

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	10
2.4 Стохастический алгоритм	11
2.5 Модификации и ускорения	11
3 Реализация	12
3.1 Общая структура приложения	12
3.2 Загрузка полигональной модели	12
3.3 Построение BVH-дерева	12
3.4 Расчет излучательности	12
3.5 Тонирование	12
3.6 Отображение результатов	12
4 Тестирование	13
4.1 Оценка производительности	13
4.2 Оценка результатов	13
4.3 Отладка	13
Заключение	14
Список литературы	15

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]:

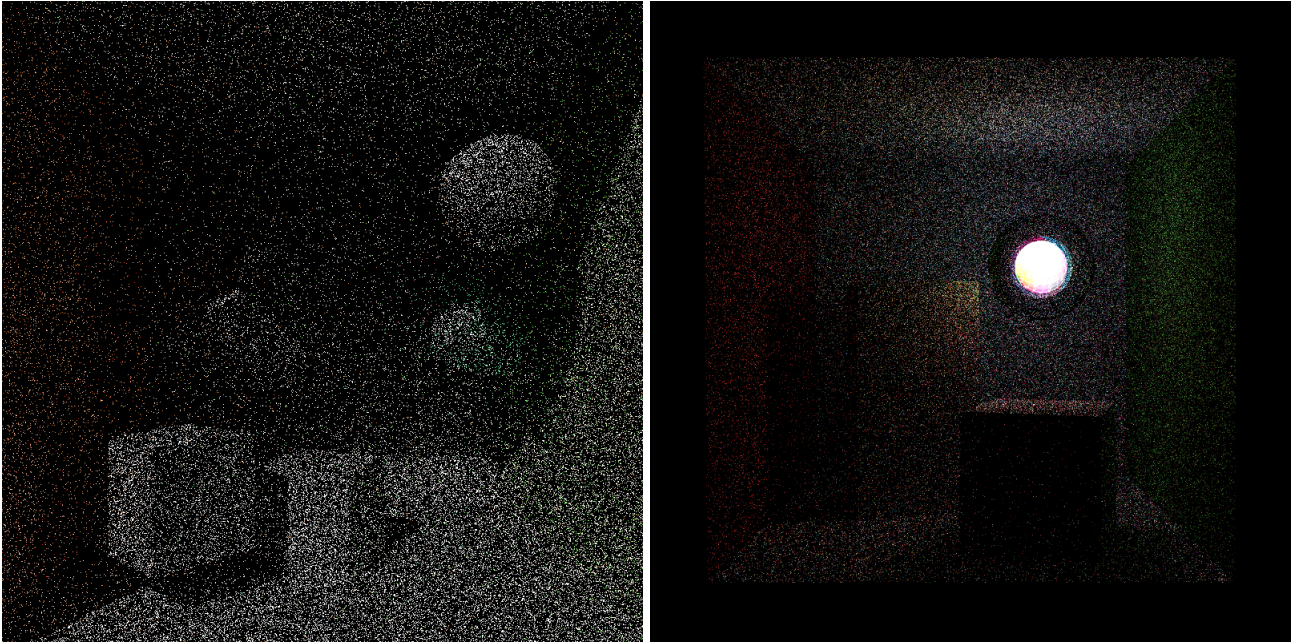


Рисунок 1 — Неэффективно рассчитанные трассировкой путей сцены

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.

