23.756	0.949
ProFill	3.906	24.677	<b>0.950</b>		—	—	—
Ours	<b>3.524</b>	<b>25.175</b>	0.944		<b>3.474</b>	<b>25.311</b>	<b>0.950</b>
Разрешение	1024 × 1024				1536 × 1536		
Модель Метрика	L1	PSNR	SSIM		L1	PSNR	SSIM
DeepFill v2	5.546	21.834	0.946		5.669	21.697	0.948
DFNet	4.345	23.424	0.948		4.633	22.915	0.950
HiFill	4.403	23.696	0.944		4.391	23.710	0.947
Photoshop 2020	4.047	23.863	<b>0.952</b>		4.153	23.659	0.952
Предл. метод	<b>3.447</b>	<b>25.374</b>	<b>0.952</b>		<b>3.434</b>	<b>25.412</b>	<b>0.954</b>
Разрешение	1792 × 1792				2048 × 2048		

## 6. БЛАГОДАРНОСТИ

Исследования выполнены при финансовой поддержке РФФИ в рамках научного проекта 19-01-00785 а. Для обучения моделей использовался вычислительный кластер IBM Polus факультета вычислительной математики и кибернетики Московского государственного университета имени М.В. Ломоносова.

## СПИСОК ЛИТЕРАТУРЫ

1. Drori I., Cohen-Or D., Yeshurun H. Fragment-based image completion // ACM Transactions on Graphics. 2003. V. 22. № 3. P. 303–312.
2. Criminisi A., Perez P., Toyama K. Region filling and object removal by exemplar-based image inpainting // IEEE Transactions on Image Processing. 2004. V. 13. № 9. P. 1200–1212.
3. Barnes C., Shechtman E., Finkelstein A., Goldman D. PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing // ACM Transactions on Graphics (Proc. SIGGRAPH). 2009. V. 28. № 3.
4. Yakubenko A.A., Kononov V.A., Mizin I.S., Konushin V.S., Konushin A.S. Reconstruction of Structure and Texture of City Building Facades // Programming and Computer Software. 2011. V. 37. № 5. P. 260–269.
5. Liu G., Reda F.A., Shih K.J., Wang T.C., Tao A., Catanzaro B. Image Inpainting for Irregular Holes Using Partial Convolutions // The European Conference on Computer Vision (ECCV). 2018. P. 85–100.
6. Yu J., Lin Z., Yang J., Shen X., Lu X., Huang T.S. Generative Image Inpainting with Contextual Attention // IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2018. P. 5005–5514.
7. Hong X., Xiong P., Ji R., Fan H. Deep Fusion Network for Image Completion // Proceedings of the 27th ACM International Conference on Multimedia. 2019. P. 2033–2042.
8. Zeng Y., Lin Z., Yang J., Zhang J., Shechtman E., Lu H. High-Resolution Image Inpainting with Iterative Confidence Feedback and Guided Upsampling // arXiv preprint arXiv:2005.11742, 2020.
9. Molodetskikh I., Erofeev M., Vatolin D. Perceptually Motivated Method for Image Inpainting Comparison // CEUR Workshop Proceedings. 2019. V. 2485. P. 131–135.
10. Goodfellow I., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y. Generative adversarial nets // Advances in neural information processing systems. 2014. V. 27. P. 2672–2680.
11. Yu J., Lin Z., Yang J., Shen X., Lu X., Huang T. Free-Form Image Inpainting With Gated Convolution // Proceedings of the IEEE International Conference on Computer Vision. 2019. P. 4471–4480.
12. Yi Z., Tang Q., Azizi S., Jang D., Xu Z. Contextual Residual Aggregation for Ultra High-Resolution Image Inpainting // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020. P. 7508–7517.
13. Ronneberger O., Fischer P., Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation // International Conference on Medical image computing and computer-assisted intervention. 2015. P. 234–241.
14. Simonyan K., Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition // arXiv preprint arXiv:1409.1556, 2014.
15. Timofte R., Gu S., Wu J., Van Gool L., Zhang L., Yang M.H. NTIRE 2018 Challenge on Single Image Super-Resolution: Methods and Results // The IEEE Conference

- on Computer Vision and Pattern Recognition (CVPR) Workshops. 2018. P. 965.
16. *Ioffe S., Szegedy C.* Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift // arXiv preprint arXiv:1502.03167, 2016.
  17. *Johnson J., Alahi A., Fei-Fei L.* Perceptual Losses for Real-Time Style Transfer and Super-Resolution // European conference on computer vision. 2016. P. 694–711.
  18. *Kingma D.P., Ba J.* Adam: A Method for Stochastic Optimization // arXiv preprint arXiv:1412.6980, 2014.
  19. *Bradley R.A., Terry M.E.* Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons // *Biometrika*. 1952. V. 39. № 3/4. P. 324–345.
  20. *Wang Z., Bovik A., Sheikh H., Simoncelli E.* Image Quality Assessment: From Error Visibility to Structural Similarity // *IEEE transactions on image processing*. 2004. V. 13. № 4. P. 600–612.
  21. *Soman K.* GIMP-ML: Python Plugins for using Computer Vision Models in GIMP // arXiv preprint arXiv:2004.13060, 2020.

## ГИБРИДНЫЙ МЕТОД ПОДАВЛЕНИЯ ОСЦИЛЛЯЦИЙ ГИББСА НА ИЗОБРАЖЕНИЯХ МАГНИТНО-РЕЗОНАНСНОЙ ТОМОГРАФИИ

© 2021 г. М. А. Пенкин<sup>а,\*</sup>, А. С. Крылов<sup>а,\*\*</sup>, А. В. Хвостиков<sup>а,\*\*\*</sup>

<sup>а</sup> Московский государственный университет имени М.В. Ломоносова,  
Факультет вычислительной математики и кибернетики,  
Лаборатория математических методов обработки изображений,  
119991 Москва, Ленинские горы, д. 1, Россия

\*E-mail: penkin97@gmail.com

\*\*E-mail: kryl@cs.msu.ru

\*\*\*E-mail: khvostikov@cs.msu.ru

Поступила в редакцию 10.10.2020 г.

