

Метод восстановления текстуры объектов сцены

Д. Д. Жданов, С. И. Куприянов

Университет ИТМО, Санкт-Петербург, Россия

Аннотация. В статье представлен метод восстановления текстуры диффузных поверхностей объектов сцены. Для восстановления оптических свойств объектов создается проекция UV-развертки треугольной сетки на текстурную плоскость, для которой формируются карты яркостей, полученные в результате проекции изображений сцены с каждой камеры на текстурную плоскость. Полученная карта яркостей и оригинальные изображения в сочетании с параметрами геометрии и источников света сцены позволяют рассчитать первичное и оценить вторичное освещение сцены. На основе полученных данных рассчитывается начальный цвет поверхности. Далее запускается процесс оптимизации, который рассчитывает на текстурных картах вторичное освещение методом прямой стохастической трассировки лучей и с учетом вычисленного распределения вторичной освещенности выполняет переоценку оптических свойств сцены. Процесс повторяется до тех пор, пока отклонение яркости полученного изображения от яркости исходного изображения не окажется в заданных пределах. Предложенный подход требует заранее определенной геометрии сцены, параметров источников света и камер, а также оригинальных изображений для заданных камер. В работе приводятся примеры восстановления и оценки точности восстановления оптических свойств текстурированных и нетекстурированных объектов сцены.

Ключевые слова: оптические свойства, прямое освещение, вторичное освещение, яркость, освещенность, цвет, развертка геометрии, текстурирование, оптимизация.

Method of texture reconstruction for scene objects

D.D. Zhdanov, S. I. Kupriyanov

ITMO University, Sant-Petersburg, Russia

Abstract. The article presents a method for restoring the texture of diffuse surfaces of scene objects. To restore the optical properties of objects, a projection of the uv scan of a triangular grid onto a texture plane is created, for which luminance maps are generated, obtained as a result of the projection of images of the scene from each camera onto the texture plane. The resulting luminance map and original images, combined with the parameters of the geometry and light sources of the scene, allow us to calculate the primary and evaluate the secondary illumination of the scene. Based on the data obtained, the initial surface color is calculated. Next, the optimization process is started, which calculates secondary illumination on texture maps using direct stochastic ray tracing and, taking into account the calculated distribution of secondary illumination, performs a reassessment of the optical properties of the scene. The process is repeated until the luminance deviation of the received image from the luminance of the original image is within the specified limits. The proposed approach requires a predefined geometry of the scene, parameters of light sources and cameras, as well as original images for the specified cameras. The paper provides examples of restoring and evaluating the accuracy of restoring the optical properties of textured and non-textured scene objects.

Keywords: optical properties, direct illumination, secondary illumination, luminance, illuminance, color, geometry scanning, texturing, optimization.

Введение

Задача восстановления оптических свойств сцены находит применение не только в приложениях виртуальной и дополненной реальностей, где физическая корректность не является определяющим аргументом, но и в приложениях виртуального прототипирования, требующих физическую корректность модели оптических свойств. В качестве основного решения, обеспечивающего физическую корректность восстановления оптических свойств, как правило, используется дифференцируемый рендеринг. Дифференцируемый рендеринг представляет собой метод, обеспечивающий математически формализованное вычисление градиентов целевых функций по параметрам трехмерной сцены (в данном случае оптические характеристики материалов). Он устанавливает дифференцируемое соответствие между параметрами сцены и цветовыми значениями результирующего изображения.

Расчет дифференциалов яркости по параметрам сцены позволяет рассчитывать градиенты яркости по параметрам сцены, что дает возможность применить методы оптимизации на основе градиентного спуска в задачах определения свойств сцены по целевым метрикам. Такой подход позволяет не только реконструировать геометрию сцены и характеристики источников света, но и восстановить параметры модели оптических свойств объектов сцены.

Для восстановления параметров модели оптических свойств сцены дифференцируемому рендерингу необходимо иметь ряд дополнительных данных сцены. К этим данным относятся геометрия объектов сцены, изображения сцены, информация о камерах, для которых были получены эти изображения, данные источников света. Необходимо отметить, что все дополнительные данные могут быть восстановлены средствами дифференцируемого рендеринга, но проблема их восстановления не является целью данного исследования.

В основе данной работы лежит идея уменьшения объема данных для последующего применения алгоритмов дифференцируемого рендеринга. То есть восстановленная текстура объекта определяет базовые параметры оптических свойств объекта (коэффициенты яркости по каналам цвета). Если оптические свойства близки к свойствам ламбертовых поверхностей, то предложенный алгоритм корректно восстанавливает их. В противном случае мы имеем первое приближение, которое позволяет уменьшить объем данных для последующего дифференцируемого рендеринга [1] и ускорить процесс определения параметров модели оптических свойств. Уменьшение объема данных достигается за счет преобразования яркости изображения в псевдоосвещенность, полученную в результате ее деления на восстановленную текстуру изображения. В результате получается достаточно равномерное изображение, к которому применяется алгоритм, представленный в работе [1].

Постановка задачи

В рамках исследования решается проблема определения оптических свойств поверхности при заданных параметрах геометрии сцены, источников света, камер и набора изображений, полученных на данных камерах. Предполагается, что камеры формируют изображения широкого динамического диапазона в единицах реальной яркости, т.е. яркость источников света и оптические свойства сцены физически корректно соотносятся с яркостью полученных изображений. Модель оптических свойств – диффузная с переменным коэффициентом отражения по цветовым каналам. Допускается вариация оптических свойств по поверхности сцены, которая восстанавливается в виде текстуры на поверхностях сцены. Если оптические свойства сцены близки к диффузным, то предложенное решение должно физически корректно их восстановить. В противном случае данное решение позволит уменьшить объем данных и ускорить процесс восстановления оптических свойств средствами дифференцируемого рендеринга.

Теория

Задача восстановления текстуры разбивается на две подзадачи: во-первых, формирование UV-развертки геометрических объектов сцены на текстурную плоскость; во-вторых, восстановление параметров текстуры, т.е. распределения коэффициентов яркости по каналам цвета для каждой точки текстуры, связанной с геометрическим объектом.

Формирование UV-развертки. UV-развертка представляет собой процесс проекции трехмерной поверхности на плоскость для построения текстуры. Это процесс позволяет установить соответствие между вершинами трехмерной модели и координатами текстуры (u, v) [2]. Математически UV-развертка может быть описана следующим образом: каждая точка $p \in M$ (где M – многообразие, представляющее 3D-объект) отображается в точку $(u, v) \in R^2$, сохраняя локальную геометрическую информацию [3].

Параметризация поверхности предполагает нахождение биективного отображения $\phi : M \rightarrow \Omega \subset R^2$, минимизирующего искажения. При биективном отображении каждому элементу одного множества соответствует ровно один элемент другого множества, при этом определено обратное отображение, которое обладает тем же свойством. Для треугольной сетки задача сводится к определению таких координат (u_i, v_i) для каждой вершины i , чтобы деформация у развертки была минимальной.

Классические алгоритмы UV-развертки включают различные методы.

Методы на основе граничных условий

Методы, использующие фиксацию границ на выпуклом многоугольнике, упрощают параметризацию за счет снижения вычислительной сложности [3]. Например, привязка граничных вершин к выпуклому контуру гарантирует однозначность отображения и предотвращает самопересечения [2]. Однако такие методы могут вызывать искажения внутри области из-за неравномерного распределения деформаций, особенно для поверхностей со значительной кривизной [4].

