Задание 4. Трассировка лучей

Автор задания: Груздев Алексей

Цель задания - реализовать визуализацию сцены с полигональной геометрией при помощи алгоритма трассировки лучей. Предлагается визуализировать сцену интерьера, например, гостинную, спальню и т.п. В задании требуется использовать сложные геометрические модели с разнообразными типами материалов. Необходимо реализовать эффективный метод трассировки луча в сцене за счёт использования описывающих объёмов. Также требуется обеспечить эффективное хранение высокополигональных моделей в памяти, не допуская лишнего копирования. В качестве дополнительной части предлагаются варианты по улучшению скорости и качества синтеза, а также постэффектов.

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:

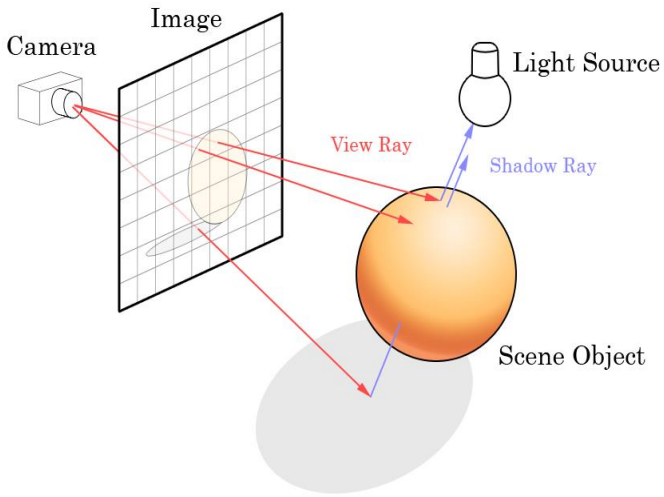
1. Папка **src** (исходный код)
   - Файлы исходного кода
   - Файлы проекта
   - **НЕ нужно** включать в архив папку **ipch,** базы данных программы **.ncb, .sdf**.
   - Проект должен собираться из папки src
2. Папка **bin** (исполняемый код - конфигурация Release, 32 бит). Обязательно проверьте, что 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. 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

Базовая часть (5 баллов)

Реализовать алгоритм трассировки лучей в сцене с полигональными объектами. Требуется создать сцену с не менее чем 10 объектами (более 200 треугольников каждый). Использование простых геометрических объектов (сферы, кубы) не засчитываются. Они могут быть использованы как вспомогательные: пол в виде плоскости, яркая сфера на месте точечного источника света и т.п. Разрешение результирующего цветного изображения должно быть не менее 512x512 пикселей.

1) Модели должны быть загружены из внешних файлов (.3ds, .obj и пр.) Допускается генерация моделей в коде, но дожны использоваться математические формулы. Чтение из массива “захардкоденных” вершин не разрешается. У каждой модели должны быть корректные **гладкие** нормали.

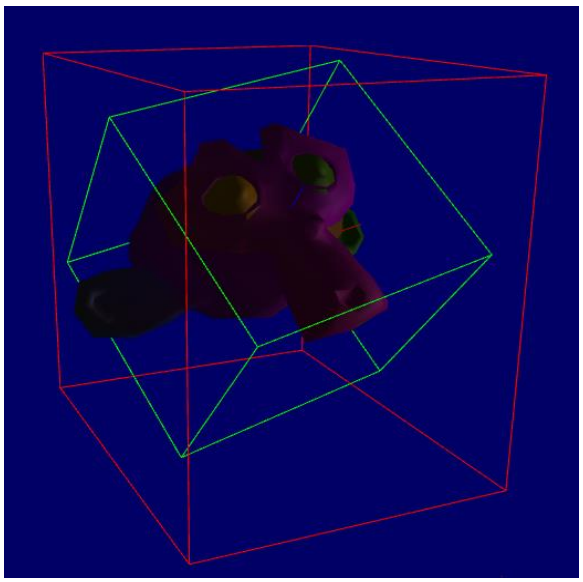
2) Необходимо реализовать простейший вариант камеры.



Начало луча выбирается в позиции камеры, а направление выбирается для каждого пикселя, как бы проходя через текущий пиксель через виртуальную картинную плоскость перед камерой. Расстояние до виртуальной плоскости определяется выбранным углом обзора камеры (рекомендуется брать между 45 и 60 градусами, если не уверены с чего начать). Камера не должна быть наклонена на бок.

