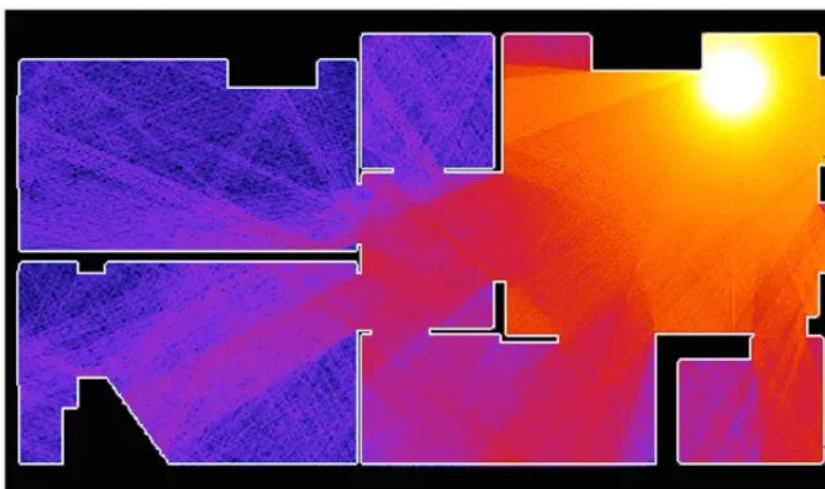


К

Задание 3. Визуализация зоны покрытия WiFi

Автор задания: Владимир Афанасьев

Цель задания – провести моделирование физической величины на воксельной сетке и реализовать визуализацию полученных значений при помощи алгоритма трассировки лучей. Предлагается визуализировать зону покрытия помещения сигналом WiFi. Необходимо реализовать метод трассировки лучей через сцену, содержащую полигональную геометрию и воксельную сетку. Предлагается простейшая модель распространения сигнала, не учитывающая волновых эффектов.



Пример реализации. База, вид сверху.

Правила оформления работы

Внимание! При невыполнении указанных требований работа может не проверяться!

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
 - Проект должен собираться из папки **src** в вашей среде разработки, но на другой машине
2. Папка **bin**, если на Windows (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.

3. Папка **img** (визуализированные изображения сцены, минимум 3 штуки для приведённых файлов настроек)

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

Как известно, WiFi сигнал является радиоизлучением на частоте 2.4 или 5 ГГц. Качество приёма сигнала устройствами зависит в первую очередь от мощности радиоизлучения в конкретной точке. Антенна передатчика излучает сигнал некоторой мощности, затем мощность падает с расстоянием, при прохождении через стены, при отражении от непрозрачных стен. Также при огибании препятствий и наложении сигналов друг на друга существуют волновые эффекты, но для простоты рассматривать здесь их не будем. Распространение радиоволн в пространстве предлагается моделировать с помощью трассировки прямолинейных лучей.

Базовая часть (до 10 баллов)

В базовой части требуется реализовать алгоритм обратной трассировки лучей в сцене с полигональными объектами (стенами) и воксельной сеткой (накапливает мощность излучения в пространстве).

На вход подаётся:

1. Модель помещения в файле в формате .obj
2. Трёхмерные координаты антенны излучателя (роутера)
3. Мощность сигнала в децибеллах
4. Положение и ориентация камеры, с которой хотим получить изображение.

Результат – изображение зоны покрытия помещения сигналом, выполненное в специальной цветовой гамме (на выбор), где цвета соответствуют мощности сигнала.

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. Один ракурс должен быть видом **сверху по центру помещения**. Как минимум один вариант настроек должен позволять получить изображение не дольше, чем за 2 минуты на средней по мощности машине.

Все размеры задаются в метрической системе.