Глобальные методы: Angle-Based Flattening (ABF)

Метод ABF минимизирует угловые деформации через решение системы нелинейных уравнений, сохраняя локальные углы между треугольниками [5].

Многосеточные подходы

Многосеточные методы адаптируют разрешение, комбинируя грубые и детальные сетки [6]. Например, на уровне низкого разрешения решается задача с аппроксимированной геометрией, после чего результаты уточняются на высокодетализированных уровнях [7]. Это позволяет снизить сложность алгоритмов с $O(N^3)$ до $O(N \log N)$, где N – число вершин. Такие методы эффективны для сложных поверхностей, таких как органические формы или анатомические структуры [8].

Современные методы: машинное обучение

Нейросетевые архитектуры, такие как MeshVAE, обучаются на датасетах 3D-моделей для предсказания UV-координат [9]. Эти модели минимизируют зависимость от ручной настройки, автоматизируя развертку для неструктурированных сеток.

Кроме того, для построения развертки существуют готовые библиотеки, например xatlas. Библиотека проста в реализации и применении. Алгоритм работы библиотеки следующий:

1. Сегментация поверхности – разделение сетки на связные области (чарты) с минимизацией метрики угловых искажений. Для этого используются методы спектрального анализа, дополненные условиями сохранения локальной геометрической согласованности;

2. Оптимизация UV-размещения – упаковка чартов в текстурное пространство с применением итеративного алгоритма, снижающего перекрытия и растяжения за счет решения систем уравнений, учитывающих тензорные деформации на проекции.

Основным преимуществом xatlas является способность генерировать UV-развертки для высокодетализированных сеток с сохранением топологии, включая модели с нерегулярной триангуляцией и геометрическими аномалиями. Библиотека демонстрирует высокую устойчивость к артефактам, что критично для последующего построения HDR-карт яркостей.

К основным ограничениям можно отнести склонность к фрагментации на моделях с множеством тонких элементов и отсутствие поддержки параллельных вычислений, что ограничивает производительность при работе с большими сетками.

Формирование текстуры. После построения развертки геометрии извлекается текстура (в данном случае цвет) из набора изображений. При наличии известной геометрической развертки (UV-развертки) объекта задача построения текстурной карты сводится к проецированию информации с множества изображений, снятых под разными углами, на соответствующую параметризованную поверхность. Этот процесс требует решения задач согласования данных, коррекции освещения и устранения артефактов. Классический метод предполагает проецирование точек каждого из изображений на UV-карту с использованием матриц камеры и данных о геометрии. Для текстурных координат (u, v) значение цвета текстуры определяется как средневзвешенное значение по всем изображениям, которые он видит:

$$T(u, v) = \frac{\sum_{i=1}^N w_i \cdot L_i(x_i, y_i)}{\sum_{i=1}^N w_i}, \quad (1)$$

где $L_i(x_i, y_i)$ – яркость пикселя в i -м изображении; w_i – веса, учитывающие двунаправленную функцию отражения, условия освещения и наблюдения [10].

Кроме того, для устранения шума применяются различные фильтры (например, медианный), а для HDR-текстур используется линейное сочетание экспозиций [11]. Поскольку двунаправленная функция отражения неизвестна, то основная задача сводится к определению истинного значения цвета текстуры, т.е. цвета текстуры для некоторого опорного источника света, как правило, D65. Если предположить, что модель оптических свойств поверхности близка к ламбертовым, то истинное значение цвета текстуры это практически есть двунаправленная функция рассеивания в каждой ее точке. Это приводит нас к возможному решению [1], позволяющему ускорить процесс восстановления оптических свойств сцены средствами дифференцируемого рендеринга для объектов с постоянными свойствами (нормализация яркости изображения на «истинное» значение цвета текстуры позволяет получить однородные значения яркости изображения).

Рассмотрим идею метода восстановления оптических свойств равноярких объектов сцены. Исходя из предположения, что нам известны изображения сцены для заданных параметров камеры, геометрия сцены с наложенной на нее текстурой в виде карты реальных ее яркостей (методы построения UV-развертки на геометрии сцены и проекция карты яркостей изображений на данную развертку представлены выше) и параметры источников света, мы можем представить яркость в виде суммы яркостей прямого и вторичного освещений, зависящих от одной величины $K_d(p_j, c)$ – коэффициента яркости в каждой точке p_j поверхности по каналам цвета c . Поскольку между точками текстуры на поверхности сцены и точками изображения имеется взаимно-однозначное соответствие, то расчет яркости изображения $L(p_j, c)$ можно выполнять в координатах сцены p_j , непосредственно связанных с точками текстуры, где j – индекс точки текстуры.

$$L(p_j, c) = \frac{1}{\pi} K_d(p_j, c) \cdot \left(\sum_{k=1}^{K_L} E_{d_k}(p_j, c) + E_i(p_j, c) \right), \quad (2)$$

где $E_{d_k}(p_j, c)$ – значение прямой освещенности от k -го источника света в j -й точке текстуры (суммирование происходит по всем K_L источникам света); $E_i(p_j, c)$ – значение вторичной освещенности в j -й точке текстуры.

Поскольку геометрия сцены и параметры источников света известны, то вычисление значения $E_{d_k}(p_j, c)$ не представляет большого труда. Для точечных источников света $E_{d_k}(p_j, c) = \frac{I_k}{\|p_j - p_k\|^2} \cos \gamma$, где γ – угол между направлением освещения и нормалью к поверхности; p_k – координата точечного источника света; I_k – сила света k -го источника света на освещаемую точку. Для протяженных источников света $E_{d_k}(p_j, c)$ может быть вычислена интегрированием по поверхности источника с применением метода выборки точек на поверхности источника света. Для бесконечно удаленных источников света применяются аналогичные выражения с единственной разницей, что величина $\frac{I_k}{\|p_j - p_k\|^2}$ заменяется на плотность светового потока.

В результате из выражения (2) можно определить оптические свойства поверхности:

$$K_d(p_j, c) = \pi \frac{L(p_j, c)}{\sum_{k=1}^{K_L} E_{d_k}(p_j, c) + E_i(p_j, c)}. \quad (3)$$

Как было отмечено, $\sum E_{d_k}(p_j, c)$ может быть вычислена, $L(p_j, c)$ – известная яркость изображения. Единственная неизвестная – это $E_i(p_j, c)$, которая некоторым образом может быть оценена. Оценка этой величины осуществляется из общих соображений. Например, эта величина составляет некоторую долю прямого освещения в данной точке. Если прямого освещения нет, то выражение принимает следующий вид:

$$K_d(p_j, c) = \pi \frac{L(p_j, c)}{E_i(p_j, c)}. \quad (4)$$

В данном случае вторичное освещение может рассматриваться как некоторая доля среднего освещения сцены, например половина. Естественно, данные оценки не являются корректными, и первое условие для возможной корректировки этих значений – интегральное отражение поверхности не может превышать 1.

$$0 \leq K_d(p_j, c) \leq 1. \quad (5)$$

Если данное условие нарушается, то вторичное освещение корректируется до уровня (5). Например, если мы хотим скорректировать $K_d(p_j, c)$ до уровня $K_{d_0}(p_j, c)$, то

$$E_i(p_j, c) = \frac{\pi L(p_j, c) - K_{d_0}(p_j, c) \sum_{k=1}^{K_L} E_{d_k}(p_j, c)}{K_{d_0}(p_j, c)}.$$

Как было отмечено, выражения (2) и (3) имеют две зависимые переменные. Это коэффициент отражения и вторичная освещенность. Кроме того, в данных выражениях отсутствует формальное

взаимное влияние точек изображения друг на друга, что позволяет осуществлять поиск коэффициента яркости каждой точки объекта сцены независимо от коэффициентов яркости других точек и решать проблему оптимизации независимо для каждой точки объекта с использованием единого критерия – среднеквадратичного отклонения яркости по всем точкам изображения. Естественный параметр оптимизации – коэффициент яркости в каждой точке поверхности (текстуре), однако для его вычисления необходимо определить освещенность (прямую и вторичную). Прямая освещенность вычисляется либо аналитически, либо с использованием численного интегрирования яркости по поверхности источника света. Для расчета вторичной освещенности был реализован метод прямой трассировки лучей с сохранением результата на текстурной карте (обратная трассировка в общем случае неприменима, поскольку может оценить освещенность только в направлении наблюдения, и даже в случае ламбертовых поверхностей пересчет яркости в освещенность может создавать артефакты на текстурных картах).

