

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Институт №8 «Компьютерные науки и прикладная математика»
Кафедра №806 «Вычислительная математика и программирование»

Курсовая работа
по курсу «Параллельная обработка данных»

Обратная трассировка лучей (Ray Tracing) на GPU

Выполнил: О. А. Мезенин
Группа: М8О-406Б-21
Преподаватель: А. Ю. Морозов

Москва, 2025

Условие

Цель работы.

Использование GPU для создание фотореалистической визуализации. Рендеринг полужеркальных и полупрозрачных правильных геометрических тел. Получение эффекта бесконечности. Создание анимации.

Задание.

Сцена. Прямоугольная текстурированная поверхность (пол), над которой расположены три платоновых тела. Сверху находятся несколько источников света. На каждом ребре многогранника располагается определенное количество точечных источников света. Грани тел обладают зеркальным и прозрачным эффектом. За счет многократного переотражения лучей внутри тела, возникает эффект бесконечности.

Камера. Камера выполняет облет сцены согласно определенным законам.

Требуется реализовать алгоритм обратной трассировки лучей с использованием технологии CUDA. Выполнить покадровый рендеринг сцены. Для устранения эффекта «зубчатости», выполнить сглаживание (например с помощью алгоритма SSAA). Полученный набор кадров склеить в анимацию любым доступным программным обеспечением. Подобрать параметры сцены, камеры и освещения таким образом, чтобы получить наиболее красочный результат. Провести сравнение производительности gpu и cpu (т.е. дополнительно нужно реализовать алгоритм без использования CUDA).

Вариант 10.

На сцене должны располагаться три тела: Октаэдр, Додекаэдр, Икосаэдр.

Программное и аппаратное обеспечение

Характеристики графического процессора:

- Compute capability: 8.9
- Name: NVIDIA GeForce RTX 4070 SUPER
- Total Global Memory: 12584550400
- Shared memory per block: 49152
- Registers per block: 65536
- Warp size: 32
- Max threads per block: (1024, 1024, 64)
- Max block: (2147483647, 65535, 65535)
- Total constant memory: 65536
- Multiprocessors count: 56

Характеристики центрального процессора:

- Version: AMD Ryzen 7 7700
- Max Speed: 5389 MHz
- Core Count: 8
- Thread Count: 16

Характеристики оперативной памяти:

- Size: 2x16 GB

- Type: DDR5
- Speed: 4800 MT/s

Характеристики жесткого диска:

- Device Model: ADATA SU635
- User Capacity: 480 103 981 056 bytes [480 GB]
- Sector Size: 512 bytes logical/physical
- SATA Version is: SATA 3.2, 6.0 Gb/s (current: 6.0 Gb/s)

Программное обеспечение:

- OS: Linux
- Kernel: 6.10.13-3-MANJARO
- Текстовый редактор: Kate
- Компилятор: nvcc V12.4.131

Метод решения

Трассировка лучей — один из наиболее популярных и мощных методов для рендеринга изображений в компьютерной графике. Метод основан на моделировании поведения света в виртуальной среде, что позволяет достичь высокой степени реализма.

Обратная трассировка лучей

Обратная трассировка лучей — это метод, используемый для определения цвета каждого пикселя на экране. Вместо того чтобы отслеживать путь каждого луча света от источников до камеры, как это делают методы прямой трассировки, обратная трассировка начинается с камеры и "прослеживает" путь лучей до света. Этот подход значительно сокращает количество вычислений, поскольку хорошо подходит для обнаружения видимых объектов и вычисления их цвета.

Обратная трассировка лучей вызывает луч из камеры через каждый пиксель изображения, и, если этот луч пересекает объект в сцене, вычисляется освещение и цвет в точке пересечения. Это делается при помощи учета различных характеристик материала и источников света в сцене.

Рекурсивная трассировка лучей

Рекурсивная трассировка лучей — это расширение метода обратной трассировки, позволяющее добиться дополнительных эффектов, таких как отражения и преломления. После попадания луча в объект создаются новые "вторичные" лучи, которые либо отражаются, либо преломляются в зависимости от свойств поверхности объекта.

Рекурсия позволяет моделировать сложные взаимодействия между объектами, такие как множественные отражения, прозрачность и другие физические свойства света. Каждое рекурсивное посчитанное значение добавляется к текущему цвету пикселя, что позволяет получить реалистичное изображение.

Затенение по Фонгу

Затенение по Фонгу — это одна из моделей освещения, используемых для вычисления цвета поверхности в точках пересечения лучей с объектами. Эта модель состоит из трех компонентов: диффузного, отраженного и окружающего (ambient) освещения.

1. Диффузное освещение моделирует рассеивание света на шероховатой поверхности. Это основной компонент, который зависит от угла падения света.
2. Отраженное освещение (спекулярное) отвечает за блеск поверхности и формируется благодаря отражениям интенсивного света, находящегося под острым углом к наблюдателю.
3. Окружающее освещение моделирует равномерный фоновый свет, который присутствует даже при отсутствии прямых источников света.

