

Аннотация

В данной работе исследованы математические методы моделирования распределения яркости в трехмерных сценах, с особым акцентом на "локальные оценки метода Монте-Карло". Это метод моделирования, который на текущий момент малоизвестен и не используется на практике, но обладает большим потенциалом, подтвержденным, в том числе в данной работе.

Целью работы является повышение точности и эффективности визуализации синтетических изображений путем устранения проблем "зашумленности", связанных с принципом работы метода Монте-Карло, и ускорения процесса рендеринга.

В данном исследовании проведен подробный аналитический обзор актуальных научных публикаций по теме моделирования света и визуализации, определены наиболее эффективные методы предварительной обработки данных и алгоритмы для решения задачи рендеринга. Был протестирован метод "Локальных оценок" при помощи аналитической задачи в среде Matlab, а также единственная на текущий момент, реализующая моделирование "Локальными оценками", отечественная программа. В рамках практической части была разработана программа для тестирования методов визуализации, производящая моделирование света трехмерной сцены при помощи трассировки лучей методом Монте-Карло. Формируемые изображения через заданные промежутки времени подвергались устраниению шума при помощи нейросетевого денойзера. Применимость и эффективность реализованных методов была проверена и подтверждена.

В результате по итогу работы было разработано программное обеспечение для моделирования освещения с использованием метода Монте-Карло, демонстрирующее возможности устраниния искажений изображения и ускорения визуализации. Полученные результаты показывают

значительное улучшение качества визуализации и сокращение времени рендеринга.

Выпускная квалификационная работа изложена на 88 страницах, содержит 18 рисунков, список использованных источников из 22 наименований, 2 приложения.

Abstract

This work investigates mathematical methods for modeling brightness distribution in three-dimensional scenes, with a particular focus on the "local estimation method of Monte Carlo." This modeling method is currently little known and not used in practice but has great potential, as confirmed by this study.

The aim of the work is to improve the accuracy and efficiency of synthetic image visualization by eliminating noise issues associated with the Monte Carlo method and accelerating the rendering process.

This study provides a detailed analytical review of current scientific publications on light modeling and visualization, identifying the most effective data preprocessing methods and algorithms for solving the rendering task. The "Local Estimation" method was tested using an analytical task in Matlab, as well as the only currently available domestic program that implements modeling using "Local Estimations." In the practical part, a program was developed for testing visualization methods, performing light modeling of a three-dimensional scene using the Monte Carlo ray tracing method. The generated images were subjected to noise reduction at specified intervals using a neural network denoiser. The applicability and effectiveness of the implemented methods were verified and confirmed.

As a result, software was developed for lighting modeling using the Monte Carlo method, demonstrating the ability to eliminate image distortions and accelerate visualization. The obtained results show significant improvement in visualization quality and reduction in rendering time.

The thesis is presented on 88 pages, contains 18 figures, a list of references with 22 titles, and 2 appendices.

Содержание

Введение	9
1. Математическое моделирование и визуализация изображения	12
1.1 Создание реалистического изображения	12
1.2 Методы математического моделирования освещения	16
1.3 Решение уравнения глобального освещения методом трассировки лучей	22
1.4 Метод Монте-Карло	28
1.5 Аналитические решения уравнения глобального освещения	32
1.6 Локальные оценки	34
1.7 Выводы раздела	39
2. Инструменты для моделирования света и визуализации	40
2.1 Библиотека Embree	40
2.2 Варианты оптимизации процесса моделирования	43
2.3 Использование нейросетей для денойзинга	49
2.4 Библиотека Intel Open Image Denoise	57
3. Практическая часть	64
Заключение	73
Список используемой литературы	75

Введение

В современном мире компьютерная графика играет ключевую роль в различных областях, начиная от разработки игр и фильмов до научных исследований и проектирования. Одной из важных задач в этой области является создание реалистического изображения, то есть стремление к воссозданию на экране картинки неотличимой от реальности, корректно демонстрирующей все физические явления, такие как тени от объектов, отражения, преломления, и т.д. Это прежде всего достигается за счет света. Распределение яркости по сцене — это и есть процесс в компьютерной графике, который определяет, как свет воздействует на объекты в виртуальном мире. Физически достоверное освещение в сцене предоставляет ряд преимуществ, делая изображения более реалистичными и привлекательными для зрителя. Оно легко воспринимается и понимается человеческим мозгом. Благодаря использованию физически обоснованных моделей света, изображения соответствуют естественным явлениям, что улучшает визуальный опыт зрителя. Физически достоверное освещение также позволяет достичь различных визуальных эффектов, таких как реалистичные блики, тени и отражения. С коммерческой точки зрения, такое освещение позволяет создавать более реалистичные и привлекательные визуальные эффекты. Это имеет важность в различных индустриях, например в развлекательной, рекламной и дизайнерской сферах. Благодаря более реалистичным изображениям, компании могут привлекать больше внимания к своим продуктам или услугам, улучшая их презентацию и привлекательность для потенциальных клиентов. Это может способствовать увеличению продаж и конкурентоспособности бренда на рынке. С научной точки зрения, такое освещение играет важную роль в исследованиях и разработках в области компьютерной графики и компьютерного зрения. Использование реалистичных моделей освещения помогает ученым и инженерам лучше понять физические процессы, происходящие в

виртуальных и реальных сценах, и разрабатывать более эффективные методы моделирования и отображения изображений. Это может привести к созданию новых технологий и инноваций, которые могут быть применены в различных областях, таких как медицина, инженерия, наука о материалах и другие.

Существует множество методов моделирования освещения, каждый из которых имеет свои преимущества и недостатки. В работе рассмотрены такие методы, как трассировка лучей, а также локальные оценки метода Монте-Карло, которые представляют собой эффективный подход к решению задачи визуализации.

Последние в свою очередь являются редким, но перспективным методом, по которому на текущий момент очень мало исследований и имеются нерешенные проблемы. Для составления конкуренции коммерчески популярным и часто используемым методам, локальные оценки требуется доработать и ускорить получение результатов. Актуальность текущей работы заключается в исследовании перспективного метода, и в нахождении способа, способного решить проблему локальных оценок. В работе рассмотрены математические основы, исследованы возможности для ускорения визуализации синтетических изображений, в том числе различные варианты оптимизации процесса моделирования, включая использование графических процессоров и нейронных сетей. В частности, такой метод ускорения как денойзинг, и библиотека Intel Open Image Denoise. Полученные результаты имеют практическую ценность и актуальность, так как библиотека на текущий момент находится на стадии разработки и не применяется повсеместно, однако обладает конкурентоспособными возможностями.

На практике была написана программа для рендеринга освещения, моделирующая искажения визуализации, возникающие у метода локальных оценок, а также применен денойзер, устраняющий эти искажения и ускоряющий получения результата. Полученные данные были подвергнуты объективному оцениванию и результаты были отражены в конце работы.

Подводя итог, объектом исследования данной работы является математическое моделирование распределения яркости по сцене.

Цель работы заключается в исследовании возможностей ускорения существующих методов визуализации синтетических изображений, в частности локальных оценок метода Монте-Карло.

Задачи исследования:

- 1) Провести аналитический обзор научных публикаций для изучения процесса математического моделирования яркости и визуализации.
- 2) Рассмотреть теоретическую основу инструментов визуализации, а также ее оптимизацию. В частности, библиотеки Intel Embree и Intel Open Image Denoise.
- 3) Использовать полученные знания для практической реализации программного обеспечения, способного продемонстрировать работу найденных возможностей по улучшению эффективности визуализации.
- 4) Оценить полученные результаты и сформулировать окончательный вывод.

Исходя из цели исследования, в первом разделе рассмотрены возможности визуализации изображений и определены основные проблемы, с которыми приходится сталкиваться на практике. Второй раздел посвящен способам решения этих проблем. Третий раздел демонстрирует применение на практике найденных решений. Заключение содержит основные выводы из работы.

1. Математическое моделирование и визуализация изображения

1.1 Создание реалистического изображения

В процессе создания изображения трехмерного пространства на компьютере важно помнить, что мы работаем с математической моделью, которая хранится в памяти компьютера. Фактически на экране отображается растровое изображение, которое является двухмерным представлением трехмерной модели. Таким образом, целью алгоритмов компьютерной графики является установление соответствия между трехмерным и двухмерным мирами, чтобы достичь наиболее реалистичного визуального восприятия.

С развитием компьютерной графики в 60-70-е годы прошлого века, основными направлениями стали методы рисования линий и поверхностей, а также удаление скрытых элементов изображения. Проект Sketchpad, представленный в 1963 году, сыграл ключевую роль, позволяя пользователям взаимодействовать с компьютером в интерактивном режиме через световое перо и создавать векторные рисунки на экране. В то время в СССР началась разработка библиотеки графических программ "Графор" на языке программирования Фортран. Эта библиотека позволяла выводить графические примитивы на граffопостроитель и дисплей, что открыло новые возможности в инженерном деле. В последующие годы компьютерная графика продолжала развиваться, включая методы моделирования сглаженных кривых, воспроизведение поверхностей с цветовой градацией и создание виртуальной реальности на основе проектирования объектов. В 1980-х стандартные библиотеки компьютерной графики были созданы в ISO, что способствовало стандартизации и распространению методов и инструментов для работы с ней.

Таким образом, развитие компьютерной графики вело к созданию широкого спектра инструментов и методов, включая обработку изображений,

создание виртуальной реальности, алгоритмы затенения, закрашивания. Физически достоверное освещение в сцене в компьютерной графике достигается с помощью различных инструментов и методов, которые моделируют поведение света в соответствии с естественными физическими законами.

Одними из ключевых инструментов для моделирования света являются локальное и глобальное освещение. Эти два важнейших понятия в компьютерной графике и компьютерном зрении играют решающую роль в визуализации трехмерных сцен. На рисунке (рисунок 1) демонстрируются тестовая сцена Cornell Box, в которой наглядно увидеть различия между локальным и глобальным освещением с точки зрения визуализации.

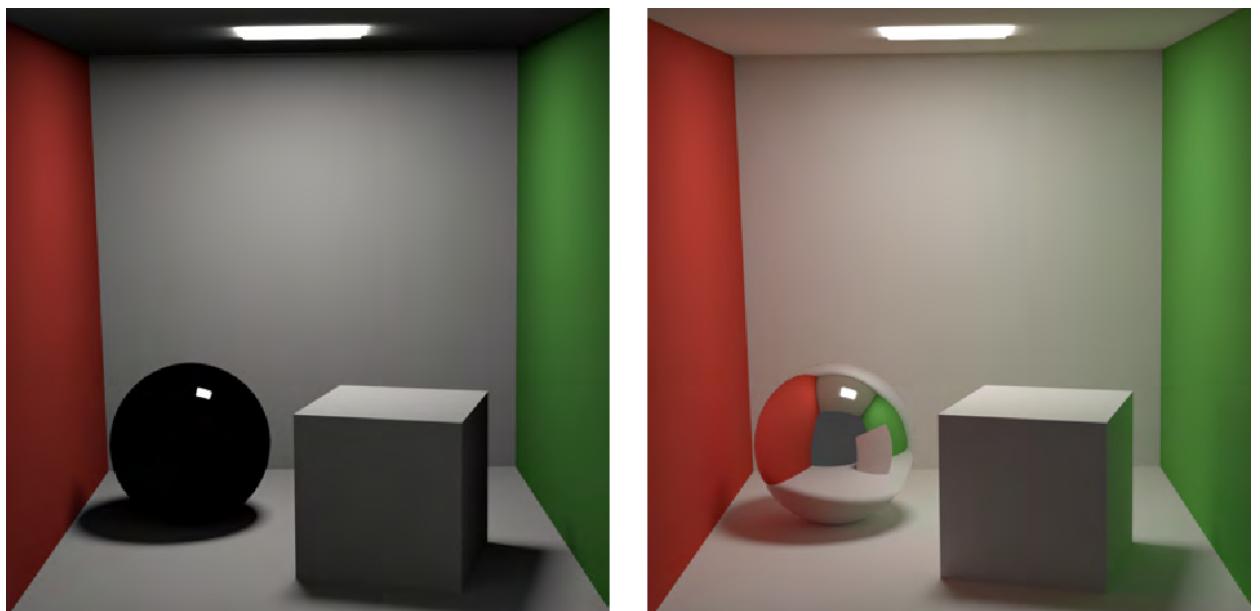


Рисунок 1 – Слева локальное освещение, а справа – глобальное.

Локальное освещение фокусируется на расчете взаимодействия света в ограниченном контексте, обычно рассматривая только прямые источники света и свойства поверхности в определенной точке сцены. При этом используются упрощенные модели, такие как модель отражения Фонга, которая включает компоненты амбиентного, диффузного и спекулярного отражения. Окружающее отражение представляет собой равномерно рассеянный свет, способствующий общей яркости сцены. Диффузное

отражение описывает отражение света от поверхностей в различных направлениях на основе нормалей поверхности, а спекулярное отражение определяет отражение в определенных направлениях, обычно ассоциируемое с глянцевыми поверхностями.

Локальное освещение, хотя и является эффективным с точки зрения вычислений методом рендеринга сцен, имеет ряд ограничений по сравнению с глобальным освещением. Во-первых, ему не хватает реалистичности, поскольку оно не может передать всю сложность реальных сценариев освещения. Это связано с тем, что модели локального освещения не учитывают косвенные эффекты освещения, такие как отражение света от объектов и его преломление, которые очень важны для достижения фотореалистичной визуализации. Следовательно, сцены, отрисованные с использованием только локального освещения, могут выглядеть плоскими и лишенными визуальной насыщенности. Более того, алгоритмы локального освещения учитывают только прямое взаимодействие света в конкретной точке сцены, что ограничивает их возможности по моделированию взаимодействия света с поверхностями за пределами непосредственной близости. Это может привести к неточностям в изображении эффектов освещения, особенно в сценах со сложной геометрией и материалами. Кроме того, тени, создаваемые моделями локального освещения, часто нереалистичны, лишены мягкости и имеют резкие края из-за отсутствия моделирования непрямого освещения. Методы локального освещения не могут точно смоделировать такие явления глобального освещения, как цветовое кровотечение (color bleeding), когда на цвет одной поверхности влияет отраженный свет от соседних поверхностей. Это ограничение приводит к тому, что в сценах отсутствует тонкое взаимодействие света и цвета, наблюдаемое в реальном мире. Материалы со сложными оптическими свойствами, такие как полупрозрачные или рассеивающие подповерхностный свет, также создают проблемы для моделей локального освещения, поскольку требуют учета рассеивания света в объеме материала.