Для запуска процесса оптимизации необходимы начальные значения коэффициентов яркости поверхности. Возможно несколько решений, как определить эти коэффициенты. Во-первых, можно установить некоторые константные значения для всех объектов сцены (не очень удачное решение для текстурированных объектов). Во-вторых, можно взять их пропорциональными яркости текстуры (также не очень удачное решение, поскольку яркость сцены может меняться на порядки, что не соответствует вариации реальных коэффициентов) или пересчитать их с учетом прямого освещения (решение не подходит для затененных участков). В-третьих, можно «определить» вторичную освещенность как некоторую долю прямой освещенности (или средней освещенности для затененных объектов сцены) и исходя из нее вычислить коэффициенты яркости в каждой точке объектов сцены (решение неоптимальное, но, как показал опыт, на рассматриваемых сценах наиболее удачное). Важным положительным моментом последнего решения является то, что вторичная освещенность, как правило, не имеет значительной вариации освещенности по поверхностям объектов сцены, и ее расчет может быть выполнен с меньшими вычислительными затратами, например на меньшем разрешении или с использованием фильтрации.

В результате был предложен и реализован общий алгоритм восстановления цвета поверхности в точке. Алгоритм состоит из следующих шагов:

1. Определение исходных параметров. Построение текстурных карт истинной яркости на сцене. Расчет прямого освещения для всех точек текстурных карт. Начальная оценка вторичной освещенности для всех точек текстурных карт (либо некоторая доля прямой освещенности, либо доля средней освещенности, если прямое освещение отсутствует).

2. Вычисление коэффициента яркости для всех точек текстурных карт по формулам (3) и (4). Для проверки и коррекции коэффициентов яркости используется условие (5). На данный момент сцена полностью сформирована и готова к расчету вторичного освещения.

3. Запуск процесса рендеринга. Вычисляется только вторичное освещение, прямое освещение не меняется.

4. Расчет яркости и оценка результатов вычислений. Для каждой из камер яркость вычисляется по выражению (2). Оцениваются разность рассчитанной яркости и исходная яркость в каждой точке текстуры, а также среднеквадратичное отклонение исходной и рассчитанной яркостей по всей текстуре.

5. Если точность достигнута или выполнено заданное число шагов оптимизационного процесса, формируется сигнал прекращения расчета.

6. На основе полученной разности исходной яркости текстуры и вычисленной яркости формируется ошибка оценки вторичной освещенности. На основе этой ошибки для каждой точки текстуры определяется необходимое изменение значения вторичной освещенности, для которой новые значения коэффициентов яркости поверхности $K_d(p_j, c)$, вычисленные по выражениям (3) и (4), должны приблизить вычисленную яркость к исходной.

7. Переход на пункт 2.

Общая схема алгоритма представлена на рисунке 1.

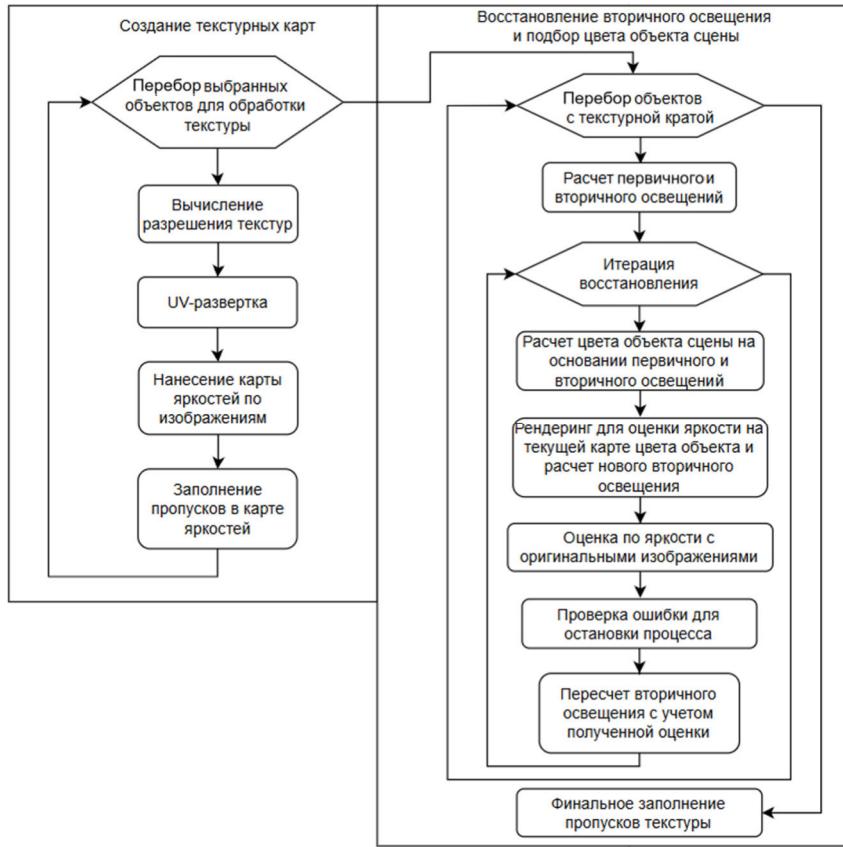


Рисунок 1. Схема алгоритма расчета коэффициентов яркости объектов сцены

Основной частью процесса оптимизации является расчет оценочной функции и выбор шага оптимизации. Оценочная функция для текстурной точки p_j и компоненты цвета c считается достаточно просто как разность оригинальной яркости, усредненной по всем видимым изображениям сцены $L_{orig}(p_j, c)$, и усредненной яркости, полученной в результате рендеринга для аналогичных изображений сцены $L_{render}(p_j, c)$:

$$\Delta L(p_j, c) = L_{orig}(p_j, c) - L_{render}(p_j, c). \quad (6)$$

В предположении ламбертовских свойств поверхности яркость изображения может быть вычислена не только на изображении, но и на текстуре, поскольку яркость вторичного освещения вычисляется как $\frac{1}{\pi} K_d(p_j, c) E_i(p_j, c)$, а значение $E_i(p_j, c)$ вычисляется непосредственно на текстуре методом прямой стохастической трассировки лучей. Яркость прямого освещения $\sum_{k=1}^{K_L} E_{d_k}(p_j, c)$ также может быть вычислена на текстуре, поскольку для каждой точки текстуры имеется обратное преобразование ее координат в систему координат сцены. Оригинальная яркость изображения сцены также существует в текстурных координатах (это первое действие, переводящее яркость изображения в текстурную карту яркости). В результате оценка производится не на изображении, а на текстуре, и оценочная функция для каждой точки текстуры – независимая величина. То есть нет необходимости формирования единой оценочной функции, алгоритм позволяет работать с большим числом независимых оценочных функций, зависящих только от одной переменной, что на порядки упрощает решение данной проблемы и сводит ее к решению большого числа однопараметрических задач.

