

Министерство образования и науки Российской Федерации Федеральное государственное бюджетное
образовательное учреждение высшего профессионального образования

«Московский государственный технический университет имени Н.Э. Баумана»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Теоретическая информатика и компьютерные технологии»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОМУ ПРОЕКТУ НА ТЕМУ:

***«Алгоритм глобального освещения для сцены с диффузными
поверхностями»***

Студент ИУ9-52

(Подпись, дата)

А.А. Олохтонов

Руководитель курсового проекта

(Подпись, дата)

И.Э. Вишняков

Москва, 2017 г.

Содержание

Введение	3
1 Алгоритмы глобального освещения	4
1.1 Трассировка лучей	4
1.2 Метод излучательности	6
2 Разработка алгоритма	9
2.1 Стандартный метод излучательности	9
2.2 Итеративный подход	10
2.3 Метод локальных линий	12
2.4 Стохастический алгоритм	14
2.5 Модификации и ускорения	16
3 Реализация	17
3.1 Общая структура приложения	17
3.2 Загрузка полигональной модели	17
3.3 Построение BVH-дерева	17
3.4 Расчет излучательности	17
3.5 Тонирование	17
3.6 Отображение результатов	17
4 Тестирование	18
4.1 Оценка производительности	18
4.2 Оценка результатов	18
4.3 Отладка	18
Заключение	19
Список литературы	20

Введение

Генерация реалистичных изображений — направление компьютерной графики, ставящее своей целью преобразование описания произвольной сцены в изображение, минимально отличающееся от полученного при фотографировании реальной сцены (если таковая существует или может быть сконструирована). Для получения такого изображения необходимо произвести симуляцию процесса распространения света, и, с использованием полученных данных, рассчитать для каждой выбранной единицы дискретизации (пикселя, вершины полигона и т.п.) значение энергетической яркости (англ: radiance).

В действительности самая полная на данный момент модель, описывающая поведение световых частиц — кватновая — слишком подробна для такой относительно простой цели, как генерация изображений. Упрощенный вариант квантовой модели, называемой *волновой*, описывает процессы распространения света и его взаимодействия с объектами, сравнимыми с длиной световой волны (проявления такого взаимодействия можно наблюдать в таких эффектах как дифракция, интерференция и поляризация), с помощью уравнений Максвелла; но для целей компьютерной графики в большинстве случаев допустимо предположение, что все объекты много больше длины световой волны.

Добавив к описанному предположения о том, что свет распространяется в абсолютно прозрачной среде и преодолевает любую дистанцию мгновенно, можно получить т.н. модель *геометрической оптики*, опирающуюся на понятие светового луча и подчиняющуюся всем знакомым законам отражения и преломления. Все описанные в данной курсовой работе алгоритмы используют именно такую модель, которая, не смотря на все упрощения, позволяет генерировать фотореалистичные изображения.

Для генерации реалистичных изображений сцены, объекты которой обладают ничем не ограниченным набором известных заранее отражающих свойств, существует множество алгоритмов, как основанных на классическом алгоритме трассировки лучей: трассировка путей (англ: path tracing), трассировка света (англ: light tracing), так и принципиально иных, таких как метод фотонных карт (англ: photon mapping), кэш освещенности (англ: irradiance caching) и др. Однако в случае, если все объекты сцены обладают диффузными поверхностями (распространяют отраженный свет по закону Ламберта), для расчета освещенности сцены возможно использовать иной алгоритм, называемый *методом излучаемости* (англ: radiosity). В установленных ограничениях этот метод позволяет получить физически корректные значения освещенности, коль скоро точно заданы параметры сцены.

Целью данной курсовой работы является подробное изучение и реализация метода излучаемости, а так же обнаружение, анализ и последующее решение проблем, свойственных данному алгоритму. В работу также входит подробное тестирование полученной реализации, включающее в себя оценку производительности и описание созданных отладочных инструментов. В ходе выполнения курсовой работы решаются следующие задачи: обзор алгоритмов глобального освещения, определение специфики сцен с диффузными поверхностями, разработка алгоритма, основанного на методе излучаемости, и анализ возможных модификаций, разработка структуры приложения, проработка и описание деталей реализации и тестирования.

1 Алгоритмы глобального освещения

Алгоритмами глобального освещения называют семейство алгоритмов, направленных на реалистичную симуляцию процессов отражения, рассеивания, поглощения и преломления света, т.е. на расчет распределения света в рамках модели геометрической оптики. Но что скрывается под фразой “реалистичная симуляция” и что именно рассчитывается в алгоритмах глобального освещения? Ответ на вторую часть этого вопроса напрямую зависит от выбранного алгоритма, однако проблема “симуляции” световых явлений была сформулирована математически Д. Кажия в 1986 г. [1] в виде интегрального уравнения, называемого “уравнением рендеринга”, современная форма которого имеет следующий вид:

$$L_0(x, \omega_o, \lambda, t) = L_e(x, \omega_o, \lambda, t) + \int_{\Omega} f_r(x, \omega_i, \omega_o, \lambda, t) L_i(x, \omega_i, \lambda, t) (\omega_i \cdot n) d\omega_i, \quad (1)$$

где $L_0(x, \omega_o, \lambda, t)$ — спектральная энергетическая яркость в точке x на длине в направлении ω_o на длине волны λ в момент времени t , $L_e(x, \omega_o, \lambda, t)$ — излучаемая энергетическая яркость (не равна нулю только для точек, излучающих свет), $\int_{\Omega} d\omega_i$ — интеграл по единичному полушарию Ω вокруг нормали n к поверхности в точке x , $f_r(x, \omega_i, \omega_o, \lambda, t)$ — двулучевая функция отражательной способности (англ. bidirectional reflectance distribution function, BRDF) в точке x , параметризованная направлением к позиции наблюдателя ω_o и направлением к источнику света ω_i . $L_i(x, \omega_i, \lambda, t)$ описывает спектральную энергетическую яркость, направленную в точку x с направления ω_i , а скалярное произведение $(\omega_i \cdot n)$ — ослабление освещенности, вызванное увеличенным углом падения.