После доработки 20.10.2020 г.

Принята к публикации 12.01.2021 г.

Подавление артефактов ложного оконтуривания на изображениях (эффектов ложного оконтуривания, *англ.* ginging) — это распространенная задача области восстановления изображений. Осцилляции Гиббса возникают из-за методики визуализации изображений магнитно-резонансной томографии, при которой исходные данные, поступающие в частотной области, отображаются в пространственную область с помощью дискретного преобразования Фурье. Появление осцилляций Гиббса обусловлено неполнотой получаемой информации, связанной в том числе с обрезкой высоких частот Фурье-спектра. В данной статье предлагается гибридный метод подавления артефактов ложного оконтуривания на изображениях магнитно-резонансной томографии, заключающийся в объединении моделей глубокого машинного обучения и классического необучаемого алгоритма подавления осцилляций Гиббса, основанного на поиске оптимальных субпиксельных сдвигов.

DOI: 10.31857/S0132347421030080

### 1. ВВЕДЕНИЕ

Подавление осцилляций Гиббса (ложного оконтуривания, *англ.* ginging) — актуальная задача математических методов обработки изображений. Они часто наблюдаются при изменении разрешения изображений, при повышении резкости изображений, а также при визуализации данных магнитно-резонансной томографии (МРТ) (см. рис. 1). Повышение качества МРТ изображений является важной для медицинской диагностики задачей, например, при работе с атласами мозга [1].

Феномен Гиббса был открыт Генри Уилбрахамом в 1848 г., а затем повторно открыт Дж. Уиллардом Гиббсом в 1898 г. Интересным является факт, что в отличие от других математических областей, исследование эффекта Гиббса было не очень активным вплоть до 1977 года [2]. Затем работы на эту тему стали появляться чаще [3–5]. Отчасти такая динамика объясняется развитием магнитно-резонансной томографии (МРТ), основанной в 1973 году, когда профессор химии Пол Лотербург опубликовал в журнале Nature статью “Создание изображения с помощью индуцированного локального взаимодействия; примеры на основе

магнитного резонанса”. Математические методы получения изображений МРТ были затем усовершенствованы Питером Мэнсфилдом, и в 2003 году обоим исследователям была присуждена Нобелевская премия по физиологии и медицине.

Для демонстрации причин осцилляций Гиббса с математической точки зрения, рассмотрим сле-

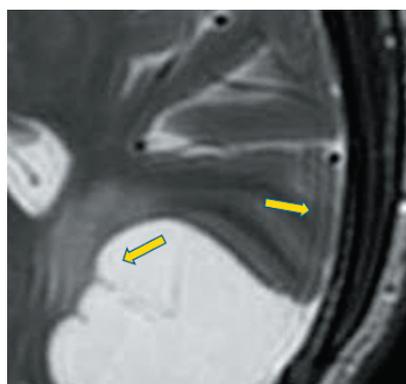


Рис. 1. Примеры осцилляций Гиббса на изображениях МРТ (указаны стрелками).

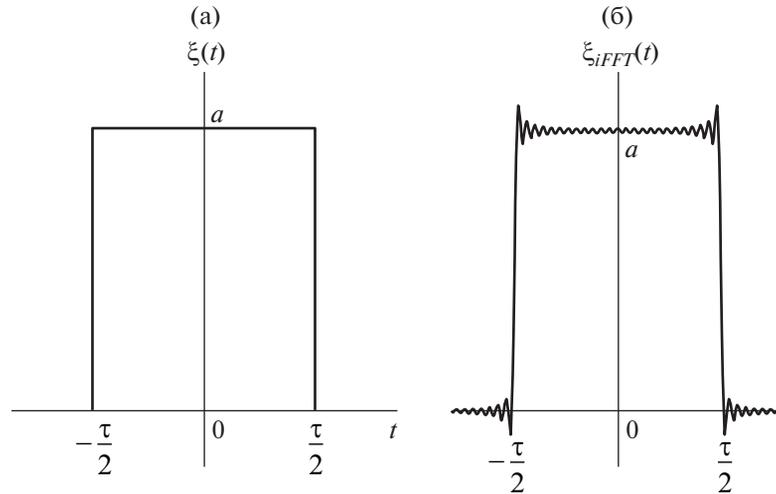


Рис. 2. Пример осцилляций Гиббса: (а) – исходная функция  $\xi(t)$ , (б) – функция, соответствующая обрезанному Фурье-спектру функции (а).

дующую вещественную периодичную модельную функцию:

$$\xi(t) = \begin{cases} a, & t \in [-\tau/2, \tau/2] \\ 0, & t \in [-T/2, T/2] \setminus [-\tau/2, \tau/2] \end{cases}, \quad (1.1)$$

$$\xi(t) = \xi(t + T),$$

где  $\xi(t)$  – модельный пример конечного разрыва;  $a$  – величина скачка;  $T$  – период модельной функции;  $\tau$  – ширина импульса.

Используя преобразование Фурье в комплексной форме и предполагая что  $T = 2\tau$ , (1.1) может быть записано в виде:

$$\xi(t) = \sum_{k=-\infty}^{+\infty} d_k e^{i\omega_k t}, \quad (1.2)$$

где  $d_k = \frac{1}{T} \int_{-\tau/2}^{\tau/2} \xi(t) e^{-i\omega_k t} dt$ ,  $\omega_k = \Omega k$ ,  $\Omega = 2\pi/T$ .

$$\begin{aligned} \xi(t) &= 2 \cdot \frac{a}{2} \sum_{k=0}^{+\infty} (-1)^k \cdot \frac{1}{(2k+1)\pi} \cdot \cos(2k+1) \cdot \frac{2\pi}{T} t = \\ &= \frac{a}{\pi} \left( \cos \Omega t - \frac{1}{3} \cos 3\Omega t + \dots \right), \end{aligned} \quad (1.3)$$

где  $\Omega = 2\pi/T$ .