Выразим дифференциал оценочной функции от значения вторичной освещенности. Для каждой точки текстуры j и компоненты цвета c

$$\begin{aligned}
 F(p_j, c) &= L_{orig}(p_j, c) - L_{render}(p_j, c) = \\
 &= L_{orig}(p_j, c) - \frac{K_d(p_j, c)}{\pi} \cdot \left(\sum_{k=1}^{K_L} E_{d_k}(p_j, c) + E_i(p_j, c) \right).
 \end{aligned} \quad (7)$$

Для каждой компоненты цвета производная по вторичному освещению

$$\frac{\partial F(p_j, c)}{\partial E_i(p_j, c)} = \frac{\partial}{\partial E_i(p_j, c)} \left(L_{orig}(p_j, c) - \frac{K_d(p_j, c)}{\pi} \cdot \left(\sum_{k=1}^{K_L} E_{d_k}(p_j, c) + E_i(p_j, c) \right) \right). \quad (8)$$

Градиент оценочной функции становится равным коэффициенту диффузного отражения точки поверхности для заданной компоненты цвета $-\frac{K_d(p_j, c)}{\pi}$. Тогда изменение вторичного освещения становится пропорциональным градиенту вторичного освещения для заданной компоненты цвета:

$$E_{i_{mod}}(p_j, c) = -\alpha \cdot \frac{K_d(p_j, c)}{\pi} \cdot F(p_j, c). \quad (9)$$

Новое значение вторичной яркости в точке j определяется как

$$E_{i_{new}}(p_j, c) = E_i(p_j, c) - E_{i_{mod}}(p_j, c),$$

где α – шаг оптимизации, определяющий скорость коррекции. Для некоторого шага n $\alpha_n = \frac{\alpha_0}{1+n}$, где α_0 – начальный шаг (в проводимых экспериментах начальный шаг был равен 2).

По разработанной методике были реализованы алгоритмы и выполнены тестовые расчеты по восстановлению параметров оптических свойств поверхностей.

Результаты экспериментов

Для демонстрации работы алгоритма было создано несколько вариантов сцены cornel box с разными оптическими свойствами поверхностей (однородными и текстурированными). Для поверхностей этой сцены было выполнено восстановление оптических свойств. Первый вариант сцены – это классическая сцена cornel box. На рисунке 2 показаны изображения сцены, для которых производилось восстановление оптических свойств.

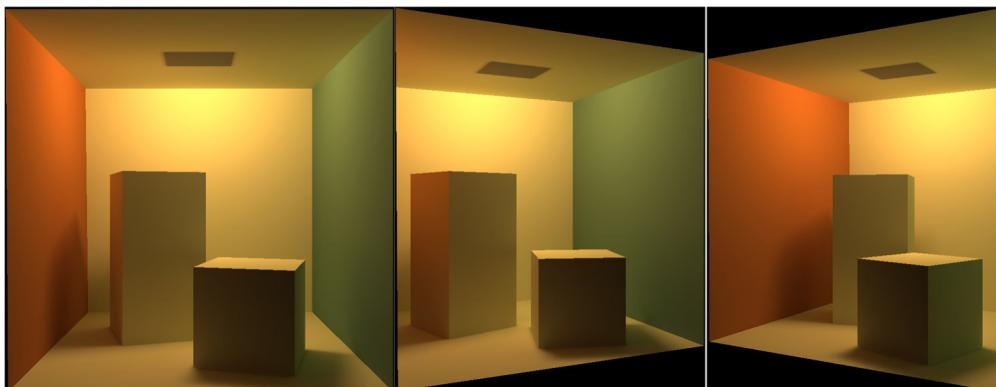


Рисунок 2. Сцена cornel box для трех ракурсов камеры

С использованием библиотеки xatlas была построена UV-развертка геометрических объектов сцены на текстурную плоскость, на которую затем было спроецировано изображение сцены и построена карта яркостей. Пример UV-развертки и карты яркостей для правой зеленой стенки сцены представлен на рисунке 3.

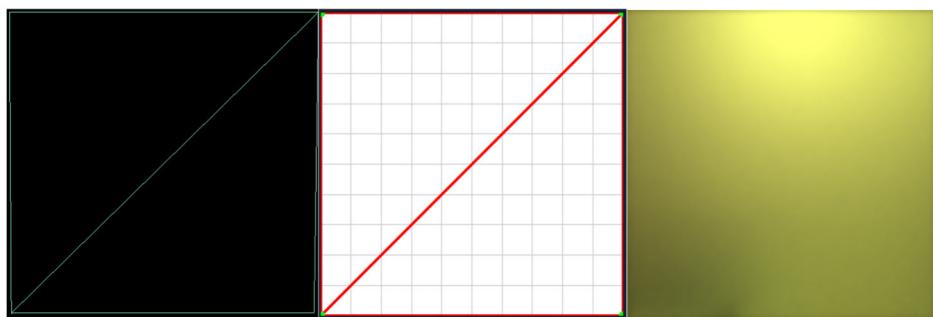


Рисунок 3. Геометрия правой (зеленой) стенки сцены (слева), текстурная развертка (по центру) и исходная карта яркости (справа)

Первые эксперименты с восстановлением оптических свойств были проведены для нетекстуированных поверхностей, однако алгоритм восстановления не был ориентирован на постоянство оптических свойств и для каждой точки поверхности восстановление осуществлялось независимым образом.

В соответствии с приведенным выше алгоритмом был выполнен первый шаг оптимизации, т.е. расчет яркости прямого освещения и оценка коэффициента яркости по компонентам цвета. После выполнения данного шага запустился итерационный процесс восстановления оптических свойств. На рисунке 4 приведены результаты первых четырех шагов процесса восстановления. Результат представлен в виде относительной погрешности восстановления, усредненной по всем компонентам цвета $\frac{|L_{orig}(p_j,c) - L_{render}(p_j,c)|}{L_{orig}(p_j,c)} * 100$.

