***Сравнение алгоритмов трассировки лучей***

***<Титулы>***

Аннотация.

В представленной работе рассмотрены некоторые популярные рендеры, реализующие различные алгоритмы трассировки лучей. Эти алгоритмы делают возможным получение фотореалистичных изображений с помощью моделирования различных световых свойств, таких как глобальное освещение, тени, каустики, дисперсия, глубина резкости, подповерхностное рассеивание и т.д. Все алгоритмы объединяет идея симулирования пути света, но отличают реализации способов это сделать. В зависимости от этого алгоритм может по-разному изображать различные световые эффекты, и объём вычислений будет различным. Благодаря высокой вычислительной мощности современных процессоров и видеокарт становится актуальным рассмотрение достаточно сложных алгоритмов и программ, реализующих их. В представленной работе сравниваются различные алгоритмы и их реализации. Сделаны выводы о качестве получаемого изображения и реализованной многопоточности вычислений.

Оглавление

Глава 1 Физические модели и законы.

Глава 2 Уравнение рендеринга и PBR.

Глава 3 Рендеры и алгоритмы трассировки.

* appleseed
* Cycles
* LuxCore

Глава 4 Способ сравнения

Глава 5 Эксперименты

* 1
* 2
* 3
* 4
* Световод 5
* Бассейн 6

Глава 5 Построение графиков, сравнение, выводы

Глава 6 Заключение

Список сокращений

|  |  |
| --- | --- |
| Сокращение | Расшифровка |
| PBR | physically-based rendering |
| ДФОС (BRDF) | двулучевая функция отражательной способности  (bidirectional reflectance distribution function) |
| PSNR | peak signal-to-noise ratio |
|  |  |

Введение

Обоснование выбора темы диплома и актуальность

На сегодняшний день на рынке существует много программ трехмерной визуализации, который используют для реалистичного синтеза различные алгоритмы трассировки лучей. Эти программы применяются в кино и теле индустрии, дизайне, архитектуре и т.п. На сегодняшний день наибольшей популярностью пользуются платные пакеты, такие как Maxwell Render, Lumion, Solidworks Visualize, Marmoset Toolbag, Corona Render, Iray, V-Ray, Indigo Render. Для некоторых молодых компаний такой расход может быть критическим, поэтому они склонны искать альтернативные бесплатные аналоги. К счастью, такие существует, но тогда встает вопрос об качестве их работы. Ведь помимо цены, важными остаются такие характеристики, как время рендеринга и фотореалистичность получающегося изображения. В данной работе сравниваются эти характеристики у некоторых популярных бесплатных рендеров.

Также важным фактором для времени работы играет реализация распараллеливания алгоритмов рендеринга. Ведь они, благодаря идее имитации пути нескольких световых лучей, легко поддерживают эту возможность. С вычислительной мощности процессоров актуально рассмотреть и реализацию многопоточности у этих алгоритмов.

Формулировка цели и задач исследования

Цель исследования: сравнение реализованных алгоритмов трассировки лучей, использующихся в различных бесплатных программах для трехмерной визуализации.

Задачи:

1 построение трехмерных сцен

2 настройка идентичных параметров для каждой сцены

3 получение эталонной сцены, с которой будет проводится сравнение

4 запуск рендеров с ограничением по времени и/или потокам

5 получение графиков скорости и качества с помощью метрики PSNR (peak signal-to-noise ratio).

Понятия объекта и предмета исследования

Объектом исследования в данной работе является компьютерная графика, а предметом – алгоритмы трассировки лучей и их реализация.

Литературный обзор.

Глава 1. Физические модели и законы

используемый набор физических моделей и правил  
имеется в виду: простая геометрическая лучевая оптика, геометрич. оптика с поправками на рассеяние, фотонная модель (каждый фотон - упругий мячик), модель Гюйгенса (волновые фронты), полная волновая модель (позволяет увидеть дифракцию и интерференцию). Возможны и комбинации моделей. Эти (и, возможно, другие) модели опишите и раскройте их разницу с позиций качества и трудоёмкости.

Перед тем, как говорить о реалистичном отображении внешнего мира определимся с понятием «реалистичность». Все что видит человеческий глаз – это отраженный от разных поверхностей свет с разными длинами волн, лежащими в диапазоне от 380 нанометров до 780 нанометров. Поэтому для реалистичного рендеринга алгоритмам необходимо руководствоваться законами физической оптики, т.е. как волновыми свойствами света, так и лучевыми.

Законы геометрической оптики.

Геометрическая оптика описывает распространение электромагнитного поля как распространение лучей в пространстве. Луч – это линия(кривая или прямая), вдоль которой распространяется энергия светового поля.

* Закон прямолинейного распространения света (принцип Ферма)
* Оптическая длина луча (произведение геометрического пути светогого луча и показателя преломления среды, в которой распространяется свет) между двумя точками минимальна по сравнению со всеми другими линиями, соединяющими эти две точки. Отсюда очевидно следует, что в однородной среде световой луч движется по прямой линии.
* Закон независимого распространения лучей
* Если через точку пространства проходит несколько лучей, то каждый луч никак не влияет на любой другой луч. То есть он ведет себя так, будто бы никаких других лучей нет.
* Закон преломления света (Закон Снеллиуса)
* При переходе светового луча из одной среды в другую, на границе двух сред световой луч разделяется на два: отраженный и преломленный. Угол падения света на поверхность связан с углом преломления соотношением ,

где показатель преломления первой среды,

показатель второй среды, в которую свет попадает, пройдя границу раздела,

угол между падающим лучом и нормалью к поверхности в первой среде,

угол между лучом и нормалью к поверхности во второй среде.

* Законы отражения света
* **Диффузное отражение**. При отражении света от шероховатой поверхности лучи отражаются в разные стороны.
* **Зеркальное отражение**. Падающий и отраженный лучи, а также нормаль к отражающей поверхности в точке падения лежат в одной плоскости и угол падения равен углу отражения.
* **Полное внутреннее отражение**. Ситуация, при которой интенсивность преломленного луча становится нулевой. Возникает на границе двух сред, где волна падает из среды с большим показателем преломления (т.е. более плотной).
* Закон обратимости светового луча.
* Траектория пути лучей не зависят от направления распространения. Другими словами, если луч, выпущенный из точки А в точку Б, пустить в обратном направлении от Б к А, то он будет иметь точно такую же траекторию.

. О

Электромагнитные свойства света.

Свет (или световое поле) – это электромагнитные волны в оптическом диапазоне длин волн (0 – 40 мкм). В этом диапазоне выполняются законы геометрической оптики и свет слабо взаимодействует с веществом. Оптический диапазон состоит из 4 видов излучения: рентгеновское, ультрафиолетовое, инфракрасное и видимое. В данной работе имеет значение только последнее.

Видимое излечение – волны, воспринимаемые человеческим глазом, лежащие в диапазоне от 380 нм до 780 нм.

**В этой главе написать еще о:**

1 Вписать\выделить среди законов закон Френеля

2 расписать про эффект подповерхностного рассеяния и другие эффекты, которые не поддерживает уравнение рендеринга <https://en.wikipedia.org/wiki/Rendering_equation>

В настоящее время не существует способа полностью сымитировать поведение сложное света, учитывая все его физические свойства. Однако в 1986 году на конференции SIGGRAPH Джеймс Т. Каджия предложил интегральное уравнение, которое является лучшей моделью симулирования освещения на сегодняшний день.

Глава 2 Уравнение рендеринга и PBR.

Написать про уравнение рендеринга, разжевать каждую букву в нем: <https://habr.com/ru/post/426123/>

Потом написать про уравнение освещения и технику ф-к рендеринга.

* Теория микрограней, закон сохранения энергии, закон френеля

Фотореалистичный синтез изображений основан на расчете интеграла освещенности (уравнение рендеринга):

*-* количество излучения заданной длины волны , исходящего вдоль направления  из заданной точки

- излучаемый материалом свет в направлении длины волны в точке

– интеграл по полусфере входящих направлений

- дифференциальный телесный угол, порожденный направлением .

