

ОТКРЫТОЕ СОРЕВНОВАНИЕ РЕНДЕР-СИСТЕМ

В.А. ФРОЛОВ^{1,2}, Д.С. ПАВЛОВ², М.А. ТРОФИМОВ², П.А. КАЗБЕЕВ²,

В.А. ГАЛАКТИОНОВ^{1*}

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

²Московский Государственный университет имени М.В. Ломоносова. Москва, Россия

*Ответственный автор. E-mail: vlgal@gin.keldysh.ru

DOI: 10.20948/mathmontis-2019-45-9

Ключевые слова: Фотореалистичный рендеринг, интеграл освещённости, трассировка лучей методом Монте-Карло.

Аннотация. Сравнение алгоритмов и методов — краеугольный камень любого исследования, позволяющий обосновать, что предложенный подход делает шаг вперёд по сравнению с предыдущими исследованиями. В таких областях, как компьютерное зрение или машинное обучение, создано огромное количество открытых наборов данных, на которых исследователи могут тестировать свои алгоритмы и соревноваться друг с другом. В компьютерной графике и фотореалистичном рендеринге в настоящий момент ситуация обстоит по-другому. Открытых наборов сцен, на которых различные исследователи в разных рендер-системах могли бы получать совпадающие (или хотя бы близкие) изображения, не существует. Это приводит к тому, что в научных работах практикуется т. н. “cherry-picking” — педантичный выбор сцен и условий освещения таким образом, чтобы продемонстрировать преимущества разработанного алгоритма. Такой подход сильно понижает практическую значимость исследований — даже если новый метод хорошо работает на определённом классе сцен, это ещё не значит что от него можно добиться хороших результатов в других случаях. По этой причине многие рендер-системы до сих не выходят за рамки базовых алгоритмов, не доверяя результатам исследований.

В данной работе мы начали заполнять этот пробел. Нами был создан специальный набор сцен (т. н. «бенчмарк»), позволяющий оценить производительность интегрирования освещения в различных ситуациях и, таким образом, проявить положительные и отрицательные стороны рендер-систем и алгоритмов, используемых ими в различных условиях (изображения некоторых сцен представлены на рис. 1). Мы воссоздали многие сцены из известных статей по компьютерной графике и добавили некоторые сцены, исходя из своего опыта. Наша цель — сделать как можно более полное покрытие, задействовав как можно меньше сцен и функциональности рендер-системы для того, чтобы такое сравнение было легко воспроизвести в любой существующей системе.

Для валидации нашего подхода было проведено пилотное сравнение среди 4-х популярных продуктов для Autodesk 3ds Max (VRay, Corona, Octane, Hydra Renderer) на скорость интегрирования освещения на различных сценариях. Несмотря на то, что 3 из 4-х систем являются закрытыми коммерческими продуктами, нам удалось получить для всех сцен совпадающие или близкие изображения, что свидетельствует о жизнеспособности предлагаемого нами подхода сравнения.

2010 Mathematics Subject Classification: 68U05, 68M20, 62M05.

Key words and Phrases: Photorealistic rendering, integration of luminance, Monte-Carlo ray tracing.

OPEN RENDERING BENCHMARK

V. FROLOV^{1,2}, D. PAVLOV², M. TROFIMOV², P. KAZBEEV²,
V. GAVAKTIONOV^{1*}

¹Keldysh Institute of Applied Mathematics Russian Academy of Sciences
Moscow, Russia

² Moscow State University (MSU). Moscow, Russia

*Corresponding author. E-mail: vlgal@gin.keldysh.ru

DOI: 10.20948/mathmontis-2019-45-9

Summary. Comparison of algorithms and methods is the cornerstone of any research which allows to prove that the proposed approach takes a step forward with a previous one. In areas such as computer vision or machine learning a huge amount of open data sets have been created, in which researchers can test their algorithms and compete with each other. In computer graphics and photorealistic rendering at the current moment the situation is different. Open sets of scenes in which different researchers using different render systems could get matching (or at least close) images do not exist. This leads to the fact that in scientific papers so-called “Cherry-picking” is practiced. This one is a pedantic selection of scenes and lighting conditions in such a way as to demonstrate the advantages of the developed algorithm. Such approach greatly reduces the practical significance of research - even if the new method works well on a certain class of scenes, it does not mean it will work for other cases. For this reason, many render systems still do not go beyond the basic algorithms, not trusting the results of research.

In this paper, we began to fill this gap. We have created a special set of scenes (the so-called “benchmark”), which allows us to evaluate the performance of lighting integration in various situations and, thus, to show the positive and negative aspects of render systems and algorithms used by them in various conditions. We recreated many scenes from well-known computer graphics papers and added some scenes based on our experience. Our goal is to make the most complete coverage possible, using as few scenes and functionality of the render system as possible so that such a comparison can be easily reproduced in any existing system.

To validate our approach, we conducted a pilot comparison among 4 popular products for 3D Studio Max (VRay, Corona, Octane, Hydra Renderer) for the speed of integrating the lighting on various scenarios. Despite the fact that 3 out of 4 systems are closed commercial products, we managed to get the same or similar images for all scenes, which indicates the viability of our proposed approach.

1 ВВЕДЕНИЕ

Сравнение систем расчёта глобального освещения по многим причинам является нетривиальной задачей. На конференции SIGGRAPH 2018 известный учёный Ярослав Криванек выделил следующие фундаментальные проблемы, возникающие при проведении такого сравнения [1]:

1. В современном фотореалистичном рендеринге не существует стандартов. Все рендер-системы вычисляют в пределах разные эталонные изображения.

2. Коммерческие продукты зачастую реализуют огромное количество “физически некорректных” функций. Причём, каждая рендер-система делает это по-своему. Пример подобной функциональности — карты нормалей (normal mapping) [2].

3. Наконец, метрика качества для изображений, определяющая качество изображения в настоящий момент является открытой научной проблемой [3–5].