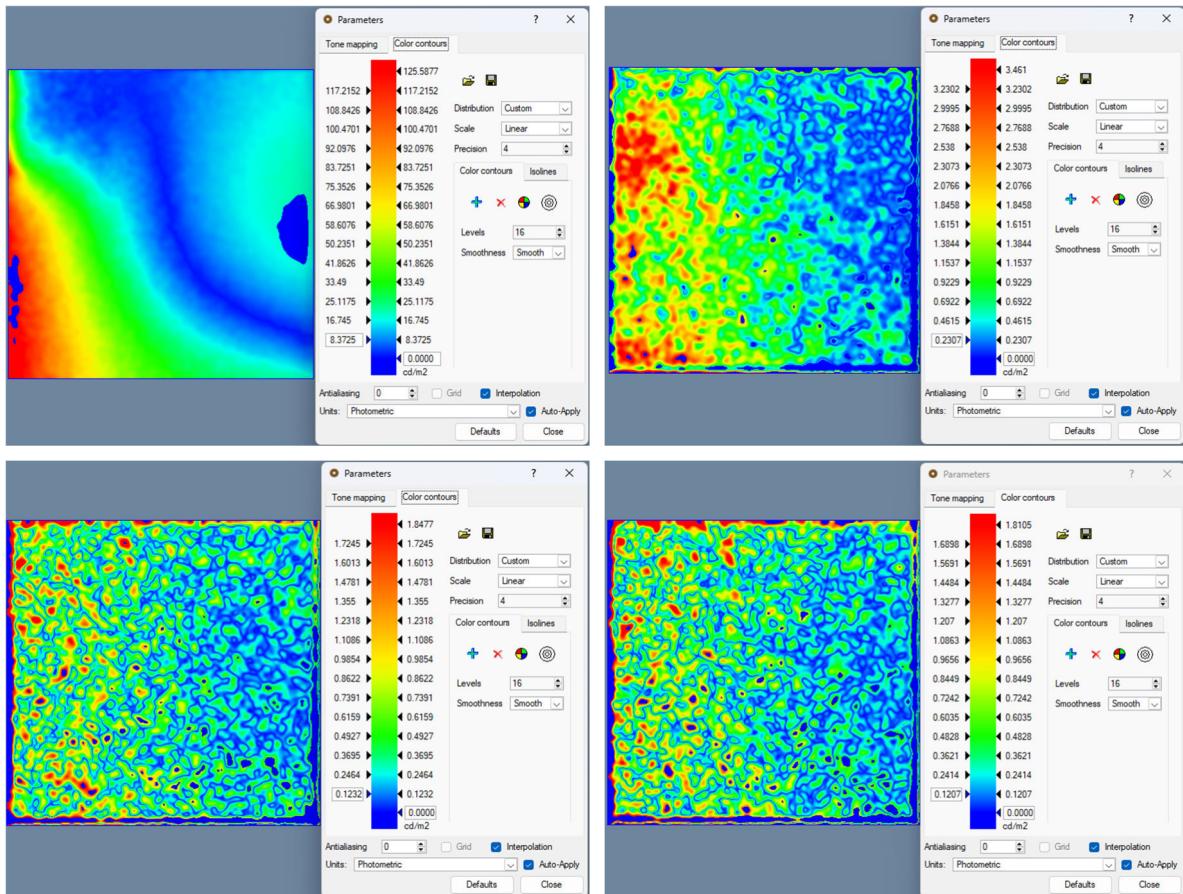


Рисунок 4. Карты нормализованной ошибки: на первом шаге (верхний левый рисунок), на втором шаге (верхний правый рисунок), на третьем шаге (нижний левый рисунок), на четвертом шаге (нижний правый рисунок)

Видно, что за первые четыре шага максимальная ошибка не превышает 2 %, что вполне допустимо и визуально не различимо на изображении. На рисунке 5 показаны оригинальное изображение и изображение после восстановления цвета.

Был проведен анализ соответствия исходного и восстановленного цвета поверхности. Анализ производился по результатам рендеринга в области, показанной на рисунке 5. Результаты анализа представлены в таблице 1.

Кроме того, была проведена оценка скорости сходимости компонент цвета поверхности к истинным значениям. Рисунок 6 демонстрирует скорость сходимости по шагам расчета.

Очевидно, что трех шагов достаточно для корректного восстановления цвета поверхности.

В приведенном примере правая стенка практически не содержит полностью или частично затененных участков. Для проверки работы алгоритма на затененных областях сцены было проведено восстановление свойств поверхности на левой стенке и потолке. Рисунок 7 демонстрирует восстановление цвета поверхности левой стенки.

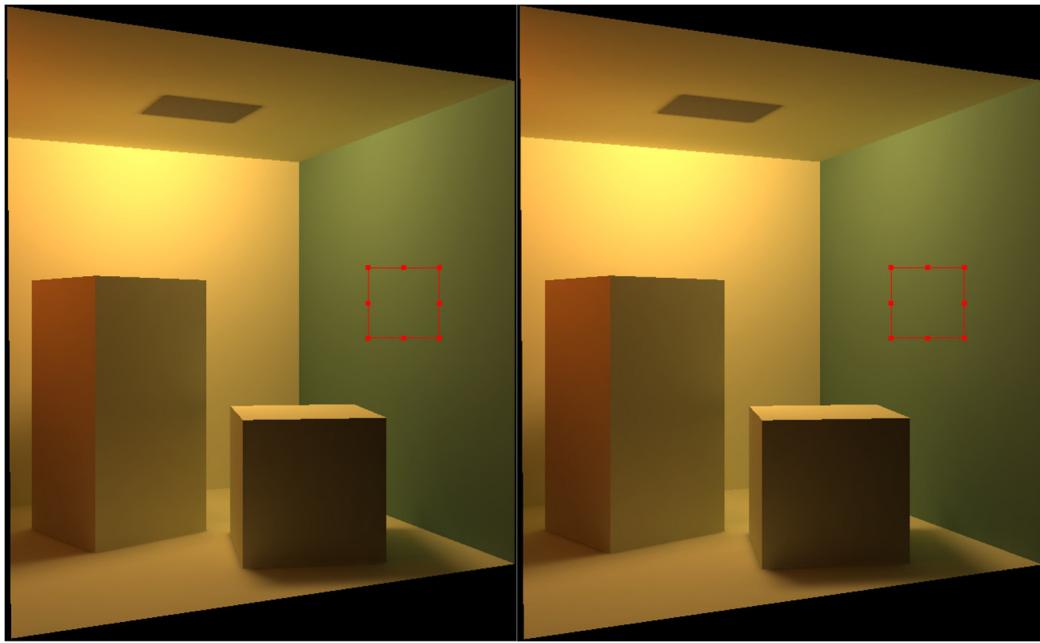


Рисунок 5. Оригинальное изображение (слева), результат рендеринга, полученный после восстановления цвета (справа)

Таблица 1. Сравнение средних значений яркости и цвета для области 90×90 точек на оригинальном изображении и изображении, полученном с помощью рендеринга с восстановленными оптическими свойствами зеленой стенки

Изображение	Оригинал	Рендеринг на полученной текстуре
Средняя яркость (cd/m^2)	0.392047	0.391723
Среднеквадратичное отклонение яркости (cd/m^2)	0.0513821	0.0516017
Цвет RGB (cd/m^2)	0.415489, 0.417419, 0.0716505	0.415249, 0.417045, 0.0715795
Цветность xy (CIE)	0.391717, 0.460432	0.391744, 0.460422

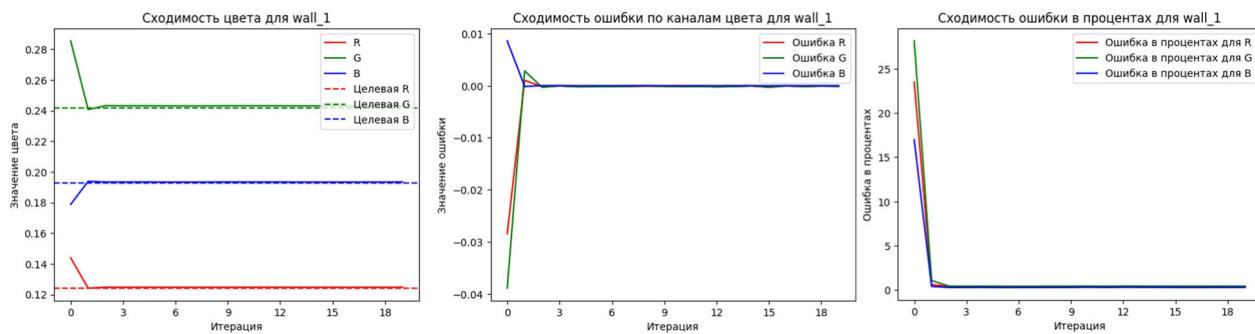


Рисунок 6. График сходимости среднего значения цвета к истинному значению цвета правой зеленой стенки (слева), абсолютная ошибка (по центру) и относительная ошибка (справа)