Наконец, достижение визуально привлекательных результатов с помощью локального освещения часто требует обширной ручной настройки параметров освещения и материалов, что может отнимать много времени и не всегда приводит к удовлетворительным результатам, особенно в сложных сценах. В итоге несмотря на то, что локальное освещение обеспечивает эффективность вычислений, оно не в состоянии передать реалистичность и сложность освещения в реальном мире по сравнению с методами глобального освещения.

Глобальное же освещение направлено на более реалистичное моделирование взаимодействия света в сцене, учитывая косвенные эффекты освещения. В отличие от алгоритмов локального освещения, алгоритмы глобального прослеживают лучи света через сцену, учитывая множество взаимодействий, включая отражение и преломление от поверхностей, а также рассеивание в различных средах, например в тумане или воде.

Такое реалистичное воспроизведение трехмерной сцены с учетом всех эффектов многократных коллизий луча на поверхностях объектов является сложной математической задачей. Для этого необходимо учитывать различные типы отражений, такие как диффузные, зеркальные и направленно-диффузные, а также учитывать взаимодействие света с различными материалами и текстурами. Одной из ключевых задач глобального освещения является моделирование теней и отражений сцены. Это позволяет создавать более реалистичные изображения, где тени и отражения влияют на общее визуальное восприятие сцены. При учете глобального освещения изменяется и понимание фотометрических величин.

Например, светимость поверхности теперь зависит не только от ее собственного излучения, но и от отраженного от других объектов в сцене света. Важным для физически достоверного освещения является и использование реалистичных моделей материалов, которые учитывают различные типы поверхностей и их взаимодействие со светом.

Глобальное освещение включает учет всех возможных эффектов в лучевом приближении, но фактически эта задача невыполнима из-за бесконечного количества взаимодействующих между собой объектов во Вселенной. И так как большинство объектов в сцене оказывают несущественное влияние на освещение, их можно исключить и ограничиться значимыми коллизиями, чтобы построить освещение в сцене.

1.2 Методы математического моделирования освещения

Глобальное освещение в компьютерной графике часто реализуется с использованием трассировки лучей. Это метод, который используется для моделирования пути света в трехмерной сцене. Он работает следующим образом: из каждой точки наблюдения (обычно камеры) выпускаются лучи света, которые проходят через каждый пиксель изображения и распространяются по сцене. Когда лучи сталкиваются с объектами в сцене, они могут отражаться, преломляться или поглощаться, в зависимости от свойств поверхностей объектов. Так как глобальное освещение описывает вклад света, который приходит в сцену не только от прямых источников света, но и от отраженных и рассеянных лучей, то при помощи трассировки лучей возможно учитывать такие сложные эффекты взаимодействия света с окружающей средой. Для создания глобального освещения с помощью трассировки, кроме лучей, выпущенных из камеры, генерируются дополнительные лучи, которые представляют отраженные или рассеянные лучи света. Эти дополнительные лучи следуют в направлении, отраженном от поверхности объектов, или в случайном направлении для учета рассеянного света. Часто используются различные оптимизации и приближения, чтобы уменьшить вычислительную нагрузку и ускорить процесс рендеринга.

Рендеринг (англ. rendering) — это фундаментальный компонент компьютерной графики, процесс конвертации описания трехмерных объектов в изображение. Алгоритмы для работы с графикой должны быть пропущены через определенную стадию рендеринга для того, чтобы их результат мог стать видимым на изображении.

Рендеринг можно разделить на два типа: первый - с результатом в реальном времени, в этом случае изображение рендерится со скоростью 30 кадров в секунду или выше, и второй - отложенный рендеринг, когда нет необходимости поддерживать высокую частоту кадров в секунду, а кадры

собираются в единый поток на этапе сборки. Первый вариант используется для создания интерактивной 3D-графики (игры, симуляторы, дополненная реальность), второй - для 3D-графики в киноиндустрии, визуализации, рекламной продукции. Основной критерий качества у рендеринга — это фотореалистичность создаваемого изображения, и это напрямую зависит от выбранной технологии рендеринга. В рендеринге решается специальное уравнение под соответствующим названием «Уравнение рендеринга».

В 1986 году Джеймс Каджия представил статью под названием «Уравнение рендеринга и его использование в компьютерной графике» [1]. На семи страницах Каджия описал способ математического вычисления физических свойств света, и именно это открытие изменило 3D-рендеринг.

Уравнение рендеринга — это интегральное уравнение, которое определяет количество светового излучения в определённом направлении как сумму собственного и отражённого излучения [2]. Физической основой уравнения является закон сохранения энергии. Пусть L — это яркость по заданному направлению в заданной точке пространства. Тогда количество исходящего излучения (L_o) — это сумма излучённого света (L_e) и отражённого света. Отражённый свет может быть представлен как сумма приходящей яркости (L_i) по всем направлениям, умноженной на коэффициент отражения из данного угла.

Уравнение рендеринга представлено на рисунке (рисунок 2).

$$L_o(\mathbf{x}, \omega, \lambda, t) = L_e(\mathbf{x}, \omega, \lambda, t) + \int_{\Omega} f_r(\mathbf{x}, \omega', \omega, \lambda, t) L_i(\mathbf{x}, \omega', \lambda, t) (-\omega' \cdot \mathbf{n}) d\omega'$$

Рисунок 2 – Уравнение рендеринга.

Где:

λ — Длина волны света.

t — Время.

L_o — Свет с заданной длиной волны λ излучаемый в направлении ω во время t , из заданной точки x .

f — Двунаправленная функция распределения отражения, яркость отражённого от ω' к ω в точке x , во время t , на длине волны λ .

L_e — Излучённый свет.

L_i — Длина волны λ по входящему направлению к точке x из направления ω' во время t .

($-\omega' * n$) — Поглощение яркости по заданному углу.

\int — Интеграл по полусфере входящих направлений.

До этого еще одним важным вкладом было получение Парри Муном (Parry Hiram Moon, 14.02.1898, Бивер Дам – 4.03.1988, Бостон) уравнение в упрощенном виде: (рисунок 3).

$$M(\mathbf{r}) = M_0(\mathbf{r}) + \frac{\sigma(\mathbf{r})}{\pi} \int_{\Sigma} M(\mathbf{r}') F(\mathbf{r}, \mathbf{r}') \Theta(\mathbf{r}, \mathbf{r}') d^2 r'$$

Рисунок 3 - Уравнение излучательности.

Где:

$M(r)$ — Светимость точки поверхности r .

$M_0(r)$ — Светимость точки r , полученная непосредственно от прямых источников света.

$\Theta(r, r')$ — Функция видимости элемента $d^2 r'$ из точки r .

$F(r, r')$ — Элементарный форм-фактор.

Достоинством его стала зависимость функции $M(r)$ только от координат поверхности. Оно определяется не только собственным свечением поверхности, но и отражением падающего с других поверхностей света, а потому характеризует не только саму поверхность, но и ее положение и свойства других поверхностей. В литературе оно получило название уравнение излучательности или радиосити (radiosity).

Уравнение глобального освещения (УГО) — это математическое выражение, используемое в компьютерной графике для моделирования освещения в трехмерных сценах. Оно описывает распределение яркости и цвета поверхностей в сцене, учитывая различные источники света, отражения от поверхностей и преломление света. УГО представляет собой интегральное уравнение, в котором искомая функция яркости точек поверхности зависит от множества факторов, таких как расположение источников света, свойства поверхностей и их взаимодействие друг с другом.

УГО можно представить в виде интегрального уравнения Фредгольма II рода для определения поля яркости точек к поверхности Σ с учетом многократных переотражений (рисунок 4), которое учитывает все кратности диффузно-зеркального отражения, пропускания и собственное излучение, что и является предметом теории глобального освещения. В этом (рисунок 4) виде УГО записано для яркости и описывает процесс переноса излучения в объеме с многократными отражениями в пятимерном пространстве координат и направлений, процесс получения которого подробно описывается в источнике [3].

$$L(\mathbf{r}, \hat{\mathbf{l}}) = L(\mathbf{r}, \hat{\mathbf{l}}) + \frac{1}{\pi} \int_{\Sigma} L(\mathbf{r}', \hat{\mathbf{l}}') \sigma(\mathbf{r}; \hat{\mathbf{l}}, \hat{\mathbf{l}}') F(\mathbf{r}, \mathbf{r}') \Theta(\mathbf{r}, \mathbf{r}') d^2 r'$$

Рисунок 4 – Уравнение глобального освещения.

Где:

$L(r, \hat{l})$ – яркость в точке r в направлении \hat{l} .

$\sigma(r; \hat{l}, \hat{l}')$ – двунаправленная функция рассеяния (отражения или преломления).

$\Theta(r, r')$ – Функция видимости элемента $d^2 r'$ из точки r .

$F(r, r')$ – Элементарный форм-фактор.

Аналитических решений УГО не найдено, а численное решение стало возможным только с появлением быстродействующих компьютеров. Среди численных методов решения используется прямое моделирование, прямая и обратная трассировка лучей, метод конечных элементов (radiosity), метод фотонных карт и другие. При этом математически строгое обоснование имеет только метод конечных элементов, основанный на диффузном приближении, а остальные методы носят преимущественно феноменологический характер. У различных алгоритмов и методов решения есть как свои преимущества, так и ограничения.

Метод последовательных приближений представляет собой итеративный процесс, в ходе которого решение уравнения уточняется на каждом шаге. Этот метод является наиболее стабильным для решения сложных уравнений, таких как УГО, и позволяет достичь высокой точности результата.

Метод излучательности (radiosity) — это численный подход к моделированию освещенности в трехмерных сценах. Он основан на представлении сцены в виде набора конечных пучков лучей, которые исходят от источников света и распространяются по сцене, взаимодействуя с поверхностями объектов. Метод часто включает в себя итерационные методы решения, такие как методы Якоби и Гаусса-Зейделя [4]. Эти методы

позволяют приблизительно решать уравнения излучательности с большой точностью. Сходимость таких методов зависит от физических свойств системы, в частности, от коэффициентов отражения и поглощения, и может быть обеспечена при выполнении определенных условий, таких как условие $k_{ij} < 1$, где k_{ij} - коэффициент переноса излучения от элемента i к элементу j . Также важно определить последовательность элементов для вычисления на каждой итерации, чтобы повысить скорость сходимости метода. Общая схема метода излучательности включает представление сцены в виде сетки граней, вычисление форм-факторов между ними, решение уравнения излучательности итерационными методами, и наконец, проецирование результатов на экран для получения изображения сцены. Для улучшения качества изображения и устранения «гранёности» сцены может потребоваться применение алгоритмов аппроксимации значений излучательности и распределение нераспределенного светового потока между гранями сцены. В контексте этого метода, освещенность сцены описывается в терминах радиационной энергии, которая распространяется в пространстве. Метод особенно эффективен для диффузных поверхностей, где яркость не слишком зависит от угла наблюдения. В итоге он позволяет достаточно точно моделировать освещенность поверхностей и создавать трехмерные изображения сцен с приемлемым качеством за разумное время. Недостатком является неспособность учета зеркальных отражений света. Метод часто базируется на методе конечных элементов.

Метод конечных элементов (МКЭ) основан на идеи аппроксимации непрерывной функции дискретной моделью, состоящей из кусочно-непрерывных функций, определенных на конечном числе подобластей (конечных элементах). Элементы геометрической области разбиваются таким образом, чтобы на каждом из них неизвестная функция аппроксимировалась некоторой функцией. Эти функции должны удовлетворять граничным условиям задачи. МКЭ позволяет получить приближенное решение задачи с

высокой точностью путем решения [5] системы уравнений, описывающих физические явления в каждом элементе.

1.3 Решение уравнения глобального освещения методом трассировки лучей

Для численного решения УГО часто применяется метод трассировки лучей, который позволяет моделировать путь световых лучей от источников освещения до камеры или точки наблюдения. В работе [6] авторами приводится решение. Суть его заключается в том, что уравнение глобального освещения можно представить в операторной форме, что облегчает его численное решение. В такой форме используется интегральный оператор, который выражает связь между освещенностью в заданной точке и ее значениями в соседних точках. Для численного решения уравнения применяется итерационный метод (рисунок 5), основанный на разложении решения в ряд Неймана. Каждая итерация метода соответствует следующему приближению решения, что позволяет уточнить результат с каждым шагом. Для вычисления интегралов в итерационном методе трассировки лучей используются квадратурные формулы, заменяющие интегралы суммами. Это позволяет численно оценить освещенность в каждой точке сцены.

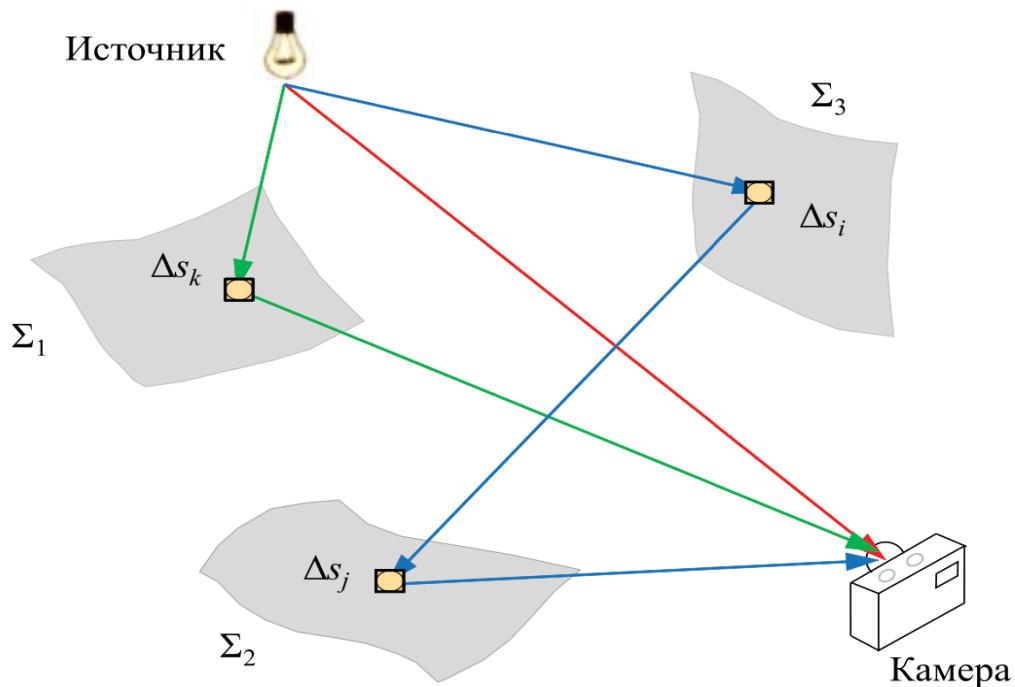


Рисунок 5 - Геометрическая интерпретация решения УГО методом итераций.