Расчёт проходит в 3 этапа:

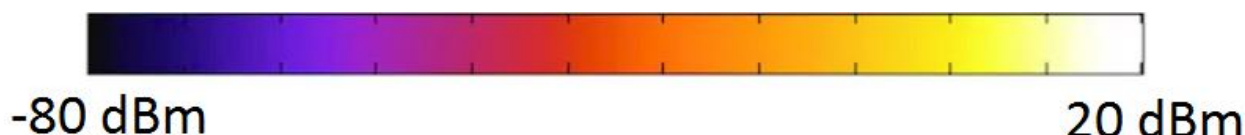
1. Строим воксельную сетку (разрешением не менее 100^3) и заполняем её значениями мощности сигнала.
Для этого нужно найти описывающий прямоугольный параллелепипед для заданной полигональной модели стен. Можно сделать параллелепипед со сторонами, параллельными осям координат.
Далее с помощью метода прямой трассировки лучей от антенны (считаем её точечным источником лучей) заполняем воксельную сетку значениями мощности сигнала. Лучи следует распределять от антенны в случайных направлениях. У

каждого луча есть стартовая мощность, равная мощности антенны. При прохождении луча через воздух его мощность должна убывать с квадратичной зависимостью от расстояния. Обратите внимание, что тут речь идёт о линейной шкале мощности, а не логарифмической, где она измеряется в децибеллах! В помещении есть 2 типа стен: полупрозрачные (пропускают лучи, отнимая фиксированное значение от мощности) и непрозрачные (отражают лучи, также уменьшая мощность). Для простоты, можно считать, что стены имеют нулевую толщину.

Мощность, которую несёт каждый луч, должна записываться в воксели на его пути. Для этого предлагается применить метод Ray marching: идём с некоторым фиксированным шагом вдоль луча и каждый раз вычисляем текущие координаты вокселя, в который попала новая точка. В каждом вокселе следует действовать по правилу: более мощный сигнал замещает более слабый.

2. Сглаживаем воксельную карту. Это необходимо для того, чтобы убрать шумы в сетке. В базовой части можно сделать сглаживание Vox-фильтром.
3. Визуализируем модель помещения и полученную воксельную сетку с помощью обратной трассировки лучей из камеры. Для этого в каждом пикселе камеры генерируем луч. Его направление определяется ориентацией камеры, её пирамидой видимости и положением пикселя на матрице. Каждому лучу присваивается цвет RGB (изначально возьмём его равным 0), который затем запишется в пиксель. Далее трассируем луч через сцену до его пересечения с первым полигональным объектом. Цвет этого объекта, рассчитанный по модели Фонга, присваивается лучу. Антенна должна быть представлена каким-нибудь объектом, например, кубом или сферой.

После этого с помощью того же метода Ray marching проходим вдоль луча от пересечения с объектом до камеры. В каждом очередном вокселе нужно взять значение мощности сигнала, найти соответствующий ему цвет из цветовой схемы и произвести альфа-смешивание между текущим цветом луча и полученным цветом. Так как эта операция будет выполняться много раз вдоль луча, коэффициент непрозрачности α у воксельной сетки должен быть достаточно мал, иначе сетка будет выглядеть абсолютно непрозрачной.



Пример цветовой схемы

Результат работы трассировщика должен сохраняться во внешний файл формата .bmp или .png рядом с исполняемым файлом. Просьба не использовать форматы, сжимающие с потерями.

Дополнительная часть (до 23 баллов)

Внимание! Все реализованные пункты дополнительной части должны конфигурироваться через внешний текстовый 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.

- **Текстуры стен и пола (+2)**
Нужно добавить в сцену не менее 2 других объектов, состоящих из примитивов, использованных в базе – это плоскости или сферы. Объекты могут быть покрашены сплошным цветом, либо быть зеркальными, либо можно реализовать на них модель освещения (например, Фонга).
- **Сложные полигональные объекты – мебель (+3)**
Нужно добавить не менее 2 дополнительных объектов из не менее чем 1000

треугольников. Как реализовывать пересечение луча с такими объектами, описано ниже. Аналогично стенам и полу, эти объекты могут быть сплошного цвета или текстурированы, освещены по модели Фонга.