- излучение, попадающее на поверхность по входящему направлению

- косинус угла между направлением на источник и нормалью

(bidirectional reflectance distribution function) - двулучевая функция отражательной способности (ДФОС) - четырёхмерная функция, определяющая, как свет отражается от непрозрачной поверхности. Это характеристика конкретного материала, показывающая долю энергии, отраженной от материала в зависимости от направления на источник света и на камеру и длины волны света (см. Рис.1)

- косинус угла между нормалью  и вектором на источник .

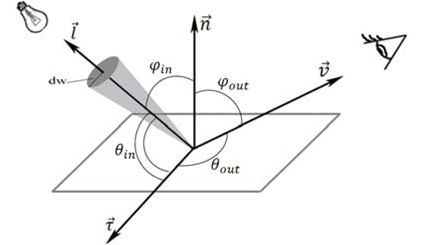


Рис. 1 ДФОС - двулучевая функция отражательной способности

Однако уравнение рендеринга все же не учитывает некоторые свойства света:

<https://en.wikipedia.org/wiki/Rendering_equation>

Глава 3 Рендеры и алгоритмы трассировки.

**Описать каждый рендер в отдельных подглавах**

* Подробно расписать алгоритмы, которые в них используются: вместе с псевдокодом… со всем
* Описать BRDF которые в них используются\реализованы. Тоже подробно.
* Кратко написать про всякие фичи, которые не проверяются в этой работе.
* Другие алгоритмы выкинуть

Рассмотрим некоторые алгоритмы синтеза, использующихся в рендерах:

1 Алгоритм рейкастинга (Ray-casting)

Не является полноценным алгоритмом глобального освещения (учитывающим перенос света между поверхностями). Тем не менее, он позволяет помимо расчета первичного освещения производить расчет теней. Идея заключается в том, что из камеры испускаются лучи, для каждого луча ищется первое пересечение с объектом. Далее из каждой точки пересечения испускается луч до источника света. Если этот луч пересечется с каким-то другим объектом по пути к источнику, то это означает, что точка находится в тени. (см. Рис. 2) Иначе можно рассчитать долю энергии, которая доходит от источника до камеры.[]

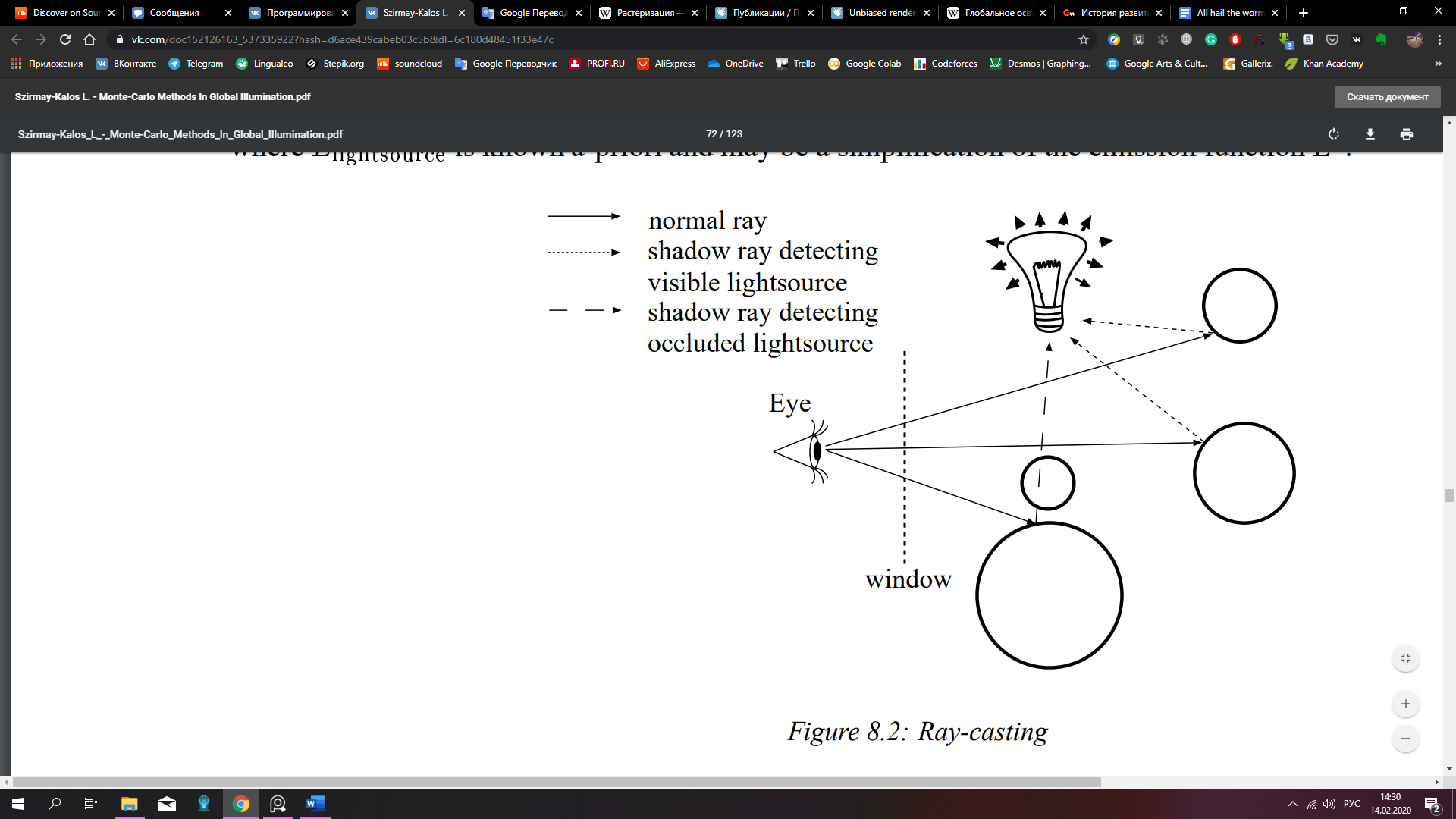


Рис. 2 Схема алгоритма рейкастинга

2 Алгоритм трассировки видимых лучей (visibility ray-tracing)

Идея состоит в том, что луч из камеры претерпевает несколько идеальных отражений и преломлений, прежде чем будет произведена проверка нахождения точки в тени (последний луч всегда идет в источник, как и в ray-casting). Преломленные и отраженные векторы рассчитываются по известным законам идеального отражения и преломления. Стоит заметить, что в этом алгоритме считается, что материалы имеют постоянные коэффициенты отражения и преломления (независимо от угла падения и отражения/преломления света). Для других материалов отражением и преломлением пренебрегают, луч идет сразу в источник света. (см. Рис. 3)