На практике часто используется не полный Фурье-спектр сигнала, а лишь его часть для минимизации времени сбора данных, что приводит к осцилляциям Гиббса, характерным для аппроксимации тригонометрическим рядом функций с разрывами. В таких случаях говорят, что проявляется эффект ложного оконтуривания (см. рис. 2).

Важно заметить, что максимальная амплитуда осцилляций Гиббса постоянна и не зависит от выбранной частоты обрезки ряда [6].

Для подавления осцилляций Гиббса на изображениях были предложены различные подходы: как с использованием необучаемых классических алгоритмов, так и с применением обучаемых глубоких свёрточных нейронных сетей.

Одним из наиболее эффективных подходов, основанных на применении необучаемых моделей, является работа [7], в которой ложное оконтуривание подавляется с помощью поиска оптимальных субпиксельных сдвигов. Этот метод был выбран нами в качестве классического алгоритма в предлагаемом гибридном подходе. Для краткости будем называть его – алгоритм Кельнера. Предложенный метод авторы сравнивали с другими классическими подходами, например, с медианной фильтрацией, фильтрацией Ланцоша – и визуально было доказано превосходство описанного в статье [7] алгоритма Кельнера.

Другим распространенным подходом к уменьшению артефактов ложного оконтуривания на изображениях является решение задачи вариационного исчисления по поиску оптимальной функции, минимизирующей целевой функционал, среди функций некоторого предопределенного класса (например,  $L_1$  или  $L_2$ ). Постановка задачи выглядит следующим образом:

$$J(u) = \frac{1}{2} \|u - u_0\|^2 + \lambda \int_{\Omega} |\nabla u(x)| dx \rightarrow \min_{u \in U}, \quad (1.4)$$

где  $u_0$  – входное изображение с артефактами ложного оконтуривания;  $u$  – искомое оптимальное изображение среди изображений класса  $U$ ;  $\Omega$  – область задания изображения;  $\lambda$  – параметр регуляризации (может зависеть от ближайшей контрастной границы, как например в [8]).

Также, известны методы одновременного подавления осцилляций Гиббса и оценки степени подавляемого ложного оконтуривания с использованием разреженных представлений [9].

В настоящее время все большую популярность приобретают методы глубокого обучения в области математических методов обработки изображений. В первую очередь – свёрточные нейронные сети. Свёрточные сети представляют возможность строить отображения изображений в обучаемые признаковые пространства большой размерности, фильтровать полученные признаки и преобразовывать обработанные признаки представления в целевое цветовое пространство. Свёрточные нейронные сети успешно решают массу прикладных задач обработки биомедицинских изображений и компьютерного зрения, таких как сегментация биомедицинских изображений [10, 11], уменьшение уровня шума на медицинских изображениях [12], и, в том числе, подавление осцилляций Гиббса на изображениях магнитно-резонансной томографии [13–15].

Одной из актуальных свёрточных моделей подавления артефактов ложного оконтуривания на изображениях МРТ, является GAS-CNN [14], которая была выбрана в качестве базовой. С результатами, полученными этой моделью, сравнивается предлагаемый гибридный подход.

GAS-CNN представляет собой очень глубокий свёрточный ансамбль (32 свёрточных блока), являющийся развитием модели увеличения разрешения EDSR [16]. Основными отличительными особенностями GAS-CNN стали:

- добавление внешних пространственных связей (*англ.* skip connectios), подобных используемым в архитектуре U-Net [10] между кодировщиком и декодировщиком;
- уменьшение размерности признакового пространства, в котором происходит фильтрация изображения (в GAS-CNN оно равно 64);
- отказ от использования слоев уменьшения пространственной размерности, например, свёртки с шагом 2 или слоя субдискретизации с функцией максимума (*англ.* max pooling), в силу локальности артефактов ложного оконтуривания.

GAS-CNN отображает входное изображение с осцилляциями Гиббса в признаковое пространство глубины 64, в котором затем осуществляется фильтрация с помощью 32-х остаточных свёрточных блоков (*англ.* ResBlock) [17]. Сеть преобразует ставшее результатом свёрточной фильтрации признаковое представление в изображение целевого цветового пространства – оттенки серого (*англ.* greyscale), с помощью заключительного реконструирующего свёрточного слоя.

Авторы GAS-CNN провели сравнение своей модели с билатеральной фильтрацией [18], нелокаль-

ной фильтрацией (*англ.* non-local means filtering) [19] и свёрточной нейронной сетью GARCNN [13].

Тем не менее, авторы GAS-CNN не исследовали возможную избыточность обучения 32-х свёрточных блоков для решения задачи уменьшения осцилляций Гиббса на изображениях МРТ. В данной статье предлагается новый, гибридный метод подавления артефактов ложного оконтуривания на изображениях магнитно-резонансной томографии, заключающийся в совместном использовании обучаемой глубокой свёрточной нейронной сети и необучаемого классического метода подавления осцилляций Гиббса. В качестве обучаемой части используется упрощение модели GAS-CNN, а в качестве необучаемой классической части выступает алгоритм Кельнера. Обучаемая часть предлагаемой модели состоит из 9 свёрточных блоков, вместо 32-ух у GAS-CNN, однако, благодаря использованию вспомогательных признаков от предварительно обработанных алгоритмом Кельнера входных изображений, наша гибридная модель сохраняет обобщающую способность на тестовом наборе данных. Такой гибридный подход, основывающийся на совместном использовании обучаемого и классического алгоритмов, ведет к улучшению итогового результата как визуально, так и по метрике оценки качества изображений PSNR [20]. Уменьшение числа свёрточных блоков также уменьшает риск переобучения модели.