Так как подынтегральная величина L_i сама в свою очередь выражается аналогично, для получения решения этого уравнения даже в одной точке необходимо вычислить бесконечномерный интеграл, что не представляется возможным. По этой причине любое “решение” уравнения рендеринга представляет собой лишь приближение, получаемое численными методами. Из выбора такого метода и проистекают различные алгоритмы глобального освещения.

1.1 Трассировка лучей

При описании алгоритмов глобального освещения нельзя не упомянуть алгоритм трассировки лучей, истоком которого принято считать работу Т. Виттеда [2], представленную в 1980 г. Однако описанная модель была неполной, и не решала уравнение рендеринга (которое на тот момент еще не было сформулировано), а включала в себя только обнаружение видимых граней и отображение идеальных отражателей и преломителей. Позже алгоритм был модифицирован (в том числе самим Д. Кажия в [1]) для решения полного уравнения рендеринга и, как следствие, с его помощью стало возможно получать изображения световых явлений любой степени сложности в рамках геометрической модели оптики.

Одним из самых популярных современных вариантов алгоритма трассировки лучей является так называемый алгоритм “трассировки путей” (англ. path tracing), в ходе которого для каждого пикселя вы-

числяется приближенное интеграла

$$L_{\text{пиксель}} = \int_{\text{экран}} L(x \rightarrow e) h(p) dp, \quad (2)$$

где p пробегает все точки экрана внутри данного пикселя, а x — точка сцены, которую “видно” при взгляде с позиции e в направлении точки p . Чаще всего функция $h(p)$ представляет собой такую фильтрующую функцию, что итоговая энергетическая яркость пикселя равняется среднему арифметическому всех вычисленных значений внутри данного пикселя, однако возможны и более сложные модели, в которых роль виртуальной камеры играет уже не стеноп (англ: pinhole), а более реалистичная модель с виртуальной линзой [3].

Подстановка на место $L(x \rightarrow e)$ из уравнения (2) правой части уравнения (1) дает интеграл, решение которого можно непосредственно (после тонирования) использовать для отображения получившейся картинки.

Для решения описанного интеграла используют метод численного интегрирования Монте-Карло: расчет приближенного значения интеграла производится с помощью набора случайных точек из области интегрирования, значения в которых взвешенно усредняются. Ключевым в данном случае является выбор оценочной функции, от которого зависит как то, будут ли получаемые решения сходиться к верному при увеличении выборки, так и скорость такой сходимости. Простейшим примером использования численного интегрирования Монте-Карло для решения уравнения (2) является равномерная выборка, при которой точки выбираются из области интегрирования с вероятностью $p = \frac{1}{M}$, где M — мера области выборки (таковой может являться площадь пикселя или рассматриваемый телесный угол), и сумма полученных значений делится на N . В таком случае оценочная сумма для интеграла из уравнения (1) принимает вид

$$\frac{1}{N} \sum_{i=1}^N \frac{L(x \leftarrow \omega_i) f_r(x, \omega, \omega_i) \cos(\omega_i, N_x)}{pdf(\omega_i)}, \quad (3)$$

где ω_i — направление из полушария вокруг нормали N_x к поверхности сцены в точке x , $f_r(x, \omega, \omega_i)$ — ДФОС в точке x при взгляде по направлению, противоположному ω , $\cos(\omega_i, N_x)$ — фактор затухания, а $pdf(\omega_i)$ — вероятность выбора направления ω_i (т.н. плотность вероятности, англ: probability density function), которая зависит от выбранной параметризации полушария. Значения ДФОС, плотности и затухания извлекаются непосредственно из описания сцены, а значение $L(x \leftarrow \omega_i)$ вычисляется по аналогичной формуле, где на месте ω будет стоять уже ω_i . На практике такие расчеты чаще всего реализуются с помощью рекурсии или итерации, условием остановки которых является превышение максимального количество итераций или глубины рекурсии, либо попадание в точку, имеющую ненулевую собственную излучаемость L_e (т.е. являющуюся источником света).

В описанном виде алгоритм трассировки путей крайне неэффективен и для большинства сцен генерирует за любое разумное время очень “шумное” изображение, так как большое количество путей так и не попадает в источник света. При увеличении выборки получаемые изображения рано или поздно сойдутся к верному решению, однако во многих случаях необходимое для этого количество времени

представляется непрактичным: к примеру, отрисовка сцены, освещаемой удаленным и небольшим, но очень ярким источником света может занять недели или даже месяцы. За разумное же время без дополнительных модификаций трассировка путей генерирует следующие изображения [4, 5]:

