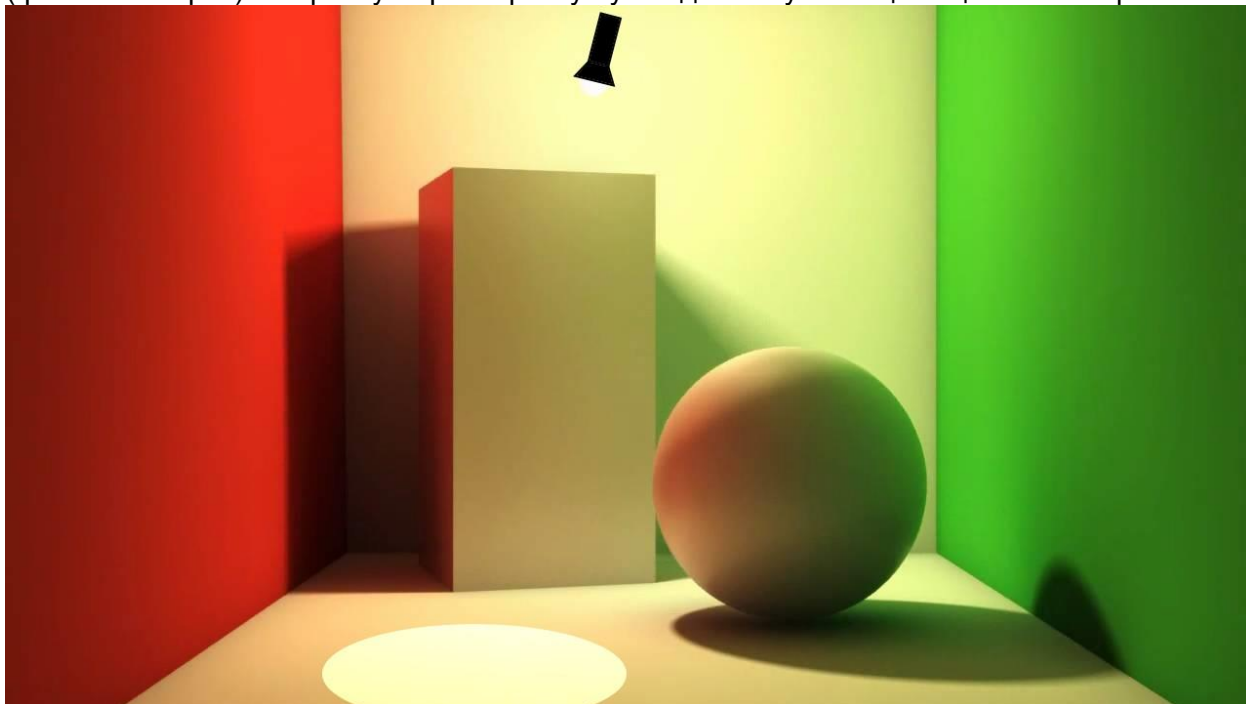


Задание 4. Трассировка лучей

Автор задания: Владимир Афанасьев

Цель задания - реализовать визуализацию сцены с аналитически заданной геометрией при помощи трассировки лучей и фотонных карт. Предлагается визуализировать пятно света от фонарика – оптической системы, состоящей из источника освещения, линзы и/или зеркала. Необходимо реализовать прямую трассировку лучей от источника света до освещаемой сцены, аккумуляцию освещённости в буфере на поверхности объектов (фотонной карте) и обратную трассировку лучей для визуализации сцены в изображении.



Пример реализации. База, мягкие тени и вторичное освещение.

Правила оформления работы

Внимание! При невыполнении указанных требований работа может не проверяться! Работы будут проверяться автоматизированно, несоблюдение этих правил усложнит нам и вам жизнь.

Архив с заданием в формате **zip** должен быть залит в систему курса. В случае превышения максимального размера архива в системе нужно разбить его на части средствами архиватора. Заливать архив на файлообменники можно только в случае невозможности залить его в систему, по предварительному согласованию с проверяющими.