## 2. АЛГОРИТМ КЕЛЬНЕРА

Алгоритм Кельнера [7] – это неитеративный необучаемый метод подавления осцилляций Гиббса на изображениях, основанный на поиске оптимальных субпиксельных сдвигов с целью минимизации функционала, оценивающего величину осцилляций. В качестве такого функционала используется полная вариация:

$$TV(f, \Omega) = \int_{\Omega} |\nabla f(x)| dx, \quad (2.1)$$

где  $f$  – функция, полная вариация ( $TV$ ) которой вычисляется в области  $\Omega$ .

Метод имеет два параметра  $k_1, k_2$ , определяющих “радиус”  $[k_1, k_2]$  пиксельного окна, в котором минимизируется полная вариация при различных субпиксельных сдвигах. Алгоритм Кельнера позволяет эффективно уменьшать эффект Гиббса с минимальным итоговым размытием, и его вычислительная сложность невелика.

Метод Кельнера осуществляет сначала поиск оптимальных субпиксельных сдвигов по двум одномерным направлениям: по горизонтали и по вертикали, а затем совмещает полученные результаты в Фурье области по следующей формуле:

$$K = \text{iFFT} \{ \text{FFT}(K_x) \cdot G_x + \text{FFT}(K_y) \cdot G_y \}, \quad (2.2)$$

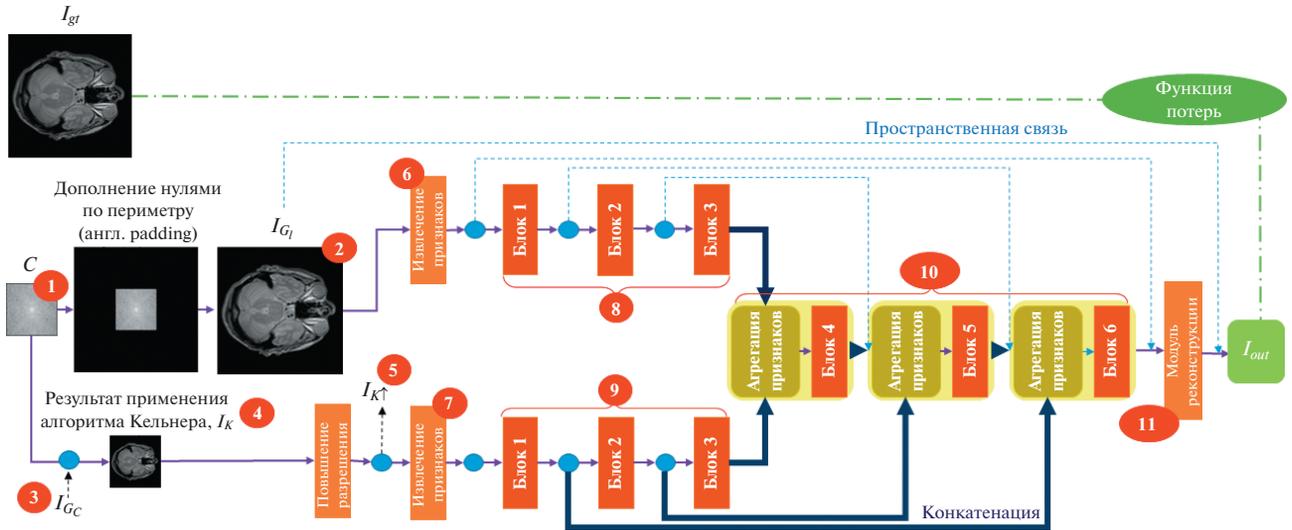


Рис. 3. Схема предлагаемой гибридной архитектуры подавления осцилляций Гиббса – DGAS9-CNN.

где  $K$  – итоговый результат алгоритма Кельнера;  $K_x$  – результат одномерного алгоритма Кельнера вдоль оси  $OX$ ;  $K_y$  – результат одномерного алгоритма Кельнера вдоль оси  $OY$ ; FFT – прямое дискретное преобразование Фурье; iFFT – обратное дискретное преобразование Фурье;  $G_x, G_y$  – весовые функции, имеющие вид:

$$G_x(t_x, t_y) = \frac{1 + \cos t_y}{(1 + \cos t_x) + (1 + \cos t_y)}, \quad (2.3)$$

$$G_y(t_x, t_y) = \frac{1 + \cos t_x}{(1 + \cos t_x) + (1 + \cos t_y)}, \quad (2.4)$$

где  $t_x \in [-\pi, \pi]$ ,  $t_y \in [-\pi, \pi]$ .

Рассмотрим подробнее одномерный алгоритм Кельнера с математической точки зрения. Пусть  $I$  – исходный сигнал и  $\{C_k\}$  –  $k$ -я комплексная амплитуда. Тогда, исходный сигнал можно приблизить следующим разложением в дискретный ряд Фурье:

$$I(n) = \frac{1}{N} \sum_{k=0}^{N-1} C_k e^{\frac{2\pi i k n}{N}}. \quad (2.5)$$

Для каждого пикселя  $n$  с помощью теоремы о сдвиге преобразования Фурье получим  $2M$  его значений в окрестности для дальнейшего выбора оптимального субпиксельного сдвига:

$$I_s(n) = \frac{1}{N} \sum_{k=0}^{N-1} \left( C_k \cdot e^{\frac{2\pi i k \cdot s}{N \cdot 2M}} \right) \cdot e^{\frac{2\pi i k n}{N}}, \quad (2.6)$$

где  $s \in [-M, M - 1]$ ,  $s \in \mathbb{Z}$ . Авторы метода Кельнера предлагают произвести сначала поиск оптимального сдвига вдоль положительного и отрицательного направлений независимо:

$$TV_s^+(n) = \sum_{k=k_1}^{k_2} |I_s(n+k) - I_s(n+(k-1))|, \quad (2.7)$$

$$TV_s^-(n) = \sum_{k=k_1}^{k_2} |I_s(n-k) - I_s(n-(k-1))|, \quad (2.8)$$

$$r^+(n) = \arg \min_s TV_s^+(n), \quad (2.9)$$

$$r^-(n) = \arg \min_s TV_s^-(n), \quad (2.10)$$

где  $k_1, k_2$  – параметры алгоритма, определяющие “радиус” пиксельной окрестности, по которой минимизируется полная вариация при различных субпиксельных сдвигах. Авторы пришли к выводу, что для большинства рассматриваемых ими прикладных случаев оптимальными входными параметрами являются:  $k_1 = 1, k_2 = 3$ .

