

Задание 1: Трассировка лучей Whitted-a

Владимир Фролов, Юрий Фетисов

21 февраля 2020 г.

Аннотация

Цель задания: закрепить на практике основы трёхмерной математики и базовые понятия компьютерной графики.

- Изучение основ построения изображений;
- Изучение алгоритма трассировки лучей.
- (Дополнительно) Изучение алгоритма трассировки путей.

1 Базовая часть: 15 баллов

Необходимо реализовать базовый алгоритм обратной трассировки лучей Whitted-a и визуализировать им некоторую сцену. При невыполнении хотя бы одного требования из базовой части, проверяющий имеет право уполовинить баллы.

- Необходимо реализовать локальное освещение по модели фона или другим аналогичным моделям.
- Необходимо реализовать тени.
- Необходимо реализовать зеркальные отражения.
- Необходимо использовать минимум 3 различных материала.
- Необходимо использовать минимум 2 различных примитива (например, треугольник и сфера).
- В сцене должен быть хотя бы 1 источник света.
- Рендеринг одного изображения не должен занимать больше 1 секунды для трассировки лучей (для процессора уровня 6-ядерного AMD Ryzen 5 3600 или GPU уровня Nvidia GTX1070).

- Рендеринг одного изображения не должен занимать больше 1 минуты для трассировки путей (для процессора уровня 6-ядерного AMD Ryzen 5 3600 или GPU уровня Nvidia GTX1070).
- Минимально допустимое разрешение выходного изображения – 512x512.

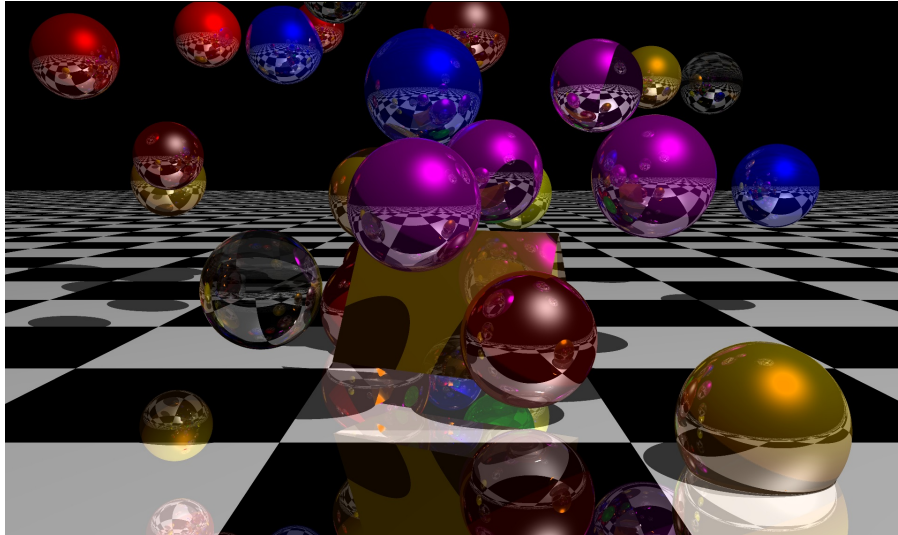


Рис. 1: Пример выполненного задания (трассировка лучей). Итоговая оценка – 20 баллов: 15 база, +1 за примитив бесконечной плоскости, +1 за использование текстуры, +1 за преломления, +1 за приятный внешний вид, +1 за устранение ступенчатости.

2 Дополнительная часть: до 30 баллов

Функциональные элементы:

- Использование текстур (+1).
- Субъективная реалистичность сцены (от +1 до +2).
- Произвольные (на ваше усмотрение) 3D модели в виде треугольных (или других) мешей (от +4). Необходимо, чтобы модель содержала как минимум несколько тысяч треугольников (или других примитивов).
- Использование поверхностей второго порядка – сплайны, NURBS и.т.д. (+4).
- Устранение ступенчатости (+1).
- Использование дополнительных геометрических примитивов. (+1) за каждый примитив, (+2) максимум.