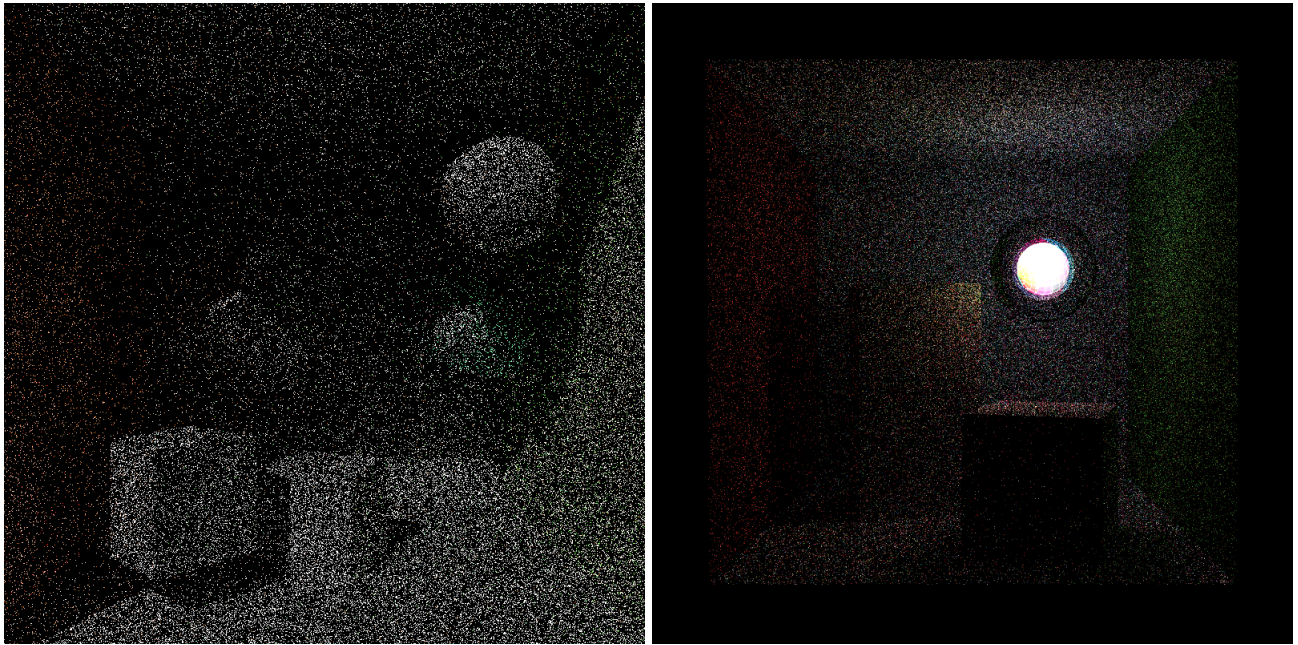


Рисунок 1 — Неэффективно рассчитанные трассировкой путей сцены

Производительность алгоритма трассировки путей может быть многократно увеличена при помощи использования техник уменьшения дисперсии, программных оптимизаций и различных эвристик, некоторые из которых затронуты в последующих главах данной курсовой работы.

Из главного преимущества алгоритма трассировки лучей — универсальности — следует и его главный недостаток: решение уравнения (2) подразумевает фиксацию позиции наблюдателя и, следовательно, представляет собой неподвижное изображение. Теоретически, алгоритм возможно адаптировать для расчета значений в вершинах сцены, заданной полигональной сеткой, однако практика показывает, что такой подход крайне неэффективен.

Для сцен с диффузными поверхностями, т.е. таких, где все ДФОС во всех точках имеет вид $f_r(x) = \frac{\rho}{\pi}$, алгоритм трассировки путей также работает крайне неэффективно по той причине, что для расчета излучаемости в таких точках требуется очень большая выборка, которая, из-за рекурсивной природы алгоритма, приводит к запредельным временам расчета для большинства нетривиальных сцен.

1.2 Метод излучательности

Помимо алгоритмов, основанных на трассировке лучей, существуют и другие, решающие схожую, но не такую же задачу: расчет освещенности сцены, где каждая точка (поверхность) является рассеива-

ющим отражателем (ДФОС во всех точках равна $\frac{\rho}{\pi}$). Для такой сцены уравнение (1) принимает вид

$$L(x) = L_e(x) + \int_{\Omega} f_r(x) L(x \leftarrow \omega_i) \cos(\omega_i, N_x) d\omega_i, \quad (4)$$

то есть не зависит от позиции наблюдателя. В указанных ограничениях возможно получить решение, которое не привязано к одному направлению взгляда и может (по окончании расчетов) быть использовано для составления интерактивной трехмерной модели.

Одним из таких алгоритмов является *метод излучаемости*, основанный на алгоритме расчета теплопередачи и впервые адаптированный для генерации реалистичных изображений в 1984 г. [6]. Основная идея алгоритма заключается в том, что поверхности сцены разбиваются на участки, излучательность на которых полагается константной, и распределение энергии рассчитывается между этими участками. Для каждого такого участка входными данными является его собственная излучаемость B_i^e (описывающая его яркость) и отражательная способность ρ_i (в диапазоне от нуля до единицы). За участки часто принимаются полигоны, описывающие сцену, если такая информация доступна.

Следующим шагом решается система линейных уравнений, связывающая излучаемость всех участков сцены. Итогом решения такой системы являются значения B_i для каждого участка, которые затем переводятся в яркость процессом тонирования и могут быть отображены на экране с помощью растеризации. Так как вычисленные значения не зависят от позиции наблюдателя, использование графических ускорителей позволяет визуализировать “освещенную” сцену в реальном времени с произвольной позиции.