А затем из двух полученных оптимальных сдвигов  $r^+(n), r^-(n)$  выбрать тот, что обеспечивает минимальную полную вариацию:

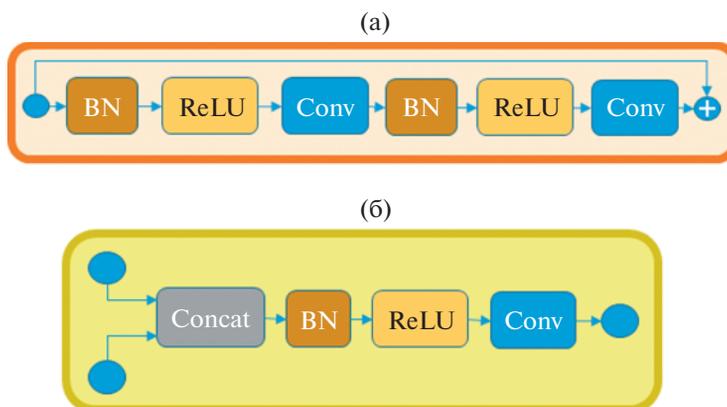
$$r(n) = \arg \min_{r^+, r^-} \{TV_{r^+(n)}^+(n), TV_{r^-(n)}^-(n)\} \quad (2.11)$$

Окончательное значение в полученной точке  $n - r(n)/(2M)$  вычисляется авторами с помощью линейной интерполяции.

### 3. ПРЕДЛАГАЕМЫЙ МЕТОД

Предлагаемая архитектура представлена на рис. 3.

Будем называть ее DGAS9-CNN (англ. Dual Gibbs-ringing artifact suppression CNN).



**Рис. 4.** (а) – остаточный блок; (б) – блок агрегации признаков. Conv – свёрточный слой; BN – пакетная нормализация; ReLU – функция активации ReLU; Concat – слой конкатенации вдоль последней размерности (форма тензора описывается по стандарту: BHWС).

Предлагаемая гибридная модель:  $C \rightarrow I_{out}$ , имеет следующую структуру (номера соответствуют номерам на рис. 3):

1. Вход: обрезанный спектр Фурье ( $C$ ).
2. Получение изображения с артефактами Гиббса, являющегося входным в свёрточную нейронную сеть:  $C \rightarrow I_{G_i}$ :
  - дополнение нулями по периметру (*англ.* zero-padding)  $C$ , для соответствия размерам референсного изображения без осцилляций Гиббса  $I_{gt}$ ;
  - применение обратного преобразования Фурье.
3. Получение изображения с осцилляциями Гиббса, являющегося входным в алгоритм Кельнера:  $C \rightarrow I_{G_c}$ :
  - применение обратного преобразования Фурье к входному спектру  $C$  без дополнения его нулями по периметру в силу требований классического метода.
4. Применение алгоритма Кельнера к  $I_{G_c}$ :  $I_{G_c} \rightarrow I_K$ .
5. Повышение разрешения  $I_K$  до соответствия с размерами референсного изображения:  $I_K \rightarrow I_{K\uparrow}$ .
6. Модуль преобразования изображения  $I_{G_i}$  в признаковое представление глубины 64.
7. Модуль преобразования изображения  $I_{K\uparrow}$  в признаковое представление глубины 64.
8. Модуль глубокого кодирования признаков изображения  $I_{G_i}$  в признаковом пространстве глубины 64.
9. Модуль глубокого кодирования признаков изображения  $I_{K\uparrow}$  в признаковом пространстве глубины 64.

10. Модуль восстановления признаков целевого изображения  $I_{out}$ , в котором агрегируются признаки изображений  $I_{G_i}$  и  $I_{K\uparrow}$ .

11. Модуль реконструкции целевого изображения без эффекта Гиббса  $I_{out}$ .

Предлагаемая гибридная модель DGAS9-CNN имеет 9 структурных остаточных блоков: 3 блока в глубоком кодировщике признаков изображения  $I_{G_i}$ , 3 блока в глубоком кодировщике признаков изображения  $I_{K\uparrow}$  и 3 блока в модуле восстановления признаков целевого изображения  $I_{out}$ . Схема остаточного блока предлагаемой модели приведена на рис. 4 (а).

В [17] показано, что использование остаточных блоков в качестве структурных архитектурных составляющих позволяет преодолеть одну из основных проблем обучения глубоких свёрточных нейронных сетей – затухание градиента [21]. Стоит отметить, что выбранная базовая модель GAS-CNN также состоит из остаточных блоков, однако в отличие от GAS-CNN используемый нами остаточный блок включает в себя слои пакетной нормализации [22] (*англ.* batch normalization), позволяющие еще более эффективно вести обучение глубоких архитектур [23].

Особенностью модуля восстановления признаков целевого изображения  $I_{out}$  является блок агрегации признаков изображений  $I_{G_i}$  и  $I_{K\uparrow}$ . Его схема приведена на рис. 4 (б). Он представляет свёртку с ядром  $3 \times 3$  предварительно конкатенированных и нормализованных представлений изображений  $I_{G_i}$  и  $I_{K\uparrow}$ .