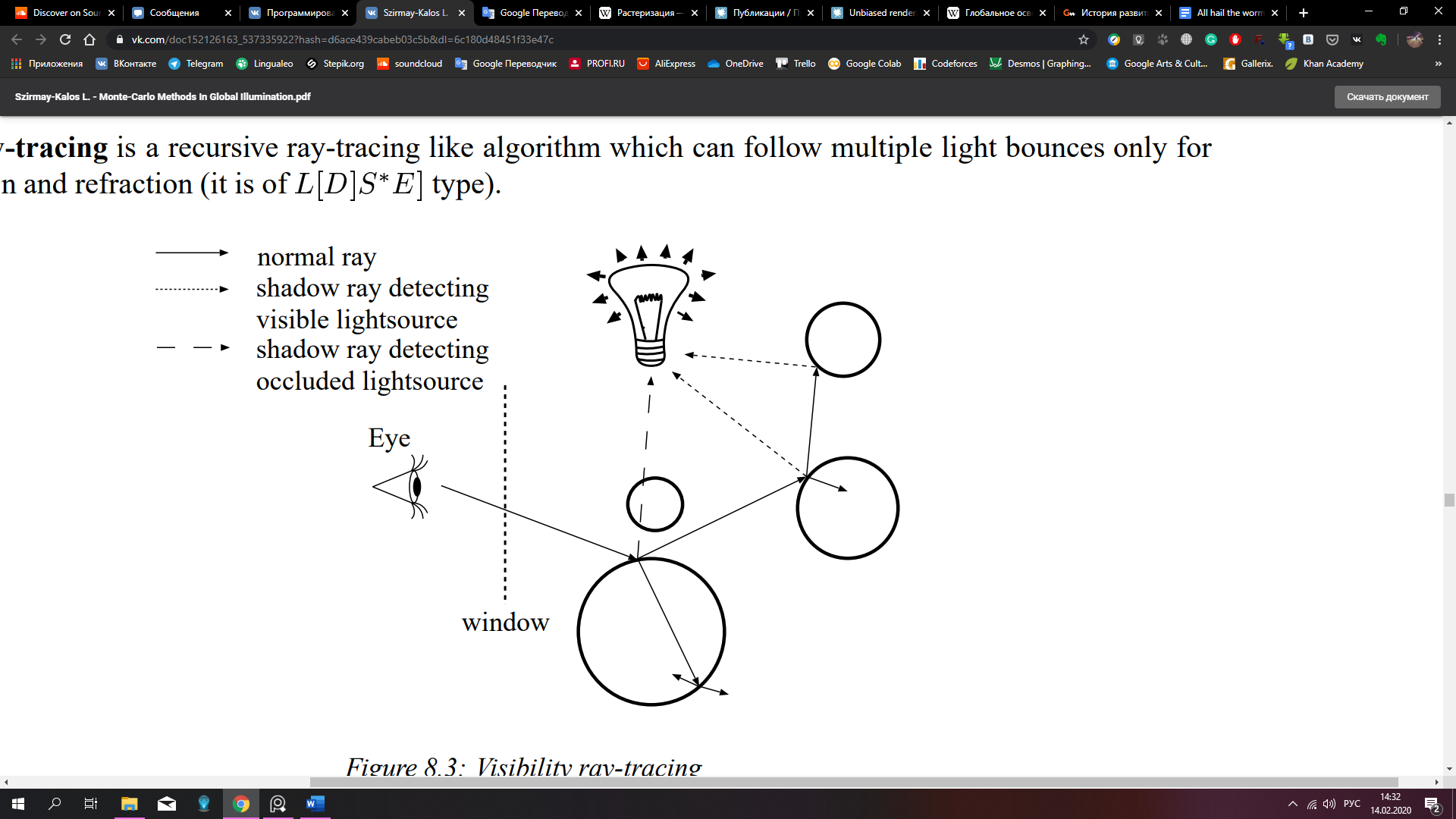


Рис. 3 Схема алгоритма трассировки видимых лучей

3 Алгоритм трассировки фотонов (Photon tracing)

Трассировка фотонов – инверсия алгоритма трассировки видимых лучей т.е. частицы фотонов испускаются из источника света. Этот алгоритм использует аналогичные допущения. При каждом попадании луча на поверхность проверяется, имеет ли она идеальное отражение, преломление и некогерентное отражение или преломление. В направлениях идеального отражения или преломления отслеживание продолжается запуском новых дочерних лучей.

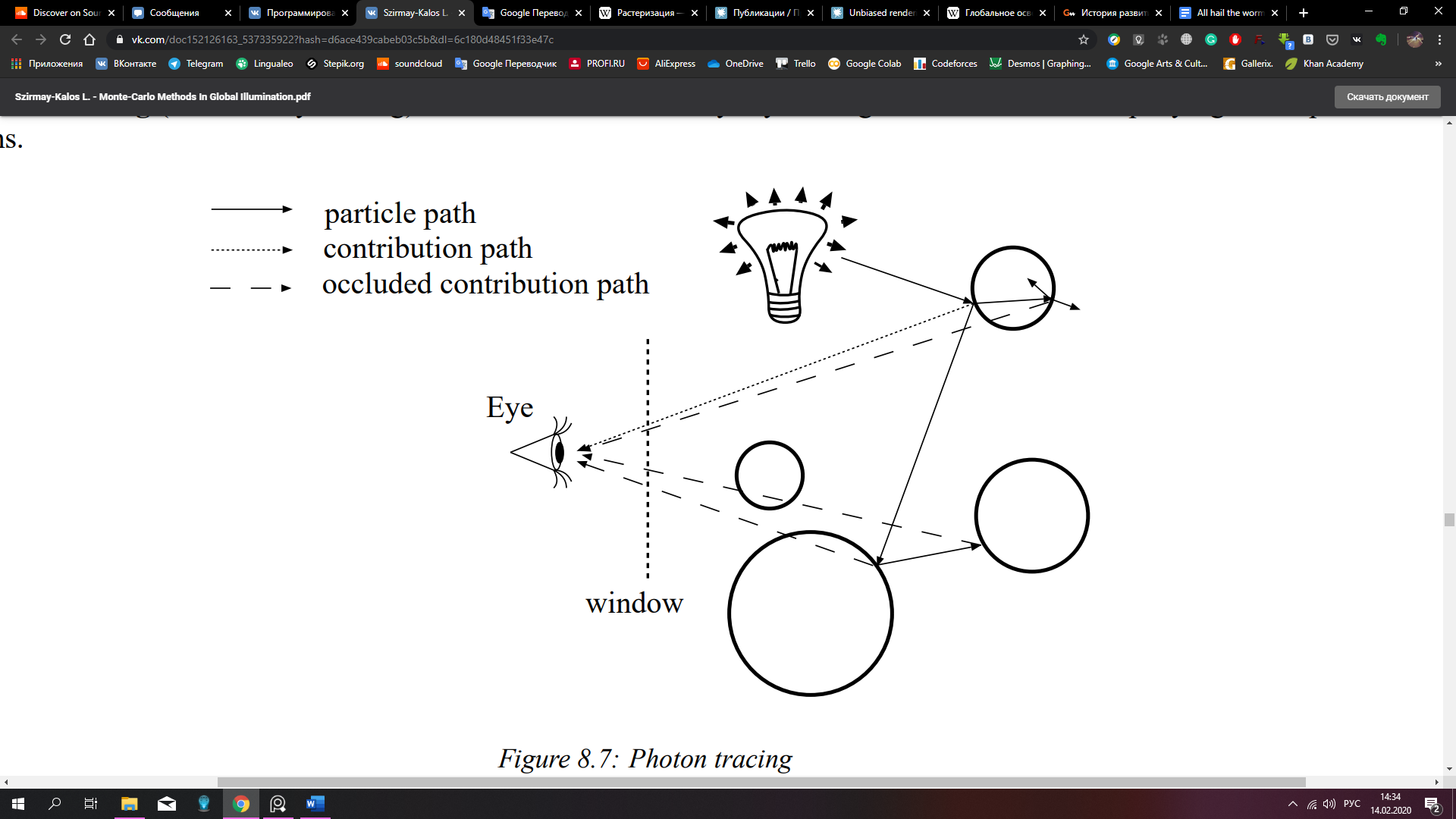


Рис. 4 Схема алгоритма трассировки видимых лучей

4 Распределенная трассировка лучей

Этот алгоритм глобального освещения может моделировать все возможные пути т.е. он позволял визуализировать материалы с произвольными двулучевыми функциями отражения. Качество этого алгоритма самое высокое из всех рассмотренных выше. Однако из-за сэмплирования (процесс испускания множества лучей называется сэмплированием ДФОС) алгоритм имеет и самую высокую вычислительную сложность.