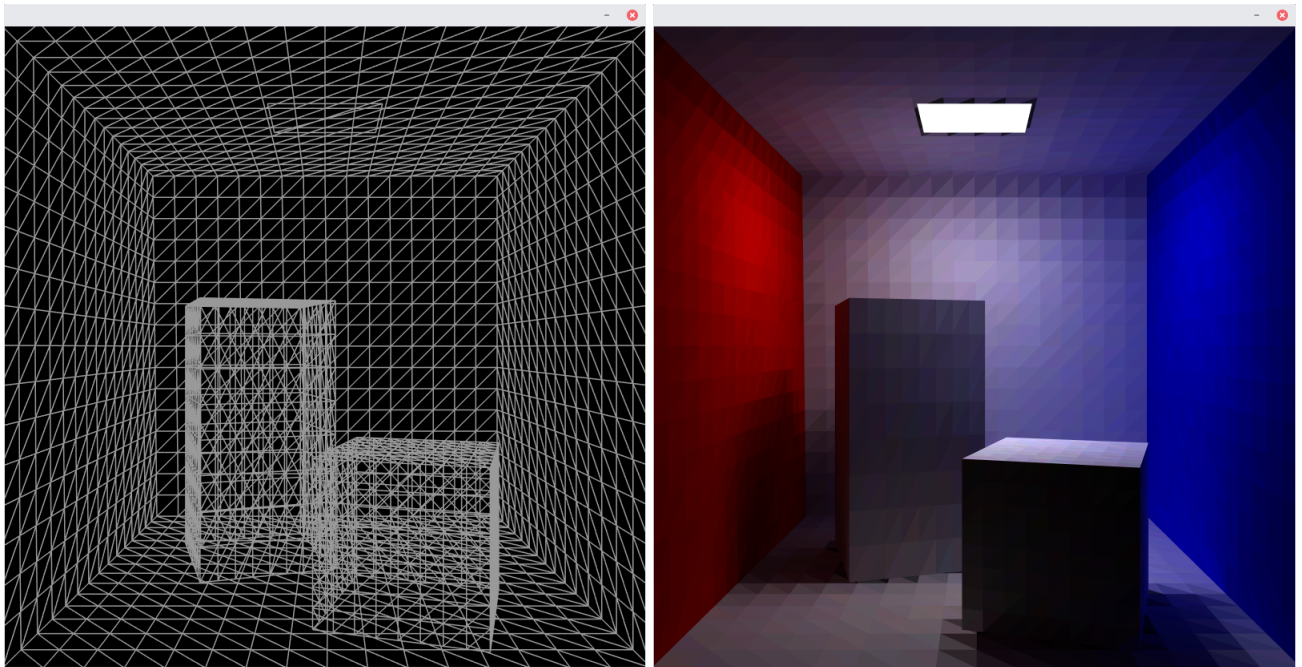


Рисунок 2 — Метод излучательности

Классический метод излучательности представляет собой частный случай более общего численного метода — *метода конечных элементов*, широко использующегося в решении задач гидродинамики,

электродинамики и теплообмена. Вкратце, математическое основание метода заключается в следующем: из уравнения (4) выводится СЛАУ путем переформулировки интеграла по полушарию на интеграл по всем поверхностям сцены:

$$L(x) = L_e(x) + \rho(x) \int_S K(x, y) L(y) dA_y, \quad (5)$$

ядром интегрирования которой служит функция $K(x, y)$, равная произведению $G(x, y)V(x, y)$, где

$$G(x, y) = \frac{\cos(\theta_{xy}, N_x) \cos(-\theta_{xy}, N_y)}{\pi r_{xy}^2} \quad (6)$$

описывает взаимную ориентацию поверхностей в пространстве, а $V(x, y)$ равняется 1, если точки x и y взаимно видимы, и нулю в противном случае. Затем, полученное равенство используется для выражения средней излучаемости B_i произвольной поверхности, после чего излучаемость полагается константой на каждом выделенном участке, и интеграл можно переписать как сумму [7]:

$$\begin{aligned} B_i &= B_{ei} + \rho_i \sum_j F_{ij} B_j \\ F_{ij} &= \frac{1}{A_i} \int_{S_i} \int_{S_j} K(x, y) dA_y dA_x, \end{aligned} \quad (7)$$

систему которых (для всех i) называют *системой уравнений метода излучательности*. Коэффициенты F_{ij} обычно называют *форм-факторами*: их вычисление и хранение служат причиной большинства проблем, связанных с реализацией метода. Как только коэффициенты системы рассчитаны, остается применить один из методов решения СЛАУ для получения искомых величин.

Следующая глава данной курсовой работы посвящена описанию и анализу классического метода излучательности, проблем, сопровождающих реализацию одного, и представлению различных модификаций, решающих эти проблемы.

2 Разработка алгоритма

2.1 Стандартный метод излучательности

Концептуально, классический метод излучательности состоит из следующих шагов:

1. дискретизация входных данных на участки, излучаемость на которых полагается константной (для данной длины волны);
2. расчет форм-факторов F_{ij} для всех пар (i, j) участков получившейся сцены;
3. численное решение системы уравнений (7);
4. отображение решения (включает в себя как процесс тонирования, так и преобразование спектральных данных в необходимое требуемое пространство).

Каждый из приведенных шагов может быть реализован с использованием различных методов, выбор которых определяет не только время работы алгоритма, но и количество необходимых для его работы ресурсов: так, форм-факторы могут быть рассчитаны заранее и сохранены в памяти для дальнейшего использования, а могут рассчитываться заново по необходимости.

Последний, четвертый, шаг алгоритма представляет меньше трудностей нежели остальные: физический смысл происходящего прост и понятен, а реализация сводится к выбору оператора тонирования, подробно описанному в главе 3.5. Первым же трем шагам сопутствуют трудности, преодоление некоторых из которых как минимум нетривиально. Далее будут описаны проблемы, сопровождающие реализацию метода излучательности в том виде, в котором он представлен в начале настоящей главы.

На первый взгляд это может показаться неочевидным, но даже реализация первого этапа алгоритма вызывает трудности. Для достижения реалистичного результата участки, на которые разбивается сцена, должны быть достаточно небольшими, чтобы передавать все детали освещения (например, на границе резких теней), однако излишнее измельчение сцены приведет к неадекватным временным затратам и требованиям к доступной памяти. В идеале процесс дискретизации сцены должен гарантировать с заданной точностью, что на каждом полученном участке излучаемость будет постоянной.

Больше всего проблем, бесспорно, связано со вторым этапом алгоритма: расчетом и хранением форм-факторов. Во-первых, форм-фактор представляет собой двойной интеграл по поверхности, т.е. для получения искомого значения необходимо решить дифференциальное уравнение четвертой степени, которое “даже самого терпеливого преисполнит отвращения” [8]. Более того, аналитически решение возможно получить только в частом случае взаимно видимых поверхностей, практически не встречающемся в задаче генерации реалистичных изображений. В остальных же случаях интеграл вычислим лишь численными методами, так или иначе включающими в себя многократную проверку взаимной видимости двух произвольных точек сцены. Не облегчает ситуации и тот факт, что подынтегральная функция может иметь разрывы разных степеней, и r_{xy}^2 в знаменателе может стремиться к нулю для двух смежных