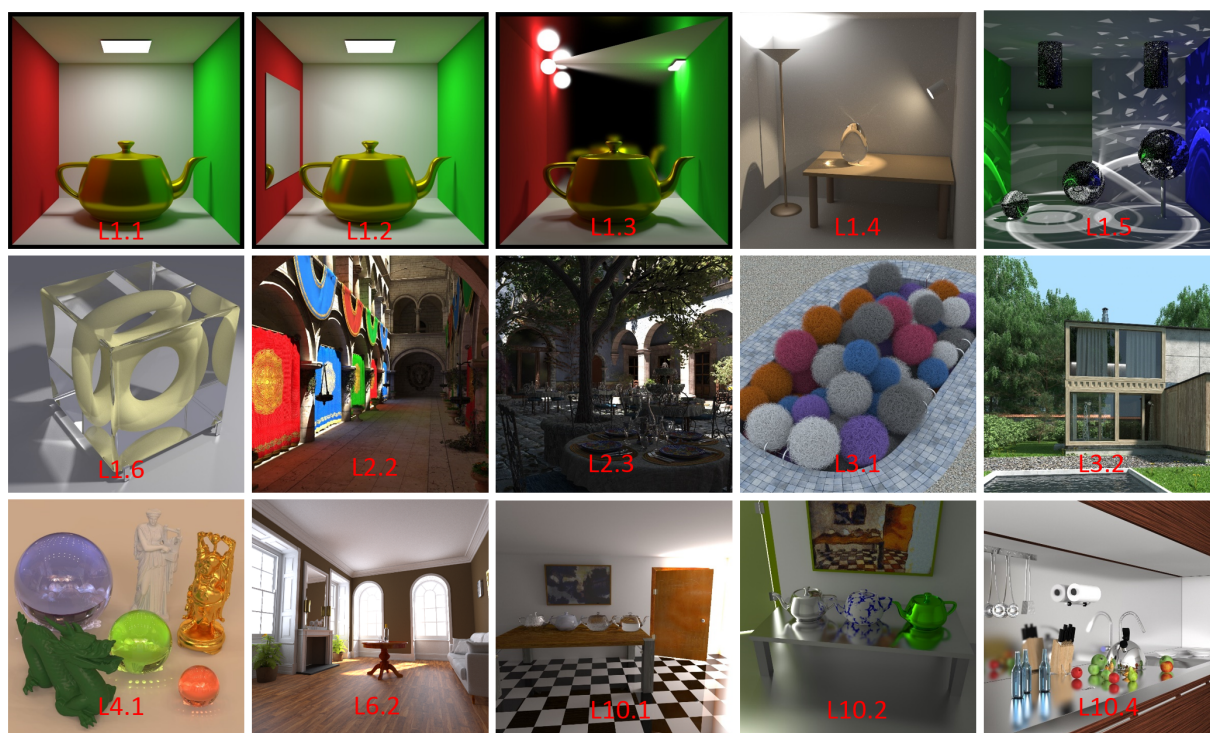


Рис. 1: Изображения некоторых сцен из созданного набора.

Краткий обзор существующих работ свидетельствует о том, что сравнения зачастую проводятся на нескольких сценах, выбранных случайным образом [6–8]. Это не позволяет проанализировать характеристики устойчивости рендера к различным сценариям: массивная геометрия, оптически сложные материалы, сложное освещение, наличие трудновычисляемых феноменов освещённости [9-11]. Время расчёта, которое при этом пишется в подобных сравнениях, в действительности не демонстрирует ровным счётом ничего, поскольку качество изображения фиксируется “на глаз” лишь приблизительно. В то же время, едва заметная для глаза разница в шуме на изображении может означать разницу в скорости в разы. Это особенно верно при приближении к “финальному качеству”, когда изображение становится максимально чистым, поскольку качество в физически корректных рендерах, как правило, пропорционально квадрату времени расчёта.

Открытые наборы сцен [12,13], с другой стороны, как правило, сохраняют сцены в разрозненном виде. В лучшем случае из них можно получить информацию о геометрии и некоторое изображение, полученное неизвестной рендер-системой в прошлом.

Материалы и условия освещения, при которых возникают, например, трудновычислимые феномены освещённости при этом безнадежно теряются.

В 2014 году впервые был представлен [14] более точный анализ производительности для нескольких рендер-систем под Autodesk 3ds Max на 7 различных упрощённых сценариях. Для того чтобы решить проблему отличающихся эталонов, в этой работе предлагается использовать для каждой системы свой собственный эталон, получаемый именно этой системой за достаточно большое время и не содержащий шума [14]. При этом считается, что изображения эталонов визуальнo совпадают. То есть рендер-системы отличаются лишь в “реализации отдельных незначительных моментов”. Общий уровень яркости, геометрическая форма объектов на изображениях и характер освещения должны совпадать. Далее вводился некоторый индекс производительности, зависящий от среднеквадратичной ошибки (MSE) между изображением за некоторое фиксированное время и эталоном. Таким образом, этой в работе фактически измеряется то, насколько быстро алгоритм/реализация/рендер-система сходится к своему собственному эталону [14].

Хотя в этом сравнении сцен было существенно больше, чем в существующих ранее сравнениях, данный набор нельзя считать объективным. Кроме того, среди использованных сцен не было задействовано ни одной популярной в научном сообществе сцены. Это делает такое сравнение менее интересным для специалистов, т. к. не позволяет хотя бы приблизительно сопоставить существующие реализации в статьях и индустриальных программах.

В данной работе мы ставим задачу создания т. н. “бенчмарка” — способа объективного сравнения производительности различных алгоритмов интегрирования освещённости и их реализаций. Наша глобальная цель заключается в том, чтобы предоставить возможность сравнения производительности всех существующих рендер-систем в мире, а также отдельных алгоритмов интегрирования освещённости и их реализаций, решающих эту проблему численно и с высокой точностью (т. е. мы не рассматриваем приближённые алгоритмы глобального освещения).

2 ПОСТАНОВКА ЗАДАЧИ И ОБОСНОВАНИЕ ВЫБОРА РЕНДЕР СИСТЕМ

2.1 Постановка задачи

С математической точки зрения все рендер-системы в той или иной мере решают проблему глобальной освещённости [15] (формула 1). Эта проблема сводится к вычислению многомерного интеграла, называемого интегралом освещённости (соответственно решение этой проблемы называют “интегрированием освещённости”). Размерность интеграла зависит от количества переотражений света, которые необходимо учитывать, т. к. значение функций L в одних точках трёхмерной сцены зависит от значения самого интеграла освещённости в других точках (формула 1).

$$I(\phi_r, \theta_r) = \int_{\phi_i, \theta_i} L(\phi_i, \theta_i) R(\phi_i, \theta_i, \phi_r, \theta_r) \cos(n, l_{\phi_i, \theta_i}) d\phi_i d\theta_i \quad (1)$$

Формула 1. Уравнение рендеринга или т. н. интеграл освещённости [15]. Пара (ϕ_r, θ_r) задаёт направление на наблюдателя, пара (ϕ_i, θ_i) — направление на источник освещения. Функция L — яркость падающего освещения, R — двунаправленная функция отражательной способности (ДФО). Важно отметить, что значение функций L в одних точках трёхмерной сцены зависит от значения самого интеграла освещённости в других точках сцены. Поэтому данная проблема трудная, и эквивалентна решению интегрального уравнения Фредгольма 2 рода.

Системы Corona [16] и V-Ray [17] являются адептами “старой школы”. Они работают на CPU (Central Processing Unit или центральный процессор) и для ускорения используют так называемые алгоритмы со смещённым решением [18–22]. В этих алгоритмах численная точность не ставится во главу угла. Кроме того, они не обладают способностью достигать правильного решения в пределе, что можно считать недостатком. С другой стороны, типичный пользователь рендер-системы в Autodesk 3ds Max беспокоится не столько о правильности решения, сколько о реалистичности и красоте. Однако здесь есть нюанс: смещённые алгоритмы зачастую дают артефакты, неприемлемые для пользователя [23].

Ocane [24] и Hydra [25] во многом противоположны рассмотренным ранее программам Corona и V-Ray. Они ориентированы на расчёт без допущений (несмещённая оценка) и работают полностью на GPU. Кроме того, обе системы используют методы рендеринга на основе Марковских цепей. Программа Ocane использует алгоритм PMC/ERPT [26,27] (Population Monte-Carlo/Energy Redistribution Path Tracing), Hydra – MMLT [18] (Multiplexed Metropolis Light Transport).