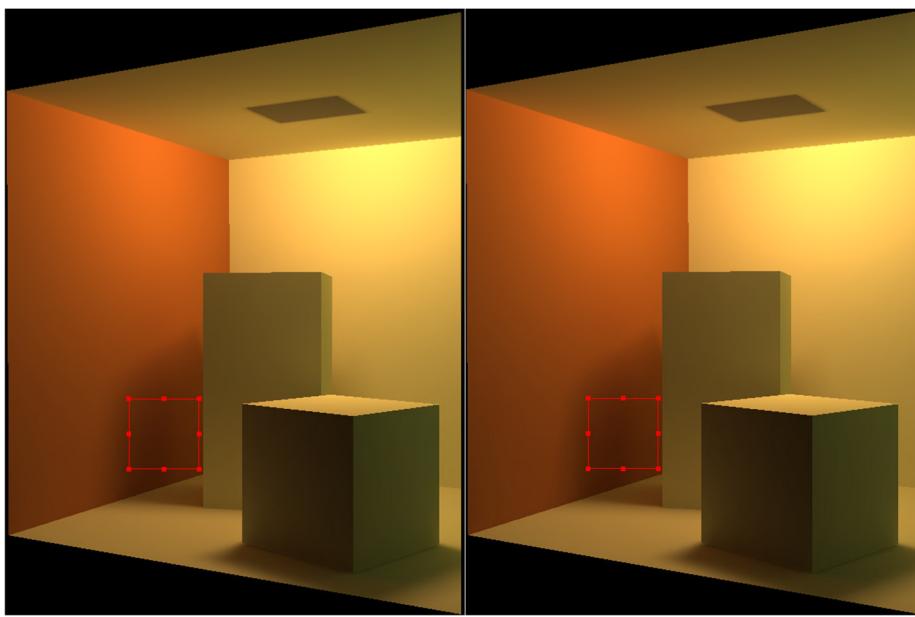


Рисунок 7. Оригинальное изображения (слева), результат рендеринга, полученный после восстановления цвета (справа)

Также был проведен анализ соответствия исходного и восстановленного цвета поверхности. Анализ проводился по результатам рендеринга в области, показанной на рисунке 7. Результаты анализа представлены в таблице 2.

Таблица 2. Сравнение средних значений яркости и цвета для области 90×90 точек на оригинальном изображении и изображении, полученном с помощью рендеринга с восстановленными оптическими свойствами красной стенки

Изображение	Оригинал	Рендеринг на полученной текстуре
Средняя яркость (cd/m^2)	0.0893196	0.0892520
Среднеквадратичное отклонение яркости (cd/m^2)	0.0181009	0.0179845
Цвет RGB (cd/m^2)	0.276969, 0.0424213, 0.00120226	0.276751, 0.0423917, 0.001120092
Цветность xy (CIE)	0.562333, 0.38754	0.562329, 0.387545

Рисунок 8 демонстрирует восстановление цвета поверхности на потолке.

Анализ проводился по результатам рендеринга в области, показанной на рисунке 8. Результаты анализа сравнения оригинального и восстановленного изображений представлены в таблице 3.

Аналогичные эксперименты были проведены и для реально текстурированных поверхностей. На рисунке 9 приведен результат восстановления истинного цвета текстуры. Восстановление было выполнено за 4 шага расчета.

Максимальная относительная ошибка не превысила 2.7 %. Поскольку сцена имеет достаточно сложное освещение с точки зрения цвета прямого и вторичного освещений, то истинный цвет текстуры может сильно отличаться от наблюдаемого. Рисунок 10 демонстрирует, как исходный цвет текстуры отличается от наблюдаемого и насколько он близок к восстановленному.

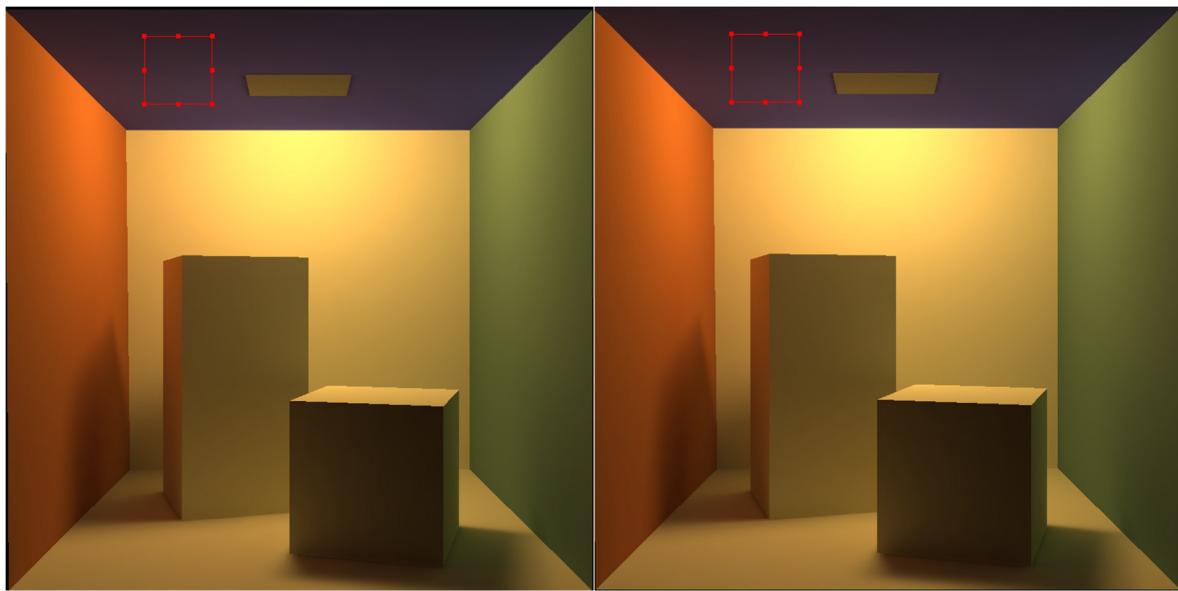


Рисунок 8. Оригинальное изображение (слева), результат рендеринга, полученный после восстановления цвета (справа)

Таблица 3. Сравнение средних значений яркости и цвета для области 90×90 точек на оригинальном изображении и изображении, полученном с помощью рендеринга с восстановленными оптическими свойствами потолка

Изображение	Оригинал	Рендеринг на полученной текстуре
Средняя яркость (cd/m ²)	0.0781676	0.0781493
Среднеквадратичное отклонение яркости (cd/m ²)	0.0214328	0.0215023
Цвет RGB (cd/m ²)	0.140054, 0.0590389, 0.085381	0.13989, 0.0590521, 0.085481
Цветность xy (CIE)	0.357999, 0.296823	0.357786, 0.296719