участков. Однако самой серьезной проблемой для любых нетривиальных сцен всегда является количество вычисляемых форм-факторов: в более сложных моделях количество полигонов может исчисляться миллионами, а количество вычисляемых форм-факторов, следовательно, — триллионами, что далеко выходит за рамки возможностей современных ПК. Вместе с тем, как замечено ранее, недостаточная гранулярность разбиения приводит к потерям деталей и нарушению реализма:

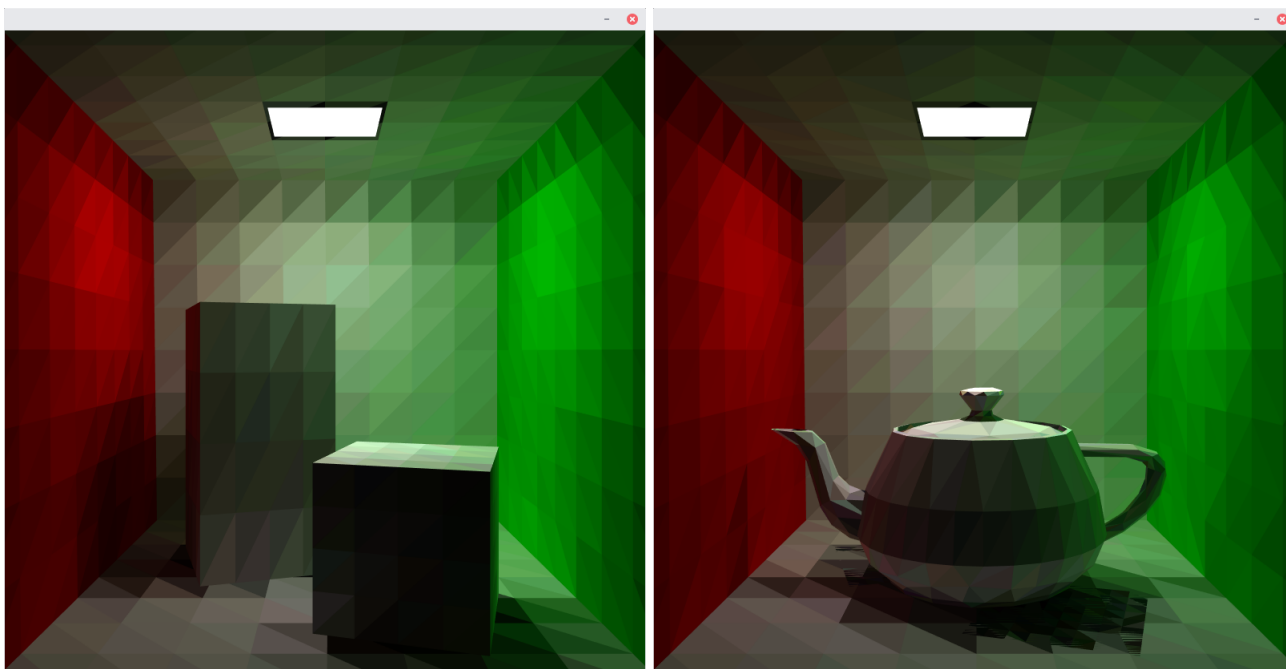


Рисунок 3 — Артефакты при недостаточном разбиении

Решение системы уравнений излучательности (7) сложной сцены детерминистскими методами также нецелесообразно из-за необходимости хранения форм-факторов.

2.2 Итеративный подход

Заметим, что матрица коэффициентов системы уравнений излучательности обладает свойством диагонального преобладания [9], что позволяет использовать итеративный метод Якоби для получения решения с любой требуемой точностью. Использование итеративного подхода допускает расчет только одной строки форм-факторов в любой момент времени, так как в процессе итерации уточненные значения рассчитываются для каждого участка независимо. Таким образом, проблема хранения квадратичного числа форм-факторов решается в корню. Более того, каждая такая итерация отвечает одному отражению всех световых лучей сцены, что интуитивно понятнее, чем расчет “всех” отражений для каждого луча по очереди, и позволяет тривиально организовать процесс предпросмотра получаемого решения.

Более конкретно, итеративный процесс Якоби применим к решению СЛАУ излучательности следующим образом: начальными значениями излучаемости для каждого участка полагается их собственная излучаемость (положительная для источников света и ноль для остальных). Далее, значения последова-

тельно уточняются с использованием приближений, полученных на предыдущей итерации. Применив к системе (7) использование итеративного метода Якоби дает для каждого участка i следующее выражение:

$$\begin{aligned} B_i^{(0)} &= B_{ei} \\ B_i^{(k+1)} &= B_{ei} + \rho_i \sum_j F_{ij} B_j^{(k)}. \end{aligned} \quad (8)$$

С использованием метода полукуба [10], позволяющим получить сразу всю строку форм-факторов для данного участка, становится возможной генерация следующего изображения [10]:



Рисунок 4 — Решение СЛАУ излучательности итеративным методом Якоби

Применение итераций Якоби для решения системы уравнений излучательности позволяет не только избавиться от необходимости хранения всех пар форм-факторов в явном виде, но и предоставляет интуитивно понятный процесс получения последовательных приближений. Так, для расчета локальной модели освещения достаточно остановить итеративный процесс после первого же приближения (см. рис. 5).

Однако, хоть проблема хранения и решается описанным методом, сохраняет свою актуальность вопрос вычисления форм-факторов: для получения одного приближения необходимо произвести квадратично зависящее от N число вычислений: $N \cdot (N - 1) / 2$ двойных интегралов по поверхностям. Сверх того, отказ от хранения форм-факторов приводит к тому, что на каждой итерации значения рассчитываются заново! Из этого следует, в том числе, что при недостаточно точно вычисляемых значениях форм-факторов, используемые итерации могут вообще не дать годного к использованию решения за адекватное время.

В следующей главе представлен т.н. *метод локальных линий*, в котором форм-факторы рассматриваются как вероятности, и потому могут быть приближены с использованием статистических методов и, что самое главное, вычисляться неявно, в следствие чего проблемы хранения и повторного вычисления форм-факторов попросту исчезают.