Отдельно следует сказать несколько слов про рендер-систему Hydra Renderer. Эта система, как и Ocane, позиционирует скорость в качестве своего основного преимущества. Однако, в отличие от остальных является полностью открытым программным обеспечением. Поэтому, хотя она и не является столь распространённой как три оставшиеся программы, для наших целей она подходит. Открытость системы позволяет нам анализировать её исходные коды и используемые алгоритмы для того, чтобы создавать “предельные случаи” – сцены, на которых определённые алгоритмы работают плохо, либо, наоборот хорошо, т. к. они создавались именно для таких сцен.

2.2 Используемые рендер-системы в деталях

Так как V-Ray, Corona и Ocane являются закрытым программным обеспечением, мы не можем со 100% уверенностью утверждать что-либо об их внутренней реализации. Тем не менее, обладая опытом разработки аналогичных систем, анализируя публикации и артефакты, производимые различными алгоритмами в этих системах, можно с достаточной степенью достоверности делать предположения об использовании тех или иных методов рендеринга.

2.2.1 Система V-Ray

Система V-Ray впервые стала доступной для широкой общественности в 2000 году. Ключевыми средствами повышения производительности являются кэш освещенности (называемый в V-Ray термином "Irradiance Maps") и некоторый закрытый алгоритм, называемый в V-Ray термином "Light Cache". Каустики в V-Ray могут рассчитываться при помощи фотонных карт с накоплением [20,21]. Таким образом, алгоритмы, используемые в данной системе, были разработаны в 80–90 годах прошлого века и в настоящий момент могут считаться морально устаревшими (что, тем не менее, не означает, что они не могут быть с успехом использованы).

Начиная с версии 3.5, V-Ray изменила направление развития и перешла в сторону GPU и несмещённых алгоритмов (V-Ray Next). В настоящий момент система проходит бета-тестирование и информации по ней не так много. По этой причине для тестирования GPU за точку отсчёта была выбрана система Octane.

2.2.2 Система Corona

Corona набрала популярность в течение 2012 года. Эта программная система использует такие алгоритмы как Radiance Cache [18,19] и фотонные карты [20,21] но, наряду с этим, реализует более современный метод — Vertex Connection Merging (VCM [22]), являющийся гибридом фотонных карт и двунаправленной трассировки путей (Bidirectional Path Tracing [28]). По сравнению с V-Ray используемые подходы являются более современными (Radiance Cache разрабатывался в 2000-ых годах, а VCM был представлен в 2012 на SIGGRAPH Asia).

2.2.3 Система Octane

Octane ориентирован на расчёт без допущений (unbiased rendering), позиционируется как самая быстрая рендер-система в мире и работает полностью на GPU. Она также впервые появилась в 2012 году. В настоящий момент поддерживаются только GPU с Nvidia CUDA, несмотря на неоднократные заявления о поддержке OpenCL. Одним из существенных преимуществ в плане алгоритмов является использование методов рендеринга на основе Марковских цепей — PMC/ERPT [26,27]. Однако разработчики ограничились лишь однонаправленной версией PMC на основе обратной трассировки путей.

2.2.4 Система Hydra

Hydra Renderer, также как и Octane, ориентирована на расчёт без допущений. Алгоритмы расчёта глобального освещения реализованы целиком на GPU посредством технологии OpenCL. Отличительной особенностью является использование двунаправленной трассировки путей и двунаправленного алгоритма Multiplexed MLT (MMLT), впервые представленного в 2014 году [29]. В отличие от рассмотренных ранее систем Hydra Renderer является полностью открытым программным обеспечением. Таким образом, она позиционируется как некоторое ПО, соединяющее научные разработки в области рендеринга и параллельных вычислений на GPU с индустрией. При этом данная система не обладает таким же большим объёмом функциональности, как упомянутые ранее коммерческие системы. Но в нашем сравнении этого и не

требуется.

3 РАЗРАБОТАННЫЙ НАБОР СЦЕН И ОГРАНИЧЕНИЙ

Одним из вызовов при подготовке сравнения является создание объективного набора тестовых сцен. Здесь существует несколько трудностей, которые необходимо решать.

3.1 Стандартизация, спецификация

Первая проблема, как уже было отмечено, это отсутствие стандарта для физически-корректных рендер-систем. Например, в некоторых системах могут быть реализованы математические модели материалов или источники света, отсутствующие в других. Чтобы минимизировать различие в изображениях и чтобы каждая рендер-система была способна в принципе отобразить эти сцены так, как они должны выглядеть, мы вводим набор ограничений при создании сцен (т. е. стремимся минимизировать используемый набор функций):

1. Материал может быть только следующих типов:
 - i. Полностью диффузный (Ламбертов) материал;
 - ii. идеально зеркальный материал;
 - iii. матовые (glossy) отражения с использованием т. н. “исправленной модели Фонга”;
 - iv. простое стекло (идеально-зеркальное, т.е. не матовое) с использованием формул Френеля для диэлектриков;
 - v. материал, обладающей суммой компонент: (i и ii) или (i и iii);
 - vi. материал, смешивающий компоненты (i и ii) или (i и iii) при помощи формул Френеля для диэлектриков в одной-двух сценах;
 - vii. Допускается использование техники bump mapping в нескольких сценах. При этом разрешается использовать только готовые карты нормалей (а не карты высот, из которых рендер-системы самостоятельно оценивают нормаль).
 - viii. Материал с транслюценцией – пропусканием света под поверхность. Такой материал разрешается использовать только на тонких объектах вроде листьев и штор. Отношение отражённой энергии к пропущенной должно быть строго 1 к 1.
2. Источник света может быть только следующих типов:
 - i. Площадный источник света прямоугольной формы не нулевого размера;
 - ii. сферический источник света не нулевого размера;
 - iii. направленный источник света;
 - iv. точечный источник-прожектор (spot распределение);
 - v. окружение, заданное константой, т.е. цветной фон без текстуры;
 - vi. окружение в виде HDR панорамы в одной-двух сценах.

Выбор подобных ограничений – это компромисс между упрощением для исследователей и интересностью полученных результатов для индустрии. Изначально мы планировали оставить в материалах только пункты i – iv, а источники ограничить пунктами i и ii. Это бы значительно упростило реализацию алгоритмов для будущих исследователей (т.к. такой набор позволяет однозначно и легко реализовать физически корректный рендеринг) и позволило бы более легко добиться совпадения изображений в нашем сравнении. Однако в действительности это было бы слишком сильным упрощением задачи глобального освещения по сути, поскольку:

1. Материалы, смешивающие несколько элементарных компонент, используются повсеместно. Кроме того, многокомпонентные материалы увеличивают размерность пространства интегрирования, поскольку интегратору необходимо делать выбор между компонентами и использовать под это ещё одно случайное число, ещё одну размерность. Поэтому запрет такой функциональности был бы слишком сильным упрощением. С другой стороны, полноценное дерево компонент выглядит чрезмерно сложным для оценки производительности. Поэтому мы ограничили такое смешивание всего двумя заранее известными компонентами.