При трассировке лучей могут использоваться два основных подхода:

- Метод прямой трассировки лучей, который заключается в запуске лучей из источников света и отслеживании их пути до камеры или наблюдателя.

- Метод обратной трассировки лучей заключается в запуске лучей из камеры или наблюдателя и отслеживании их пути через сцену. Это позволяет определить, какие объекты в сцене освещены, и каким образом свет распространяется через сцену. Этот метод обеспечивает более эффективное использование вычислительных ресурсов, поскольку рассматриваются только лучи, которые попадают в область наблюдения. Он используется чаще всего.

На рисунке (рисунок 6) демонстрируются методы прямой и обратной трассировки.

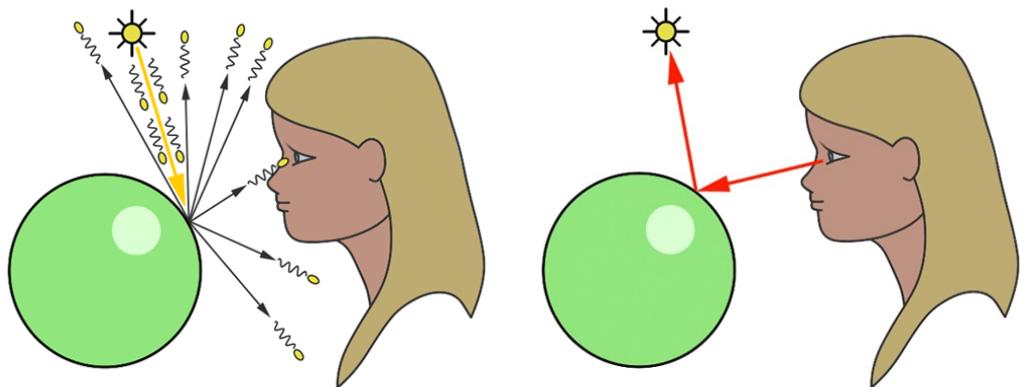


Рисунок 6 – Направление луча при прямой трассировке слева и при обратной справа.

Процесс построения траектории луча при обратной трассировке начинается с определения направления луча от каждого пикселя на ПЗС-матрице камеры к оптическому центру сцены. Затем луч ищет точку пересечения с поверхностью объекта, которая описывается математическими уравнениями. В зависимости от формы поверхности (плоской, многоугольной, сферической) условия пересечения будут различаться. Обнаружив поверхность, луч принимает во внимание её особенности для правильного определения точки пересечения и вычисления освещенности. На этом моменте происходят такие явления как отражение и преломление луча.

Когда луч света пересекает границу между двумя средами, он может изменить свое направление.

Преломление происходит, когда луч света переходит из одной среды в другую, имеющую разные оптические свойства, такие как показатели преломления. Это изменение направления луча объясняется законом Снеллиуса [7], который устанавливает зависимость между углами падения и преломления луча, а также показателями преломления сред. Чтобы понять, куда именно направится луч после преломления, используется векторный подход. Это позволяет выразить новое направление луча как комбинацию векторов, описывающих исходный луч и нормаль к поверхности, через которую происходит преломление. Такой метод помогает предсказать, как изменится направление луча после перехода через границу между средами.

Отражение происходит, когда луч света отражается от поверхности, не проникая в среду. Закон отражения гласит, что угол падения луча равен углу отражения. Это явление также можно объяснить с использованием векторного подхода, который позволяет определить новое направление луча после отражения от поверхности. На рисунке (рисунок 7) продемонстрировано изменение траектории луча при отражении и преломлении.



Рисунок 7 – Отражение и преломление луча на границе двух сред.

После взаимодействия с поверхностью из одного луча может образоваться целый веер лучей. Этот процесс можно представить как создание вторичных

источников света на каждой поверхности. При этом, с увеличением числа рассеяний, количество вычислений возрастает геометрически. В таких условиях даже мощный компьютер может не справиться. Однако многие траектории света начинают становиться похожими друг на друга, что позволяет рассчитывать только наиболее типичные из них. Это приводит к идеи стохастической трассировки лучей, где используется статистика траекторий для усреднения результатов. Для эффективного вычисления таких интегралов на практике применяется метод Монте-Карло, который позволяет получить приближенное значение интеграла, не зависимо от сложности функции подынтегрального выражения.

1.4 Метод Монте-Карло

Метод Монте-Карло используется для численного решения различных задач, основанных на случайности или вероятности. Он применяется в различных областях, включая физику, финансы, биологию, инженерные науки и многие другие. Принцип работы метода Монте-Карло заключается в генерации случайных чисел и использовании их для приближенного численного решения задачи. Например, для вычисления интегралов, определения вероятностей, моделирования физических систем и т. д. Важно отметить, что точность метода Монте-Карло зависит от количества случайных чисел, сгенерированных в процессе, поэтому чем больше чисел используется, тем более точные результаты можно получить. Этот метод позволяет оценить характеристики системы, которые могут быть сложны для аналитического решения или требуют слишком большого объема вычислений.

Термин «метод Монте-Карло» появился в 1949 году в статье Николаса Метрополиса и Стэнислава Улама в журнале *Journal of the American Statistical Association* [8]. Название метода происходит от столицы княжества Монако, известного своими казино, где ruletka является одним из популярных генераторов случайных чисел. Станислав Улам упоминает, что название предложил Николасу Метрополису в честь его дяди, азартного игрока. Идеи, лежащие в основе метода, развивались в 1930-х годах учеными, такими как Энрико Ферми и Джон фон Нейман, а затем были доработаны Станиславом Уламом в Лос-Аламосе в 1940-х годах. Они поняли, что стохастический подход может быть применен для аппроксимации многомерных интегралов в уравнениях переноса, возникающих при рассмотрении движения нейтронов в изотропной среде. Андрей Колмогоров также внес вклад в развитие теории, доказав связь цепей Маркова с интегро-дифференциальными уравнениями.

Понятие вероятности имеет свои корни в азартных играх. Первые попытки решения комбинаторных задач датируются 10-м веком, когда

епископ Виболд из Камбре решил задачу о возможных вариантах броска трех костей, насчитав 56 вариантов. Однако это число не отражало равнозначность различных исходов. В 13 веке Ришар де Форниваль внес поправки в эти рассуждения, учитывая, например, что одинаковое количество очков на трех kostях можно получить разными способами. Это позволило установить, что общее число равно вероятным вариантам составляет 216. Вероятность события определяется как отношение площади благоприятных исходов к общей площади возможных исходов.

На практике, в контексте случайных явлений, приходится часто сталкиваться со случайными величинами. Согласно теории вероятностей, случайная величина может быть непрерывной или дискретной.

— Непрерывная случайная величина описывает явления, которые могут принимать бесконечное количество значений в определенном диапазоне. Например, время, потраченное на прохождение света через определенную среду, может быть представлено как непрерывная случайная величина.

— Дискретная случайная величина, напротив, принимает только конечное или счетное количество значений. Например, результат броска кубика или количество бракованных деталей на конвейере.

Метод Монте-Карло основывается на трех основных положениях теории вероятности и математической статистики:

1. Неравенство Чебышева и закон больших чисел.

Данное положение позволяет численно вычислить математическое ожидание как среднее арифметическое выборки. Согласно неравенству Чебышева, среднее значение случайной величины из выборки будет ближе к математическому ожиданию, если увеличить количество экспериментов.

2. Центральная предельная теорема Муавра.

Она позволяет оценить распределение случайной величины при большом количестве независимых экспериментов. Центральная предельная теорема Муавра-Лапласа говорит о сходимости

последовательности случайных величин к нормальному распределению.

3. Математическая статистика.

Все рассуждения о теории вероятностей относятся к бесконечным пространствам, но на практике имеют дело с конечными выборками случайных величин. Здесь важна эффективность оценок, которые учитывают дисперсию и несмешённость.

При анализе конечных выборок для оценки математического ожидания и дисперсии используется среднее арифметическое. Это подтверждается, даже несмотря на некоторые математические выкладки, которые подчеркивают, что замена n элементов в выборке на $n-1$ не вносит значительных изменений в расчеты, но может быть важным при небольшом количестве данных.

Метод Монте-Карло для расчета интегралов основан на использовании случайных чисел и математических ожиданиях. Принцип работы метода заключается в генерации случайных точек в области интегрирования и вычислении значений функции в этих точках. Затем среднее значение функции считается как среднее арифметическое значений функции в сгенерированных точках, умноженное на площадь области интегрирования. Для реализации метода Монте-Карло требуется генерировать случайные точки согласно определенному распределению вероятности в области интегрирования. Затем значения функции оцениваются в этих точках, и их среднее значение вычисляется. Основное преимущество метода Монте-Карло заключается в том, что он может использоваться для вычисления интегралов в многомерных пространствах и для функций, которые сложно аналитически интегрировать.

Метод Монте-Карло применяется в методе однородного марковского процесса блужданий лучей для моделирования распространения света в сцене. Он основан на случайных блужданиях, где каждый луч света представлен последовательностью случайных шагов. Вероятность перехода из одной точки в другую определяется функцией, что позволяет

приближенно вычислять интегралы, описывающие распределение яркости света в сцене. Каждый шаг в блуждании луча моделируется выбором начальной точки, определением направления и распределением интенсивности света. Коэффициент рассеяния света может быть учтен с использованием статистических весов, что позволяет учитывать различные оптические эффекты, такие как отражение, преломление и поглощение света на поверхности.

1.5 Аналитические решения уравнения глобального освещения

Аналитические решения УГО позволяют лучше понять физические процессы, связанные с освещением, а также используются для тестирования алгоритмов и программ решения уравнения. На сегодняшний день найдены только два аналитических решения уравнения радиосити:

1. Точечный изотропный источник внутри фотометрической сферы [9]: это решение описывает освещение, создаваемое точечным источником света, находящимся внутри фотометрической сферы. Фотометрическая сфера является оптическим компонентом, представляющим собой полую сферическую полость с диффузным белым отражающим покрытием. Решение позволяет определить освещенность поверхности сферы в результате многократных отражений света от внутренней поверхности сферы.

2. Точечный изотропный источник между двумя параллельными однородными диффузными поверхностями: это решение позволяет аналитически выразить освещенность на обеих поверхностях в зависимости от расстояния до источника и характеристик поверхностей.

Истоком второго варианта является Задача В.В. Соболева [10], связанная с использованием белого камуфляжа для светомаскировки войск зимой, возникшая в годы ВОВ. Соболев разработал аналитический подход к решению этой задачи, связанной с использованием фотоосветительных авиабомб (ФОТАБ) для освещения местности при ночном воздушном фотографировании. ФОТАБ представляет собой авиационную бомбу, создающую мощную кратковременную световую вспышку. Она освещает местность для фотографирования с самолета на высотах до 10 000 метров при скоростях полета до 1000 км/ч. Основной заряд ФОТАБ состоит из порошкообразного алюминиево-магниевого сплава, который воспламеняется и сгорает с использованием кислорода воздуха, создавая вспышку с силой света более 2 миллиардов кандел. В задаче рассматривается точечный

изотропный источник света между двумя параллельными диффузными поверхностями, что имитирует использование ФОТАБ для освещения местности сильно облачной зимой.

В работе [11] описывается процесс получения аналитического решения. Процесс состоит в том, что уравнение для освещенности каждой из плоскостей, между которыми находится источник, составляет систему уравнений, которая анализируется с помощью преобразования Фурье. Затем уравнения принимают вид функций, выраженных через преобразование Ганкеля. Решение системы уравнений и последующее обратное преобразование Фурье позволяют получить распределение облученности по плоскости, учитывая как прямую освещенность от источника, так и многоократные переотражения света между поверхностями.

Конечное уравнение можно видеть на рисунке (рисунок 8):

$$E_1(r) = \frac{h_1}{(h_1^2 + r^2)^{3/2}} + \rho_2 \int_0^{\infty} \frac{e^{-h_2 k} + \rho_1 e^{-h_1 k}}{1 - \rho_1 \rho_2 k^2 K_1^2(k)} K_1(k) J_0(kr) k^2 dk$$

Рисунок 8 – точное аналитическое решение уравнения излучательности.

Где:

$E_i(r)$ – освещенность i -ой плоскости ($i=1,2$).

ρ_i – Коэффициент i -ой плоскости.

h_i – Расстояние от i -ой плоскости до источника.

$h_1+h_2=1$.

$K_1(k)$ – Модифицированная функция Бесселя чисто мнимого аргумента или функция Мак-Дональда первого порядка.

J_0 – Функция Бесселя нулевого порядка.

1.6 Локальные оценки

Метод локальных оценок основан на построении марковской цепи, которая описывает перемещение лучей света по сцене. На каждой итерации вычисляется яркость в интересующей нас точке, учитывая направление освещения и характеристики поверхности. Данный метод позволяет эффективно оценивать яркость в различных точках сцены, включая многоократные отражения света. Этот подход обеспечивает математически строгое решение для УГО что открывает новые возможности для создания программных инструментов, позволяющих моделировать световое поле в реалистичных трехмерных сценах.

УГО (Рисунок 4) не является удобным для статистического моделирования, так как искомая функция под интегралом стоит в точке r' , а определяется в точке r . При этом его переменные r' и \hat{l}' связаны соотношением. При интегрировании по пространству и заменив отражения на диффузные, получится уже упомянутое выше уравнение Излучательности (Рисунок 3). В результате оценка I_ϕ для $L(r, \hat{l})$ принимает вид, представленный ниже (рисунок 9). Это выражение получило название локальная оценка метода Монте-Карло [12].

$$I_\phi = M \sum_{n=0}^{\infty} Q_n k(\mathbf{r}, \mathbf{r}')$$

Рисунок 9 – Локальная оценка.

где $k(\mathbf{r}, \mathbf{r}')$ – Ядро УГО.

Q_n – Вес марковской цепи.

M – Математическое ожидание различных случайных траекторий лучей.

Марковская цепь моделирует последовательный ряд случайных лучей, блуждающих по сцене. Вес первого луча Q_1 определяется начальной яркостью источника $L_0(r, \hat{l})$. При этом $k(r,r')$ определяет вероятность перехода из точки марковской цепи в заданную точку r . [12]. Со всех узлов траектории, где произошло ее пересечение с поверхностью сцены, рассчитывается вклад в освещение сцены в точке r на основе ядра уравнения $k(r,r')$ в выражении (Рисунок 9). При этом все распределения вероятностей должны отвечать условиям нормировки. Величины $L_0(r, \hat{l})$ и $k(r,r')$ будут определять статистические веса марковской цепи Q_n . В случае диффузной модели каждый последующий статистический вес будет умножаться на коэффициент отражения.