Содержимое архива:

1. Папка `src` (исходный код)

- Файлы исходного кода
- Файлы проекта
- **НЕ нужно** включать в архив папку `ipch`, базы данных программы `.ncb`, `.sdf`.
- Проект должен собираться из папки `src`

2. Папка `bin` (исполняемый код - конфигурация Release, 32 бит). Обязательно проверьте, что программа запускается из папки `bin`. Желательно, на другой машине.

- Исполняемый файл
- Библиотеки, необходимые для запуска

- Данные (модели, текстуры, файл настроек). Дублировать данные в папке src не нужно.
- 3 файла настроек.
- **Программа должна поддерживать запуск из командной строки с параметрами следующего вида: “task4.exe settings.txt result.png”** т.е. первый аргумент – имя файла настроек, второй – имя выходного изображения.

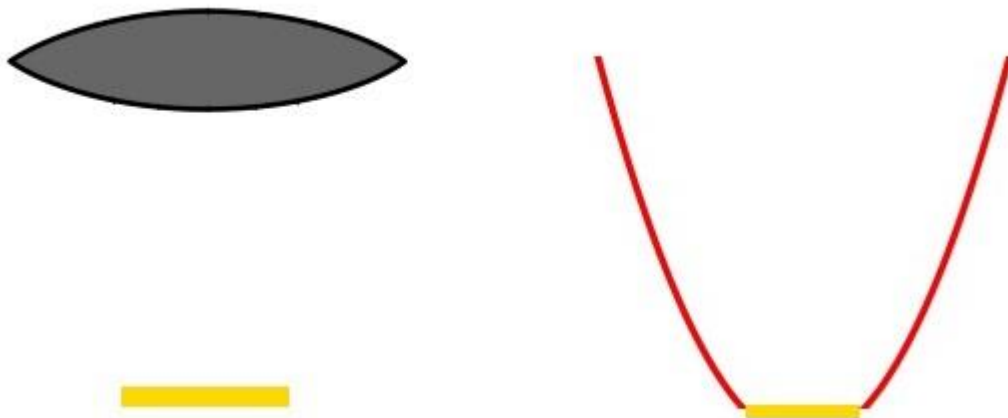
3. Папка img (визуализированные изображения сцены)

4. Файл Readme.txt

- Фамилия, имя, отчество, группа
- Операционная система
- Оборудование (процессор, видеокарта, объём памяти)
- Управление программой (формат задания настроек в файле настроек, описание интерфейса)
- Время работы программы для каждого варианта настроек
- Реализованные пункты из бонусной части

Введение

Всем известно, что карманный фонарик служит для освещения чего-либо пятном света. Как правило, источником света в фонарике служит маленькая лампочка или светодиод – источник света, близкий к точечному. Чтобы не рассеивать его свет по всему пространству и сфокусировать на нужном объекте, используется выпуклая линза, вогнутое зеркало или система линз и зеркал.



На рисунке показаны 2 принципиальные схемы карманного фонарика.

Базовая часть (до 10 баллов)

Требуется реализовать алгоритмы:

1. Прямой трассировки лучей от источника света до объектов сцены с аккумуляцией освещённости в буфере на поверхности непрозрачных объектов
2. Обратной трассировки лучей из камеры и генерации финального изображения сцены, освещённой после работы предыдущего алгоритма

Сцена состоит из:

1. Cornell box [1] – прямоугольный параллелепипед с 5 стенками разных цветов, внутри которого должны находиться ещё непрозрачный параллелепипед и шар, они должны стоять на полу. Через отсутствующую боковую стенку камера снимает сцену.
2. Фонарика, находящегося в свободном пространстве внутри Cornell box и освещающего пол под некоторым углом (не вертикально). У фонарика задано

положение **LightPos** и направление **LightDir**. Вокруг вектора направления фонарик может быть повернут произвольно на ваше усмотрение.

Фонарик – это:

- a. Светодиод, плоская квадратная площадка размера **S×S мм**, излучающая свет равномерно в полупространство. Центр площадки лежит в точке **LightPos**, площадка перпендикулярна вектору **LightDir**.
- b. Либо сферическая линза – стекло с коэффициентом преломления **RI**, ограниченное с двух сторон сферическими поверхностями с центрами на оси фонарика, смещёнными от **LightPos** на **C1** и **C2 мм** и радиусами **R1** и **R2 мм**.
- c. Либо вогнутое зеркало – часть эллиптического параболоида с круглым сечением, в фокусе которого (или близко) находится **LightPos** и ось совпадает с **LightDir**. Коэффициент отражательной способности зеркала **Reflectance**.