3) Трассировка лучей должна быть оптимизирована с помощью проверки пересечения с описывающими объёмами. Не допускается поиск пересечения сразу со всеми треугольниками сцены. Каждый объект должен быть помещён в описывающую сферу или параллелепипед(Axis aligned bounding box). Сначала ищется пересечение луча со всеми описывающими объёмами в сцене. Затем среди всех найденных выбирается наиболее близкий и рассчитывается пересечение луча со всеми полигонами объекта заключённого в выбранный объём. Если пересечение не найдено, то выбирается следующий по глубине объект. Обратите внимание, что красный параллелепипед со сторонами, параллельными осям координат (Axis aligned bounding box) предпочтительнее, так как с ним проще искать пересечение луча.

Для простоты можно размещать объекты на расстоянии друг от друга так, чтобы описывающие объёмы не пересекались.



4) Требуется реализовать инстанциирование. Т.е. в сцене должно быть несколько объектов, использующих общий массив треугольников, а не собственные копии. Так как треугольники хранятся в локальных координатах, а луч должен пересекаться с объектами в мировых координатах, то необходимо переводить луч в локальные координаты каждого объекта, вместо того чтобы преобразовывать каждый треугольник.
   Пусть $T = Translate * Rotate * Scale$ - модельное преобразование локальных координат объекта. Луч в мировых координатах задан парой позиции и направления: $Ray_{world} = (Origin, Direction)$. Чтобы получить луч в локальных координатах, необходимо воспользоваться матрицей обратной к матрице преобразования. Причём домножить направление луча надо на матрицу без учёта переноса, т.е.:
   $T_{dir}^{-1} = (Rotate * Scale)^{-1}$;
   $Ray_{local} = \left(T^{-1} * Origin, T_{dir}^{-1} * Direction\right)$.
   К объектам должны быть применены различные преобразования. Требуется, чтобы было как минимум 2 повёрнутых и отмасштабированных объекта. Объекты не должны быть слишком маленькими, то есть они должны быть различимы в сцене на глаз.

5) Обекты должны иметь как минимум 3 разных материала. Например, полностью диффузный, отражающий, бликующий, преломляющий. Также допускается

использование разных моделей освещения: Фонг, Ламберт, Кук-Торранс. Можно использовать табличную ДФО. (http://steps3d.narod.ru/tutorials/lighting-tutorial.html). В сцене должен присутствовать хотя бы один источник света. Все источники света могут быть точечными любого цвета.

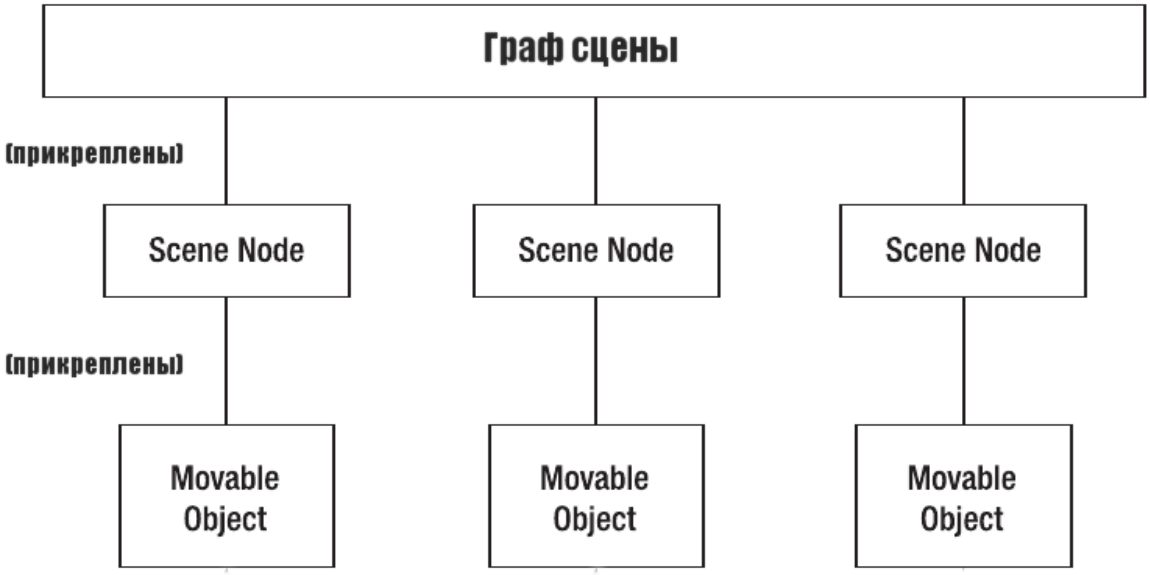
6) Результат работы трассировщика должен сохраняться во внешний файл формата .bmp или .png.