Локальная оценка позволяет проводить оценку освещенности в заданной точке сцены. Таким образом, для расчета освещенности в некоторой заданной точке r необходимо построить марковскую цепь и на каждом акте отражения вычислять ядро $k(r,r')$ для всех заданных точек. Математическое ожидание полученной величины и будет равно освещенности.

Марковская цепь – это последовательность случайных событий с конечным или счётным числом исходов, характеризующаяся тем, что при фиксированном настоящем будущее не зависито от прошлого. В данном случае цепь строится от источника света, при этом начальный вес кладется равным силе света источника света с учетом нормировки на поток. После чего находится точка пересечения луча с элементом сцены и вес умножается на коэффициент отражения. Далее вычисляется ядро УГО для каждой из исследуемых точек, умноженное на вес луча, и суммируется с предыдущими значениями. После чего производится розыгрыш луча в соответствии с диффузным законом отражения и ищется следующее пересечение. Процесс повторяется до тех пор, пока луч либо не покинет пределы сцены, либо его вес не станет ниже заданного порога. При розыгрыше большого количества

лучей и усреднив их, получаются значения освещенности в исследуемых точках. Локальная оценка позволяет по одному лучу вычислять значение освещенности сразу в нескольких точках. Это принципиальное отличие от прямого моделирования и трассировки лучей [13].

Использование локальных оценок демонстрирует значительное ускорение вычислений по сравнению с прямым моделированием. В той же работе приводится сравнение локальных оценок и прямого моделирования при решении задачи Соболева, где метод позволяет получить результат в 80-90 раз быстрее.

Выражение на (рисунок 10) называется Двойной локальной оценкой и представляет собой метод, который расширяет возможности локальной оценки, позволяя непосредственно моделировать яркость в заданной точке сцены по заданному направлению. Этот метод основан на построении марковской цепи, описывающей перемещение лучей света, и представлении решения уравнения глобального освещения в виде ряда Неймана.

$$I_\varphi = M \sum_{n=0}^{\infty} Q_n k(\mathbf{r}, \hat{\mathbf{l}}, \mathbf{r}', \hat{\mathbf{l}}') ,$$

$$k(\mathbf{r}, \hat{\mathbf{l}}, \mathbf{r}', \hat{\mathbf{l}}') = \frac{1}{\pi^2} \sigma(\mathbf{r}'; \hat{\mathbf{l}}'', \hat{\mathbf{l}}') \sigma(\mathbf{r}; \hat{\mathbf{l}}', \hat{\mathbf{l}}) F(\mathbf{r}', \mathbf{r}'').$$

Рисунок 10 – Двойная локальная оценка.

В данном выражении угловая особенность пропадает в результате интегрирования, а независимые переменные \mathbf{r} , $\hat{\mathbf{l}}$, \mathbf{r}' , $\hat{\mathbf{l}}'$, \mathbf{r}'' , $\hat{\mathbf{l}}''$ соответствуют геометрии распространения луча [12], что и приводит к возможности моделировать УГО и вычислению яркости в заданной точке в заданном направлении от кратностей выше первой.

На сегодняшний день двойная локальная оценка является единственным методом, позволяющим получать значения яркости в любой точке пространства трехмерной сцены [13].

Однако практическая реализация метода локальных оценок сталкивается с проблемами, связанными с сингулярностями в ядре УГО. Эти сингулярности возникают, когда точка столкновения луча с поверхностью находится очень близко к точке оценки яркости, что приводит к "всплескам" искажений в оценке. Такие ситуации могут возникать вблизи ребер соседних граней при использовании сеточного представления объектов сцены. Это означает, что прямая реализация метода локальных оценок в виде программы для моделирования переноса яркости в сложных сценах становится менее эффективной из-за вероятности увеличения "всплесков" искажений с увеличением сложности сцены.

В работе [14] авторами приводится метод усреднения искомой величины в некоторой окрестности исследуемой точки. Например, для точечного изотропного источника света в диффузном приближении, локальная оценка освещенности в точке определяется интегралом по объему окрестности рассматриваемой точки. Поскольку количество лучей и столкновений конечное, этот интеграл можно переписать, а его значение оценить методом статистических испытаний.

Предложенный метод устранения сингулярностей открывает новые возможности для развития локальных оценок метода Монте-Карло в области моделирования переноса яркости и его реализация в ПО для моделирования освещения способна значительно повысить качество визуализации.

На основе всего вышесказанного можно сделать вывод, что метод Локальных оценок становится новым вектором в развитии светотехнических расчетов. Двойная локальная оценка впервые в светотехнической практике позволяет вычислять непосредственно яркость в заданной точке по заданному направлению. Это открывает принципиально новые горизонты светотехнического проектирования и возможность пересмотра

существующей нормативной базы, основанной на существующих методах моделирования. Двойная локальная оценка может стать новым численным методом, используемым при реалистичной визуализации трехмерных сцен.

1.7 Выводы раздела

Развитие компьютерной графики в области моделирования освещения прошло долгий путь. В данной работе были упомянуты исторический события о простых методах рисования линий и поверхностей, а также были рассмотрены сложные алгоритмы, учитывающие физические свойства света, что привело к значительному прогрессу в создании реалистичных изображений.

Для решения уравнения глобального освещения, которое и является фундаментом в расчёте яркости в сцене, были рассмотрены различные методы. Метод трассировки лучей, который моделирует путь световых лучей от источников света до камеры, позволяя создавать изображения с высокой степенью реализма. Метод излучательности, основанный на представлении сцены в виде набора конечных пучков лучей, также позволяет моделировать освещенность поверхностей, но не учитывает отражения.

Последними из рассмотренных, но не по важности, были методы локальных оценок, в частности двойная локальная оценка, которые представляют собой эффективные направления в моделировании светового поля. Благодаря им открываются новые возможности для светотехнического проектирования и визуализации, позволяя непосредственно вычислять яркость в заданной точке сцены по заданному направлению, что имеет коммерческое применение, но до сих пор никем не было использовано в полноценном программном обеспечении.

В этой части была рассмотрена математическая сторона визуализации, но помимо этого, требуется упомянуть про инструменты и процесс практической реализации вышеупомянутых методов и алгоритмов, где и проявляются сложности вычислений и оптимизации. Этому посвящен следующий раздел данной работы.

2. Инструменты для моделирования света и визуализации

2.1 Библиотека Embree

Трассировка лучей представляет собой мощный метод создания реалистичных изображений в компьютерной графике, и существует множество библиотек и инструментов для ее реализации. Одной из таких библиотек является Intel Embree. Embree — это высокопроизводительная библиотека трассировки лучей, разработанная Intel. Она предоставляет API для реализации трассировки лучей на процессорах Intel с использованием различных методов оптимизации, что делает ее идеальным выбором для широкого спектра приложений, включая компьютерные игры, виртуальную реальность, анимацию и визуализацию [15]. Одной из главных особенностей библиотеки Embree является ее высокая производительность. Она оптимизирована для работы на многоядерных процессорах Intel и использует эффективные алгоритмы и структуры данных для быстрой обработки лучей. Это позволяет достигать высокой скорости рендеринга даже при работе с большими и сложными сценами. Кроме того, Embree предоставляет различные возможности для оптимизации трассировки лучей, а также поддерживает различные типы материалов и световых источников, что делает ее универсальным инструментом для создания разнообразных визуальных эффектов. Библиотека интегрируется с различными программными пакетами и фреймворками для компьютерной графики, такими как Blender, Maya, Unity и Unreal Engine, что обеспечивает ее широкое использование в индустрии. Для обеспечения эффективности и быстродействия в Embree есть иерархические структуры BVH (Bounding Volume Hierarchy), что позволяет эффективно управлять сложностью сцены и ускорить процесс поиска пересечений лучей с объектами.

BVH — это древовидная структура данных, используемая для ускорения поиска пересечений лучей с объектами в сцене. Она разбивает пространство сцены на иерархию ограничивающих объемов, таких как сферы или параллелепипеды, что позволяет быстро отсекать части сцены, которые не пересекаются с лучами.

При построении BVH сначала создается корневой ограничивающий объем, который охватывает всю сцену. Затем этот объем рекурсивно разделяется на два подобъема, которые в свою очередь также разделяются, пока не будет достигнута определенная глубина разделения или не будет выполнен критерий остановки. В результате каждый лист дерева BVH содержит ссылки на объекты в сцене или на их ограничивающие объемы. Одним из преимуществ BVH является его способность быстро и эффективно отсекать части сцены, которые не пересекаются с лучами, что существенно уменьшает количество проверок пересечений и ускоряет процесс трассировки лучей. Кроме того, BVH обладает хорошей масштабируемостью и может быть эффективно использован для сцен с различными типами объектов и геометрией.

На рисунке (рисунок 11) можно наглядно про наблюдать рекурсивное разделение объема, описанного выше.

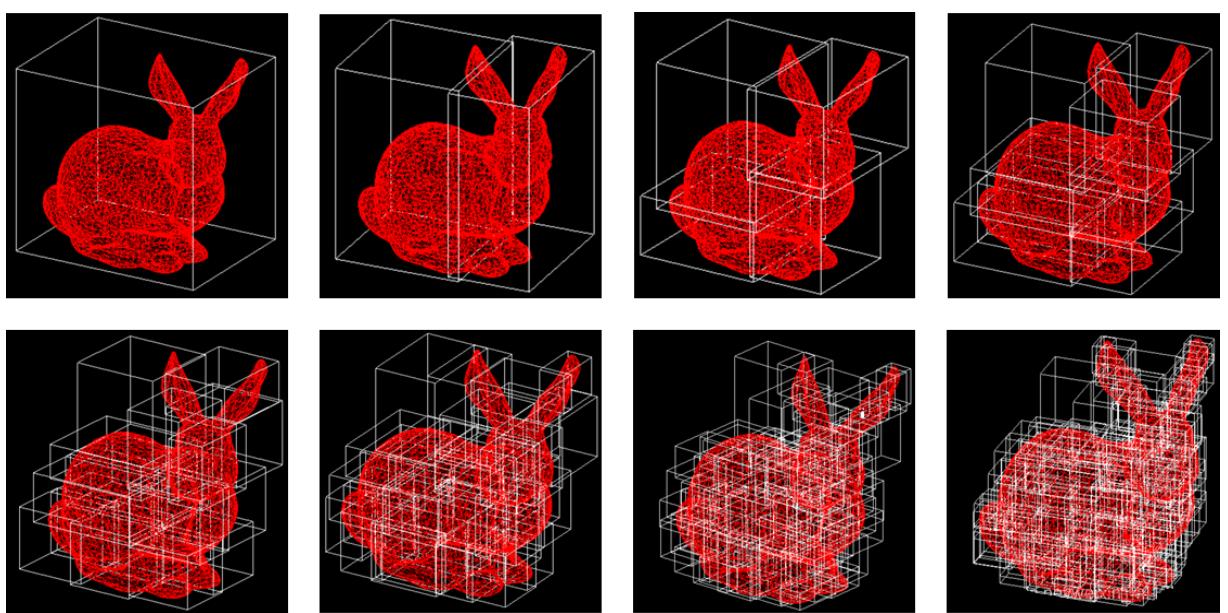


Рисунок 11 – Демонстрация рекурсивного деления в BVH.

В исследовании [16] автор экспериментирует с различными методами построения BVH и проводит оценку производительности нескольких методов разделения, а также описывает способы визуализации полученного результата. Один из рассмотренных методов разделения, называемый методом медианного разделения, привел к созданию деревьев с более худшим качеством результатов по сравнению с другими методами при использовании высокополигональных сеток. Использование же принципа площади поверхности (Surface Area Heuristic, SAH) привело к созданию деревьев с более лучшим качеством результатов, но заняло больше времени для построения BVH.

На основе вышеописанного можно сделать вывод, что библиотека Embree, разработанная Intel, представляет собой высокопроизводительную библиотеку для трассировки лучей, которая включает в себя оптимизированные реализации алгоритмов трассировки лучей и поддержку BVH. Она предоставляет API для быстрого и эффективного рендеринга сцен с использованием трассировки лучей. Одной из ключевых особенностей Embree является ее оптимизация под архитектуру современных многоядерных процессоров, что позволяет достичь высокой производительности при рендеринге сложных сцен. Это делает Embree идеальным выбором для широкого спектра приложений, включая компьютерные игры, виртуальную реальность, анимацию и визуализацию. Библиотека интегрируется с различными программными пакетами и фреймворками для компьютерной графики, что обеспечивает ее широкое использование в индустрии.

2.2 Варианты оптимизации процесса визуализации

Процесс рендеринга включает в себя несколько этапов, каждый из которых играет свою роль в формировании конечного изображения:

1. Подготовка сцены. В этом этапе определяются объекты, свет и камера сцены. Объекты могут быть созданы вручную или импортированы из других программных средств или файлов. Свет исходит от различных источников в сцене, таких как источники света, отраженный свет и глобальное освещение. Камера устанавливается для определения точки обзора сцены.

2. Трассировка лучей. Это ключевой этап процесса рендеринга, где создается изображение путем трассировки лучей от камеры через каждый пиксель изображения в сцену. Каждый луч может пересекаться с объектами в сцене, отражаться, преломляться или поглощаться, в зависимости от свойств поверхностей объектов. Результатом этого этапа является получение освещенности для каждого пикселя изображения.

3. Теневая обработка. На этом этапе определяется, какие части сцены находятся в тени, и как это влияет на освещение объектов. Это включает в себя проверку видимости между источниками света и объектами сцены.

4. Вычисление материалов. Каждый объект в сцене обычно имеет свои материалы, которые определяют, как они отражают и пропускают свет. На этом этапе происходит вычисление вклада каждого материала в освещение сцены и его взаимодействие с окружающей средой.

5. Постобработка. Наконец, изображение подвергается постобработке, которая может включать в себя применение фильтров, коррекцию цветовой гаммы, наложение эффектов и т. д., чтобы улучшить его визуальное качество и достичь желаемого визуального эффекта.

В процессе рендеринга изображения в компьютерной графике существуют свои проблемы. Основные из них: низкая скорость рендеринга и наличие шума на изображении.

Скорость рендеринга может страдать из-за сложности самой сцены. Когда сцена содержит множество объектов со сложной геометрией или большое количество источников света, трассировка лучей становится более затратной по времени. Каждый луч должен пройти через сцену, взаимодействуя с каждым объектом, что требует значительных вычислительных ресурсов. Долгий процесс рендеринга замедляет разработку и тестирование графических проектов, ограничивает способность разработчика (художника, дизайнера) к визуализации своих идей, тормозит рабочий процесс.