Размеры светодиода, линзы или зеркала должны быть подобраны так, чтобы светодиод не пересекал оптическую систему. Светодиод не обязательно ставить точно в фокус оптической системы, параметры можно подобрать исходя из качества получаемого пятна.

При преломлении луча света в линзе необходимо учитывать, что его яркость падает согласно формулам Френеля (код вычисления коэффициента пропускания есть ниже). При отражении луча от зеркала яркость также падает.

В базовой части необходимо получить изображение освещённой фонариком сцены с заданной камеры. Достаточно, если фонарик освещает только пол Cornell box.

Освещение сцены вне пятна света должно быть выполнено по модели Фонга. Точечным источником света для неё нужно считать фонарик. Сам фонарик должен быть визуализирован простейшим образом, например, в виде чёрного параллелепипеда так, чтобы было видно его направление.

Разрешение результирующего цветного изображения должно быть не менее 512x512 пикселей. Разрешение, как и другие основные параметры, должно задаваться в файле настроек. **Необходимо сделать 3 или более различных файла настроек**, демонстрирующих разные ракурсы и варианты визуализации. Как минимум один вариант настроек должен позволять получить изображение не дольше, чем за 2 минуты на средней по мощности машине.

Все размеры задаются в миллиметрах.

Дополнительная часть

Внимание! Все реализованные пункты дополнительной части должны конфигурироваться через файл настроек. У проверяющего должна быть возможность легко модифицировать файл. В readme должно быть пояснение формата файла, как с ним работать. За каждый пункт указано максимальное число баллов, которое можно получить при сложной и качественной реализации.

- **Система зеркал и линз (+3)**

Нужно сделать комбинацию из 2 и более линз, либо линзы и зеркала и настроить параметры так, чтобы получалось более-менее равномерное пятно света.

- **Освещение фонариком всех объектов в сцене (+5)**

Дополнительные баллы даются за освещение параллелепипеда, максимальный балл – за параллелепипед и шар.

- **Тени от объектов (+3)**

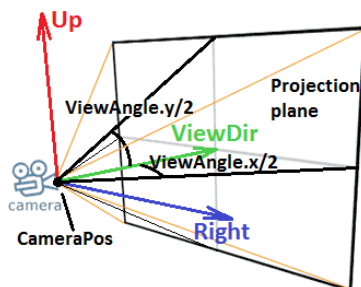
Тени должны быть на всех объектах. При этом можно считать фонарик также точечным или площадным источником освещения, от которого и образуются тени. За мягкие тени максимальный балл.

- **Параллелизм (+5)**

- **+2** за простую реализацию на OpenMP, стандартных потоках и т.п
- **+4** за Optix, CUDA, OpenCL или шейдеры OpenGL и **+1** за анимацию движущегося фонарика
- **Антиалиасинг (+1)**
- **Продвинутые методы фотонных карт (+5)**
 Аккумуляция освещённости более сложным способом, чем в текстуру. Например, способом, описанным в [2] с финальным сбором.
- **Вычисление эффективности оптической системы фонарика и оптимизация (+5)**
 Эффективностью оптической системы можно считать долю излучения светодиода, дошедшую до объектов сцены. Очевидно, что максимальная эффективность 100% будет достигнута, если убрать всю оптическую систему – тогда весь свет достигнет сцены. Но тогда не будет и пятна света, всё будет освещено равномерно. Поэтому нужно наложить некоторые ограничения на получаемое пятно света, на угол расхождения лучей и т.п., и с учётом их оптимизировать положение линз, зеркал, их кривизну. Данный пункт довольно творческий, идеи приветствуются.
- **Освещение дыма (+5)**
 Эффект объёмного рассеивания лучей. Должен быть виден весь луч от фонарика или его часть. Дым может быть равномерным или неравномерным.
- **Сложные триангулированные модели (+5)**
 Добавить в сцену объект, состоящий из более чем 1000 треугольников. Для оптимизации трассировки использовать KD-дерево или другую ускоряющую структуру.
- **Прозрачные объекты в сцене (+3)**
 Сделать прозрачными оба объекта в сцене либо добавить аналогичные им другие прозрачные объекты. Максимальный балл – если эти объекты преломляют и свет фонарика, и лучи от камеры.