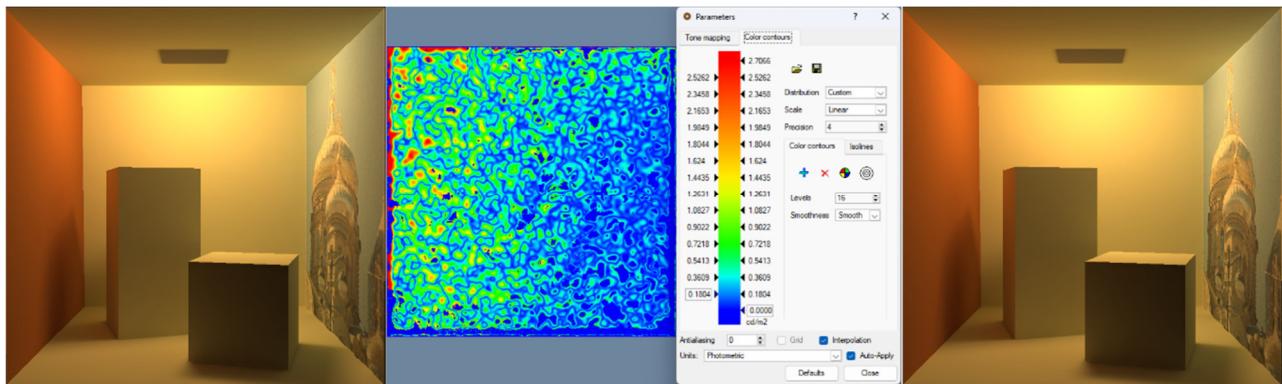


Рисунок 9. Исходное изображение церкви (слева), карта ошибок (по центру) и результат рендеринга с восстановленной текстурой (справа)

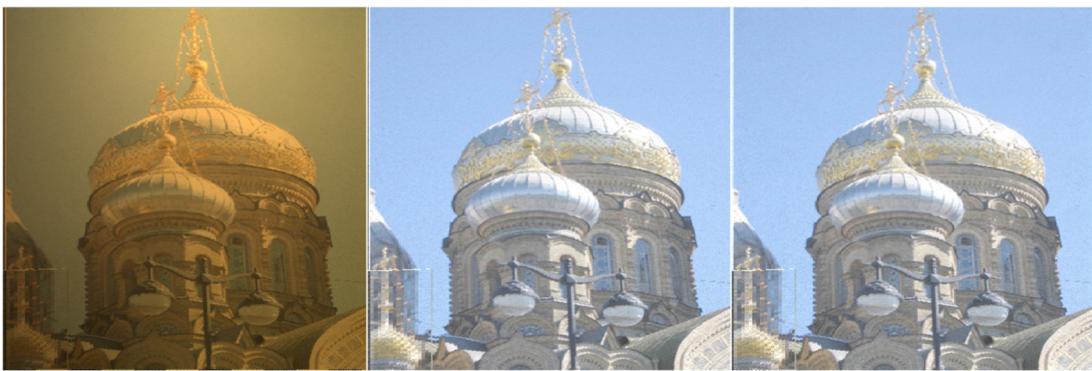


Рисунок 10. Слева – оригинальное изображение, по центру – расположена оригинальная текстура, а справа – восстановленная

Видно, что даже мелкие детали текстуры (дефект в нижней левой четверти – маленькая белая линия) были восстановлены.

Также в качестве эксперимента было проведено восстановление трех различных текстур, назначенных на стены и потолок. Рисунок 11 демонстрирует исходную и восстановленную сцены.

Исходные и восстановленные текстуры представлены на рисунке 12.

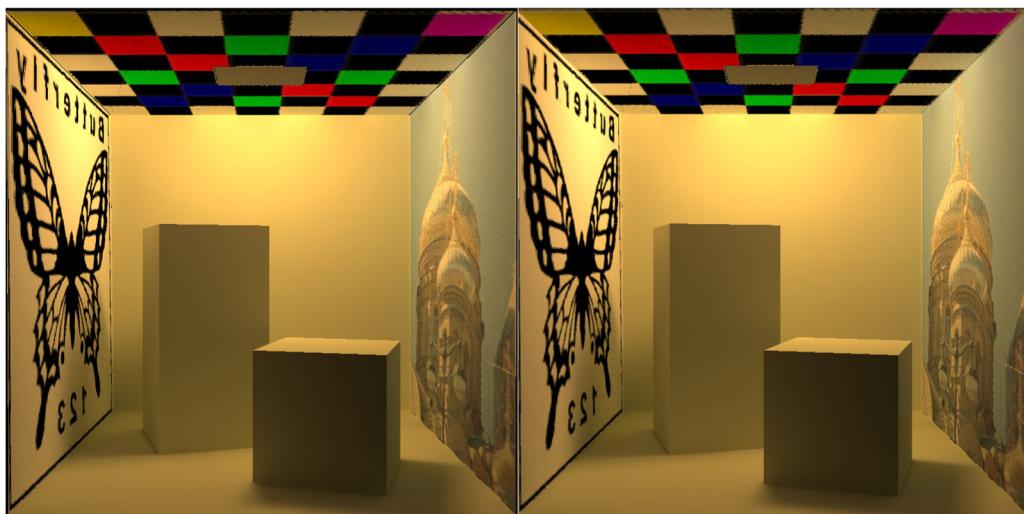


Рисунок 11. Слева – оригинальное изображение, справа – результат рендеринга для восстановленных текстур

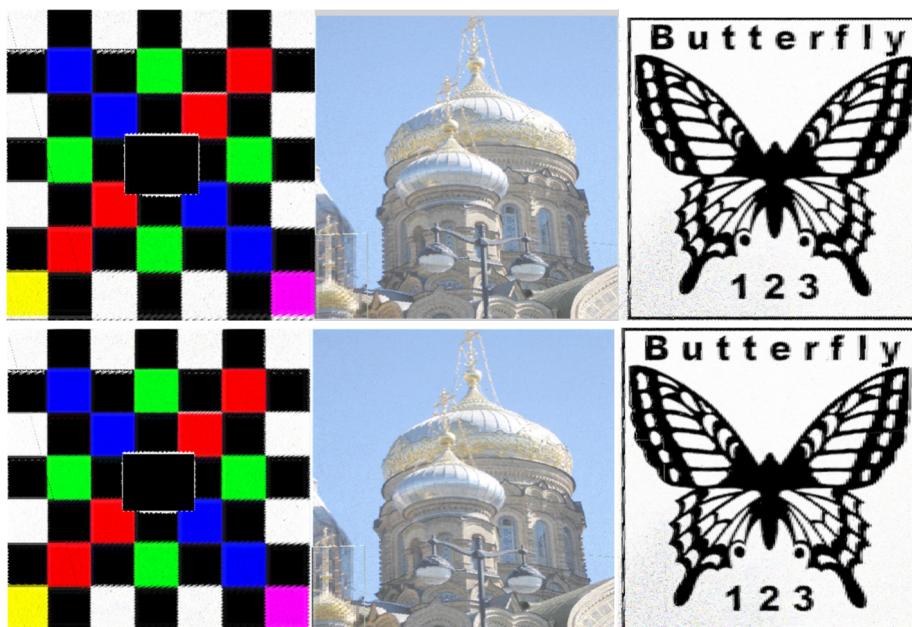


Рисунок 12. Верхний ряд – исходные текстуры, нижний ряд – текстуры после восстановления

Обсуждение результатов

Приведенные результаты были получены на компьютере AMD RYZEN 9, 5000 series, 32 Gb RAM. Для расчета вторичной освещенности использовалось 100,000,000 световых лучей на одну фазу расчета. Исследование показало, что 300,000,000 лучей позволяют повысить точность восстановления, однако замедление расчета не оправдывает незначительное повышение качества восстановления для данных сцен. Время, затрачиваемое на восстановление текстурного объекта, составляет порядка 120 секунд, в то время как Mitsuba затрачивает порядка 900 секунд на восстановление текстуры для данной сцены [12].