Дополнительная часть (10 баллов)

Внимание! Выполнение любого пункта дополнительной части не заменяет любой пункт базовой части! Если реализована трассировка путей, это не значит что не надо делать трассировку лучей. Отрисовка в реальном времени не заменяет сохранение в файл. Поэтому все реализованные пункты дополнительной части должны конфигурироваться через внешний текстовый (не бинарный!) файл настроек. У проверяющего должна быть возможность легко модифицировать файл, так чтобы в программе выполнялась только база. В ридми должно быть пояснение формата файла, как с ним работать.

- **Текстурированные объекты (1-2.5 балла)**
  Для объекта должны быть корректно заданы текстурные координаты и аккуратно проинтерполированы внутри треугольников
  Обратите внимание на необходимость реализации как минимум билинейной интерполяции при выборке из текстуры. При полном отсутствии фильтрации баллы за текстурирование могут быть снижены на 0.5.
  Билинейная - 1 балл
  Трилинейная - 1.5 балла
  Анизатропная - 2.5 балла

- **Реализация графа сцены (сценграфа) (2 балла)**



Позиционирование объектов в сцене происходит с помощью прикрепления объектов к узлам дерева. Для каждой вершины могут быть заданы локальные преобразования (сдвиг, масштаб, поворот) относительно положения родительского узла. Перед отрисовкой для каждой вершины высчитывается полное преобразование, с учётом всех родительских узлов. Каждый объект использует полное преобразование того узла, к которому он прикреплён. К одной вершине графа может быть прикреплено любое количество объектов.
В базе достаточно хранить все объекты в простом контейнере и задавать преобразования для каждого объекта отдельно. При наличие сценграфа преобразования задаются не для объектов, а только для вершин графа.
В сцене должен явно использоваться сценграф. Например может быть задан составной объект, состоящий из отдельный деталей. (Дерево, робот, автомобиль и.т.д.)

- **Ускоряющие структуры (1-3 балла)**
  Использование kd-дерева, BVH и других ускоряющих структур для ускорения трассировки лучей. kd-дерево - 1-2 балла (в зависимости от алгоритма выбора плоскости разбиения; BVH - 2-3 в зависимости от эффективности описывающего объёма (хороший описывающий объём - точно охватывающий геометрию и достаточно простой для поиска пересечения)

  (http://www.ray-tracing.ru/articles181.html, http://www.ray-tracing.ru/articles184.html)
- **Моделирование глубины резкости (1.5 балл)**
  Реализация через модель камеры с линзой. Изображение должно иметь область резкости и нечёткие области ближе и дальше фокуса, а не быть целиком размытым. Постобработка в виде размытия в зависимости от глубины не будет засчитана. (см.

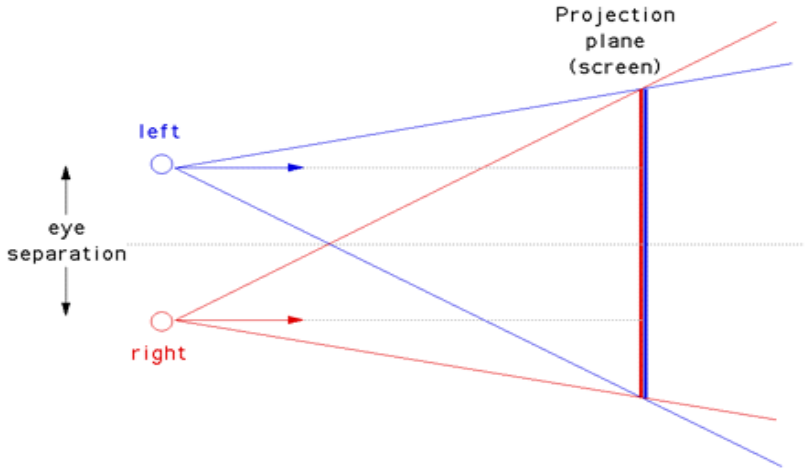
пункт постэффектов)

- **Ambient Occlusion (2 балла)**
  http://www.ray-tracing.ru/articles232.html

- **Трассировка путей (3-4 балла)**
  Альтернативный алгоритм синтеза изображения вместо трассировки лучей. Не заменяет базовую часть, должна быть возможность переключения на трассировку лучей из базы.
  http://www.ray-tracing.ru/articles216.html

  - **HDR панорамы для моделирования фонового освещения (+1 балл) в дополнение к Ambient Occlusion и/или Path tracing**
    Использование куб-мапа или сферы вокруг сцены с натянутой текстурой панорамы. Панорама считается бесконечно удалённой и, при выходе луча за пределы сцены, значение освещённости выбирается из панорамы в соответствии с направлением луча.