Подсказки к решению

1. **Вычисления**
 Для расчётов на CPU предлагается использовать библиотеку **glm** [3] для векторных вычислений.
2. **Генерация лучей. Прямая трассировка.**
 Предлагается реализовать случайную генерацию лучей. Выбирается случайная точка источника света и в случайном направлении бросается луч. Имеем в виду, что источник освещает только одно полупространство. Распределение направлений и точек испускания лучей должно быть равномерным (либо можно использовать диаграмму направленности от реального светодиода). Далее выпущенный луч трассируется через оптическую систему и попадает в сцену.
 Изначально у каждого луча есть некоторая яркость, которую он теряет при каждом отражении и преломлении. Во время преломления луч расщепляется на 2 новых так, что выполняется закон сохранения энергии.
3. **Генерация луча. Обратная трассировка. Модель камеры**



Камера задаётся центром проекции *CameraPos*, ортогональной тройкой $(\vec{Up}, \vec{Right}, \vec{ViewDir})$ и углами обзора по горизонтали и вертикали. Результирующее

изображение имеет разрешение $W \times H$ пикселей, и для того, чтобы не было искажения пропорций (пиксели квадратные), нужно соответствие углов обзора разрешению.

Каждый луч, проходящий через пиксель изображения с координатами (cx, cy) задаётся начальной точкой $CameraPos$ и направлением

$$\overrightarrow{ViewDir} + \left(\frac{cx + 0.5}{W} - 0.5 \right) \overrightarrow{Right} + \left(\frac{cy + 0.5}{H} - 0.5 \right) \overrightarrow{Up}$$

Для того, чтобы это направление было верно, нужно сделать длину $\overrightarrow{Up}, \overrightarrow{Right}$ равной соответствующему размеру плоскости проекции, а длину $\overrightarrow{ViewDir}$ – равной расстоянию от центра камеры до плоскости проекции. В числителях к координатам пикселя добавлено 0.5 для того, чтобы луч проходил через центр пикселя, а не его угол.

4. Пересечение луча с объектами

а. Пересечение с плоскостью

Предлагается решить задачу в параметрическом виде: луч задан начальной точкой A и направляющим вектором \vec{m} , плоскость задана точкой P и нормалью \vec{n} . Решение нужно искать как $A + \vec{m}t$, где $t \in [0, 1]$ – неизвестный параметр. Для проверки пересечения с треугольником или прямоугольником нужно дополнительно наложить условие принадлежности точки пересечения этой фигуре. Также есть более быстрые способы, например, описанные в [4].

Для пересечения луча с объектом, состоящим из многих примитивов, можно перебрать их все и среди точек пересечения выбрать ближайшую к началу луча.

б. Пересечение луча и сферы

Предлагается также прибегнуть к параметрическому заданию прямой и решить квадратное относительно t уравнение пересечения этой прямой и сферы. Уравнение даст 2 решения в случае пересечения, нужно выбрать минимальное $t > 0$. [5] Для того, чтобы узнать, пересекает ли луч поверхность сферической линзы, нужно пересечь луч с ближайшей сферой и проверить, лежит ли точка пересечения внутри второй сферы.

с. Пересечение луча и параболоида

Уравнение симметричного параболоида из этого задания $z = k(x^2 + y^2)$ в системе координат, где осью является ось Z . Предлагается перевести луч в эту систему координат и далее аналогично пересечению со сферой решить квадратное уравнение относительно параметра t . После этого проверить ограничения (параболоид не бесконечный) и найти ближайшее пересечение, если оно есть.