Также при использовании методов трассировки лучей появляется эффект зернистости изображения, свойственный статистической природе метода Монте-Карло что также называется шумом. Шум на изображении может возникать из-за недостаточного количества сэмплов, используемых для оценки освещенности в каждой точке изображения. Низкое качество сэмплирования приводит к зернистости или размытости изображения из-за неправильной оценки сложных освещенных эффектов. Эти проблемы могут усугубляться использованием сложных материалов или неэффективных алгоритмов рендеринга.

Существует множество методов и техник для смягчения этих проблем.

Для решения проблемы скорости и ускорения процесса используются:

1. Параллельное вычисление на многопроцессорных системах. Использование многопроцессорных систем позволяет распараллеливать вычисления и ускорять обработку данных.
2. Оптимизация алгоритмов трассировки лучей. Путем оптимизации алгоритмов трассировки лучей можно значительно снизить время рендеринга. Это включает в себя, например методы ускорения пересечения лучей с объектами и эффективное управление памятью.

3. Использование приближенных методов рендеринга. Приближенные методы рендеринга, такие как алгоритмы глубокого обучения или методы предварительного вычисления освещения, могут значительно сократить время рендеринга без существенной потери качества.
4. Применение технологий аппаратного ускорения, таких как графические процессоры (GPU). Использование их может значительно ускорить процесс, особенно если это такие специализированные аппаратные средства, как, например, технология NVIDIA RTX.

Для устранения шума используются:

1. Метод с увеличением числа сэмплов лучей. Он является одним из распространенных способов и включает в себя генерацию большего количества лучей для каждого пикселя изображения, что позволяет более точно оценить освещение и уменьшить статистический шум, характерный для методов Монте-Карло.
2. Использование приближенных методов рендеринга. Это также является эффективным способом уменьшения времени, необходимого для генерации изображения, за счет сокращения вычислительной сложности алгоритма. Эти методы являются включением различных оптимизаций, таких как адаптивное сэмплирование или аппроксимация сложных явлений освещения.
3. Денойзеры. Для устранения шума на изображениях применяются различные алгоритмы денойзинга, которые позволяют улучшить качество изображений без необходимости увеличивать количество сэмплов. Денойзеры работают путем анализа и фильтрации шума, сохраняя детали изображения.

Денойзеров существует несколько типов. В статье [17] автором рассматриваются различные вариации. Можно выделить следующие типы:

- 1) SVGF. Spatio-Temporal Variance Guided Filter (SVGF) — это денойзер, который использует пространственно-временную репроекцию вместе с буферами характеристик, такими как нормали, глубина и расчет дисперсии, чтобы управлять двусторонним фильтром для размытия областей с высокой дисперсией. Разновидность технологии SVGF используется в компьютерной игре Minecraft RTX с добавлением кэширования облучения, использованием длины лучей для лучшего отражения и разделением рендеринга для пропускающих поверхностей, таких как вода. SVGF, хотя и очень эффективен, имеет временную задержку, которая заметна в игре.
- 2) A-SVGF. Адаптивная пространственно-временная фильтрация, управляемая дисперсией (A-SVGF), улучшает SVGF за счет адаптивного повторного использования предыдущих образцов, которые были пространственно репроектированы в соответствии с временными эвристиками, такими как изменение дисперсии, угол обзора и т. д., закодированными в буфере моментов, и фильтрации с помощью быстрого двустороннего фильтра. Таким образом, вместо накопления образцов, основанных на длительности истории, буфер моментов выступает в качестве альтернативной эвристики, которая использует изменение дисперсии для определения пропорции между старыми и новыми образцами, что приводит к уменьшению призрачности. Если в SVGF буфер момента использовался только для управления размытием, то в A-SVGF он используется как на этапе фильтрации, так и на этапе накопления.
- 3) ReSTIR. Spatiotemporal Importance Resampling for Many-Light Ray Tracing (ReSTIR) пытается перенести шаг пространственно-временной

репроекции в денойзерах реального времени на более ранние этапы рендеринга, используя статистику вероятностей соседних выборок.

4) DLSS. Deep Learning Super Sampling (DLSS) - это техника повышения разрешения, которая использует небольшой цветовой буфер и карту направлений для увеличения разрешения выходного изображения в 2-4 раза. Это эксклюзивная технология для предварительно одобренных разработчиков компанией NVIDIA, поэтому в настоящее время нет возможности использовать ее открыто, однако существует альтернатива под названием DirectML's SuperResolution Sample.

5) Intel OIDN. Intel Open Image Denoise (OIDN) - это автокодировщик с машинным обучением, который принимает альбето, нормали первого скачка и входное зашумленное изображение и выводит отфильтрованное изображение.

6) NVIDIA Optix 7 Denoising Autoencoder принимает те же исходные данные, что и OIDN, дополнительные альбето, нормали и входное зашумленное изображение, и выдает отфильтрованное изображение гораздо быстрее, чем решение Intel, но с потерей качества.

Из информации выше можно сделать вывод, что использование денойзеров на основе нейронных сетей при рендеринге изображений имеет ряд значительных преимуществ. Они способны эффективно уменьшать шум на изображениях, что позволяет получить более чистые и качественные результаты без необходимости увеличивать количество сэмплов, что сокращает время, необходимое для генерации изображения. Это особенно важно в случае применения метода Монте-Карло, где увеличение числа сэмплов может значительно увеличить вычислительную нагрузку. Также использование денойзеров на нейронных сетях может быть более

эффективным и гибким, чем традиционные методы денойзинга, так как нейронные сети способны автоматически адаптироваться к различным типам шумов и изображений, что позволяет достигать качественных результатов на разных сценах.

2.3 Использование нейросетей для денойзинга

Существует несколько разных типов нейросетей, которые можно классифицировать по определенным признакам. Один из самых распространенных способов классификации – это по типу связей между нейронами. Согласно этому подходу, можно выделить следующие типы нейросетей:

1. Многослойные перцептроны (Multilayer Perceptrons, MLP) – это наиболее простой тип нейросети, который состоит из одного или нескольких слоев нейронов. Каждый нейрон обрабатывает входящие данные и передает выходные данные следующему слою нейронов. MLP может использоваться для решения широкого спектра задач – от прогнозирования временных рядов до распознавания образов на изображениях.

2. Сверточные нейронные сети (Convolutional Neural Networks, CNN) – это тип нейросетей, который часто используется в задачах обработки изображений и видео. Они основаны на принципе свертки, который позволяет выявлять визуальные признаки изображения. Каждый нейрон в CNN обрабатывает только небольшую область изображения, что позволяет учитывать локальные свойства каждого фрагмента.

3. Рекуррентные нейронные сети (Recurrent Neural Networks, RNN) – это тип нейросетей, который хорошо подходит для работы с последовательными данными, например, с текстом или звуком. В отличие от MLP и CNN, RNN сохраняет информацию о предыдущих знаках в последовательности, что позволяет учитывать контекст и последовательность входных данных.

4. Глубокие нейронные сети (Deep Neural Networks, DNN) – это тип нейросетей, который состоит из множества слоев, каждый из которых

содержит множество нейронов. Глубокие нейронные сети широко используются для работы с большими объемами данных, например, в задачах распознавания речи или обработки естественного языка.

Из вышеописанного можно сделать вывод, что наиболее подходящими вариантами для работы с изображениями являются CNN и MLP.

В статье [18] автором рассматривается сравнение между традиционными многослойными перцептронами (MLP) и сверточными нейронными сетями (CNN), выявляются две основные проблемы с MLP и наглядно демонстрируется, почему CNN являются более предпочтительным выбором для обработки изображений.

1. Проблема с потерей пространственной ориентации изображения (рисунок 12): MLP использует одномерное представление изображения, что затрудняет распознавание изображений и выявление аномалий в пространственной ориентации. Например, на изображениях, представленных в одном измерении, изменение относительного положения ключевых элементов (например, нос, глаза, уши) затрудняет идентификацию объекта. В отличие от этого, CNN использует двумерное представление изображения, что позволяет сохранить пространственную ориентацию и обеспечить более эффективное распознавание объектов.

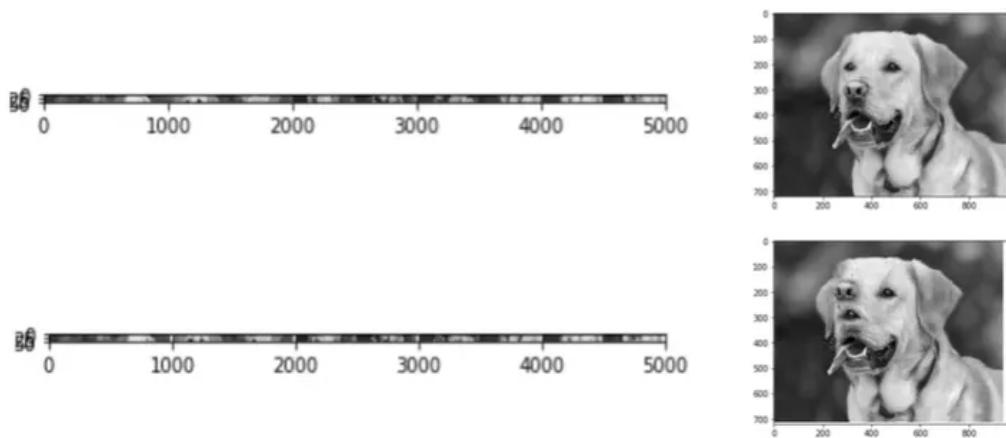


Рисунок 12 - Потеря пространственной ориентации изображения у MLP
снизу, и отсутствие проблемы у CNN сверху.

2. Проблема с исследованием параметров в нейронной сети: MLP требует огромного количества параметров для эффективной работы с изображениями больших размеров, что приводит к высокой вычислительной сложности и нежелательному использованию ресурсов. CNN решает эту проблему, используя концепцию свертки, которая позволяет снизить количество обучаемых параметров, сохраняя при этом высокую производительность в распознавании и классификации изображений.

В связи с этим CNN представляется как более эффективное решение для обработки изображений, поскольку она сохраняет пространственную ориентацию и эффективно управляет количеством обучаемых параметров, что делает ее более подходящей для широкого спектра задач компьютерного зрения и обработки изображений.

Принцип работы нейросети непосредственно в задаче денойзинга будет состоять из следующих шагов:

1. На первом этапе требуется определиться с подходящей архитектурой нейронной сети, которая должна быть способной извлекать признаки

изображений и обрабатывать их для эффективного удаления шума. На основе вышеописанного исследования это будет CNN. Она хорошо подходит для работы с изображениями благодаря своей способности к автоматическому изучению пространственной структуры данных.

2. Далее нейронная сеть обучается на наборе данных, содержащем пары зашумленных и чистых изображений. Во время обучения сеть стремится минимизировать разницу между входными зашумленными изображениями и соответствующими им чистыми версиями. Для этого она настраивает веса и параметры своих слоев, используя методы оптимизации, такие как стохастический градиентный спуск.

3. Для обучения используется набор данных, содержащий пары зашумленных и чистых изображений. Эти изображения могут быть собраны из различных источников или созданы программно. Важно, чтобы чистые изображения соответствовали зашумленным, чтобы нейронная сеть могла учиться восстанавливать чистые версии изображений из зашумленных входных данных.

4. Для оценки ошибки между предсказанными и истинными чистыми изображениями определяется функция потерь. Обычно для задачи денойзинга используется среднеквадратичная ошибка (MSE), которая измеряет среднеквадратичное отклонение между пиксельными значениями предсказанных и истинных изображений.

5. В процессе обучения нейронная сеть принимает на вход зашумленные изображения и постепенно настраивает свои веса и параметры, минимизируя функцию потерь. Это достигается с помощью алгоритма обратного распространения ошибки, который распространяет градиент ошибки через сеть и корректирует веса нейронов в соответствии с этим градиентом.

6. После завершения обучения результаты работы нейронной сети оцениваются на отдельном наборе тестовых данных, которые не участвовали

в обучении. Это позволяет оценить качество работы сети на новых, реальных данных и определить ее способность к эффективному удалению шума.

7. В процессе обучения сверточные слои нейронной сети извлекают различные уровни абстракции изображения, начиная с низкоуровневых признаков, таких как границы и текстуры, и заканчивая более высокоуровневыми признаками, связанными с содержанием и структурой изображения. В сверточных нейронных сетях (CNN), широко применяемых в компьютерном зрении, извлечение признаков осуществляется через применение сверточных слоев.

8. Сверточные слои состоят из фильтров (или ядер), которые скользят по входному изображению, производя операцию свертки. Эти фильтры обучаются находить различные низкоуровневые признаки, такие как границы, текстуры и цветовые характеристики. При этом каждый фильтр выделяет определенные характеристики изображения путем усиления значимых пикселей и подавления несущественных.

9. Следующий этап в CNN - операция подвыборки (pooling), которая позволяет уменьшить размерность изображения и создать инвариантность к незначительным пространственным сдвигам искомых признаков. Это осуществляется путем применения операции максимальной или средней подвыборки в определенных областях изображения.

10. На более высоких уровнях сети, более глубокие сверточные слои начинают извлекать более сложные и абстрактные признаки, такие как формы, структуры и объекты. На этом этапе, каждый нейрон в сверточном слое представляет собой комбинацию более низкоуровневых признаков, которые формируют более высокоуровневые абстракции.

11. Этапы извлечения признаков в CNN обеспечивают постепенное переход от извлечения простых, низкоуровневых признаков к более сложным

и абстрактным характеристикам изображения. Этот процесс позволяет нейронным сетям эффективно моделировать и анализировать сложные структуры визуальных данных и применять их в том числе для денойзинга.

12. После завершения обучения нейронная сеть, предназначенная для фильтрации шума на изображениях, применяется к новым зашумленным входным данным. Этот процесс начинается с передачи зашумленного изображения в сеть в качестве входных данных. Затем, на основе ранее изученных признаков, полученных в ходе обучения, нейронная сеть прогнозирует чистую версию изображения.

13. В процессе фильтрации шума нейронная сеть применяет различные математические операции к каждому пикселью входного зашумленного изображения. Эти операции включают в себя свертки, которые позволяют сети анализировать окрестности каждого пикселя с помощью фильтров различных размеров.

14. После этого применяются функции активации, которые вводят нелинейность в выходные данные каждого слоя, обеспечивая сети способность обучаться более сложным зависимостям между входными и выходными данными.