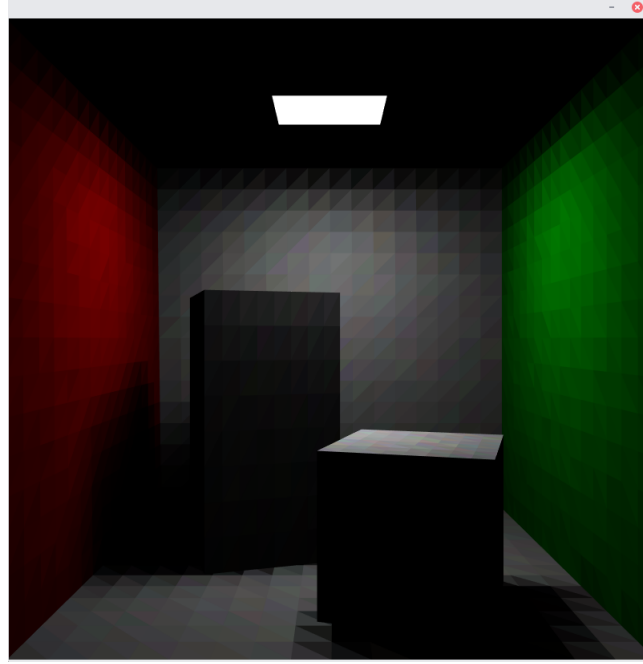


Рисунок 5 — Результат работы одной итерации алгоритма

2.3 Метод локальных линий

Рассмотрим формулу для форм-факторов из уравнения (7): значение подынтегральной функции, а значит и самого интеграла, всегда принимает неотрицательные значения. Взглянем теперь на сумму всех форм-факторов F_{ij} для фиксированного участка i :

$$\begin{aligned}\sum_j F_{ij} &= \frac{1}{A_i} \int_{S_i} \sum_j \int_{S_j} K(x, y) dA_y dA_x = \\ &= \frac{1}{A_i} \int_S K(x, y) dA_y dA_x.\end{aligned}\tag{9}$$

Переформулировав интеграл обратно в термины направлений, получим

$$\begin{aligned}\sum_j F_{ij} &= \frac{1}{A_i} \int_{S_i} \frac{1}{\pi} \int_{\Omega_x} \cos(\theta_{xy}, N_x) d\theta_{xy} dA_x \\ &= \frac{1}{A_i} \int_{S_i} \frac{\pi}{\pi} dA_x = 1.\end{aligned}\tag{10}$$

Таким образом, сумма всех форм-факторов для данного участка равняется единице (в случае незамкнутой сцены сумма может принимать меньшие значения). Наконец заметим, что простая перестановка

порядка интегралов позволяет убедиться, что для любых i, j

$$A_i F_{ij} = A_j F_{ji}. \quad (11)$$

Перечисленный набор свойств позволяет рассматривать форм-факторы F_{ij} как набор *вероятностей*. Вспомним теперь, что (за вычетом собственной излучаемости) форм-фактор F_{ij} описывает *долю* излучаемости участка i , происходящую от участка j . Учитывая, что энергия P_i и излучаемость B_i данного участка различаются лишь зависимостью от площади: $P_i = A_i B_i$, систему уравнений излучательности можно переформулировать с использованием энергии:

$$P_i = P_{ei} + \sum_j P_j F_{ji} \rho_i, \quad (12)$$

где F_{ji} обозначает часть энергии участка j (за вычетом излучаемой), отвечающей участку i , или, что то же самое, F_{ij} обозначает часть энергии участка i (за вычетом излучаемой), попадающей на участок j . Такая формулировка позволяет получить приблизительное значение форм-факторов F_{ij} для участка i при помощи простейшей симуляции, выпустив N_i лучей из случайных точек участка и распределенных согласно закону косинусов в полушаре вокруг нормали. Тогда отношение N_j/N_i попавших в участок j лучей к общему их числу будет приблизительно равно форм-фактору F_{ij} [11].

Сделать нормальный скриншот

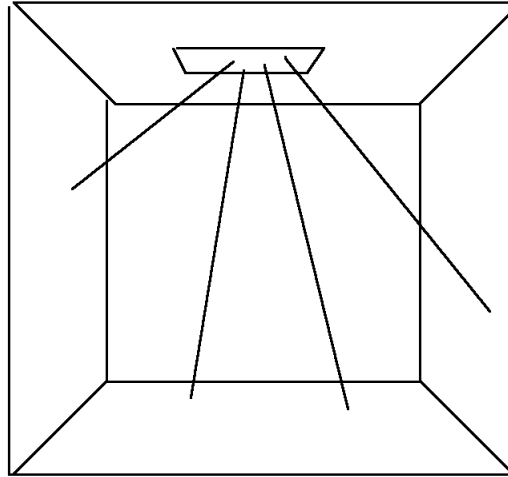


Рисунок 6 — Расчет форм-факторов с использованием локальных линий

Описанный процесс симуляции не должен использоваться непосредственно, так как до сих пор подразумевает явный расчет значений форм-факторов. Ценность подхода заключается в переформулировке проблемы подсчета искомых значений: выражение форм-фактора в качестве суммы позволяет (с учетом описанных выше свойств) применить к вычислению этой суммы метод Монте-Карло, чему и посвящена следующая глава настоящей работы.

2.4 Стохастический алгоритм

Проведя дальнейший анализ, можно получить формулировку теоремы 5.1. из [12], утверждающую, что “форм-фактор F_{ij} между двумя участками i и j в дискретизированной сцене соответствует вероятности p_{ij} того, что случайно выбранный в соответствии с равномерным распределением луч с началом в случайной точке участка i первым пересечет участок j данной сцены”. Для вычисления одного значения форм-фактора данное утверждение никак не упрощает решение поставленной задачи, однако для расчета значений *всех* форм-факторов для фиксированного участка i , описанная процедура может использоваться без изменений! Точнее, используемая в алгоритме статистическая оценка может без дополнительных модификаций быть применена для расчета всей строки форм-факторов одновременно: попадание в любой участок j уточняет значение форм-фактора F_{ij} , вне зависимости j .