- **Параллелизм вычислений (1-3 балла)**
  Использование OpenMP, pthreads, TBB, C++11 threads для параллельной генерации изображения. +1 балл
  Визуализация постепенного вычисления по блокам изображения в реальном времени +2 балла



- **Генерация стереопары (1-2 балл)**
  Рендеринг пары изображений из двух виртуальных камер. Для камер должна быть реализованна корректная генерация лучей
  Позиции двух камер должны быть смещены, а картинная плоскость совпадать. У таких камер используется искажённая пирамида видимости, а не симметричная, как у простой моно камеры. - 1 балл
  Если изображения сохранены в анаглифическое изображение или в виде gif анимации, то 2 балла.

- **Антиалиасинг (1 балл)**
  Например, выборка в пикселе нескольких точек и трассировка луча для каждой (multisampling).

- **Тени (1-2 балла)**
  Резкие тени - 1 балл
  Мягкие тени - 2 балла

- **Затухание света (1 балл)**
  Интенсивность освещения от источника света должна убывать с расстоянием.

- **Несколько источников света (1 балл)**

- **Объёмные источники света (2 балла)**
  Использование источников света конечного размера. При расчёте затенения надо выбирать случайные точки на поверхности источника света

- **Bump-mapping (2 балла)**
  Должны быть корректно вычисленные tangent-space координаты для выборки из карты нормалей, иначе не будет зачтено

- **Parallax-mapping (2 балла)**
  Техника имитации рельефа смещением текстурных координат.

- **Реализация эффекта поглощения (1-2 балла)**
  Поглощение света в сплошной однородной среде описывается законом Бугера-Ламберта-Бера (экспоненциальное затухание). Для реализации эффекта поглощения в преломляющем материале нужно для каждого луча учитывать длину участков пути, пройденных ими в сплошной среде, и домножать интенсивность лучей, прошедших через объект, на рассчитанный коэффициент пропускания.
  Примерная оценка длинны пути внутри объекта - 1 балл
  Точная оценка длинны пути - 2 балла

- **Прогрессивная трассировка и графический интерфейс (4 балла)**
  Необходимо реализовать визуализацию картинки в окне (Qt, OpenCV и т.п.) На каждом кадре рассчитывать небольшое количество лучей (чтобы было как можно быстее) и добавлять к готовому изображению по реккурентной формуле : $Sum_n = \frac{n-1}{n} * S_{n-1} + \frac{1}{n} * x_n$ Рекомендуется добавить алгоритм фильтрации для результата каждого кадра перед добавлением в буфер.

- **Постобработка (1-3 балла)**
  Различные варианты постобработки готового изображения (1-2 балла) если распараллелено, то +1 балл (к общей сумме). Если реализовано несколько, то не более 3 баллов в сумме.

- Фильтрация шума (Гаусс, билатеральная) +1 балл. Если используется сгенерированная по сцене карта глубины или нормалей для сохранения границ при размытии, то ещё +1 балл.
- Эффект Bloom (+1 балл)
- Эффект GodRays (+1-2 балла) Рекомендуется использовать, когда источник попадает в видимость камеры. Тогда следует нарисовать на его позиции небольшой яркий объект перед применением эффекта GodRays (2 балла), иначе у лучей не будет явного источника.(1 балл)
- Глубина резкости как размытие с использованием карты глубины (+1 балл)
- Эффект неона (Выделение границ оператором Собеля + их подсветка и наложение сверху) +2 балла
- Другие эффекты на свой вкус (1-2 балла в зависимости от сложности)

Дополнительная библиотека

Чтобы упростить выполнение задания, а именно работу с геометрией, расчёты трансформаций и пересечений, к заданию прилагается дополнительная библиотека математики (взята из движка OGRE и немного упрощена). В ней реализованы операции над векторами размерности 2, 3,4, матрицы 3x3, 4x4, кватернионы. Реализован луч, плоскость, сфера, AxisAlignedBox. Использование библиотеки не обязательно, можно всё реализовать самостоятельно.
Ссылка: CommonMath

Полезная литература

Архив моделей: http://archive3d.net/?page=1

http://www.ray-tracing.ru/

http://steps3d.narod.ru/snippets.html

http://www.gamedev.ru/

http://en.cppreference.com/w/