15. Агрегация данных осуществляется с использованием методов, таких как пулинг или объединение, которые позволяют уменьшить размерность данных, сохраняя при этом основные характеристики изображения. Это помогает снизить вычислительную сложность и улучшить обобщающую способность сети.

16. Во время фильтрации шума нейронная сеть применяет комплексные алгоритмы, которые учитывают контекст и структуру изображения. Это

означает, что сеть анализирует не только отдельные пиксели, но и их окружение, чтобы учесть пространственные зависимости и текстурные особенности. Такой подход позволяет сети эффективно удалять шум и одновременно сохранять важные детали изображения.

Качество работы денойзера оценивается с использованием различных метрик, таких как PSNR (Peak Signal-to-Noise Ratio) и SSIM (Structural Similarity Index Measure). Эти метрики помогают определить степень сходства между исходным чистым изображением и восстановленным денойзером.

PSNR выражает отношение между максимально возможным значением сигнала и среднеквадратичной ошибкой между исходным и восстановленным изображением, предоставляя количественную оценку качества изображения. Чем выше значение PSNR, тем меньше различий между исходным и восстановленным изображениями.

SSIM, с другой стороны, измеряет структурное сходство между двумя изображениями, учитывая не только яркость, но и текстурные особенности. Он оценивает три аспекта: яркость, контраст и структуру. Высокое значение SSIM указывает на высокое сходство между изображениями, что свидетельствует о сохранении деталей и текстурных характеристик.

Обе метрики помогают определить эффективность денойзера в удалении шума и сохранении деталей изображения. Они используются для сравнения качества работы различных моделей денойзеров, а также для оптимизации параметров обучения сетей с целью достижения наилучших результатов в фильтрации шума на изображениях.

В статье [19] авторами рассматривается исследование влияния характера шума на качество его удаления с использованием нейронных сетей.

Предположение состоит в том, что обучение нейронной сети на отделенном шуме, а не на зашумленном изображении, может привести к более эффективному удалению шума. Это может уменьшить вероятность потери деталей изображения или "размытия" после денойзинга. Сравнивается использование аддитивного гауссовского шума и реальной модели шума, полученной с помощью фотокамеры Canon, которые можно увидеть на рисунке (рисунок 13). Результаты эксперимента показали, что нейронная сеть, обученная на датасете с реальным шумом, достаточно эффективно его удаляет, сохраняя при этом мелкие детали и четкость изображений. Однако сеть, обученная на гауссовском шуме, не смогла также эффективно удалить реальный шум, что подтверждает важность использования реалистичных моделей шума при обучении нейронных сетей для денойзинга. Подход, основанный на отделении шума, может быть более универсальным и гибким, поскольку не привязан к конкретному типу шума или сценарию. Также обучение на отделенном шуме может улучшить устойчивость нейронной сети к различным уровням шума и условиям освещения на изображениях. Это может сделать модель более надежной и применимой в реальных сценариях.

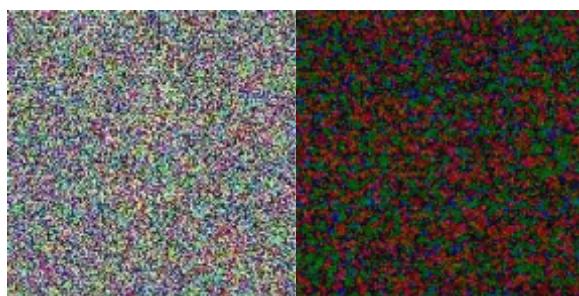


Рисунок 13 – Слева гауссовский шум, а справа шум, полученный реальной фотокамерой.

2.4 Библиотека Intel Open Image Denoise

Стоит рассмотреть подробнее готовое решение, использующее нейросети для удаления шума - библиотеку Intel® Open Image Denoise, которая упоминалась ранее.

Это библиотека с открытым исходным кодом, входящая в состав Intel Rendering Framework, в который также входит библиотека Embree, и выпускаемая под лицензией Apache 2.0. Ее цель - предоставить разработчикам высококачественное решение с открытым исходным кодом, которое значительно сокращает время рендеринга.

Библиотека обладает простым и гибким API на языке C/C++, что гарантирует возможность легкой интеграции в большинство существующих или новых решений для рендеринга.

На текущий момент реализована поддержка широкого спектра CPU и GPU от разных производителей:

- процессоры, совместимые с архитектурой Intel® 64;
- ARM64 (AArch64);
- Графические процессоры NVIDIA;
- Графические процессоры AMD;
- Apple silicon.

Благодаря тому, что Intel Open Image Denoise использует современные наборы инструкций, такие как SSE4, AVX2, AVX-512 и NEON в CPU, Intel® Xe Matrix Extensions (Intel® XMX) в GPU Intel и тензорные ядра в GPU NVIDIA, происходит оптимизация работы при помощи GPU и CPU для достижения более высокой производительности денойзинга. Работа на текущий момент над библиотекой активно ведется и на GitHub разработчиков [20] систематически появляются новые версии, добавляющие поддержку нового оборудования и оптимизирующие текущие наработки.

В основе библиотеки лежит фильтр устранения шума на основе глубокого обучения, обученный работать с широким диапазоном настроек, от одной выборки на пиксель до почти полной конвергенции. Таким образом, он подходит как для предварительного просмотра, так и для рендеринга финального кадра.

Для обучения нейросети команда Intel использовала пары изображений - одно с шумом, другое полностью сконвергированное и без шума. Нейросеть учился обесцвечивать изображения, основываясь на примерах, предоставленных разработчиками. [21]

Библиотека Intel Open Image Denoise поставляется с полностью обученной нейросетью, которую команда разработала для использования с изображениями, полученными с использованием трассировки лучей. Для различных вариантов использования, уровней шума, тематических настроек и освещения нейросеть находит решение без кропотливого ввода дополнительных параметров.

Для эффективного задействования CPU и GPU используется низкоуровневая работа с памятью. К примеру, если приложение хочет поддерживать графические процессоры, передача указателей на память, выделенную с помощью системного аллокатора (например, malloc), приведет к ошибке, поскольку графические процессоры почти во всех случаях не могут получить доступ к такой памяти, в то время как для CPU это будет совершенно стандартная операция.

Данные изображения могут быть переданы в Open Image Denoise либо через указатели на память, выделенную и управляемую пользователем, либо путем создания буферных объектов. Независимо от того, какой метод используется, данные должны быть распределены таким образом, чтобы они были доступны устройству (CPU или GPU). Использование буферов обычно является предпочтительным подходом, поскольку это гарантирует

выполнение требований к распределению данных независимо от типа устройства.

Для обеспечения совместимости с любыми устройствами, включая графические процессоры, в библиотеке создаются объекты-буферы для хранения всех передаваемых данных изображения. Память, выделенная с помощью буферов, по умолчанию будет доступна и CPU или GPU. Данные могут копироваться асинхронно с помощью соответствующих методов, что способно обеспечить более высокую производительность, чем обычная темпсру.

Если в системе есть несколько устройств одного типа, что очень частое явление (например, компьютер с CPU и одним или несколькими GPU), то у библиотеки есть функции, позволяющие учесть это, и гибко распараллелить работу между ними.

Фильтры — это основные объекты Open Image Denoise, которые отвечают за фактическое обесцвечивание. Библиотека поставляется с набором фильтров, оптимизированных для различных типов изображений и случаев использования.

Создание объектов фильтра может быть очень дорогостоящим, поэтому разработчиками настоятельно рекомендуется использовать один и тот же фильтр для денойзинга как можно большего числа изображений, если эти изображения имеют одинаковый размер, формат и характеристики (т. е. могут отличаться только места в памяти и значения пикселей). В противном случае (например, для изображений с разным разрешением), повторное использование одного и того же фильтра не принесет никакой пользы.

После создания фильтра, его необходимо настроить, указав входное и выходное изображения, а также, возможно, задав значения других параметров.

Часто производительность фильтрации важнее, чем максимально возможное качество изображения, поэтому есть возможность переключаться между несколькими режимами качества фильтра.

Фильтры могут обесцвечивать изображения, используя только цветовой («косметический») буфер с шумом. Или, чтобы сохранить как можно больше деталей, он может использовать вспомогательные буфера, такие как «альбето» и «нормали». В качестве примера можно рассмотреть готовый фильтр "RT":

Фильтр RT (Ray Tracing) — это общий фильтр денойзинга, который подходит для очистки от шума изображений, отрисованных с помощью методов трассировки лучей Монте-Карло, таких как односторонняя и двунаправленная трассировка. Он также поддерживает глубину резкости и размытие движения, но не является временно-стабильным. Основан на сверточной нейронной сети (CNN) и поставляется с набором предварительно обученных моделей, которые хорошо работают с широким диапазоном рендеринговых систем, основанных на трассировке лучей, и уровней шума.

Для денойзинга в качестве основного входного изображения принимается изображение с низким динамическим диапазоном (LDR) или высоким динамическим диапазоном (HDR).

Фильтр RT имеет определенные ограничения в отношении поддерживаемых входных изображений. В частности, он не может обесцвечивать изображения, которые не были отрендерены с помощью трассировки лучей. Другое важное ограничение связано с фильтрами сглаживания. Большинство рендеров для минимизации артефактов сглаживания используют высококачественный фильтр реконструкции пикселей, а не тривиальный бокс-фильтр (например, гауссовский, Блэкмана-Харриса) [22]. Фильтр RT поддерживает такие пиксельные фильтры, но только если они реализованы с дискретизацией по важности. Взвешенная

выборка пикселей (иногда называемая *splatting*) вносит корреляцию между соседними пикселями, что приводит к неудаче при обработке (шум не будет отфильтрован), поэтому она не поддерживается.

Использование готовых фильтров не является обязательным. Чтобы оптимизировать фильтр для конкретного рендерера, количества образцов, типа контента, сцены и т. д., можно самостоятельно обучить модель, используя прилагаемый набор инструментов для обучения и предоставленных пользователем наборов данных изображений.

Исходный дистрибутив Intel Open Image Denoise включает инструментарий для обучения нейросети на языке Python (находится в каталоге *training*), который можно использовать для обучения моделей фильтров денойзинга на наборах данных изображений, предоставленных пользователем. Это расширенная возможность библиотеки, использование которой требует некоторых базовых знаний в области машинного обучения и базового знакомства с фреймворками и инструментами глубокого обучения (например, PyTorch или TensorFlow, TensorBoard).

Инструментарий для обучения состоит из следующих скриптов командной строки:

1. *preprocess.py*: предварительно обрабатывает обучающие и проверочные наборы данных.
2. *train.py*: Обучает модель, используя предварительно обработанные наборы данных.
3. *infer.py*: выполняет вывод на наборе данных, используя указанный результат обучения.
4. *export.py*: экспортирует результат обучения в другой формат весов временной модели.
5. *find_lr.py*: инструмент для поиска оптимальных минимальной и максимальной скоростей обучения.

6. `visualize.py`: вызывает TensorBoard для визуализации статистики результатов обучения.
7. `split_exr.py`: разделяет многоканальное изображение EXR на несколько изображений признаков.
8. `convert_image.py`: конвертирует изображение в другой формат.
9. `compare_image.py`: сравнивает два изображения с заданными метриками качества.

В настоящий момент запуск обучающего инструментария возможен на Linux (другие операционные системы еще не поддерживаются). Также в требованиях фигурируют: Python 3.7, PyTorch 1.8, NumPy 1.19, OpenImageIO 2.1, TensorBoard 2.4(опционально) или более поздние версии всего вышеперечисленного.

Набор данных должен состоять из коллекции зашумленных и соответствующих им очищенных от шума эталонных изображений. В наборе данных может присутствовать несколько зашумленных версий одного и того же изображения, например, отрендеренных с разными выборками на пиксель и/или с использованием разных семплов.

Инструментарий обучения предъявляет требования к правильному расположению файлов, а также их названию.

Изображения должны храниться в формате OpenEXR (.exr). Каждая характеристика изображения (например, цвет, альбедо) должна храниться в отдельном файле изображения, т. е. многоканальные файлы изображений EXR не поддерживаются. Многоканальные EXR-файлы можно разделить на отдельные изображения для каждой характеристики с помощью прилагаемого инструмента `split_exr.py`.

Наборы обучающих и проверочных данных можно использовать только после их предварительной обработки с помощью скрипта

`preprocess.py`. Он преобразует указанные наборы обучающих и проверочных данных в формат, который может быть более эффективно загружен во время обучения.

Один и тот же набор данных можно препроцессировать несколько раз, используя различные комбинации характеристик и опций.

Фильтры требуют отдельных обученных моделей для каждой поддерживаемой комбинации вводимых признаков. Таким образом, в зависимости от того, какие комбинации признаков пользователь хочет поддерживать для конкретного фильтра, необходимо обучить одну или несколько моделей.

После предварительной обработки наборов данных можно приступить к обучению модели с помощью скрипта `train.py`.

Обучение модели включает в себя оценку с использованием проверочного набора данных и регулярное сохранение контрольных точек. Визуализация статистики обучения может быть выполнена с помощью скрипта `visualize.py`.

Результат обучения, полученный скриптом `train.py`, не может быть сразу использован основной библиотекой. Сначала его необходимо экспортовать в формат весов модели, файл Tensor Archive (TZA).

Таким образом, изучив подробнее данную библиотеку и процесс денойзинга в целом можно сделать вывод, что данная библиотека способна помочь в устраниении зернистости на изображении при использовании локальных оценок, что приведет к более быстрому получению результата за счет сокращения времени рендеринга.

3. Практическая часть

Для любого численного метода в любой области требуется оценка его точности. Получение такой оценки возможно тремя способами сравнения с:

- Уже существующим общепринятым численным методом с известной точностью;
- Экспериментом, проведенным с известной точностью;
- Аналитическим решением.

Исторически, одним из первых методов сравнения для математических методов решения уравнения глобального освещения, стала экспериментальная установка, получившая название Cornell Box.

Чтобы доказать перспективность метода локальных оценок, его требуется сравнить с другими методами. Идеальным вариантом для сравнения любого численного метода является наличие точных аналитических решений частных случаев. Для уравнения глобального освещения не существует аналитического решения в общем виде, однако для уравнения излучательности (рисунок 3), рассмотренного ранее в главе 1.2, существуют два аналитических решения. Первое из них – фотометрическая сфера. Однако для сравнения точности она слабо подходит из-за полной симметрии. Вторым частным случаем являются две бесконечные параллельные плоскости и точечный источник между ними, он же - задача Соболева, о которой уже было сказано в главе 1.5. В изначальном виде решение задачи Соболева не является удобным для непосредственных расчетов, поэтому рассмотрим решение задачи Соболева, позволяющее получить результат в более приемлемой аналитической форме. Конечное уравнение, применяемое в расчетах, было изображено ранее на рисунке 8. Для вычислений и построения графика использовалась программа matlab. Параллельно был реализован расчет

локальной оценкой с использованием 2000 лучей, $h1, h2, p1, p2 = 0.5$.