Извлечение признаков представлений изображений  $I_{G_i}$  и  $I_{K\uparrow}$  предлагаемая архитектура реализует с помощью одного свёрточного слоя ( $3 \times 3$ ) с

последующей пакетной нормализацией и активацией ReLU.

Повышение разрешения результата алгоритма Кельнера  $I_K$  реализуется интерполяцией методом ближайшего соседа с последующими двумя обучаемыми свёрточными слоями с ядрами  $(3 \times 3)$ .

Заключительный модуль реконструкции реализован через два последовательных свёрточных слоя с ядрами  $(3 \times 3)$ , осуществляющих:

- промежуточное проецирование обработанных признаков глубины 64 на подпространство меньшей размерности – 16;
- отображение полученного тензора в целевое изображение без эффекта Гиббса  $I_{out}$ .

#### 4. ЭКСПЕРИМЕНТЫ

Предлагаемая гибридная модель DGAS9-CNN была обучена и протестирована на синтезированных изображениях с фиксированной степенью осцилляций Гиббса, полученных прямым и обратным преобразованиями Фурье из набора данных IXI<sup>1</sup> референсных изображений  $I_{gt}$  по следующей схеме [14]:

1. Применить преобразование Фурье к очередному референсному изображению  $I_{gt}$  ( $256 \times 256$ ) из набора данных IXI.

2. Обрезать спектр Фурье таким образом, чтобы сохранилась лишь центральная часть частот, составляющая  $\frac{1}{9}$  площади (см. изображение  $C$  на рис. 3).

3. Получить изображение ( $I_{G_i}$ ) с осцилляциями Гиббса, являющееся входным в свёрточную нейронную сеть:

- дополнить нулями по периметру спектр  $C$  до соответствия с размерами референсного изображения  $I_{gt}$ ;
- применить обратное преобразование Фурье.

4. Получить изображение ( $I_{G_c}$ ), являющееся входным для алгоритма Кельнера:

- применить обратное преобразование Фурье к обрезанному спектру  $C$ .

В результате тренировочный, валидационный и тестовый наборы данных представляют собой множество триплетов:  $(I_{gt}, I_{G_i}, I_K)$ . Тренировочный сет состоит из 10 427 триплетов, валидационный – из 2016 триплетов и тестовый – из 2617 триплетов [14].

Следует отметить, что набор данных IXI содержит МРТ-данные трех различных модальностей: T1, T2, PD, поэтому для приведения значений пикселей изображений к единому диапазону [0,

1] была использована стандартная Min-Max нормализация каждого триплета  $(I_{gt}, I_{G_i}, I_K)$ .

Авторы базовой статьи GAS-CNN [14] не предоставили кода и весов, так что для корректного сравнения мы обучили все сравниваемые здесь модели самостоятельно. Программная реализация сравниваемых здесь архитектур написана на языке Python 3 с использованием фреймворка глубокого обучения Tensorflow 1.14<sup>2</sup>.

Модели обучались с помощью метода оптимизации Adam [24] с параметрами:  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ ,  $\epsilon = 1e-08$  на графическом ускорителе GPU NVIDIA GeForce RTX 2080 Ti. Сравнимые здесь модели оптимизировались с точки зрения классической регрессионной, устойчивой к выбросам, функции потерь  $l_1$  с  $l_2$  регуляризацией ( $\gamma = 10^{-4}$ ). Скорость обучения менялась полиномиально:

$$lr(x) = (lr_0 - lr_1) \cdot \left(1 - \frac{x}{M}\right)^p + lr_1, \quad (4.1)$$

где  $x$  – очередная итерация обучения;  $lr_0$  – исходное значение скорости обучения;  $lr_1$  – конечное значение скорости обучения;  $M = N_e \cdot N_d / bs$  – общее количество итераций обучения;  $N_e$  – число эпох обучения;  $N_d$  – число триплетов в тренировочном наборе данных;  $bs$  – пакет триплетов, подаваемых модели на очередной итерации обучения (*англ.* batch size);  $p$  – степень полинома, с которой ведется полиномиальное уменьшения скорости обучения.

Непосредственно, были использованы следующие значения представленных выше параметров:  $lr_0 = 10^{-4}$ ,  $lr_1 = 0$ ,  $N_e = 1000$ ,  $bs = 20$ ,  $p = 0.3$ .

Важно отметить, что для борьбы с потенциальной проблемой переобучения применялась не только  $l_2$  регуляризация, но и аугментация входных данных: случайными вращениями на углы, кратные 90, и отражениями по горизонтали и по вертикали. Модели обучались на случайно вырезаемых в процессе обучения патчах размера  $48 \times 48$ .

Отдельное внимание было уделено возможной избыточности обучения 32-х свёрточных блоков для решения задачи подавления осцилляций Гиббса на изображениях магнитно-резонансной томографии.

Для этого, сначала была обучена сильно упрощенная версия GAS-CNN с 6-ю свёрточными блоками (помимо модулей выделения признаков представлений входного изображения и модуля реконструкции целевого изображения – эти модули сохраняют свою архитектуру для всех исследуемых здесь моделей) – GAS6-CNN, повто-

<sup>1</sup> <http://brain-development.org/ixi-dataset/>

<sup>2</sup> <https://github.com/MaksimPenkin/DGAS9-CNN>

ряющая остаточную архитектуру GAS-CNN, однако имеющая 6 свёрточных остаточных блоков вместо исходных 32-х. Полученные средние значения метрики оценки качества изображений PSNR на тестовом наборе данных для GAS-CNN и GAS6-CNN отражены в табл. 1, а динамика средних значений PSNR на валидационном наборе данных во время обучения показана на рис. 5. Видно, что такое значительное уменьшение числа слоев модели при равных условиях обучения привело к потере обобщающей способности на тестовом наборе данных.