2. Формулы Френеля для смешивания компонент тоже используются слишком часто, чтобы их не рассматривать. Тем не менее, мы старались сократить число сцен, в которых данный вид смешивания был применён, чтобы можно было оценить производительность на большинстве сцен без реализации этой функциональности (Fresnel Blend в терминах PBRT).

3. Bump Mapping и карты нормалей [2] существенно повышают реалистичность и повсеместно используются. Кроме того, такие алгоритмы как фотонные карты и однонаправленный Kelemen Metropolis Light Transport [30] теряют свои преимущества при наличии микрорельефа в том или ином виде. Это делает их значительно менее полезными на практике, и отчасти именно поэтому они не часто используются в индустрии. Поэтому с нашей точки зрения системы и алгоритмы, которые будут не вполне справляться с микрорельефом, нужно штрафовать, а значит, эта функциональность должна быть включена. Тем не менее, мы старались не использовать карты нормалей слишком часто, т. к. они представляют собой отдельную функциональность рендера и, строго говоря, нарушают физическую корректность интерпретации сцены, поскольку изменяют нормаль к поверхности, не меняя при этом геометрию поверхности.

4. Транслюценция (просвечивание) необходима для реалистичной имитации сцен с растительностью. Заменять их полностью диффузными материалами было бы неправильно с точки зрения производительности, т.к. это существенно меняет траектории путей света в сцене.

5. Направленный источник света легко реализуется и часто встречается в научных работах. Используется в различных популярных сценах. Поэтому мы решили добавить его в базовый набор.

6. Прожектор (spot) необходим для имитации освещения от фонарных столбов и других типичных уличных светильников. Его отличительная особенность в том, что он может быть эффективно отбракован при вычислении освещения, если область находится вне освещаемого конуса.

7. Окружение в виде панорамы с текстурой также часто встречается, но уже в индустрии. Отдельный вопрос – это HDR текстура, поскольку её нужно сэмплировать пропорционально яркости пикселей. Мы решили ограничиться константным цветом в большинстве сцен, чтобы избежать ошибок с ориентацией систем координат при переводе из декартовой в сферическую. Кроме ситуации с HDR панорамой, когда именно в этом состоит смысл теста.

3.2 Покрытие

Следующая проблема заключается в составлении объективного набора сцен,

наиболее полно покрывающего как типичные случаи из индустрии, так и известные проблемы, и вызовы в науке. Мы постарались расширить подход 2014 года [14] и сделать наш набор сцен более полным, уделяя внимание не только сценам, привычным для индустрии, но и сценам, часто используемым в научных работах и демонстрирующим известные вызовы в области интегрирования освещения (т. н. Light transport challenges). Мы сгруппировали сцены по номерам. Каждому номеру соответствует некоторая группа сценариев, встречающаяся в индустрии либо некоторому известному вызову в науке. Сами группы мы выбирали не случайным образом, а формировали их в соответствии с известными вызовами и тестовыми сценами, встречающимися в научных работах по компьютерной графике.

1. Cornell Box [31] с несколькими вариациями. Геометрически простые сцены, содержащие тем не менее почти все основные эффекты: шумное первичное освещение и мягкие тени, зеркальные блики от источника освещения, отраженные каустики.

1.1 Базовый вариант, содержащий все эффекты более-менее поровну.

1.2 Базовый вариант + зеркало, создающее широкий каустик, плюс SDS (Specular-Diffuse-Specular) каустик, видимый в отражении. Тривиальное изменение сцены, кардинально меняющее, тем не менее, характер освещения.

1.3 Вариант, ориентированный на многократные глянцевые отражения.

1.4 Оригинальный Cornell Box Эрика Вича, используемый им для демонстрации Metropolis Light Transport [32]. Ярко выраженные трудновычислимые феномены освещённости. SDS каустики присутствуют в небольшом количестве (в стеклянном яйце).

2. Сильное вторичное диффузное освещение и геометрия средней тяжести.

2.1 Dabrovic Sponza [12] и вторичный свет от направленного источника. Часто используется в статьях по компьютерной графике. Недостаток оригинальной сцены в том, что все текстуры в ней однотонные, и вторичное освещение получается в значительной степени равномерным. Поэтому практически любой сколь угодно плохой алгоритм глобального освещения показывает на этой сцене высокую точность. По этой причине в итоговой версии бенчмарка мы исключили данную сцену.

2.2 Crytek Sponza [12] – усложнённый вариант предыдущей сцены, с более детальной геометрией, треугольниками с текстурой непрозрачности на листьях и (opacity для имитации растительности) и разноцветными шторами. В различных демонстрационных программах и статьях шторы (как и растительность) могут частично пропускать свет или полностью отражать его. Мы выбрали второй вариант, т.к. он является более распространённым.

2.3 San Miguel [12] – более тяжёлая геометрическая сцена с растительностью, пропускающей свет. По-прежнему яркое солнце и сильное вторичное освещение. Более сложная с геометрической точки зрения (частично из-за opacity, частично просто из-за более сложной геометрии).

2.4 Ночной вариант сцены San Miguel. Несколько источников света, более сложное первичное освещение.

3. Сцены на исключительно сложную геометрию. Но простые в плане освещения.

4. Тестирование освещения от HDR панорамы. Геометрия простая. Сложность этого сценария в сильно неравномерном первичном освещении от карты окружения (HDR панорамы).

5. Тест на небесные порталы – источники света в окнах, имитирующие рассеянный

свет с улицы. Данный сценарий проверяет систему на наличие реализации многократной выборки по значимости (Multiple Importance Sampling [28,32]), поскольку при наличии больших площадных источников явная стратегия сэмплирования (теневые лучи) становится мало эффективной.

6. Тесты на вторичное освещение от слонца, образованное относительно узкой, но яркой полоской света.

7. Трудновычислимые феномены освещённости (light transport challenge), вызванные многократными отражениями от поверхностей с различными свойствами.

8. Трудновычислимые феномены освещённости (light transport challenge), вызванные SDS каустиками (бассейн с водой).

9. Трудновычислимые феномены освещённости (light transport challenge), вызванные отражениями от глянцевых поверхностей в сцене, содержащей карты нормалей. Данный вид сцен может представлять проблему для однонаправленных алгоритмов на основе Марковских цепей, работающих в первичном пространстве путей.

10. Трудновычислимые феномены освещённости, оригинальная сцена Эрика Вича с приоткрытой дверью и ещё одна сцена, её вариация с глянцевыми отражениями и некоторые менее известные сцены, проверяющие трудновычислимые феномены освещённости.

11. Сцена с самосветящимися поверхностями.

12. Тест на большое количество источников света.

13. Известная по курсу SIGGRAPH 2014 [33] сцена красной кухни на трудновычислимые феномены освещённости, содержащая в себе элемент “зеркального коридора”.