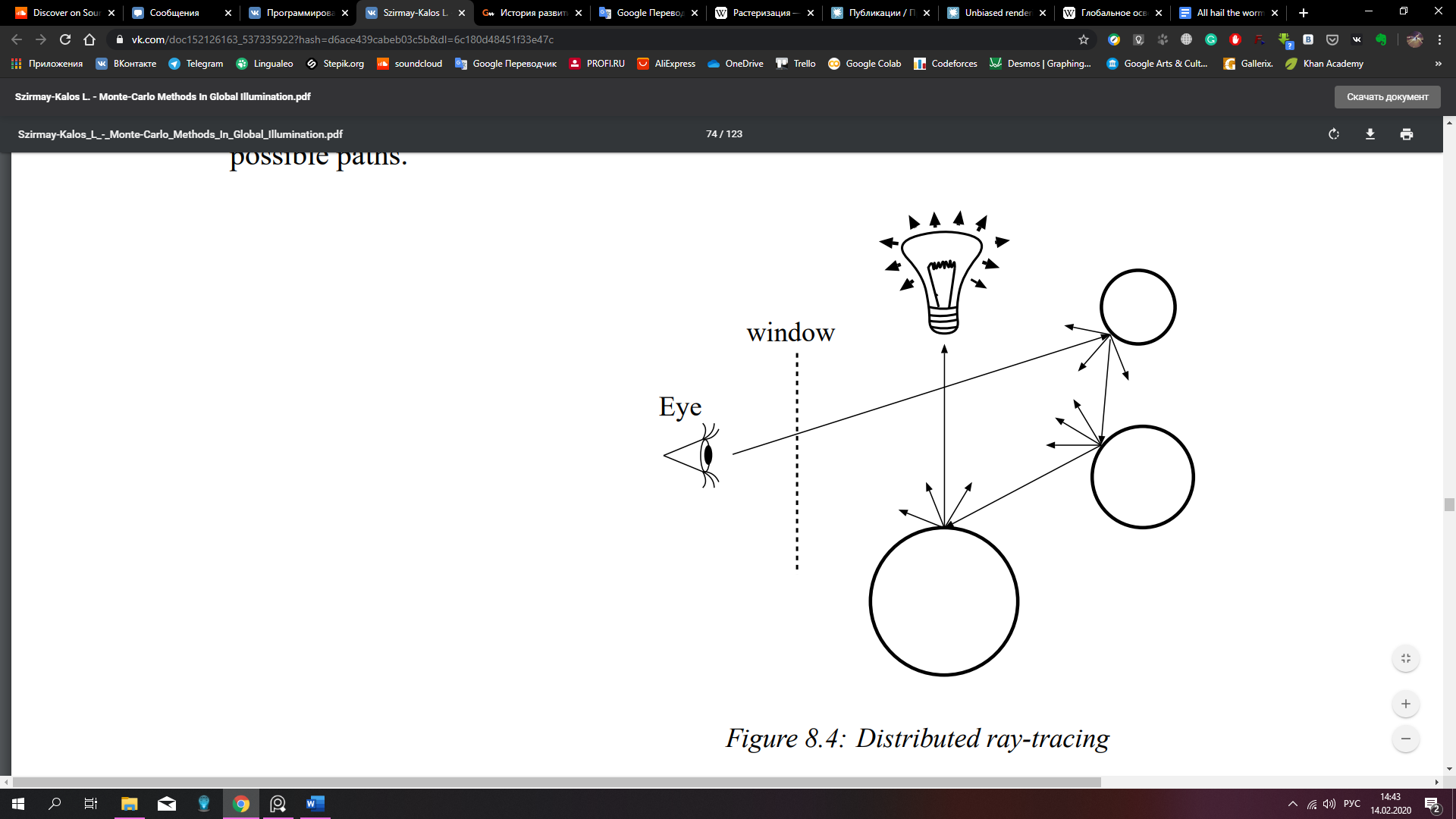


Рис. 5 Схема алгоритма распределенной трассировки лучей

В этом методе трассировка луча не прекращается при достижении поверхности, не имеющей идеального отражения или идеального преломления. После попадания луча на диффузную поверхность дочерние лучи генерируются случайным образом в соответствии с BRDF, характеризующим поверхность. Для надлежащей оценки общего взаимного отражения необходимо проследить дочерние лучи и рассчитать среднее их вкладов.

**Русская рулетка и расщепление**

Это два связанных метода, призванных улучшить сходимость метода Монте-Карло за счет повышения вероятности того, что каждая выборка внесет существенный вклад в результат.

Под «русской рулеткой» понимается стохастический процесс , основанный на методе Монте-Карло, позволяющий избавиться от «лишних» фотонов. Суть метода заключается в следующем: для конкретного фотона известно с какой поверхностью он столкнулся. Эта поверхность имеет коэффициенты диффузного () и зеркального () отражений. Для каждого отдельного фотона берётся некоторая случайная величина P и сравнивается со значениями этих коэффициентов. Рассматриваются три ситуации:

 → фотон отражается диффузно  
  
 → фотон отражается зеркально  
  
 → фотон поглощается

5 Алгоритм трассировки пути (Path-tracing)

Этот метод прослеживает путь отдельной частицы света, взаимодействующей с окружающей средой, до поглощения с использованием BRDF-семплирования. При попадании луча на поверхность испускается два новых луча: один - в произвольном направлении другой – до источника света. Доля отразившейся энергии при этом рассчитывается на основе ДФОC (либо испускается луч с одной и той же энергией по принципу «русской рулетки», при этом вероятность испускания вычисляется на основе ДФОC).

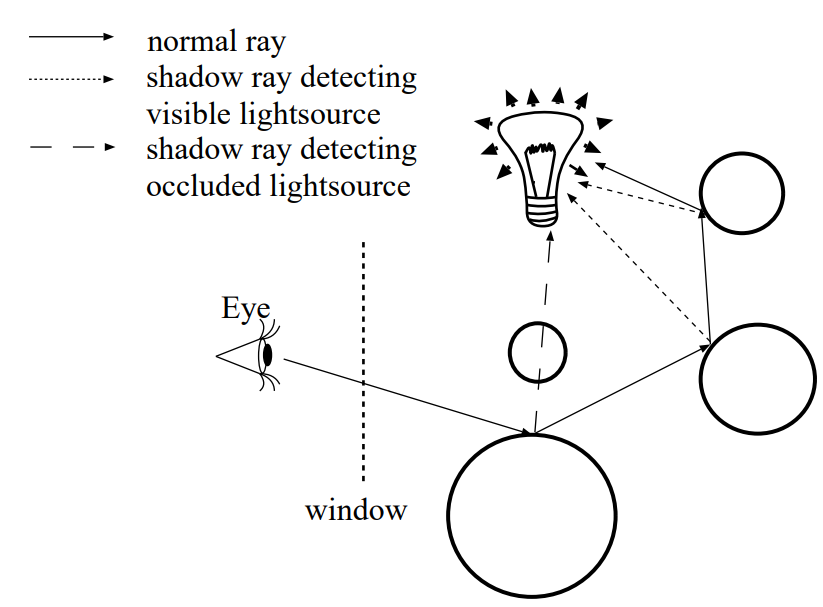


Рис. 6 Схема алгоритма трассировки пути с прямым вычислением источника света.

Данный алгоритм генерирует все пути фотона с помощью BRDF-семплирования, кроме последнего, который, в свою очередь, получается при помощи семплирования источника света. Таким образом, если поверхность, отражающая свет, является блестящей (близкой к идеальному зеркалу или идеальному рефрактору), то качество выборки по важности может быть довольно плохим. Поскольку почти идеально гладкие поверхности, фокусирующие отраженный свет, создают каустику, трассировка пути - как и другие gathering алгоритмы - плоха в симулировании этого эффекта.

6 Алгоритм двунаправленной трассировки лучей (Bi-directional path-tracing)

Двунаправленная трассировка (см. Рис. 7) подразумевает испускание лучей от камеры и от источника (максимальная глубина трассировки с каждой стороны фиксирована). Далее пути от камеры и источника начинают соединяться различными способами (так называемый детерминированный шаг алгоритма). В случае пересечения соединяющих лучей с другими объектами вклад всего пути будет равен нулю, иначе - не ноль.