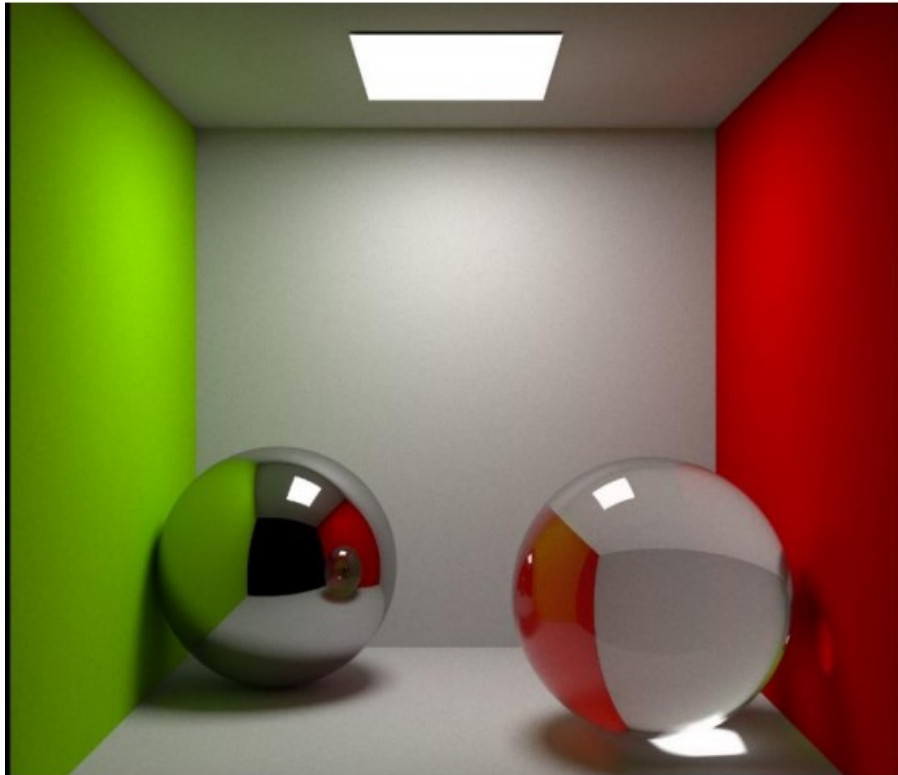


Рис. 2: Пример выполненного задания (трассировка путей). Итоговая оценка – 25 баллов: +5 за трассировку путей, +2 за стекло. +1 за приятный внешний вид, +1 за устранение ступенчатости, +1 за использование гамма-коррекции.

- Фрактальные поверхности (+2). Здесь предполагается использование т.н. Distance Aided Ray Marching.
- Использование карт окружения в виде сферической или кубической наноразмы (+1).
- Непредусмотренный бонус на усмотрение проверяющего.

Трассировка путей:

- Базовый алгоритм трассировки путей: (+5).
- Использование гамма-коррекции (+1). Помните, что для обыкновенных текстур в этом случае нужно использовать гамма-преобразование, чтобы перевести их значения в линейное пространство!
- Источник света в виде меша (сетки) (+1).
- Материал с глянцевым отражением при помощи «Исправленной Модели Фонга» (+3). В данном пункте необходимо использовать механизм выборки по значимости для генерации луча вокруг направления отражения. В противном случае будет засчитан лишь 1 балл.

- Использование других известных моделей материалов (+1 за простое использование модели, +3 за использование с выборкой по значимости). Разрешается реализовать до 2 дополнительных моделей (за большее число моделей баллы не начисляются).
- Реализация подповерхностного рассеивания (+4).
- Моделирование спектрального разложения света: (+3).
- Непредусмотренный бонус проверяющего при реализации некоторой нетривиальной техники вроде фотонных карт (до +2).

Скорость:

- Использование SIMD инструкций (+2). Аргумент командной строки: -simd 1.
- Использование многопоточности (+2). Аргумент командной строки: -threads 10.
- Реализация на GPU: (+5).

3 Порядок сдачи

Ваша программа будет запускаться в полу-автоматической проверочной системе. Поэтому запуск Вашей программы должен выглядеть следующим образом:

```
./rt -out <output_path> -scene <scene_number> -threads <threads>
```

- output_path - путь к выходному изображению (относительный).
- scene_number - номер сцены от 1 до 3. Ваша программа должна создавать до 3 сцен по выбору, на которых по Вашему усмотрению распределены все сделанные Вами дополнительные задания. Можно все допы показать на одной сцене.

Пример командной строки, используемой автоматической системой проверки:

```
./rt -out 311_ivan_yellow.bmp -scene 2 -threads 1
```

Внимание! Если вы реализуете меньше чем 3 сцены, при получении номера неподдерживаемой сцены программа должна завершаться с кодом успешного завершения (return 0), не сохраняя изображение.

Порядок компиляции:

```
mkdir build
cd build
cmake -DCMAKE_BUILD_TYPE=Release ..
make -j 4
```

Использование библиотек:

Все используемые библиотеки должны быть либо собраны в статические библиотеки для архитектуры x64 с поддержкой AVX (-mavx) под gcc, либо должны поставляться с проектом в виде исходных кодов, и фактически быть его частью.

4 Именованние архива

Архив, который вы загружаете на сайт должен называться по следующему шаблону: <номер_группы>_<фамилия>.zip. Фамилию следует писать латинскими символами.

5 Полезные ссылки

- [1]Scratchapixel 2.0. An Overview of the Ray-Tracing Rendering Technique. <https://www.scratchapixel.com/lessons/3d-basic-rendering/ray-tracing-overview/light-transport-ray-t>
- [2] http://mathinfo.univ-reims.fr/IMG/pdf/Using_the_modified_Phong_reflectance_model_for_Physically_based_rendering_-_Lafortune.pdf
- [3] Игнатенко А. В., Фролов В. А. Основы синтеза фотореалистичных изображений. Электронная публикация книги. <http://ray-tracing.ru/upload/free/ImageSyntBook.pdf>
- [4] Matt Pharr, Wenzel Jakob, and Greg Humphreys. 2016. Physically Based Rendering: From Theory to Implementation (3-rd Ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.