С учетом сказанного выше, для расчета всех форм-факторов фиксированного участка i достаточно следующей процедуры [12]:

1. инициализировать все $F_{ij} = 0$
2. для всех $k = 1, \dots, N$
 - (а) выбрать случайную точку x на участке i
 - (б) выбрать случайное направление Θ_x из полушара вокруг нормали к участку i в точке x в соответствии с распределением косинусов
 - (в) найти ближайшее пересечение построенного луча с поверхностью j сцены, запомнить j
 - (г) $F_{ij} = F_{ij} + \frac{1}{N}$

Листинг 1 — Расчет строки форм-факторов методом Монте-Карло

где N означает количество лучей, выпускаемых из участка i , и правильный выбор N позволяет избавиться от необходимости явного вычисления и, как следствие, хранения форм-факторов.

Переформулировав итерации Якоби (8) с терминов энергии (12) на термины *нераспределенной* энергии:

$$\Delta P_i^{(k+1)} = \sum_{j \neq i} \Delta P_j^{(k)} F_{ji} p_i, \quad (13)$$

приняв вероятность выбора участка i за $p_i = \Delta P_j^{(k)} / \Delta P_T^{(k)} = \sum_i \Delta P_i^{(k)}$ и использовав районированную выборку по значимости (англ: stratified importance sampling), можно получить итоговый алгоритм [12], называемый *инкрементальным стохастическим итеративным методом Якоби* (англ: *incremental stochastic Jacobi iterative method*). В дополнение к сказанному, для удобства реализации можно ввести

для каждого участка значения δP_i , обозначающие полученную на данной итерации энергию. По окончании итерации, нераспределенная энергия полагается равной полученной, а полученная обнуляется:

1. инициализировать общую энергию $P_i = B_{ei} \cdot A_i$, нераспределенную энергию $\Delta P_i = B_{ei} \cdot A_i$, полученную энергию $\delta P_i = 0$ для всех участков i , и вычислить общую нераспределенную энергию $\Delta P_T = \sum_i \delta P_i$
2. продолжать пока для $\|\Delta P_i\| \leq \varepsilon$ или не будет превышено максимальное количество итераций:
 - (а) выбрать количество лучей N
 - (б) сгенерировать случайное число $\xi \in (0, 1)$
 - (в) инициализировать $N_{\text{prev}} = 0, q = 0$
 - (г) для каждого участка i :
 - i. $q_i = \Delta P_i / \Delta P_T$
 - ii. $q = q + q_i$
 - iii. $N_i = \lfloor Nq + \xi \rfloor - N_{\text{prev}}$
 - iv. повторить N_i раз:
 - А. выбрать случайную точку x на участке i
 - Б. выбрать случайное (согласно описанному распределению) направление Θ из точки x
 - В. определить участок j , содержащий первое пересечение луча с началом в x в направлении Θ с поверхностями сцены
 - Г. увеличить $\delta P_j = \delta P_j + \frac{1}{N} \rho_j \Delta P_T$
 - v. $N_{\text{prev}} = N_{\text{prev}} + N_i$
 - (д) пройтись по всем участкам i , увеличить энергию $P_i = P_i + \delta P_i$, заменить нераспределенную энергию $\Delta P_i = \delta P_i$ и обнулить полученную энергию δP_i . На лету пересчитать общие для сцены величины P_T и ΔP_T
 - (е) отобразить полученный результат с использованием P_i .

Листинг 2 — Инкрементальный стохастический итеративный метод Якоби

В приведенном виде алгоритм всё же не может использоваться для получения цветных изображений: описанную процедуру необходимо проводить спектрально, для каждой длины волны. Более того, так как алгоритм опирается на нахождение пересечений с объектами сцены, необходимо эффективно ор-

ганизовать доступ к геометрии сцены, иначе с ростом детализации время работы алгоритма будет расти с квадратичной зависимостью от оной.

Для относительно простой же сцены, с проведением трех независимых симуляций для красного, синего и зеленого цветов, описанный метод позволяет получить следующие изображения:

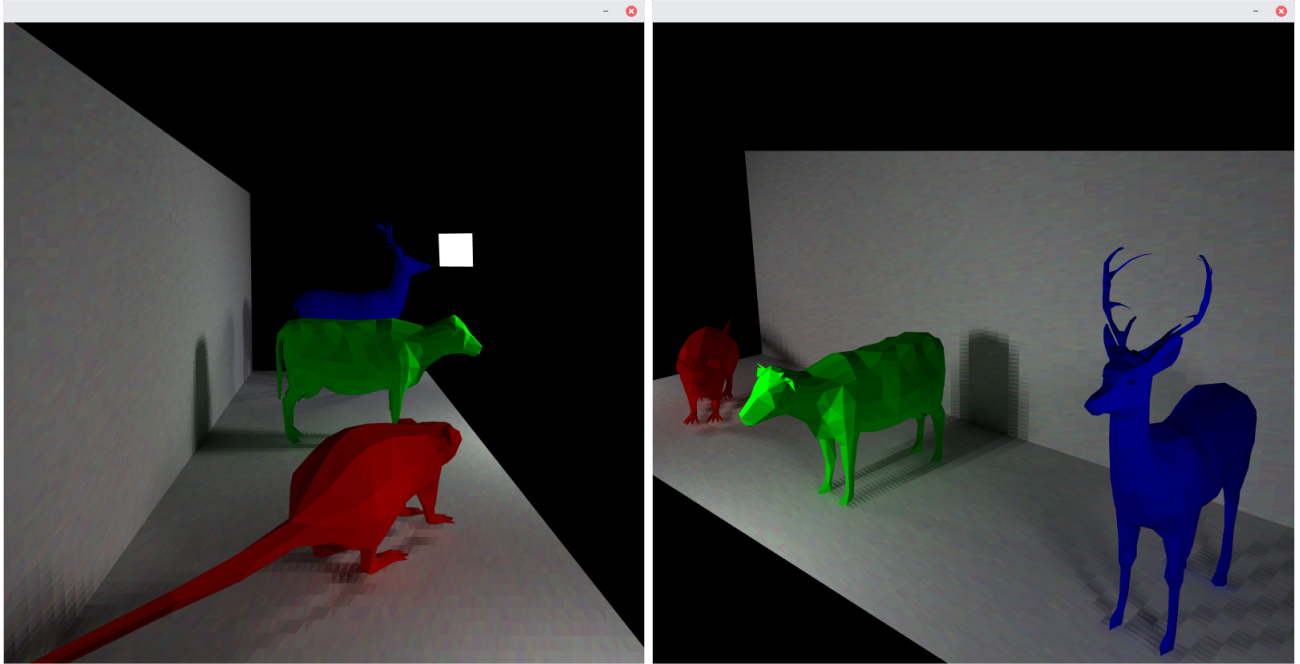


Рисунок 7 — Результат работы ИСИМЯ для трех длин волн

2.5 Модификации и ускорения

С применением простейших модификаций, описанных выше, алгоритм работает корректно, и может быть использован для получения изображений. Однако специфика выбранных методик открывает огромный простор для различных модификаций и ускорений. Так, расчет методом Монте-Карло позволяет использовать широкий набор методов для уменьшения дисперсии: выборка по значимости и районированная выборка уже включены в листинг 2; использование для генерации точек и направлений последовательностей с низкой расходимостью (англ. low-discrepancy sequence) позволяет в некоторых случаях получить ускорение вплоть до одного порядка [13]; существенно ускорить сходимость можно при помощи вычитания из искомой суммы некоторой заранее известной суммы, схожей по своей природе (англ. control variates); лучший результат дает также множественная выборка по значимости (англ. multiple importance sampling).

Помимо техник уменьшения вариации, возможно адаптировать алгоритм для параллельных вычислений: расчеты могут производиться для нескольких участков одновременно, с незначительным количеством обращений к разделяемым ресурсам и затратам на синхронизацию, а выбор и реализация структуры хранения сцены вообще играют ключевую роль в быстродействии всей программы.

3 Реализация

3.1 Общая структура приложения

3.2 Загрузка полигональной модели

3.3 Построение BVH-дерева

3.4 Расчет излучательности

3.5 Тонирование

3.6 Отображение результатов

4 Тестирование

4.1 Оценка производительности

4.2 Оценка результатов

4.3 Отладка

Заключение

Список литературы

- [1] Kajiya J. *The Rendering Equation*. / J. Kajiya // Computer Graphics (SIGGRAPH '86 Proceedings) — Dallas, 1986 — С. 143-150.
- [2] Whitted T. *An Improved Illumination Model for Shaded Display*. / T. Whitted // Communications of the ACM — 1980 — С. 343–349.
- [3] Kolb C. *A Realistic Camera Model for Computer Graphics*. / C. Kolb, D. Mitchell, P. Hanrahan // Computer Graphics (SIGGRAPH '95 Proceedings) — Los Angeles, 1995 — С. 317-324.
- [4] Yining K. *Importance Sampled Direct Lighting*. / Yining K. // Code and Visuals — 2013 — Режим доступа: <http://blog.yiningkarlli.com/2013/04/importance-sampled-direct-lighting.html> (дата обращения 11.01.2018)
- [5] Yining K. *Bidirectional Pathtracing Integrator*. / Yining K. // Code and Visuals — 2015 — Режим доступа: <http://blog.yiningkarlli.com/2015/02/bidirectional-pathtracing-integrator.html> (дата обращения 11.01.2018)
- [6] Goral C. *Modeling the Interaction of Light Between Diffuse Surfaces*. / C. Goral, K. Torrance, D. Greenberg, B. Battaile // Computer Graphics (SIGGRAPH '84 Proceedings) — Minneapolis, 1984 — С. 213-222.
- [7] Cohen M. *Radiosity and Realistic Image Synthesis* / Shir90 M. Cohen, J. Wallace — Boston: Academic Press Professional, 1993 — 381 с.
- [8] Schröder P. *A closed form expression for the form factor between two polygons* / P. Schröder, P. Hanrahan // Tech. Rep. CS-404-93 — Department of Computer Science, Princeton University, 1993.
- [9] Peters A. *Advanced Global Illumination, Second Edition* / A. Peters, P. Dutre, K. Bala — Wellesley, Massachusetts: A K Peters Ltd/CRC Press, 2006 — 384 с.
- [10] Cohen M. *The Hemi-Cube, A Radiosity Solution for Complex Environments* / M. Cohen, D. Greenberg // Computer Graphics (SIGGRAPH '85 Proceedings) — San Francisco, 1985 — С. 31-40.
- [11] Shirley P. *A Ray Tracing Method for Illumination Calculation in Diffuse-Specular Scenes*. / P. Shirley // Graphics Interface '90 — 1990 — С. 205–212. 1990.
- [12] Bekaert P. *Hierarchical and stochastic algorithms for radiosity* / P. Bekaert // Ph.D Thesis — Department of Computer Science, KU Leuven, Celestijnenlaan 200A, 3001 Heverlee, 1999 — 276 с.
- [13] Keller A. *Quasi-Monte Carlo Radiosity*. / A. Keller // Eurographics Rendering — Workshop 1996 — С. 101–110.

Термины:

'radiosity' — метод излучательности

'radiance' — энергетическая яркость

'importance sampling' — выборка по значимости

'tone mapping' — тонирование (???)

'sample' — выборка

'cosine distribution' — распределение косинусов

'estimator' — статистическая оценка

'statified sampling' — районированная выборка