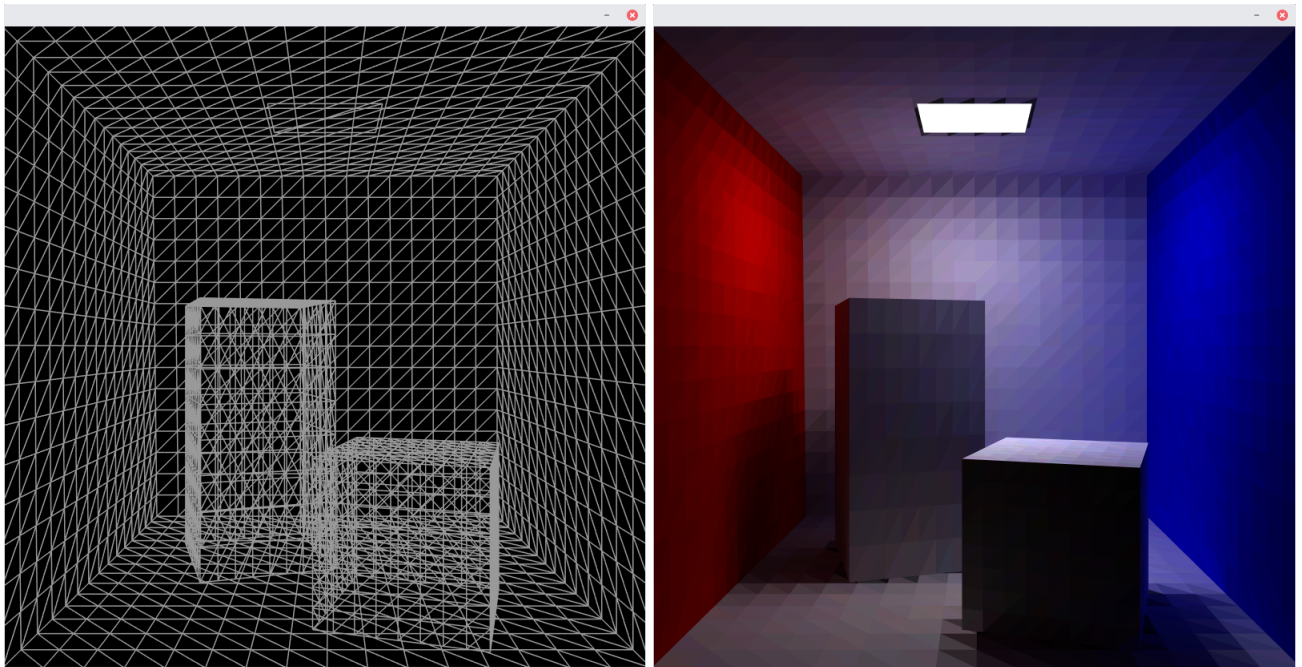


Рисунок 2 — Метод излучательности

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:

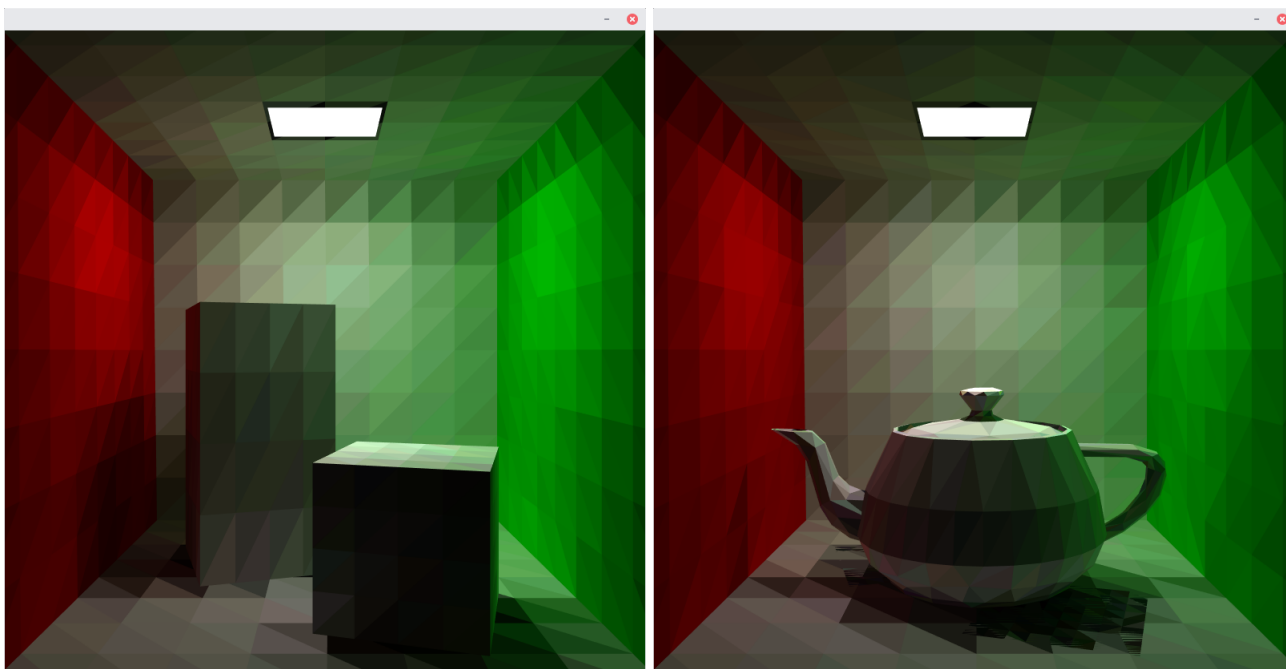


Рисунок 3 — Артефакты при недостаточном разбиении

Решение системы уравнений излучательности (7) детерминистскими методами также нецелесообразно из-за их размера (количество уравнений совпадает с количеством участком разбиения).

2.2 Итеративный подход

Происходит от итеративного подхода к решению СЛАУ. Объяснение метода + источник. Можно не хранить ВСЕ форм-факторы, а считать их по необходимости для каждой строки СЛАУ. Более подробное описание алгоритма (псевдокод), результаты.

2.3 Метод локальных линий

Формфакторы как вероятности. Мат. доказательство. Вывод -> можно семплить. Можно запустить кучу лучей и посчитать сколько куда упадет + домножить.

2.4 Стохастический алгоритм

Монте-Карло для излучательности через стохастическую релаксацию. Вообще не считать форм-факторы, а сделать чтобы все лучи несли один. количество энергии. Гораздо меньше сложность (логлинейная + НЕ НУЖНО ХРАНИТЬ ФФ). Итерации Якоби: Regular gathering, regular shooting, incremental shooting. Алгоритм для incremental shooting. Описание алгоритма на псевдокоде (выбрать стреляющий полигон, стрелять). По окончании "итерации" посмотреть сколько энергии осталось, решить когда останавливаться.

2.5 Модификации и ускорения

disc из книги: как выбирать полигон, как считать количество лучей, сколько всего брать лучей, ручная настройка. imp sampling, BVH (+ кто его придумал)

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 Proceedings, Annual Conference Series (SIGGRAPH 1995) — 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* / 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.

Термины:

'radiosity' — метод излучательности

'radiance' — энергетическая яркость

'importance sampling' — выборка по значимости

'tone mapping' — тонирование (???)

'sample' — выборка