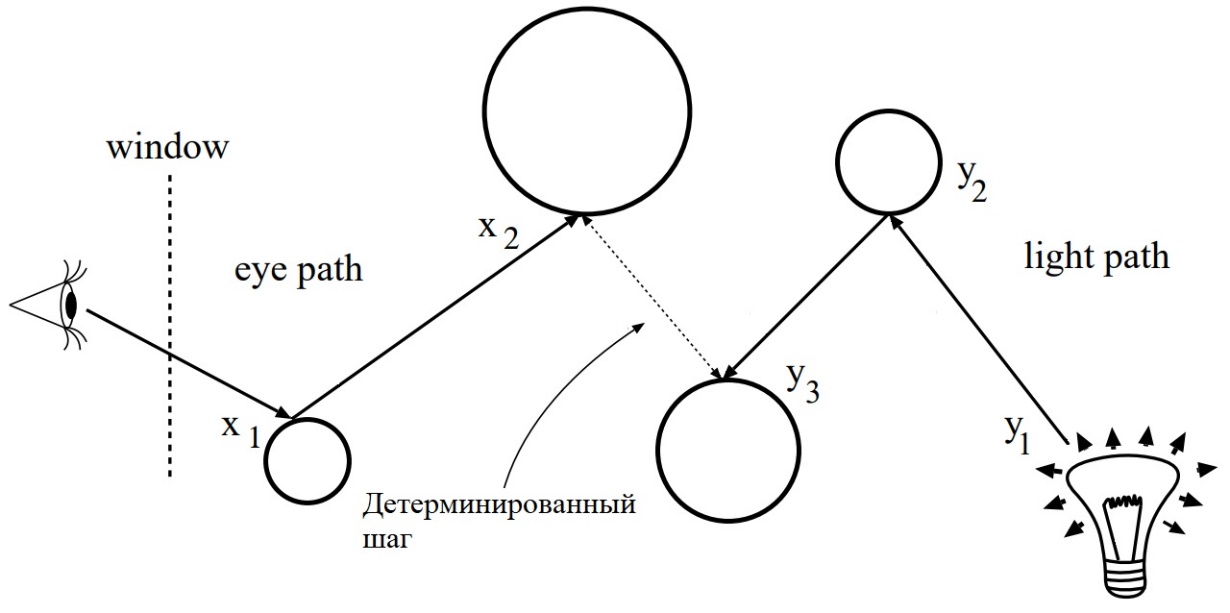


Рис. 7 Схема алгоритма двунаправленной трассировки пути с единичным детерминированным шагом.

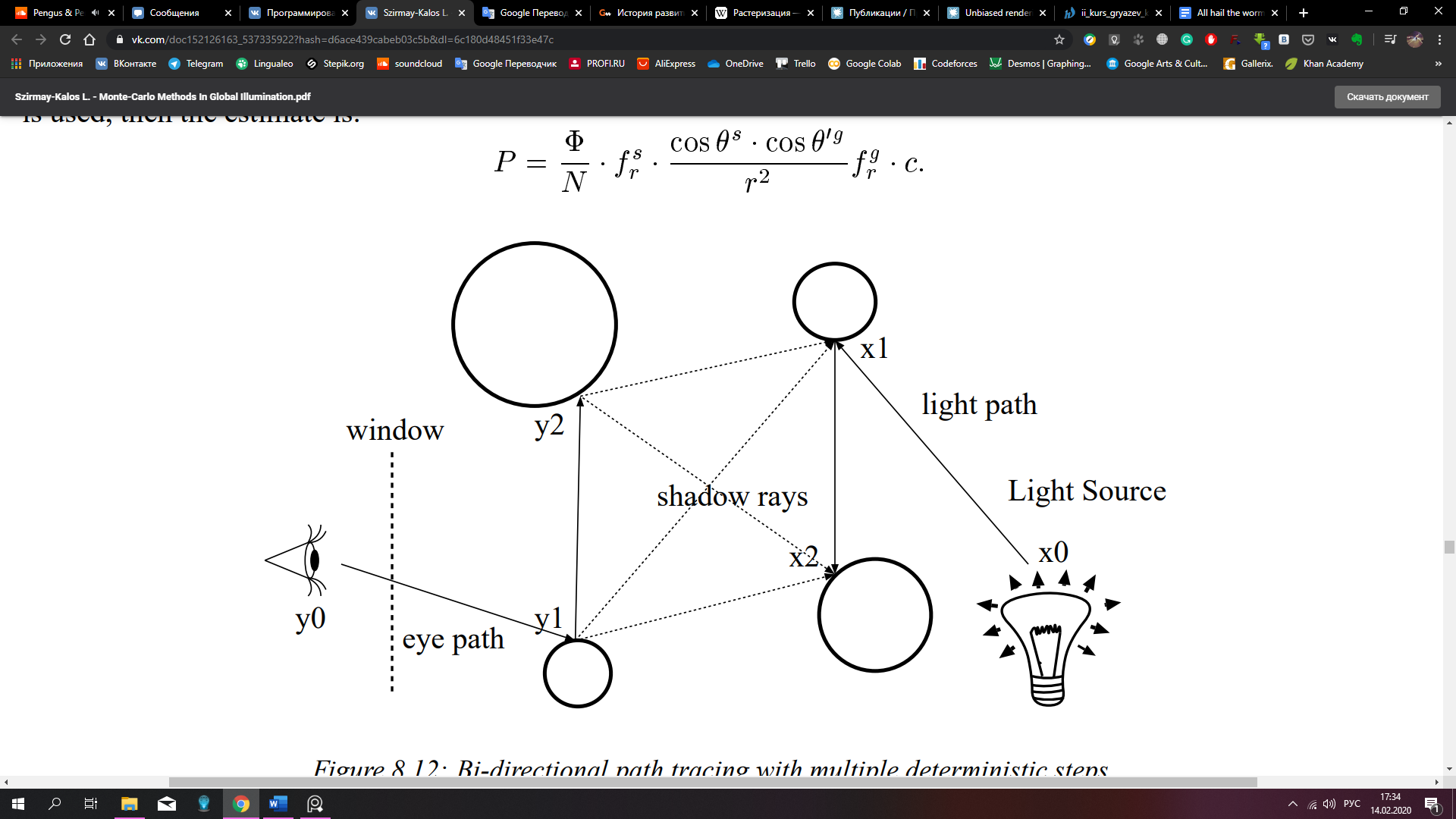


Рис. 8 Схема алгоритма двунаправленной трассировки пути с множественным детерминированным шагом.

6.1 Алгоритм Metropolis Light Transport

На первом шаге алгоритм ищет какой-то путь от источника до камеры, дающий ненулевой вклад. На втором этапе путь модифицируется специальным образом, основываясь на предыдущем пути. Новый путь обычно лучше, чем старый, если он вносит больший вклад в изображение. Если старый путь оказывается хуже, чем новый, то далее модификации подвергается новый путь, иначе – опять старый. Алгоритмы, отыскивающие «хорошие» пути потенциально быстрее сходятся, чем остальные алгоритмы трассировки.