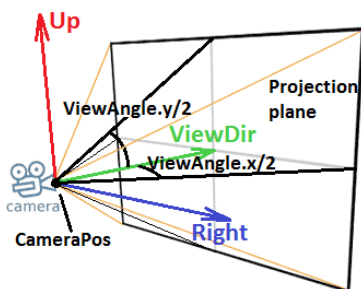
- **Пользовательский интерфейс (+3)**
Интерфейс должен позволять удобно менять некоторые настройки, например, положение антенны, параметры камеры. Удобство оценивается субъективно.
- **Параллелизм (+5)**
 - +1 за простую реализацию на OpenMP, стандартных потоках и т.п
 - +3 за Optix, CUDA, OpenCL или шейдеры OpenGL и +2 за анимацию движения передающей антенны
- **Антиалиасинг (+1)**
Предлагается реализовать с помощью метода мультисэмплинга. Через разные точки пикселя бросаются несколько лучей, их цвета усредняются.
- **Полупрозрачные полигональные объекты (+1)**
Окна. Моделируются альфа-смешиванием.
- **Более сложные фильтры сглаживания воксельной сетки (+1)**
- **Постобработка результата (+1)**
Любые фильтры, применённые к изображению, позволяющие улучшить его визуальные качества.
- **Определение наиболее выгодной точки для антенны (+6)**
Нужно оценить мощность сигнала во всей воксельной сетке (например, доля вокселей, в которых сигнал не слабее некоторой величины) и затем применить один из методов глобальной оптимизации на ваш выбор. Максимальный балл можно получить, визуализировав процесс оптимизации – интерактивно или как набор изображений, сохранённых на диск.

Подсказки к решению

1. Вычисления

Для расчётов на CPU предлагается использовать библиотеку **glm [1]** для векторных вычислений. Рекомендуется использовать точность вычислений float32.

2. Генерация луча. 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

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

Для того, чтобы это направление было верно, нужно сделать длину \vec{Up}, \vec{Right} равной соответствующему размеру плоскости проекции, а длину $\vec{ViewDir}$ – 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.

3. Пересечение луча с объектами

Луч света в предложенном подходе состоит из набора отрезков, проверка пересечений выполняется для каждого нового отрезка.

а. Пересечение отрезка и плоскости

Предлагается решить задачу в параметрическом виде: направленный отрезок \overrightarrow{AB} задан начальной точкой A и направляющим вектором $\vec{m} = B - A$; плоскость задана точкой P и нормалью \vec{n} . Решение нужно искать как $A + \vec{m}t$, где $t \in [0,1]$ – неизвестный параметр. В случае выбора глобальной системы координат так, что плоскость лежит в плоскости XY , решение упрощается.

б. Пересечение отрезка 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 . [2]

с. Пересечение отрезка и треугольника для доп. задания с полигональными объектами.

Здесь необходимо пересечь отрезок с плоскостью треугольника и провести дополнительные тесты на то, что точка лежит внутри треугольника. Это можно сделать как напрямую, так и другими способами [3]. Для пересечения луча с объектом, состоящим из многих треугольников, можно перебрать все треугольники и среди точек пересечения выбрать ближайшую к началу луча. Для ускорения поиска можно использовать KD- или BVH-деревья.

4. Антиалиасинг

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

5. Альфа-смешивание

Смешиваем два цвета: RGB_0 – цвет объекта с коэффициентом непрозрачности $\alpha \in [0,1]$ и RGB_1 – цвет исходного луча. Результирующий цвет прошедшего дальше луча будет равен $RGB = \alpha RGB_0 + (1 - \alpha) RGB_1$, где α – значение альфа-канала, нормированное в отрезок.

Ссылки

1. <https://github.com/g-truc/glm/releases/tag/0.9.8.5> Библиотека glm
2. <http://ray-tracing.ru/articles245.html> Пересечение луча со сферой
3. <http://ray-tracing.ru/articles213.html> Пересечение луча с треугольником