Затем, добавив к GAS6-CNN дополнительные признаки, кодирующие обработанные классическим алгоритмом Кельнера входные изображения, уровень обобщающей способности модели на тестовом наборе данных повысился и сравнился с результатом работы GAS-CNN, имеющей почти в 3.5 раза больше свёрточных остаточных блоков. (см. табл. 1 и рис. 5).

Для оценки вклада алгоритма Кельнера в гибридный метод, мы дополнительно обучили модель DGAS9-CNN, полностью повторяющую структуру предлагаемой DGAS9-CNN, однако без использования алгоритма Кельнера. В ней изображение  $I_K$  совпадает с  $I_G$ . Полученный результат, как можно видеть из табл. 1 и рис. 5, уступает DGAS9-CNN и на тестовом наборе данных, и на валидационной выборке.

На центральном процессоре Intel(R) Core(TM) i7-8700 базовый подход GAS-CNN обрабатывает одно изображение  $256 \times 256$  за 1.03 сек, в то время как предлагаемая нами гибридная модель DGAS9-CNN – в полтора раза быстрее, за 0.68 с (см. табл. 1), обеспечивая сравнимую с GAS-CNN обобщающую способность на тестовом наборе данных (см. табл. 1).

Визуально оценить результаты применения рассматриваемых алгоритмов можно на рис. 6.

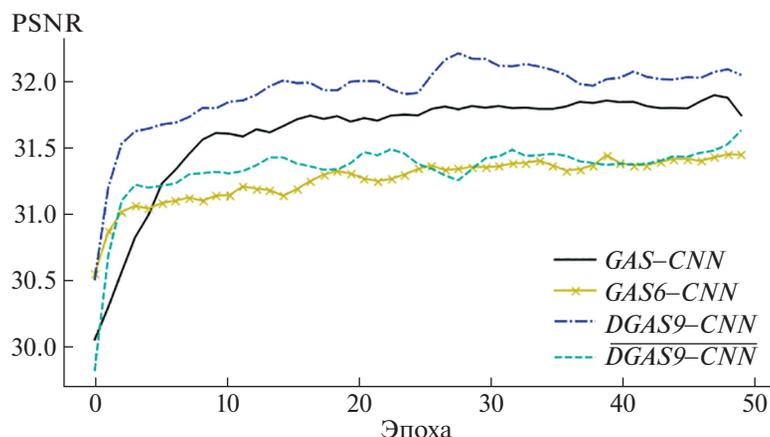
**Таблица 1.** Средние значения PSNR на тестовом наборе данных из 2617 изображений и средние времена обработки одного изображения на ЦПУ

Модель	PSNR	Время, с
Исходные изображения	29.79	–
GAS-CNN	32.04	1.03
GAS6-CNN	31.67	0.25
DGAS9-CNN	32.27	0.68
DGAS9-CNN	31.61	0.45

**Таблица 2.** Средние значения PSNR и полной вариации на тестовом наборе данных из 2617 изображений. Изображения приведены к единому размеру  $I_K$

Модель	PSNR	TV
Исходные изображения	20.70	659.68
Алгоритм Кельнера	21.68	540.18
DGAS9-CNN	29.57	611.25
GAS-CNN	29.46	620.09

Результат применения алгоритма Кельнера имеет наименьшую полную вариацию среди рассматриваемых алгоритмов, однако, минимизируя полную вариацию, метод также подавляет и детали на изображениях, что приводит к приросту только в 1 дБ метрики PSNR (см. табл. 2). Результаты применения предлагаемого гибридного алгоритма DGAS9-CNN и базового GAS-CNN имеют также меньшую полную вариацию в сравнении с исходными входными изображениями, однако они значительно лучше восстанавливают детали на изображениях, увеличивая PSNR: на 8.87 и 8.76 дБ, соответственно (см. табл. 2).



**Рис. 5.** Средние значения PSNR на валидационном наборе из 2016 изображений во время обучения.

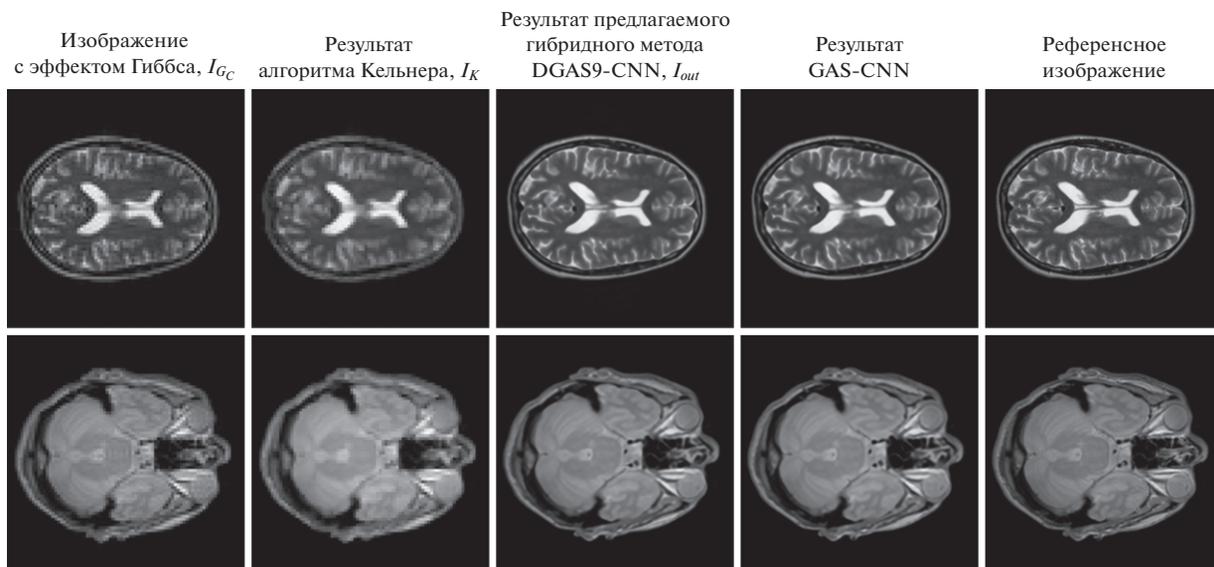


Рис. 6. Визуальное сравнение алгоритмов.