4 МЕТОДИКА ТЕСТИРОВАНИЯ

Следующая серьёзная проблема, которую необходимо решить – это отсутствие совпадения эталонов. Различные программные системы в действительности вычисляют различные изображения. Отличия вызваны отсутствием общего стандарта для фотореалистичного рендеринга и огромным количеством деталей в реализации. Более того, даже одна и та же программная система может давать отличающиеся эталонные изображения при использовании разных алгоритмов. Это может являться как следствием ошибок в реализации (которые мы, вообще говоря, не всегда хотим учитывать, если нас интересует только скорость интегрирования), так и некоторыми объективными факторами. Такими как невозможность расчёта определённых феноменов в одном из алгоритмов либо намеренное упрощение вычислений (например, так называемый “clamping”, когда выбросы ограничиваются некоторым максимальным значением).

В нашей работе мы выбрали следующий компромиссный критерий оценивания:

3. Мы требуем лишь визуального совпадения эталонов для различных систем и алгоритмов. Форма освещения и приблизительный уровень яркости должны совпадать. Допускаются незначительные отличия, не влияющие на суть вычисляемых феноменов освещённости.

4. Если пункт 1 выполнен, то для каждой системы и каждого алгоритма мы вычисляем свой эталон, получающийся при достаточно большом времени расчёта этим

алгоритмом в этой самой системе. Единственно исключение – для алгоритмов со смещённой оценкой мы выбираем в качестве эталона Path Tracing [15] (но в той же самой системе).

5. Если у какой-либо системы пункт 1 не выполнен для разных алгоритмов, тогда для этой системы за её эталон мы принимаем трассировку путей (Path Tracing [15]) реализованную в этой системе. Path Tracing, как базовый алгоритм, реализован везде, мы можем использовать его. Смысл данного пункта в том, что мы хотим штрафовать алгоритм или систему, если реализация не может рассматриваться как корректная даже с натяжкой.

6. Если по какой-то причине в одной из систем эталонное изображение значительно отличается от других, тогда на основе анализа вычисляемых феноменов освещённости выносится вердикт о некорректной реализации, и за данный тест система получает ноль баллов. Впрочем, в наших тестах таких прецедентов не было.

Для оценки ошибки и последующего вычисления баллов мы использовали известную метрику – MSE. Плюс этой метрики для нашего сравнения заключается в линейной зависимости MSE (квадрат ошибки, формула 2) от времени расчёта для систем с несмещённой оценкой. Таким образом, чтобы уменьшить в 2 раза MSE, необходимо рассчитывать изображение в 2 раза дольше. Тестирование производилось на следующем оборудовании: для GPU рендер-систем была использована видеокарта Nvidia GTX2070, для CPU рендер-систем аналогичный по стоимости процессор Intel Xeon E5 2690v2 3Ghz (10 ядер, 20 потоков).

$$MSE = \frac{1}{W * H} \sum_{i=1}^{W*H} ((R_i - R_i^{ref})^2 + (G_i - G_i^{ref})^2 + (B_i - B_i^{ref})^2) \quad (2)$$

Формула 2. Вычисление MSE между измеряемым изображением (R,G,B) и эталоном ($R^{ref}, G^{ref}, B^{ref}$), получаемым за большое время. W — размер изображения по горизонтали. H — по вертикали. Тройки (R,G,B) обозначают значения в красном, зелёном и синем каналах изображений.

$$Winner = \underset{t}{Maxcount} \left\{ \underset{render}{argmin} MSE(render, t) \right\} \quad (3)$$

Формула 3. Выбор победителя по метрике MSE (получающего 3 балла) происходит по нескольким точкам на графике, соответствующим существенно различающимся временам расчёта (*maxcount* по *t*). Далее (соревнование за 2 балла) проводится аналогично. *render* — одна из четырёх тестируемых систем.

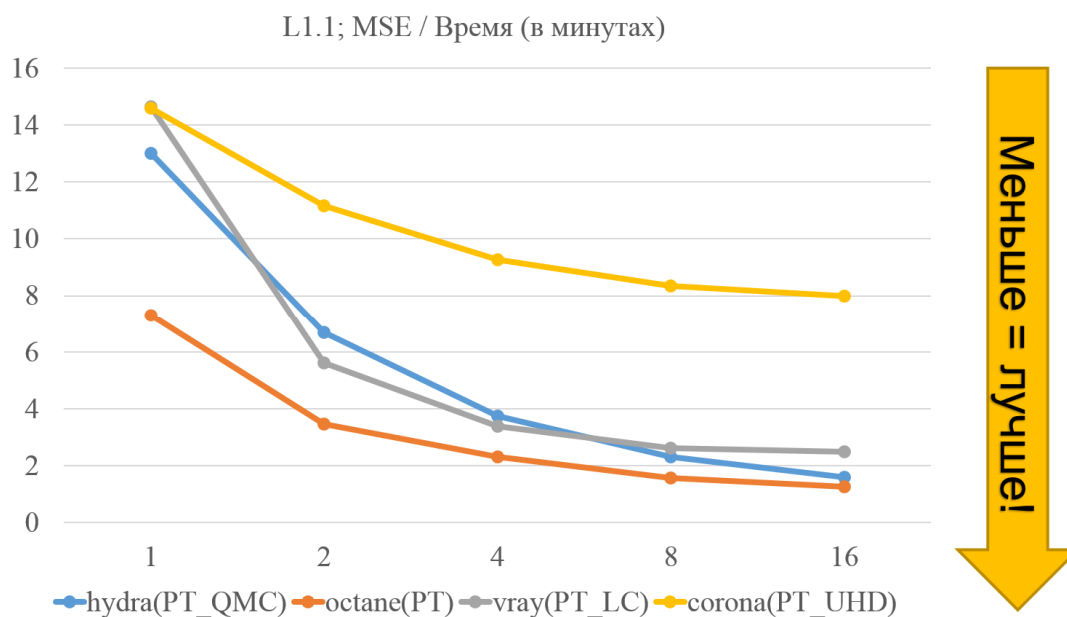


Рис. 2. Зависимость квадрата ошибки (MSE) от времени на сцене L1.1. На графике каждая следующая позиция соответствует увеличению времени в 2 раза. Можно считать, что MSE и скорость связаны линейным соотношением. Уменьшение MSE в 2 раза соответствует увеличению времени расчёта в 2 раза.