Результат изображен на графике (рисунок 14).

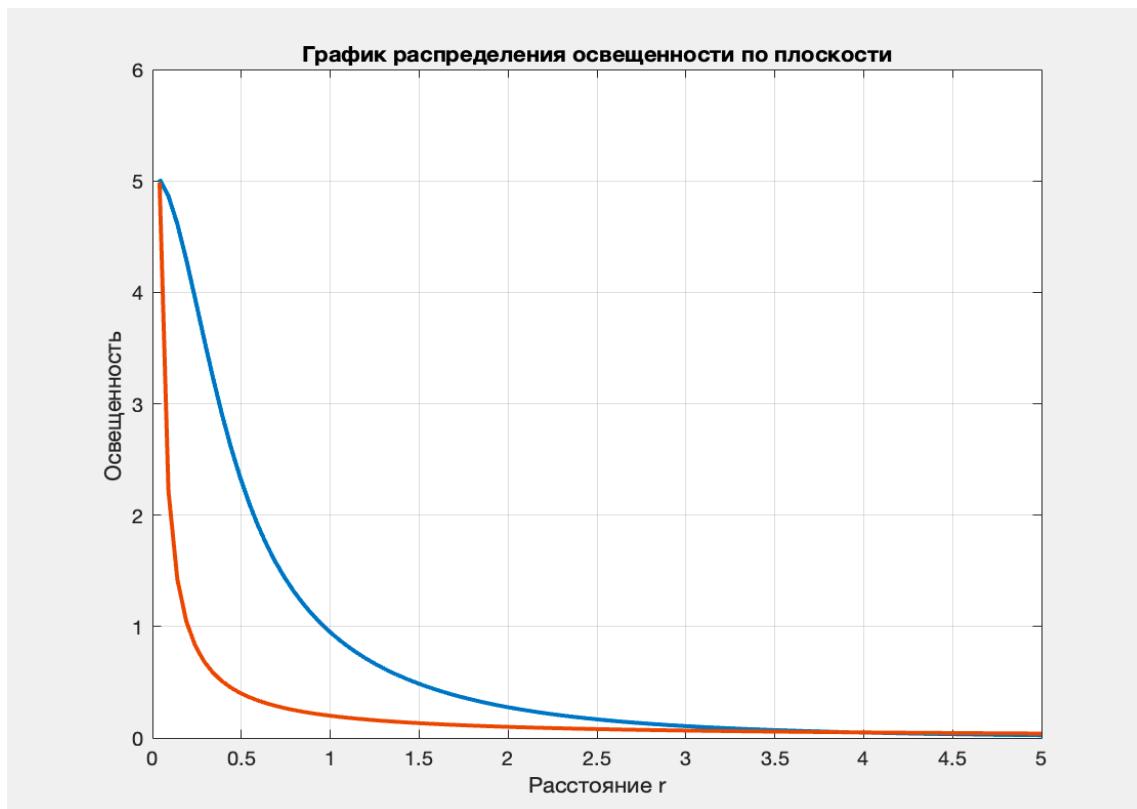


Рисунок 14 – Сравнение локальной оценки (оранжевый) и метода излучательности (голубой).

В результате замера времени получился следующий результат:

Время выполнения алгоритма локальной оценки: 0.0034 секунд

Время выполнения алгоритма метода Соболева: 0.1917 секунд

Алгоритм локальной оценки выполняется в 56.13 раз быстрее, чем метод Соболева.

Тестирование производилось на компьютере Macbook Air M1 (2020).

Единственная программа на сегодняшний день, в которой реализован расчет и моделирование сцены с помощью локальных оценок, является Light-in-Night. Это отечественная программная платформа для проектировщиков

осветительных систем, в которой используется разработанный российскими учеными кафедры светотехники НИУ «МЭИ» модуль для расчета уровней освещенности. Программа представляет собой важный шаг в области импортозамещения, способствуя развитию отечественной индустрии программного обеспечения.

Основной конкурент программы – это Dialux, которая известна своим обширным функционалом и широким спектром возможностей. Она предлагает богатый выбор инструментов для проектирования освещения и рендеринга, включая настройку параметров освещения и сценариев. Dialux также имеет долгую историю использования в индустрии и широкое распространение, что делает ее популярным выбором среди профессиональных проектировщиков.

Программа Light-in-Night в свою очередь обладает высокой производительностью и стабильной работой на устройствах с ограниченными ресурсами, что делает ее доступной и эффективной для широкого круга пользователей. Интуитивно понятный интерфейс упрощает процесс проектирования освещения, не перегружая лишними элементами. Программа предлагает широкий набор функций, а благодаря гибкости настроек может быть адаптирована под различные потребности пользователей. Благодаря активной разработке и добавлению новых функций, Light In Night представляет собой амбициозный проект с высоким потенциалом, а внедрение точного и быстрого метода моделирования, которым являются локальные оценки, у программы появляется весомое преимущество.

Стоит также упомянуть про ранее озвученную Intel Embree, которая отвечает в этой программе за оптимизацию процесса вычислений, используя эффективные алгоритмы и структуры данных, такие как BVH (Bounding Volume Hierarchy), для обеспечения высокой производительности рендеринга, в том числе в сложных сценах.

Моделирование в программе Light-in-Night начинается с создания трехмерной модели архитектурного объекта или помещения, в котором планируется осуществлять проектирование освещения. Пользователь может импортировать готовую модель из других CAD-программ или создать ее в самом Light-in-Night, используя инструменты для построения стен, полов, потолков и других элементов.

После создания модели пользователь размещает светильники и источники света в соответствии с требованиями проекта. Light-in-Night предоставляет широкий выбор светильников различных типов и моделей, что позволяет точно моделировать осветительные системы различных характеристик и конфигураций.

Затем происходит настройка параметров освещения, таких как интенсивность света, направленность лучей, цветовая температура и др. Light-in-Night позволяет проводить расчеты освещенности, учитывая различные факторы, включая отражения света от поверхностей, теневые зоны и прозрачность материалов.

После завершения настройки и расчетов пользователь может визуализировать результаты моделирования, получив трехмерное изображение сцены с учетом освещения. Light-in-Night предоставляет возможность анализировать качество освещения, выявлять теневые области и оценивать равномерность освещения в помещении.

В зависимости от потребностей проекта, пользователь может вносить корректизы в расположение светильников, параметры освещения или выбор типа светильников, чтобы достичь оптимального решения.

Визуализация в программе Light-in-Night представляет собой процесс создания трехмерного изображения освещенной сцены на основе расчетов освещенности. Этот процесс имеет ряд важных функций и применений:

1. Визуализация позволяет архитекторам, дизайнерам и инженерам получить наглядное представление о том, как будет выглядеть освещение в помещении

или на улице после установки определенных светильников и настройки параметров освещения. Это помогает лучше понять эстетические и функциональные аспекты проекта и сделать необходимые корректировки еще на этапе проектирования.

2. Визуализация помогает оценить качество освещения в различных зонах помещения или на улице. Пользователи могут анализировать равномерность освещения, наличие теневых областей, яркость и контрастность, что позволяет оптимизировать распределение света для достижения оптимальных условий в помещении или на территории.

3. Визуализация помогает в принятии решений при проектировании освещения. Путем визуального анализа результатов моделирования пользователи могут сравнивать различные варианты расположения светильников, параметры освещения и типы светильников, чтобы выбрать наиболее эффективное и удовлетворяющее требованиям решение.

4. Визуализация также является важным инструментом для взаимодействия с заказчиками или клиентами. Предоставление наглядного представления о конечном результате помогает заказчикам лучше понять концепцию проекта и принять обоснованные решения.

Таким образом, визуализация в Light-in-Night играет ключевую роль в процессе проектирования освещения, обеспечивая визуальное представление о результате и помогая принимать информированные решения для достижения оптимального освещения в зданиях и на улицах.

Результат моделирования при помощи локальных оценок можно видеть на рисунке ниже (рисунок 15).

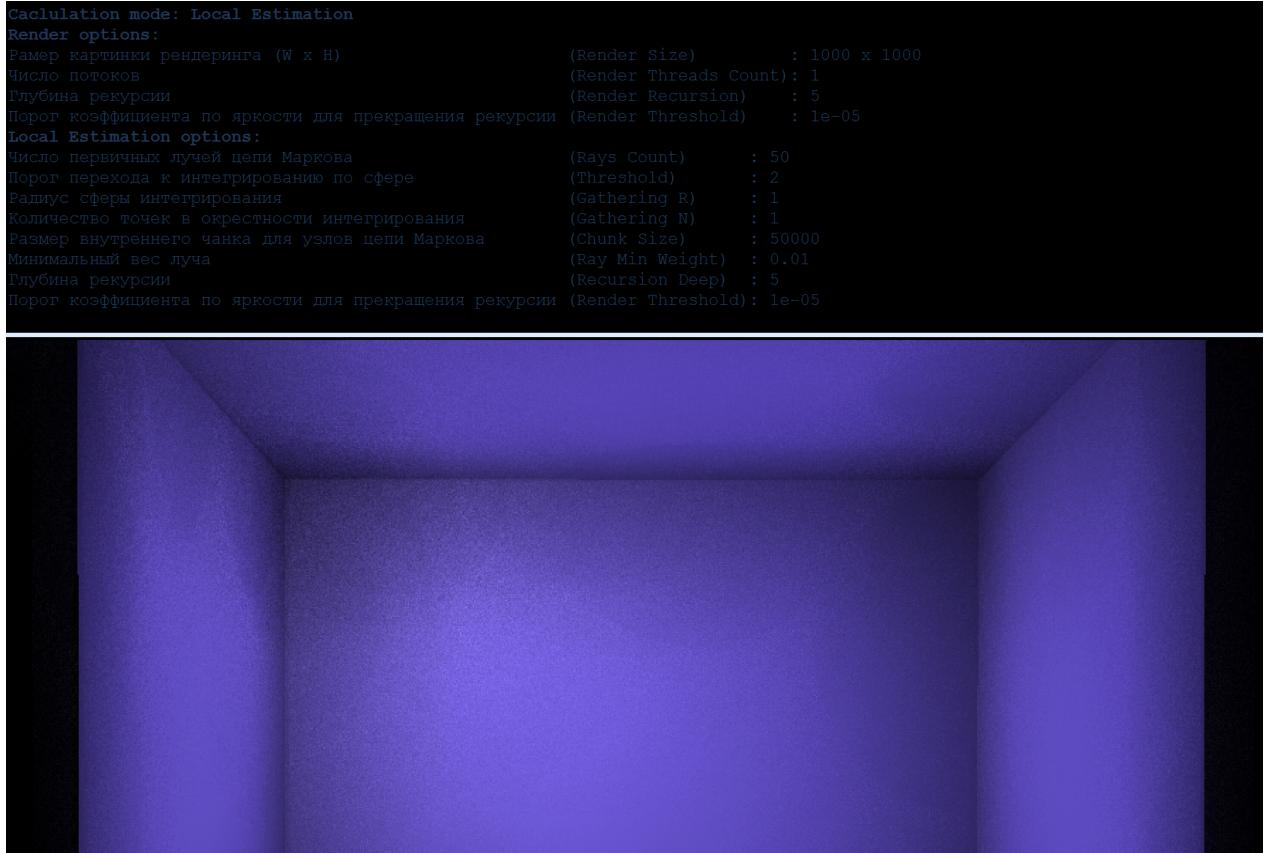


Рисунок 15 – Результат моделирования сцены при помощи локальных оценок.

Как видно по результату рендеринга, ближе к граням наблюдается повышенный эффект зашумленности, характерный для метода Монте-Карло.

В целях проверки была написана программа для моделирования разной степени зашумленности. При написании использовался язык программирования Python. Была реализована трехмерная сцена Cornell Boxes. Сцена состоит из нескольких объектов, расположенных внутри замкнутого пространства, и включает в себя источники света, объекты с различными материальными свойствами и поверхностями, а также фоновые элементы. Для визуализации этой сцены был применен метод трассировки лучей, основанный на методе Монте-Карло. Этот метод позволяет моделировать путь света от источников до камеры, учитывая его взаимодействие с объектами и материалами сцены. Путем случайной

генерации лучей света и их последующего отражения, преломления и поглощения можно получить изображение, которое визуально соответствует реальным физическим процессам освещения. На рисунке (рисунок 16) показано несколько разных промежутков времени и изменение процента зашумленности.

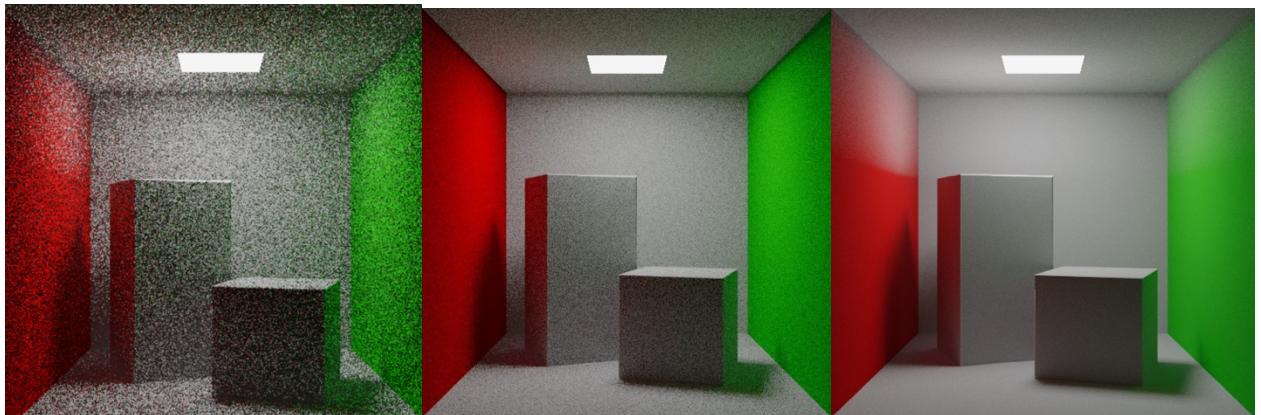


Рисунок 16 – Проявление шума разной интенсивности.

Далее, в текущую программу была внедрена библиотека Intel Open Image Denoise. Через заданное число итераций бралось изображение, и для наглядности, в отдельном окне выводилась его обработанная версия. На рисунке (рисунок 17) показана разница между зашумленным изображением и его очищенной копией. Два окна работают параллельно.