5. Коэффициент пропускания стекла по формулам Френеля

float FresnelReflectance(float IncidentCosine, float RI)

```
{
    float ci2 = IncidentCosine * IncidentCosine;
    float si2 = 1.0f - ci2;
    float si4 = si2 * si2;
    float a = ci2 + n * n - 1.0f;
    float sqa = 2.0f * sqrtf(a) * IncidentCosine;
    float b = ci2 + a;
    float c = ci2 * a + si4;
    float d = sqa * si2;
    return (b - sqa) / (b + sqa) * (1.0f - d / (c + d));
}
```

Здесь приведена функция, вычисляющая отражательную способность стекла с коэффициентом преломления RI для луча, падающего с направлением Incident, если нормаль в точке падения – Normal. Тогда вычисляем

$$IncidentCosine = abs(dot(Incident, Normal))$$

И передаём в функцию. Коэффициент пропускания

$$Transmittance = 1 - Reflectance$$

6. Фотонная карта (базовая часть)

Предлагается использовать карту освещённости, лежащую в плоскости пола Cornell box. Это матрица, каждая ячейка которой хранит освещённость в данной точке в виде float. Она задаётся углом Corner, векторами ориентации \overrightarrow{DirX} и \overrightarrow{DirY} , физическим размером W×H и разрешением ResX×ResY. При попадании луча в плоскость вычисляются координаты точки пересечения, переводятся в систему координат (Corner, \overrightarrow{DirX} , \overrightarrow{DirY}) и далее вычисляется нужная ячейка. В неё добавляется яркость, которую принёс луч.

Аналогично происходит выборка значений яркости из карты освещённости при визуализации с помощью обратной трассировки. При этом полученное из карты освещённости значение нужно складывать с основным освещением, полученным по модели Фонга.

7. Освещение объектов сцены

Это можно реализовать двумя способами:

1) На каждую грань параллелепипеда сделать свою карту освещённости, а на сфере сделать такую же карту по сферическим координатам (при этом не забыть, что тогда пиксели будут иметь разную площадь, нужно нормировать на неё освещённость)

2) Использовать более продвинутый метод фотонных карт, где освещение всех объектов будет получаться автоматически. В этом случае вы получаете баллы и за данный пункт.

8. Антиалиасинг

Предлагается устранять ступенчатость методом мультисэмплинга. Для этого через каждый пиксель камеры пропускается не один луч, а несколько – через различные точки этого пикселя, помимо центра. Результирующий цвет этих лучей усредняется и присваивается пикселю.

9. Освещение дыма

Предлагается 2 способа:

1) На этапе прямой трассировки при прохождении луча через дым в случайном месте этого луча испускать луч в камеру с некоторой яркостью. Яркость нужно разделить между лучами в соответствии с законом сохранения энергии. При этом надо иметь в виду, что на самом деле лучи ушли не только в камеру, но и во всё окружающее пространство. Яркость луча, попавшего в камеру, значительно ниже изначальной яркости луча, проходящего через дым.

Таким образом, ещё до этапа обратной трассировки в камере будет накоплена некоторая яркость, соответствующая дыму.

2) Сделать внутри Cornell box воксельную сетку, хранящую яркости. При прохождении луча через неё записывать яркость в воксели и затем при обратной трассировке так же собирать яркости. Это более вычислительно затратный способ, но и более универсальный.

Яркость, ушедшая в камеру, пропорциональна плотности дыма в данной точке.

Регулируя плотность дыма по какому-то закону или используя плотность из воксельной сетки, можно добиться неравномерной яркости дыма.

10. Тени

Во время обратной трассировки лучей при пересечении луча с поверхностью нужно пускать теневой луч в источник света. Если он пересечёт другой объект, значит, в данной точке тень.

Для реализации мягких теней нужно пускать несколько теневых лучей в разные части площадного источника света, и освещённость будет определяться попавшими в источник света лучами.

Ссылки

1. https://en.wikipedia.org/wiki/Cornell_box Cornell box
2. <http://ray-tracing.ru/articles165.html> Фотонные карты
3. <http://glm.g-truc.net/0.9.7/index.html> Библиотека glm

4. <http://ray-tracing.ru/articles213.html> Пересечение луча с треугольником
5. <http://ray-tracing.ru/articles245.html> Пересечение луча со сферой