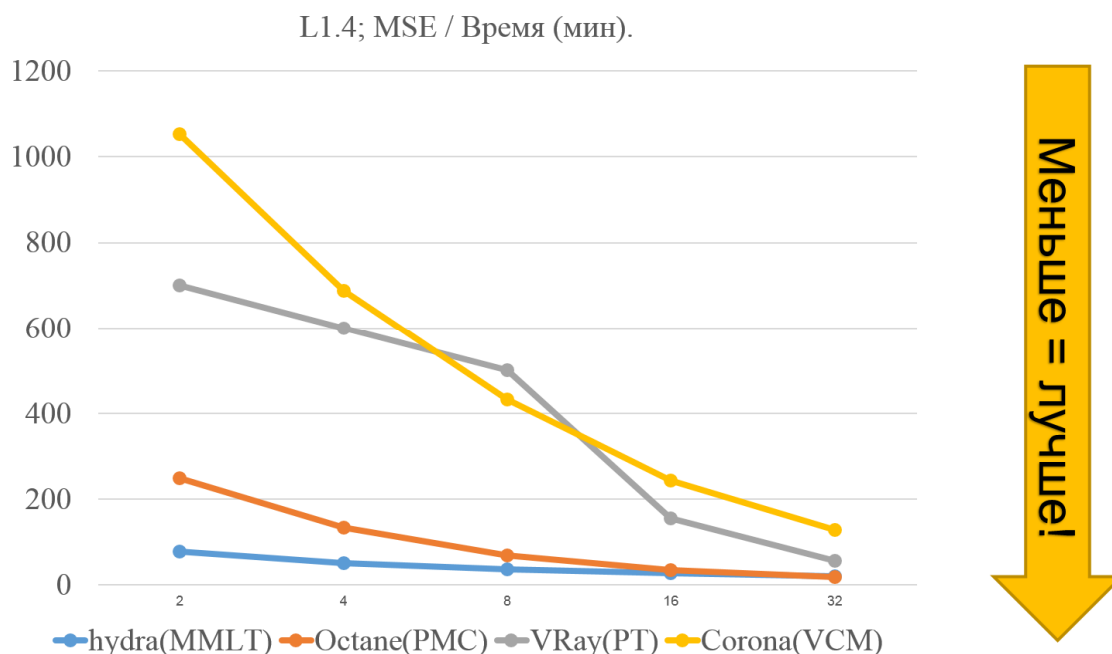


Рис. 3. Зависимость квадрата ошибки (MSE) от времени на сцене L1.4. На графике каждая следующая позиция соответствует увеличению времени в 2 раза. Можно считать, что MSE и скорость связаны линейным соотношением. Уменьшение MSE в 2 раза соответствует увеличению времени расчёта в 2 раза.

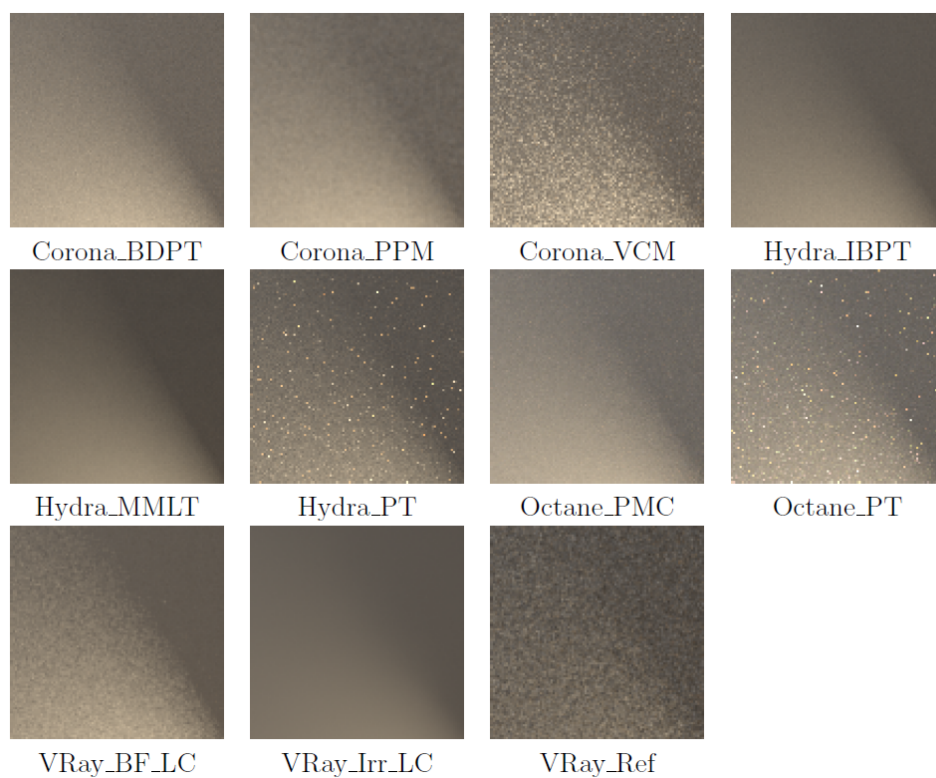


Рис. 4: Пример сопоставления рендер-систем и их алгоритмов на сцене ORB L1.4. 16 минут.

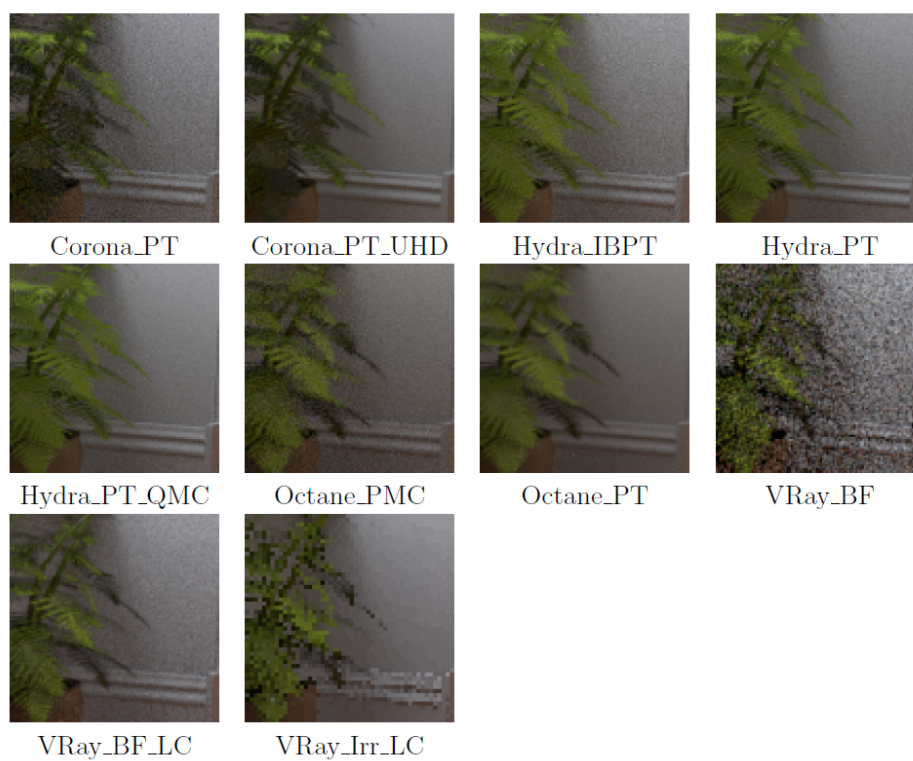


Рис. 5: Пример сопоставления рендер-систем и их алгоритмов на сцене ORB L6.2. 8 минут.

5 АНАЛИЗ РЕЗУЛЬТАТОВ

В экспертном тестировании участвовали 10 профессиональных 3D художников. Примеры визуального сравнения, по которому производилась экспертная оценка, показаны на рис 4,5. Каждому эксперту было показано 17 сопоставлений отдельных фрагментов изображений и было предложено отранжировать системы по местам – 1, 2 и 3. За первое место мы ставили 3 балла, за второе 2, а за третье 1. При этом для каждой системы выбирался только наиболее удачный алгоритм. Эксперт может поставить ноль баллов системе (то есть не назначать никакое место), если он считает, что изображение некорректно либо результат, по его мнению, “не тянет” на призовое место. Результаты экспертного ранжирования приведены на рис. 6 слева.