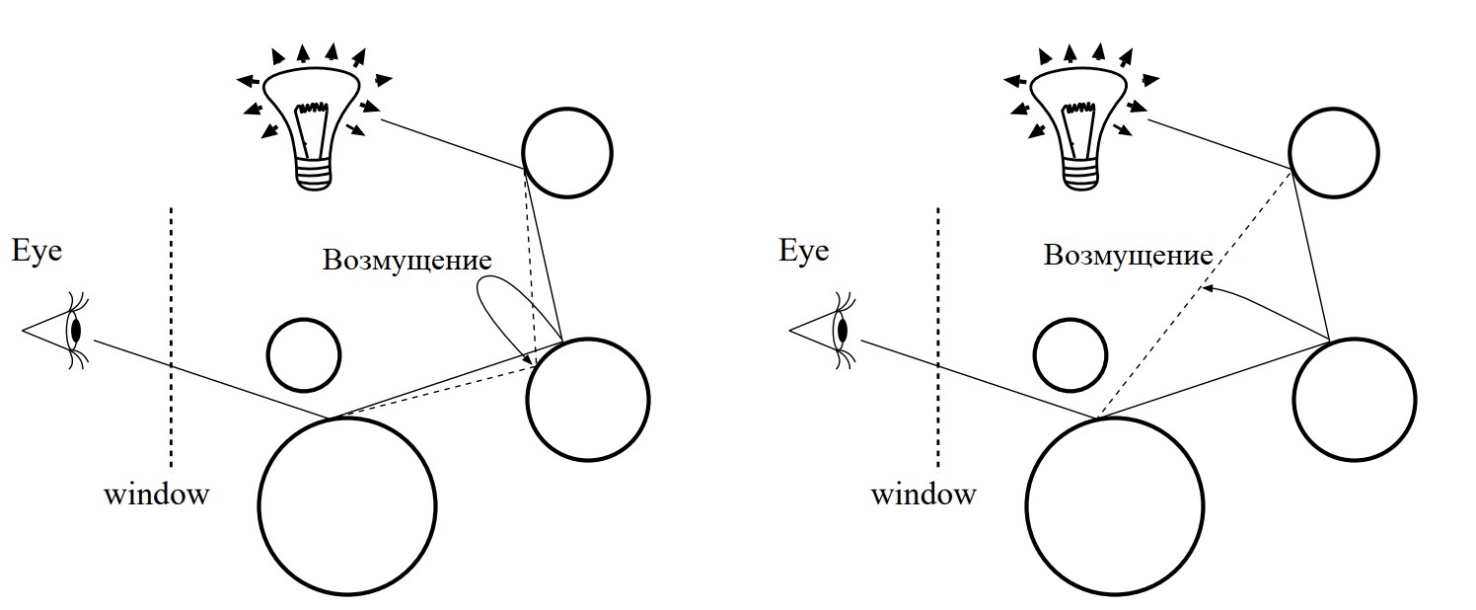
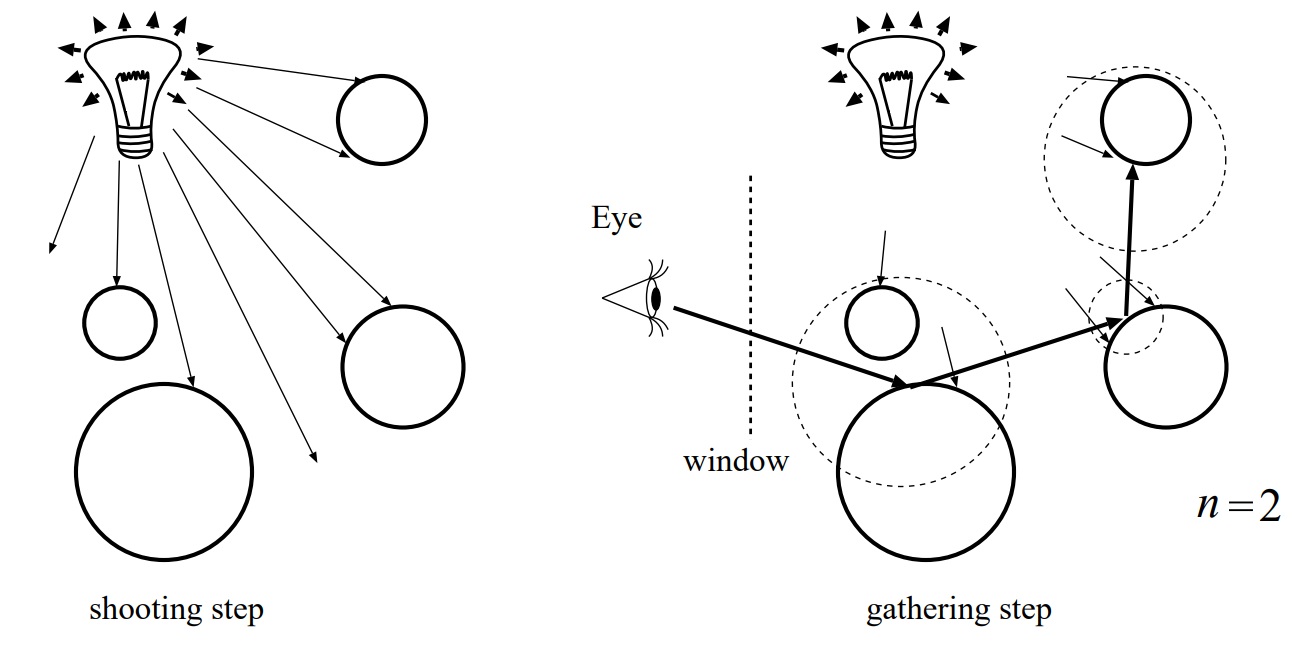


Рис. 9 Схема генерации модификаций пути в Metropolis Light Transport

7 Метод фотонных карт (Photon mapping)

Алгоритм состоит из двух этапов: трассировка и сохранение фотонов в фотонной карте и после расчёт освещённости с использованием данных из фотонной карты. На первом этапе трассировки из источника света испускаются фотоны. Фотоны испытывают ряд переотражений и преломлений, оставляя часть своей энергии на поверхностях объектов. Совокупность энергий фотонов, оставленных на поверхностях объектов, образуют фотонную карту. На следующем этапе необходимо собрать энергию в фотонных картах. Сбор освещенности можно делать по-разному. Чаще всего сбор идет по сфере по каждой отдельной поверхности объекта. В каждой точке сбора мы просматриваем окрестность этой точки и суммируем значения точек фотонной карты, попавших в эту окрестность, нормируем результат – это и будет результирующая освещенность в точке.

  
Рис. 10 Схема этапов метода фотонных карт

**2. Сравниваемые рендеры и метрики качества изображения**

В экспериментах, представленных ниже, исследованы следующие рендеры

1. Cycles Render — реализует алгоритм Path Tracing. Поддерживает как собственный язык программирования шейдеров, так и Open Shading Language (OSL)[7]. Способен воссоздать такие эффекты, как Max Bounces[8], Clamp[8], Caustics[9], смаз от движения (motion blur)[10].
2. appleseed Render – физически реалистичный рендер, поддерживающий такие алгоритмы глобального освещения, как Unidirectional Path Tracing, Stochastic Progressive Photon Mapping[11] и Light Tracing. Appleseed поддерживает Open Shading Language (OSL)[12]. Среди особенностей можно выделить возможность визуализировать RGB/спектральный/смешанный рендеринг, шевеленку(motion blur)[13], подповерхностное рассеивание (SSS)[14] и т.д.
3. LuxCore Render – LuxRender основан на PBRT, физической программе трассировки лучей. Он реализует такие алгоритмы глобального освещения, как Path Tracing, Tiled Path Tracing[15] – более оптимизированный для распараллеливания pathtracer и Bi-directional path-tracing (только для CPU). Поддерживаются такие алгоритмы выбора направления луча, как Random, Sobol[16] и Metropolis, который рекомендуется использовать в комбинации с Bidir, а также PhotonGI Cache[17]. Среди прочего он также поддерживает подповерхностное рассеивание (SSS).
4. ProRadeon Render

Основная метрика для сравнения изображений – отношение между максимумом возможного значения сигнала к мощности искажающего шума, PSNR (peak signal-to-noise ratio). Поскольку многие сигналы имеют широкий динамический диапазон, PSNR обычно измеряется в логарифмической шкале в децибелах [18].

PSNR наиболее часто используется для измерения уровня искажений при сжатии изображений. Эту меру можно связать со среднеквадратичной ошибкой (СКО, MSE, mean square error). Для двух монохромных изображений *I* и *K* размера *m×n*, одно из которых считается зашумленным приближением другого, MSE вычисляется по формуле (mse\_1):



Выражение PSNR с использованием MSE представлено в формуле (psnr\_1):



здесь – максимальное значение яркости пикселя изображения. Чаще всего, . Для цветных изображений *MSE* – это комбинация отклонений по трём каналам (красный, зелёный и синий), поэтому в аналоге формулы для цветного изображения знаменатель умножается на 3.

**3. Эксперименты**

В экспериментальной части исследования на представленном наборе данных было протестировано несколько алгоритмов классификации. Их результаты сравниваются в выводах работы. Реализациям алгоритмов трассировки предоставлялось примерно одинаковое количество времени для подготовки изображения (около 1000 – 1500 с.). Результаты их работы сравнивались с эталонным изображением.

Далее эксперимент продолжался примерно до 15000 – 20000 с. Качество полученных изображений сравнивалось на промежуточных этапах и по истечении этого времени.