Согласно результатам экспериментов, разработанный метод хорошо работает как с однородными поверхностями, так и с текстурированными. Кроме того, полученная текстура представляет ценность в контексте предложенного алгоритма отбора точек для дифференцируемого рендеринга [1]. Так как точки выбираются исходя из яркости оригинальных изображений, сильная вариация цвета объекта сцены может привести к не совсем равномерному и корректному выбору областей для расчета. Это может происходить, когда у объекта присутствуют одинаково освещенные зоны, но с большим разбросом по цвету – от практически черных областей до пересыщенно белых. Для того чтобы сделать выбор независимым от цвета объекта и тем самым устранить вариацию цвета, можно использовать текстурную карту и «обесцветить» поверхность сцены.

Для примера можно устранить вариацию цвета на стенке с изображением «церковь». Для того, чтобы обесцветить изображение, понадобится текстурная карта цвета объекта и яркость оригинального изображения. Для получения «обесцвеченного» значения в точках текстуры (с малой вариацией цвета) необходимо выполнить нормировку на цвет текстуры:

$$L_{colorless}(p_j, c) = \sum_{i=1}^N \frac{L_{render}(p_j, c)}{K_d(p_j, c) / \sum_{c=1}^C K_d(p_j, c)}, \quad (10)$$

где C – число компонент цвета. На рисунке 13 показана яркость стенки с текстурой «церковь» и соответствующая «обесцвеченная» яркость.

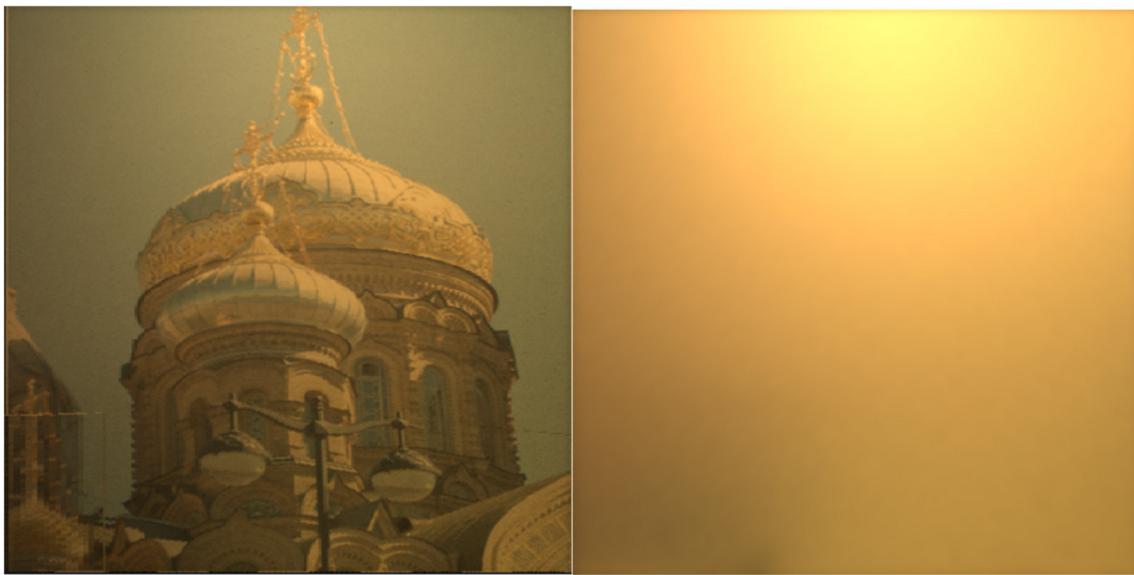


Рис. 13. Слева оригинальная яркость, справа яркость после устранения вариации цвета объекта сцены

Таким образом можно добиться устранения вариации цвета объекта и тем самым отбирать точки по яркости, более корректно используя метод, предложенный в [1].

Выводы

Предложенный алгоритм позволяет предварительно построить текстурные карты, собрать яркости по оригинальным изображениям и рассчитать прямое и вторичное освещение на развертке геометрии для ускорения процесса восстановления оптических свойств модели сцены средствами дифференцируемого

рендеринга. Кроме того, алгоритм использует эти данные для восстановления цвета равноярких поверхностей сцены.

Проведенные исследования показывают эффективность восстановления цвета поверхности на сценах как с однородными свойствами, так и со сложными цветными текстурами. Данный алгоритм позволяет убрать вариацию цвета объектов сцены и тем самым использовать предложенный ранее метод [1] для выбора наиболее информативных точек по одноцветному изображению сцены. Построенная текстура может быть передана в дифференцируемый рендеринг как вариация цвета, что позволяет снизить размерность задачи дифференцируемого рендеринга и повысить скорость и точность его работы. Кроме того, предложенный алгоритм может использоваться как независимое решение для физически корректного восстановления оптических свойств сцены, содержащих равнояркие объекты.

Список литературы

1. Куприянов С.И., Жданов Д.Д., Валиев И.В. Предобработка изображений для решения задач дифференцируемого рендеринга // Препринты ИПМ им. М.В. Келдыша. 2025. № 24. С. 1-23
2. Desbrun, M., Meyer, M., & Alliez, P. (2002). Intrinsic Parameterizations of Surface Meshes. Computer Graphics Forum, 21(3), 209–218
3. Floater, M.S., & Hormann, K. (2005). Surface Parameterization: a Tutorial and Survey. In: Advances in Multiresolution for Geometric Modelling (pp. 157–186). Springer.
4. Lévy, B., Petitjean, S., Ray, N., & Maillot, J. (2002). Least Squares Conformal Maps for Automatic Texture Atlas Generation. ACM Transactions on Graphics, 21(3), 362–371.
5. Sheffer, A., & de Sturler, E. (2001). Parameterization of Faceted Surfaces for Meshing Using Angle-Based Flattening. Engineering with Computers, 17(3), 326–337.
6. Zayer, R., Lévy, B., & Seidel, H.-P. (2007). Linear Angle Based Parameterization. In: Proceedings of the Fifth Eurographics Symposium on Geometry Processing (pp. 135–141).
7. Sorkine, O., Cohen-Or, D., & Toledo, S. (2002). High-Pass Quantization for Mesh Encoding. In: Proceedings of the Eurographics/ACM SIGGRAPH Symposium on Geometry Processing (pp. 42–51).
8. Gu, X., Luo, F., Sun, J., & Yau, S.-T. (2016). Convergence of Discrete Conformal Geometry and Computation of Teichmüller Space. Foundations of Computational Mathematics, 16(3), 649–685.
9. Tatarchenko, M., Dosovitskiy, A., & Brox, T. (2016). Multi-view 3D Models from Single Images with a Convolutional Network. arXiv preprint arXiv:1605.03557.
- 10.Debevec, P.E., Taylor, C.J., Malik, J. (1996). Modeling and Rendering Architecture from Photographs. ACM SIGGRAPH.
11. Matusik, W., et al. (2003). Image-Based 3D Photography. Eurographics Workshop on Rendering.
12. Nimier-David M., Vicini D., Zeltner T., Jakob W. (2019). Mitsuba 2: A Retargetable Forward and Inverse Renderer. Transactions on Graphics. In: Proceedings of SIGGRAPH Asia. 38(6).