Примеры численного сравнения (метрика MSE) можно увидеть на рис. 2 и 3. Однако они не наглядны, особенно если оценивать их для всех сцен (17 графиков). Поэтому для того чтобы поставить такое сравнение в один ряд с экспертным, мы использовали ту же соревновательную форму (формула 3), в которой единственным “экспертом” была метрика MSE, а баллы ставились тем же образом от 3 за первое место до 1 или 0 за последнее (рис. 6 справа).

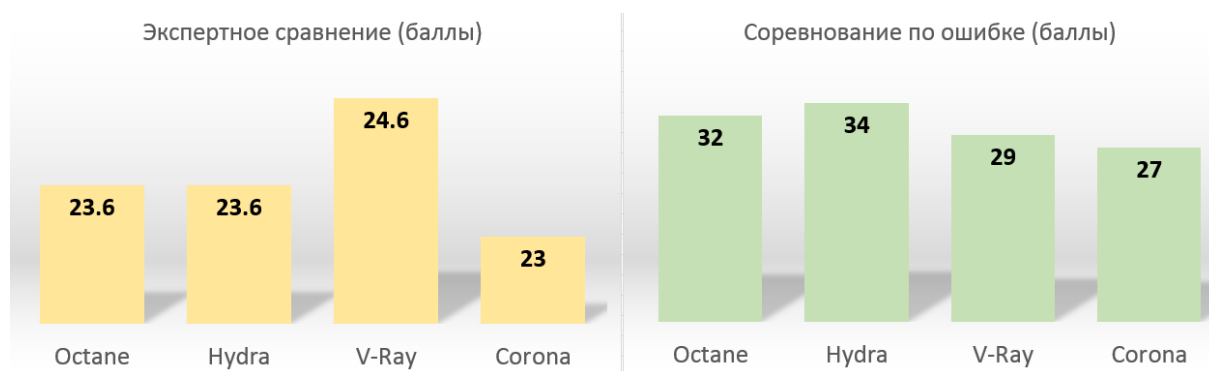


Рис. 6: Экспертное сравнение в “баллах” (слева). Соревнование по точности/ошибке (справа).

Результаты сравнения стали во многом для нас неожиданными. С одной стороны, с точки зрения ошибки победили системы с расчётом без допущений на GPU (Octane и Hydra). Это, в принципе, ожидаемо, т. к. эти системы ориентированы на точный расчёт. С другой стороны, наиболее “чистые” изображения давала система V-Ray, которая победила в экспертном сравнении. Далее, если внимательно посмотреть на цифры, видно, что разница между победителем и проигравшим в экспертном сравнении очень незначительна (1.5 балла). А в сравнении по ошибке она достигает 7 баллов. Из этого можно заключить, что измерение MSE – более чувствительный способ сравнения, чем экспертный. И более понятный, поскольку, в отличие от экспертной оценки, ошибка может быть измерена непосредственно.

Неожиданным оказалось также, что по оценке экспертов (эксперты оценивали качество изображений за одно и то же время, но мы называем это также оценкой производительности) большой разницы в производительности этих рендер-систем нет. Используя разные подходы, разные алгоритмы и разные вычислители (CPU и GPU) все продукты, за исключением отдельных сцен, достигают паритета между собой. Из этого

можно сделать как минимум 2 вывода. Первый — борьба между разными подходами, методами и системами будет продолжаться, т. к. ярко выраженного победителя нет. Второй — расчёт без допущений на GPU вплотную подобрался по скорости к “приближённым” методам на CPU, использующим смещённую или состоятельную оценку. Это означает, что за исключением небольшого количества случаев (вроде искусственного примера L1.5, рис. 1) “приближённые” методы теряют свою актуальность в фотореалистичном рендеринге, поскольку методы расчёта без допущений, как правило, универсальней и проще в поддержке (добавление новых материалов, источников и т. д.). Кроме того, расчёт без допущений делает осмысленным усложнение моделей взаимодействия света с веществом для всё большего погружения рендер-систем в физику и углубления моделирования вплоть до квантового уровня (такие работы существуют [34,35]).

При сравнении коммерческой и открытой системы на GPU мы видим также, что существенной разницы в производительности между ними нет. Причём, это показывает и ошибка, и эксперты. В одних сценариях побеждает Hydra, в других Octane. Это лишний раз подчёркивает необходимость тесного взаимодействия науки и индустрии посредством разработки открытого программного обеспечения, т. к. закрытые научные разработки (Octane) не показали преимущества над открытой реализацией хорошо известных методов.

Сравнивая Octane и Hydra, можно косвенно сравнить технологии CUDA и OpenCL, поскольку Octane реализован на CUDA, а Hydra на OpenCL. На сцене с тяжёлой геометрией, но простым освещением видно (сцена L2.2, рис. 1), что ощутимой разницы в производительности трассировки лучей нет, поскольку на такой сцене скорость интегрирования будет зависеть исключительно от скорости трассировки лучей. Следовательно, закрытым решениям, использующим аппаратные особенности конкретных GPU, вполне можно предпочесть открытую и аппаратно-независимую технологию OpenCL без существенного ущерба производительности.

6 ЗАКЛЮЧЕНИЕ

Предложенный способ сравнения позволяет провести сопоставление различных алгоритмов и различных рендер систем, даже если их эталоны не совпадают в силу различий в реализациях моделей материалов и источников освещения. Стандартизация такого “бенчмарка” в будущем позволит избежать т. н. “cherry-picking” в исследованиях по рендерингу [1] и сделает эти исследования более достоверными. Хотя, возможно, они станут менее зрелищными в силу того, что сцены всегда будут одни и те же.

Мы продолжаем расширять набор сцен и ставим своей целью создать объективное сравнение для всех доступных рендер-систем в мире. Открытость исходных кодов нашей системы (Hydra) позволяет будущим исследователям получить совпадающие изображения, поскольку все формулы, по которым ведутся вычисления в нашей системе, известны. Это даёт возможность в будущем проводить более точное сопоставление между нашими результатами и результатами других исследователей.

REFERENCES

- [1] J. Křivánek, “Open problems and research directions”, *Proceedings of SIGGRAPH-2018 conference*,