**3.1. Эксперимент 1**

В данном эксперименте на белой диффузной плоскости расположена модель граненого стакана. Материал стакана – гладкое бесцветное стекло. Фон окружения полностью черный. Исследуется корректность визуализации прозрачного стекла и оптического эффекта каустики.

Рендер Cycles использует Path tracing, LuxCore – Bidir + Metropolis, appleseed – SPPM.

Таблица 1. Время работы (с точностью до единиц) и качество (с точностью до 0.01 дБ) полученного изображения реализациями алгоритмов трассировки appleseed, Cycles и LuxCore при отрисовке модели прозрачного стеклянного стакана.

|  |  |  |
| --- | --- | --- |
| Реализация алгоритма трассировки | Затраченное время, с | Качество изображения, дБ |
| Appleseed | 1186 | 34.09 |
| 1745 | 35.52 |
| 3325 | 38.01 |
| 6247 | 41.05 |
| 11952 | 45.48 |
| Cycles | 1006 | 11.64 |
| 2116 | 11.66 |
| 4113 | 11.69 |
| 7907 | 11.76 |
| 16301 | 11.90 |
| Luxcore | 917 | 14.35 |
| 1846 | 14.52 |
| 3592 | 14.59 |
| 7261 | 14.64 |
| 14467 | 14.66 |

По результатам, представленным в таблице 1, можно сделать вывод о том, что Cycles и Luxcore дают примерно к 1000-й секунде изображения, схожие по качеству. Их результаты дальнейшем улучшаются очень слабо. За то же время Appleseed даёт существенно лучшее изображение. При этом, качество результата в начале эксперимента растёт почти линейно со временем, затем темп роста качества со временем снижается, но отличается от других реализаций на порядок.

Замеры по времени не совпадают так как алгоритмы принимают в качестве параметра необходимое количество семплов, а не время работы. Я не стал обрубать работу на каком-то определенном моменте, потому что небольшие отклонения не критически влияют на график.

**3.2. Эксперимент 2**

На плоскости с деревянной текстурой расположено 5 стеклянных бутылок: 2 с острыми углами, две без углов и одна многогранная. 2 из них окрашены в постоянный цвет. Слева расположен источник света типа Area. Исследуется корректность визуализации стекла в сложных случаях.

Рендер Cycles использует Path tracing, LuxCore – Bidir + Sobol, appleseed – Path tracing.

Таблица 2. Время работы (с точностью до секунды) и качество (с точностью до 0.01 дБ) полученного изображения реализациями алгоритмов трассировки appleseed, Cycles и LuxCore при отрисовке полупрозрачных стеклянных бутылок

|  |  |  |
| --- | --- | --- |
| Реализация алгоритма трассировки | Затраченное время, с | Качество изображения, дБ |
| Appleseed | 1318 | 38.10 |
| 3612 | 43.02 |
| 5525 | 45.38 |
| 7405 | 47.21 |
| 10710 | 50.14 |
| Cycles | 1569 | 19.20 |
| 3110 | 19.25 |
| 6319 | 19.28 |
| 12627 | 19.30 |
| Luxcore | 1304 | 20.28 |
| 2643 | 20.90 |
| 5586 | 21.26 |
| 10100 | 22.33 |

Результаты этого эксперимента схожи с предыдущим опытом: Cycles и Luxcore примерно к 1500-й секунде получают результаты, которые в дальнейшем улучшаются слабо. Темп роста качества изображения, построенного appleseed, со временем постепенно снижается, но даже к 10000-й секунде он на порядок выше, чем в других алгоритмах.

**3.3. Эксперимент 3**

В этом эксперименте исследовалась способность визуализировать материал неплотной ткани, которая должна пропускать свет. Также на этот материал нанесено несколько текстур – изображений в формате JPEG.

Рендер Cycles использует Path tracing, LuxCore – Bidir + Metropolis, appleseed – Path tracing.

Таблица 3. Время работы (с точностью до секунды) и качество (с точностью до 0.01 дБ) полученного изображения реализациями алгоритмов трассировки appleseed, Cycles и LuxCore при отрисовке полупрозрачной шторы с узором

|  |  |  |
| --- | --- | --- |
| Реализация алгоритма трассировки | Затраченное время, с | Качество изображения, дБ |
| Appleseed | 1925 | 19.76 |
| 4816 | 23.42 |
| 6327 | 24.68 |
| 18981 | 31.20 |
| Cycles | 1890 | 19.65 |
| 3800 | 19.91 |
| 7655 | 20.07 |
| 15178 | 20.15 |
| Luxcore | 1709 | 16.78 |
| 3650 | 16.98 |
| 7304 | 17.09 |
| 14742 | 17.15 |

Характер результатов эксперимента несколько отличается от результатов предыдущего опыта. В условиях ограниченного времени, примерно к 2000-й секунде все рендеры получают изображение примерно одинакового качества. При этом, темп роста качества изображений, построенных с помощью Cycles и Luxcore близок к нулю, что говорит о достижении пика своих возможностей, как и в предыдущих экспериментах. За то же время качество изображения, полученного при работе appleseed, возрастает.

**3.4. Эксперимент 4**

В данном эксперименте исследовалась способность визуализации материала полупрозрачной неплотной ткани, пропускающей прямые лучи света без рассеивания. В отличие от предыдущего эксперимента, где в качестве источника света была плоскость, на некотором расстоянии отдаленная от отверстия, здесь используется освещение окружения (World).

Рендер Cycles использует Path tracing, LuxCore – Path tracing, appleseed – Path tracing.

Таблица 4. Время работы (с точностью до единиц) и качество (с точностью до 0.01 дБ) полученного изображения реализациями алгоритмов трассировки Appleseed, Cycles и Luxcore при отрисовке модели полупрозрачной шторы

|  |  |  |
| --- | --- | --- |
| Реализация алгоритма трассировки | Затраченное время, с | Качество изображения, дБ |
| Appleseed | 1480 | 9.80 |
| 2917 | 9.85 |
| 5727 | 9.88 |
| Cycles | 625 | 12.97 |
| 1247 | 12.98 |
| 2507 | 12.99 |
| 5002 | 13.00 |
| Luxcore | 1168 | 23.79 |
| 1714 | 25.88 |
| 2376 | 27.58 |
| 4580 | 32.35 |
| 6819 | 36.69 |

При отображении этой сцены лучшим оказался Luxcore. Качество построенного им изображения лучше, чем у других рендеров, в условиях ограничения по времени (примерно к 1500-й секунде), темп роста качества изображения со временем – наилучший. Изображения, построенные Cycles и appleseed, фактически со временем не улучшаются.

**Выводы**

В представленном исследовании произведено сравнение рендеров appleseed, Cycles и Luxcore по качеству построенных изображений трёхмерных сцен на основе алгоритма трассировки лучей. Результаты их работы сравнивались при ограниченном времени работы (около 1500 с.), а также рассмотрена динамика улучшения качества изображения при увеличении времени работы.

В большинстве тестов наилучшую динамику показал рендер appleseed, который и в условиях ограниченного времени давал наиболее фотореалистичное изображение, и с течением времени демонстрировал наилучший темп роста качества изображения. При этом, рендеры Cycles и Luxcore демонстрировали схожие результаты: они дали изображения приблизительно равного качества в условиях ограниченного времени, а с течением времени качество изображения менялось мало.

При этом, в эксперименте 4 рендер Luxcore продемонстрировал лучший результат и за органиченное время, и наилучшую динамику изменения качества изображения со временем.