### 5. ЗАКЛЮЧЕНИЕ

В статье представлена гибридная модель DGAS9-CNN подавления осцилляций Гиббса на изображениях магнитно-резонансной томографии. Предлагаемый метод состоит из агрегации классического неитеративного подхода подавления осцилляций Гиббса – алгоритма Кельнера, и легкой свёрточной нейронной сети, упрощающей архитектуру сети GAS-CNN. Предлагаемая гибридная модель имеет в 3.5 раза меньше свёрточных остаточных блоков, в сравнении с GAS-CNN, однако позволяет достичь лучшие, чем GAS-CNN результаты, обрабатывая изображения в 1.5 раза быстрее.

### СПИСОК ЛИТЕРАТУРЫ

1. *Senyukova O., Zubov A.* Full anatomical labeling of magnetic resonance images of human brain by registration with multiple atlases // *Programming and Computer Software*. 2016. V. 46. № 6. P. 356–360.
2. *Gottlieb D., Orszag S.* Numerical Analysis of Spectral Methods: Theory and Application. SIAM, 1977. 176 p.
3. *Gray A., Pinsky M.* Gibbs phenomenon for Fourier-Bessel series // *Expositiones Mathematicae*. 1993. V. 11. 123 p.
4. *Pinsky M.A.* Fourier inversion for piecewise smooth functions in several variables // *Proceedings of the American Mathematical Society*. 1993. V. 118. № 3. P. 903–910.
5. *Pinsky M.A.* Pointwise Fourier inversion in several variables // *Notices of the American Mathematical Society*. 1995. V. 42. № 3. P. 330–334.
6. *Малла С.* Вэйвлеты в обработке сигналов. Пер. с англ. М.: Мир, 2005. 671 с.
7. *Kellner E., Dhital B., Kiselev V.G., Reiser M.* Gibbs ringing artifact removal based on local subvoxel-shifts // *Magnetic Resonance in Medicine*. 2016. V. 76. № 5. P. 1574–1581.
8. *Sitdikov I.T., Krylov A.S.* Variational Image Deringing Using Varying Regularization Parameter // *Pattern Recognition and Image Analysis: Advances in Mathematical Theory and Applications*. 2015. V. 25. № 1. P. 96–100.
9. *Umnov A.V., Krylov A.S.* Sparse Approach to Image Ringing Detection and Suppression // *Pattern Recognition and Image Analysis: Advances in Mathematical Theory and Applications*. 2017. V. 27. № 4. P. 754–762.
10. *Ronneberger O., Fischer P., Brox T.* U-net: Convolutional networks for biomedical image segmentation // *International Conference MICCAI 2015*. 2015. P. 234–241.
11. *Sinha A., Dolz J.* Multi-scale self-guided attention for medical image segmentation // *IEEE Journal of Biomedical and Health Informatics*. 2021. V. 25. № 1. P. 121–130.
12. *Krylov A., Karnaukhov V., Mamaev N., Khvostikov A.* Hybrid Method for Biomedical Image Denoising // *Proceedings of the 2019 4th International Conference on Biomedical Imaging, Signal Processing*. 2019. P. 60–64.
13. *Wang Y., Song Y., Xie H. et al.* Reduction of Gibbs artifacts in magnetic resonance imaging based on Convolutional Neural Network // *2017 10th International Congress on Image and Signal Processing, Biomedical Engineering and Informatics (CISP-BMEI)*. 2017. P. 1–5.
14. *Zhao X., Zhang H., Zhou Y. et al.* Gibbs-ringing artifact suppression with knowledge transfer from natural images to MR images // *Multimedia Tools and Applications*. 2019. P. 1–23.
15. *Penkin M., Krylov A., Khvostikov A.* Attention-based Convolutional Neural Network for MRI Gibbs-ringing

- Artifact Suppression // CEUR Workshop Proceedings. 2020. V. 2744. P. 1–12.
16. *Lim B., Son S., Kim H. et al.* Enhanced deep residual networks for single image super-resolution // Proceedings of the CVPR IEEE Conference. 2017. P. 136–144.
  17. *He K., Zhang X., Ren S., Sun J.* Deep residual learning for image recognition // Proceedings of the CVPR IEEE Conference. 2016. P. 770–778.
  18. *Zhang M., Gunturk B.K.* Multiresolution bilateral filtering for image denoising // IEEE Transactions on Image Processing. 2008. V. 17. № 12. P. 2324–2333.
  19. *Manj'on J.V., Coupé P., Buades A. et al.* Non-local MRI upsampling // Medical Image Analysis. 2010. V. 14. № 6. P. 784–792.
  20. *Hore A., Ziou D.* Image quality metrics: PSNR vs. SSIM // 2010 20th International Conference on Pattern Recognition. 2010. P. 2366–2369.
  21. *Hanin B.* Which neural net architectures give rise to exploding and vanishing gradients? // Advances in Neural Information Processing Systems. 2018. P. 582–591.
  22. *Ioffe S., Szegedy C.* Batch normalization: Accelerating deep network training by reducing internal covariate shift // arXiv preprint arXiv:1502.03167, 2015.
  23. *Santurkar S., Tsipras D., Ilyas A., Madry A.* How does batch normalization help optimization? // Advances in Neural Information Processing Systems. 2018. P. 2483–2493.
  24. *Kingma D.P., Ba J.* Adam: A method for stochastic optimization // arXiv preprint arXiv:1412.6980, 2014.