- Vancouver, Canada (2018). <https://cgg.mff.cuni.cz/~jaroslav/papers/2018-archvizcourse/slides/07-krivanek-research-notes.pdf>
- [2] Vincent Schüssler, Eric Heitz, Johannes Hanika, and Carsten Dachsbacher, “Microfacet-based normal mapping for robust Monte Carlo path tracing”, *ACM Trans. Graph.*, **36**(6), Article 205, (2017). doi://doi.org/10.1145/3130800.3130806
- [3] Robert Herzog, Martin Čadík, Tunç O. Aydıçin, Kwang In Kim, Karol Myszkowski, and Hans-P. Seidel, “NoRM: No-Reference Image Quality Metric for Realistic Image Synthesis”, *Comp. Graph. Forum*, 545-554 (2012). doi: 10.1111/j.1467-8659.2012.03055.x <http://dx.doi.org/10.1111/j.1467-8659.2012.03055.x>
- [4] Martin Čadík, Robert Herzog, Rafał Mantiuk, Karol Myszkowski, and Hans-Peter Seidel, “New measurements reveal weaknesses of image quality metrics in evaluating graphics artifacts”, *ACM Trans. Graph.*, **31**(6), Article 147, (2012). doi: <https://doi.org/10.1145/2366145.2366166>
- [5] Qingbo Wu, Hongliang Li, Fanman Meng, King N. Ngan, *A Perceptually Weighted Rank Correlation Indicator for Objective Image Quality Assessment*, ArXiv preprint, (2017). <https://arxiv.org/abs/1705.05126>
- [6] Render Engine Comparison: Cycles vs The Rest (2015). <https://www.blenderguru.com/articles/render-engine-comparison-cycles-vs-giants> (accessed July 06, 2019).
- [7] Render Tests, Comparisons and Guides (2019). <https://www.antonibosi.com/maya-render-tests-reviews-comparisons> (accessed July 06, 2019).
- [8] Yuichiro Yama, “Render Comparison Test 4 (Volume) - Redshift, Arnold, V-Ray, Mantra”, *Youtube.com* (2018). <https://www.youtube.com/watch?v=K0Wn6lhGNAQ> (accessed July 06, 2019).
- [9] V.A. Frolov, V.A. Galaktionov, “Memory-compact Metropolis light transport on GPUs”, *Programming and Computer Software*, **43**(3), 196-203 (2017). doi: 10.1134/S0361768817030057
- [10] S.V. Ershov, A.A. Garbul, S.G. Pozdnyakov, V.G. Sokolov, A.G. Voloboy, “Lighting simulation in anisotropic media”, *Mathematica Montisnigri*, **39**, 42-56 (2017).
- [11] S.V. Ershov, D.D. Zhdanov, A.G. Voloboy, “Calculation of luminance of scattering medium by MCRT using multiple integration spheres”, *Mathematica Montisnigri*, **44**, 122-134 (2019).
- [12] Morgan McGuire, “Computer Graphics Archive” (2017). <https://casual-effects.com/data> (accessed July 06, 2019).
- [13] Lighting Challenges (2015). <http://www.3drender.com/challenges/> (accessed July 06, 2019).
- [14] V. Frolov, V. Galaktionov, M. Trofimov, “Sravnenie industrialnyh sistem raschyota osvesheniya”, *Privolzhskij nauchnyj zhurnal*, **14**, 79-85 (2014).
- [15] James T. Kajiya, “The rendering equation”, *Proceedings of the 13th annual conference on Computer graphics and interactive techniques (SIGGRAPH '86)*, David C. Evans and Russell J. Athay (Eds.). ACM, New York, NY, USA, 143-150 (1986).
- [16] Corona Renderer, (2019). <https://corona-renderer.com/> (accessed July 06, 2019).
- [17] Chaos Group. V-Ray Home Page, (2019). <https://www.chaosgroup.com/> (accessed July 06, 2019).
- [18] J. Krivanek, et al, “Radiance caching for efficient global illumination computation”, *IEEE Transactions on Visualization and Computer Graphics*, **11**(5), 550-561 (2005).
- [19] J. Křivánek, K. Bouatouch, S. Pattanaik, and J. Žára, “Making radiance and irradiance caching practical: adaptive caching and neighbor clamping”, *Rendering Techniques*, 127-138 (2006).
- [20] H. Jensen, Realistic image synthesis using photon mapping, AK Peters/CRC Press, (2001).
- [21] H. Jensen and T. Hachisuka, “Stochastic progressive photon mapping”, *ACM Transactions on Graphics*, **28**(5), Article 14, (2009).
- [22] I. Georgiev, J. Křivánek, T. Davidovič and P. Slusallek, “Light transport simulation with vertex connection and merging”, *ACM Transactions on Graphics*, **31**(6), Article 192-1 (2012).
- [23] J. Schwarzhaupt, H.Jensen and W. Jarosz, “Practical Hessian-based error control for irradiance

- caching”, *ACM Transactions on Graphics*, **31**(6), Article 193 (2012). doi:10.1145/2366145.2366212
- [24] OTOY Octane Renderer, (2019). <https://home.otoy.com/render/octane-render/> (accessed July 06, 2019).
- [25] Ray Tracing Systems. Hydra Renderer, (2019). <https://github.com/Ray-Tracing-Systems/HydraAPI>(accessed July 06, 2019)
- [26] David Cline, Justin Talbot, and Parris Egbert, “Energy redistribution path tracing”, *Proceedings of ACM SIGGRAPH-2005 conf. (SIGGRAPH '05), New York, USA*, 1186-1195 (2005). doi: //doi.org/10.1145/1186822.1073330
- [27] Yu-Chi Lai, Shao Hua Fan, Stephen Chenney, and Charcle Dyer, “Photorealistic image rendering with population monte carlo energy redistribution”, *Proceedings of the 18th Eurographics conference on Rendering Techniques (EGSR'07)*, Jan Kautz and Sumanta Pattanaik (Eds.). Eurographics Association, Aire-la-Ville, Switzerland, 287-295 (2007). doi://dx.doi.org/10.2312/EGWR/EGSR07/287-295
- [28] E. Veach and L. Guibas, “Bidirectional estimators for light transport”, *Photorealistic Rendering Techniques, Springer*, 145-167 (1995).
- [29] T. Hachisuka , A. Kaplanyan and C. Dachsbacher, “Multiplexed metropolis light transport”, *ACM Transactions on Graphics*, **33**(4), Article 100 (2014).
- [30] C. Kelemen, et al, “A simple and robust mutation strategy for the metropolis light transport algorithm”, *Computer Graphics Forum*, **21**(3) (2002). doi: 10.1111/1467-8659.t01-1-00703
- [31] Cornell Box Data. Cornell University Program of Computer Graphics (2019). <https://www.graphics.cornell.edu/online/box/data.html> (accessed July 06, 2019).
- [32] E. Veach and L. Guibas, “Metropolis light transport”, *Proceedings of the 24th annual conference on Computer graphics and interactive techniques, ACM Press/Addison-Wesley*, 65-76 (1997).
- [33] Recent Advances in Light -Transport Simulation: Some Theory and a Lot of Practice, *Courses of 41th International conference on Computer Graphics and Interactive Techniques, Vancouver, Canada* (2014). <http://s2014.siggraph.org/attendees/courses/events/recent-advances-light-transport-simulation-some-theory-and-lot-practice.html>.
- [34] A.G. Voloboy, S.V. Ershov, S.G. Pozdnyakov “Interaktivnoe modelirovanie avtomobilnykh krasok”, *Proceedings of the 22nd International Conference on Computer Graphics and Vision (GraphiCon 2012)*, 242-247 (2012). <https://www.graphicon.ru/html/2012/conference/RU6%20-%20Graphics/gc2012Voloboy.pdf>
- [35] D. D. Zhdanov, I. S. Potemin, V. A. Galaktionov, B. Kh. Barladyan, K. A. Vostryakov, and L. Z. Shapiro ”Spectral Ray Tracing in Problems of Photorealistic Imagery Construction”, *Programming and Computer Software*, **37**(5), 236–244 (2011). doi:10.1134/S0361768811050069

Received May 15, 2019