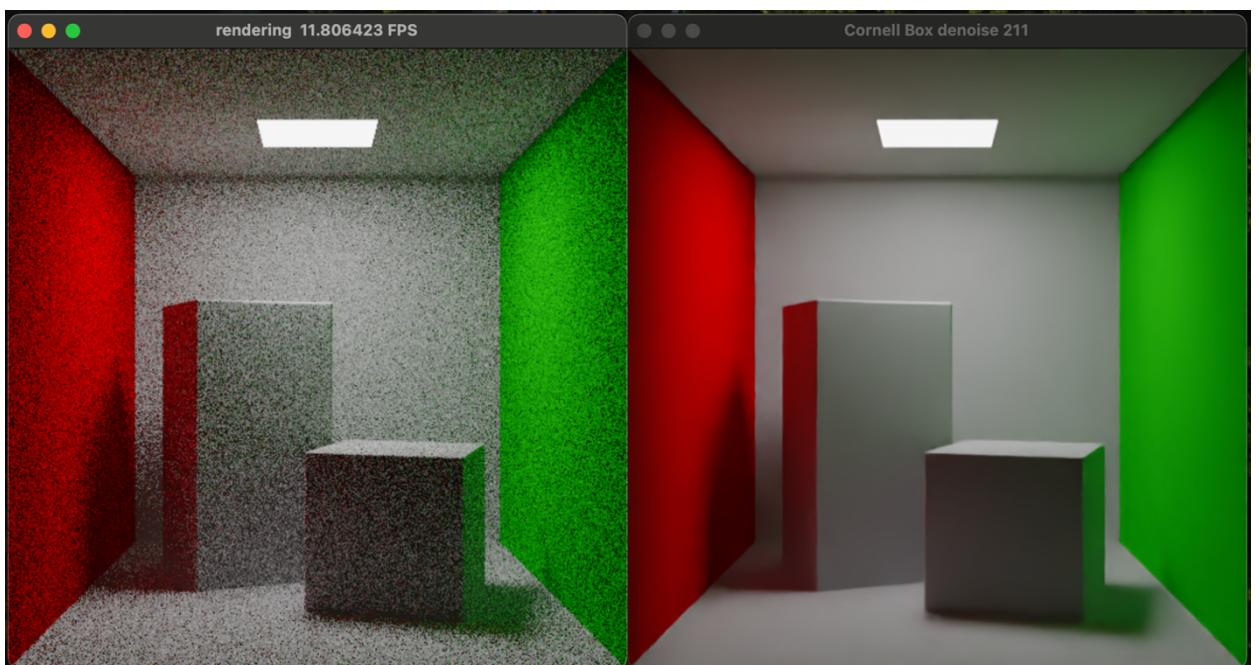


Рисунок 17 – Справа очищенная OIDN копия левого изображения.

Таким образом, благодаря комбинации метода трассировки лучей методом Монте-Карло и библиотеки Intel Open Image Denoise, удалось создать визуально привлекательное и реалистичное изображение трехмерной сцены, минимизируя при этом визуальные артефакты в виде шума.

Имея задачу измерить эффективность денойзинга, была проведена серия экспериментов, в которых сравнивались изображения до и после очистки на каждой 1000 итераций рендеринга. А также бралось первое очищенное изображение за образец, и сравнивалось с последующими очищенными для понимания, на сколько со временем изменится очистка при уменьшении влияния шумов.

Сначала пришла идея попиксельного сравнения изображений. Однако, в связи с рандомом метода Монте-Карло и нейросетевой обработкой, которая могла каждый раз незначительно изменять оттенок каждого пикселя, это делало попиксельное сравнение непрактичным, поскольку даже незначительные изменения, не воспринимаемые человеческим глазом, могли привести к большим различиям при сравнении и визуально идентичные изображения имели сходство не более 40%.

Затем было рассмотрено измерение расстоянием Хэмминга, поскольку оно позволяет оценить разницу между двумя бинарными последовательностями. Однако, данное расстояние не учитывает структурные различия между изображениями и не является достаточно чувствительным к небольшим изменениям, которые могут происходить в процессе денойзинга. Отчего результаты также не отражали объективной реальности.

В конечном итоге наиболее подходящим методом оценки эффективности денойзинга был выбран метод SSIM (Structural Similarity Index). Этот метод учитывает структурные и текстурные изменения в изображениях, а также основан на психофизиологических особенностях восприятия человека. SSIM позволяет оценить сходство между двумя изображениями не только на уровне пикселей, но и на уровне структуры и

текстуры, что делает его более чувствительным и реалистичным методом сравнения. Это делает его более точным и информативным по сравнению с попиксельным сравнением и расстоянием Хэмминга в нашей задаче. На скриншоте ниже (рисунок 18) изображены результаты сравнений с указанием процента схожести, времени и номера итерации рендеринга.

```
Образец для сравнения создан. Время 28.967365026474. Итерация 1000
SSIM между зашумленным и очищенным на каждой итерации: 74.04%. Время 29.017259120941162. Итерация 1000
SSIM между первым очищенным и очищенным на каждой итерации: 100.00%
SSIM между зашумленным и очищенным на каждой итерации: 81.48%. Время 57.64058494567871. Итерация 2000
SSIM между первым очищенным и очищенным на каждой итерации: 99.80%
SSIM между зашумленным и очищенным на каждой итерации: 85.48%. Время 86.0340428352356. Итерация 3000
SSIM между первым очищенным и очищенным на каждой итерации: 99.77%
SSIM между зашумленным и очищенным на каждой итерации: 88.01%. Время 114.50102686882019. Итерация 4000
SSIM между первым очищенным и очищенным на каждой итерации: 99.76%
SSIM между зашумленным и очищенным на каждой итерации: 89.77%. Время 142.82362985610962. Итерация 5000
SSIM между первым очищенным и очищенным на каждой итерации: 99.75%
SSIM между зашумленным и очищенным на каждой итерации: 91.07%. Время 171.19955110549927. Итерация 6000
SSIM между первым очищенным и очищенным на каждой итерации: 99.74%
SSIM между зашумленным и очищенным на каждой итерации: 92.06%. Время 199.48858880996704. Итерация 7000
SSIM между первым очищенным и очищенным на каждой итерации: 99.73%
SSIM между зашумленным и очищенным на каждой итерации: 92.85%. Время 227.8775761127472. Итерация 8000
SSIM между первым очищенным и очищенным на каждой итерации: 99.73%
SSIM между зашумленным и очищенным на каждой итерации: 93.49%. Время 256.2768929004669. Итерация 9000
SSIM между первым очищенным и очищенным на каждой итерации: 99.73%
SSIM между зашумленным и очищенным на каждой итерации: 94.03%. Время 284.6457030773163. Итерация 10000
SSIM между первым очищенным и очищенным на каждой итерации: 99.72%
SSIM между зашумленным и очищенным на каждой итерации: 94.49%. Время 312.956041097641. Итерация 11000
SSIM между первым очищенным и очищенным на каждой итерации: 99.72%
SSIM между зашумленным и очищенным на каждой итерации: 94.88%. Время 341.2959289550781. Итерация 12000
SSIM между первым очищенным и очищенным на каждой итерации: 99.72%
SSIM между зашумленным и очищенным на каждой итерации: 95.22%. Время 369.77904176712036. Итерация 13000
SSIM между первым очищенным и очищенным на каждой итерации: 99.71%
SSIM между зашумленным и очищенным на каждой итерации: 95.51%. Время 398.32773900032043. Итерация 14000
SSIM между первым очищенным и очищенным на каждой итерации: 99.71%
```

Рисунок 18 – Результаты сравнений методом SSIM.

Из результатов можно сделать вывод, что полученное на 28 секунде очищенное изображение спустя 6 минут рендеринга будет различаться всего на 0.3% от изображения, полученного при очистке на 398 секунде. Визуально изображения также выглядят идентично, что говорит об эффективности денойзинга с разным уровнем шума и возможности значительно сократить время обработки.

Заключение

В заключении следует отметить, что освещение играет фундаментальную роль в создании реалистичных визуальных эффектов в компьютерной графике. Моделирование освещения основывается на решении уравнения глобального освещения (УГО), которое описывает сложные взаимодействия света с объектами и окружающей средой. Одним из методов моделирования этого явления являются локальные оценки метода Монте-Карло, которые позволяет учитывать сложные взаимодействия света с поверхностями в сцене. В работе был подробно разобран принцип их работы, рассмотрена уникальная программа Light-in-Night, единственная на сегодняшний день, реализующая на практике обработку этим методом, а также ее ближайший конкурент в области моделирования освещения. В ходе изучения локальных оценок была выявлена проблема, подтвержденная на практике, связанная с методом Монте-Карло, привносящим в конечное изображение искажения в виде шума. В ходе поисков возможных решений были рассмотрены варианты в области денойзинга, наиболее эффективными из которых оказались свёрточные нейронные сети. Библиотека Intel Open Image Denoise, использующая эту технологию, показала себя с лучшей стороны и доказала эффективность в очистке изображений. В практической части была написана программа, создающая известную сцену Cornell Boxes и при помощи метода Монте-Карло производящая ее рендеринг. Также была использована библиотека Intel Open Image Denoise, успешно убирающая шумы параллельно с продолжающимся рендерингом. Наконец, полученные результаты были объективно измерены и эффективность денойзинга была подтверждена на практике.

Описанные технологии помимо светотехнической отрасли играют важную роль в развитии современных технологий визуализации, применяемых в иных областях, включая кино, игры, виртуальную реальность и научную визуализацию. Исследования в области моделирования освещения

продолжаются, с акцентом на совершенствовании алгоритмов, увеличении эффективности вычислений и использовании искусственного интеллекта для оптимизации процесса. Это позволяет открывать новые перспективы в создании еще более реалистичных и впечатляющих визуальных эффектов. Полученные в этой работе результаты говорят о перспективах внедрения в процесс моделирования освещения денойзеров, с которыми желаемый результат визуализации достигается в разы быстрее. В том числе это относится к методу локальных оценок и единственной, реализующей его программе Light-in-Night. Денойзинг способен показать эффективность и презентовать уникальный метод, а программе, его реализующей, сделать конкурентное преимущество.

Список используемой литературы

1. Kajiya J.T. The Rendering Equation // Proc. Conf. Computer Graphics. 1986. V. 20. No. 4. Pp. 143—150.
2. Wikipedia.org [Офиц. сайт] https://ru.wikipedia.org/wiki/Уравнение_рендеринга (дата обращения 01.03.2024).
3. Будак В.П., Макаров Д.Н. Компьютерная графика с приложением в светодизайн: Учебник для высших учебных заведений. М.: Редакция журнала «Светотехника», 2022. 75 с.
4. Фам Ван Ат, О сходимости методов Якоби и Гаусса– Зайделя, Ж. вычисл. матем. и матем. физ., 1975, том 15, номер 4, 1031–1035 https://www.mathnet.ru/links/fd521a80696c62aa2d9b17a4ae3cafd9/zvmmf6_255.pdf
5. Будак В.П., Макаров Д.Н. Компьютерная графика с приложением в светодизайн: Учебник для высших учебных заведений. М.: Редакция журнала «Светотехника», 2022. 83 с.
6. Будак В.П., Макаров Д.Н. Компьютерная графика с приложением в светодизайн: Учебник для высших учебных заведений. М.: Редакция журнала «Светотехника», 2022. 88 с.
7. Wikipedia.org [Офиц. сайт] https://ru.wikipedia.org/wiki/Закон_Снеллиуса (дата обращения 12.03.2024).
8. The Monte Carlo Method Nicholas Metropolis and S. Ulam Journal of the American Statistical Association Vol. 44, No. 247 (Sep., 1949), pp 335-341 (7 pages) Published By: Taylor & Francis, Ltd.
9. Будак В.П., Макаров Д.Н. Компьютерная графика с приложением в светодизайн: Учебник для высших учебных заведений. М.: Редакция журнала «Светотехника», 2022. 158 с.

10. Соболев В.В. Точечный источник света между параллельными плоскостями // Доклады Академии Наук СССР. 1944. Т. XLII, № 4. С. 175–176.
11. Будак В.П., Макаров Д.Н. Компьютерная графика с приложением в светодизайн: Учебник для высших учебных заведений. М.: Редакция журнала «Светотехника», 2022. 161 с.
12. Ермаков С.М., Михайлов Г.А. Курс статистического моделирования. М.: Главная редакция физико-математической литературы изд-ва «Наука», 1976. 320 с.
13. Будак В.П., Желтов В.С., Калатуцкий Т.К. Локальные оценки метода Монте-Карло в решении уравнения глобального освещения с учетом спектрального представления объектов // Компьютерные исследования и моделирование. 2012. Т. 4, № 1. С. 79.
14. В. П. Будак, А. В. Гrimайлo, В. С. Желтов, С. А. Якушев ГрафиKon 2023: 33-я Международная конференция по компьютерной графике и машинному зрению, 19-21 сентября 2023 г., стр 112
15. Github.com [Офиц. репозиторий] <https://github.com/embree/embree> (дата обращения 10.03.2024).
16. Nikolas Sanden Fileadmin.cs [Офиц. Сайт] <https://fileadmin.cs.lth.se/cs/Education/EDAN35/projects/2022/Sanden-BVH.pdf> (дата обращения 13.03.2024).
17. Alain Galvan alan.xyz [Офиц. Сайт] <https://alain.xyz/blog/ray-tracing-denoising> (дата обращения 17.03.2024).
18. Gaurav Rajpal Medium.com [Офиц. Сайт] <https://medium.com/swlh/a-comprehensive-guide-to-convolution-neural-network-86f931e55679> (дата обращения 10.03.2024).
19. Н. А. Сизов, В. П. Раевский, Д. П. Дурандин [и др.]. — Текст: непосредственный // Молодой ученый. — 2019. — № 27 (265). — С. 34-

36. — URL: <https://moluch.ru/archive/265/61326/> (дата обращения: 13.03.2024).
20. Github [Офиц. Репозиторий] <https://github.com/OpenImageDenoise/oidn-tree/master> (дата обращения 22.03.2024).
21. Intel [Офиц. Сайт] <https://www.intel.com/content/dam/develop/external/us/en/documents/intelopenimagedenoiselibRARYSAVESTIMEBOOSTSQUALITY.pdf> (дата обращения 16.03.2024).
22. Демяненко Я. М. https://edu.mmcs.sfedu.ru/pluginfile.php/49502/mod_resource/content/4/Лекция%205%20Пространственная%20обработка%20изображений.pdf (дата обращения 13.03.2024).