**Приложения**

Получившиеся в результате экспериментов изображения доступны по ссылке: https://github.com/goginyanboris/3D

**Литература**

1. Weinmann F. Comparison of Intersection Algorithms for SBR Ray Tracing on NURBS // Proceedings of the 2012 IEEE International Symposium on Antennas and Propagation. – 2012. – IEEE, Chicago, IL, USA. doi:10.1109/aps.2012.6349081
2. Wang J., Shen, Y., Ding, W. A Method for Ink-Wash Painting Rendering for 3D Scenes // Proceedings of the 2018 International Joint Conference on Information, Media and Engineering (ICIME). – 2018. – Pp. 205 – 210. – IEEE, Osaka, Japan. doi:10.1109/icime.2018.00050
3. Gu K., Liu M., Zhai G., Yang, X., Zhang, W. Quality Assessment Considering Viewing Distance and Image Resolution // IEEE Transactions on Broadcasting. – 2015. – Vol. 61(3). - Pp. 520–531. doi:10.1109/tbc.2015.2459851
4. Cho S., Shin S., Choi S. Haptic texture rendering using random fractal surface // Proceedings of the 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI). – 2017. – Pp. 290 – 292. – IEEE, Jeju, South Korea. doi:10.1109/ urai.2017.7992732
5. Shumskiy V. GPU Ray Tracing – Comparative Study of Ray-Triangle Intersection Algorithms // Transactions on Computational Science XIX. Lecture Notes in Computer Science. – 2013. – Vol. 7870. – Springer, Berlin, Heidelberg. – Pp. 78 – 91. doi:10.1007/978-3-642-39759-2
6. Zlatuska M., Havran V. Ray Tracing on GPU with CUDA - Comparative Study of Three Algorithms // WSCG 2010: Communication Papers Proceedings: 18th International Conference in Central Europe. – 2010. URL: https://www.researchgate.net/publication/228974703\_Ray\_Tracing\_on\_a\_GPU\_with\_CUDA-comparative\_study\_of\_three\_algorithms
7. Open Shading Language in Cycles [Электронный ресурс]. – Режим доступа: http://opensource.imageworks.com/?p=osl
8. Max Bounced [Электронный ресурс]. – Режим доступа: <https://docs.blender.org/manual/ru/dev/render/cycles/render_settings/light_paths.html>
9. Caustic [Электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/Caustic\_(optics)
10. Blender Motion Blur [Электронный ресурс]. – Режим доступа: https://docs.blender.org/manual/ru/dev/render/cycles/render\_settings/motion\_blur.html
11. Hachisuka, T., & Jensen, H. W. (2009). Stochastic progressive photon mapping. ACM Transactions on Graphics, 28(5), 1. doi:10.1145/1618452.1618487
12. OSL in appleseed [Электронный ресурс]. – Режим доступа: https://blenderseed.readthedocs.io/en/latest/osl/osl.html
13. appleseed Motion Blur [Электронный ресурс]. – Режим доступа: https://appleseed.readthedocs.io/projects/appleseed-blenderseed/en/latest/panels/render/motion\_blur.html
14. appleseed SSS [Электронный ресурс]. – Режим доступа: https://github.com/appleseedhq/appleseed/blob/master/docs/source/features/features.rst
15. Tiled Path (CPU/OpenCL) [Электронный ресурс]. – Режим доступа: https://wiki.luxcorerender.org/Render\_Configuration
16. И. М. Соболь. Численные методы Монте-Карло. М.: Наука, 1973 г.
17. PhotonGI cache [Электронный ресурс]. – Режим доступа: https://wiki.luxcorerender.org/PhotonGI
18. Peak signal-to-noise ratio [Электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/Peak\_signal-to-noise\_ratio

**References**

1. Weinmann F. Comparison of Intersection Algorithms for SBR Ray Tracing on NURBS// Proceedings of the 2012 IEEE International Symposium on Antennas and Propagation. – 2012. – IEEE, Chicago, IL, USA. doi:10.1109/aps.2012.6349081
2. Wang J., Shen, Y., Ding, W. A Method for Ink-Wash Painting Rendering for 3D Scenes // Proceedings of the 2018 International Joint Conference on Information, Media and Engineering (ICIME). – 2018. – Pp. 205 – 210. – IEEE, Osaka, Japan. doi:10.1109/icime.2018.00050
3. Gu K., Liu M., Zhai G., Yang, X., Zhang, W. Quality Assessment Considering Viewing Distance and Image Resolution // IEEE Transactions on Broadcasting. – 2015. – Vol. 61(3). - Pp. 520–531. doi:10.1109/tbc.2015.2459851
4. Cho S., Shin S., Choi S. Haptic texture rendering using random fractal surface // Proceedings of the 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI). – 2017. – Pp. 290 – 292. – IEEE, Jeju, South Korea. doi:10.1109/ urai.2017.7992732
5. Shumskiy V. GPU Ray Tracing – Comparative Study of Ray-Triangle Intersection Algorithms // Transactions on Computational Science XIX. Lecture Notes in Computer Science. – 2013. – Vol. 7870. – Springer, Berlin, Heidelberg. – Pp. 78 – 91. doi:10.1007/978-3-642-39759-2
6. Zlatuska M., Havran V. Ray Tracing on GPU with CUDA - Comparative Study of Three Algorithms // WSCG 2010: Communication Papers Proceedings: 18th International Conference in Central Europe. – 2010. URL: https://www.researchgate.net/publication/228974703\_Ray\_Tracing\_on\_a\_GPU\_with\_CUDA-comparative\_study\_of\_three\_algorithms
7. Open Shading Language in Cycles [Электронный ресурс]. – Режим доступа: http://opensource.imageworks.com/?p=osl
8. Max Bounced [Электронный ресурс]. – Режим доступа: <https://docs.blender.org/manual/ru/dev/render/cycles/render_settings/light_paths.html>
9. Caustic [Электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/Caustic\_(optics)
10. Blender Motion Blur [Электронный ресурс]. – Режим доступа: https://docs.blender.org/manual/ru/dev/render/cycles/render\_settings/motion\_blur.html
11. Hachisuka, T., & Jensen, H. W. (2009). Stochastic progressive photon mapping. ACM Transactions on Graphics, 28(5), 1. doi:10.1145/1618452.1618487
12. OSL in appleseed [Электронный ресурс]. – Режим доступа: https://blenderseed.readthedocs.io/en/latest/osl/osl.html
13. appleseed Motion Blur [Электронный ресурс]. – Режим доступа: https://appleseed.readthedocs.io/projects/appleseed-blenderseed/en/latest/panels/render/motion\_blur.html
14. appleseed SSS [Электронный ресурс]. – Режим доступа: https://github.com/appleseedhq/appleseed/blob/master/docs/source/features/features.rst
15. Tiled Path (CPU/OpenCL) [Электронный ресурс]. – Режим доступа: https://wiki.luxcorerender.org/Render\_Configuration
16. И. М. Соболь. Численные методы Монте-Карло. М.: Наука, 1973 г.
17. PhotonGI cache [Электронный ресурс]. – Режим доступа: https://wiki.luxcorerender.org/PhotonGI
18. Peak signal-to-noise ratio [Электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/Peak\_signal-to-noise\_ratio

Об авторах

Гогинян Борис Андреевич, студент группы КМБO-03-16 кафедры Высшей математики Института кибернетики ФГБОУ ВО «МИРЭА – Российский технологической университет» (119454, Россия, Москва, пр-т Вернадского, д. 78).

Парфенов Денис Васильевич, кандидат технических наук, доцент кафедры Высшей математики Института кибернетики ФГБОУ ВО «МИРЭА – Российский технологической университет» (119454, Россия, Москва, пр-т Вернадского, д. 78).

Петрусевич Денис Андреевич, кандидат физико-математических наук, доцент кафедры Высшей математики Института кибернетики ФГБОУ ВО «МИРЭА – Российский технологической университет» (119454, Россия, Москва, пр-т Вернадского, д. 78).

About authors

Boris A. Goginyan, student of the Higher Mathematics Chair, Institute of Cybernetics, Russian Technological University (78, Vernadskogo Pr., Moscow 119454, Russia).

Denis V. Parfenov, PhD, associate professor of the Higher Mathematics Chair, Institute of Cybernetics, Russian Technological University (78, Vernadskogo Pr., Moscow 119454, Russia).

Denis A. Petrusevich, PhD, associate professor of the Higher Mathematics Chair, Institute of Cybernetics, Russian Technological University (78, Vernadskogo Pr., Moscow 119454, Russia).