Используя нормали поверхности, направляющий вектор света и расположение наблюдателя, модель Фонга позволяет получить плавные переходы и реалистичное воспроизведение световых эффектов.

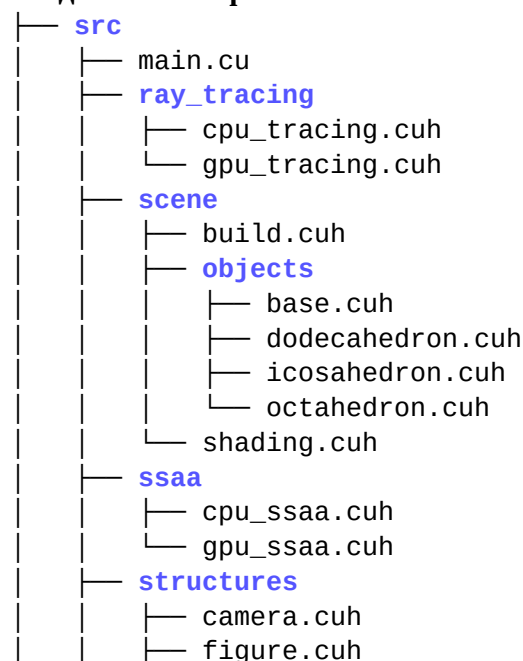
Сглаживание с помощью алгоритма SSAA

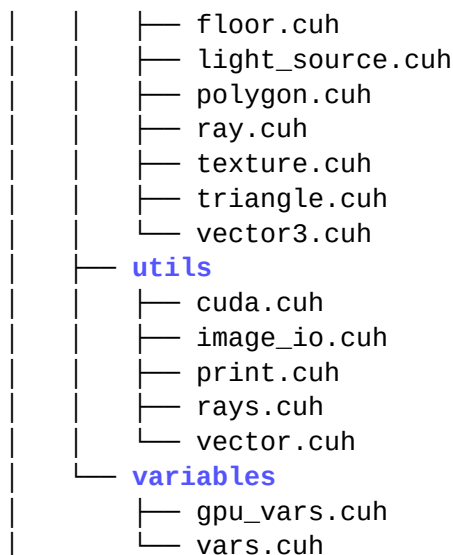
Super Sampling Anti-Aliasing (SSAA) — это метод сглаживания, который используется для уменьшения эффекта "лесенки" на границах объектов в изображении. Алгоритм работает путем отслеживания нескольких подлучей для каждого пикселя, каждый из которых слегка смещен относительно центра пикселя. Эти подлучи обрабатываются независимо и затем усредняются для получения окончательного цвета пикселя.

Подход SSAA делает края объектов более гладкими и натуральными, но при этом увеличивает количество вычислений, необходимых для рендеринга, поскольку для каждого пикселя используются несколько выборок.

Описание программы

Разделение по файлам





Файл `main.cu` считывает данные и запускает цикл по созданию кадров.

В папке `ray_tracing` находятся реализации алгоритмов трассировки лучей для CPU и GPU.

Папка `scene` содержит логику для построения сцены: `build.cuh` строит пол и платоновы тела; `shading.cuh` содержит реализацию затенения по Фонгу; а папка `objects` содержит объекты платоновых тел.

В папке `ssaa` находятся реализации алгоритма SSAA для CPU и GPU.

В папке `structures` находятся различные структуры и методы к ним: `camera.cuh` содержит параметры для камеры; `figure.cuh` содержит параметры тела (центр, цвет, радиус сферы и т.д.); `floor.cuh` содержит параметры пола (точки, текстуру и т.д.); `light_source.cuh` содержит позицию и цвет источника света; `polygon.cuh` содержит структуру полигона (координаты, цвет и т.д.) и реализует метод вычисления цвета; `ray.cuh` содержит структуру луча (позицию, направлению и т.д.); файл `texture.cuh` содержит структуру текстуры и реализует метод загрузки самой текстуры и метод получения цвета в точке; `triangle.cuh` содержит структуру треугольника (координаты, нормаль, края) и реализует методы сдвига координат и нормализации; `vector3.cuh` содержит шаблонную структуру вектора из трех составляющих и реализует множество разных методов (например, векторное и скалярное умножения).

Папка `utils` содержит несколько функций, разбитых на папки: `cuda.cuh` содержит макрос для ловли ошибок и таймер для замера времени; `image_io.cuh` содержит функции для считывания и записи изображений в формате, описанном в лабораторной работе 2; `print.cuh` содержит функцию для вывода кадра, времени и количества лучей; `rays.cuh` содержит функции для вычисления пересечения луча и полигона; `vector.cuh` содержит функции для работы с `vector3`.

Папка `variables` содержит файлы с различными макросами и константами.

Описание ядер

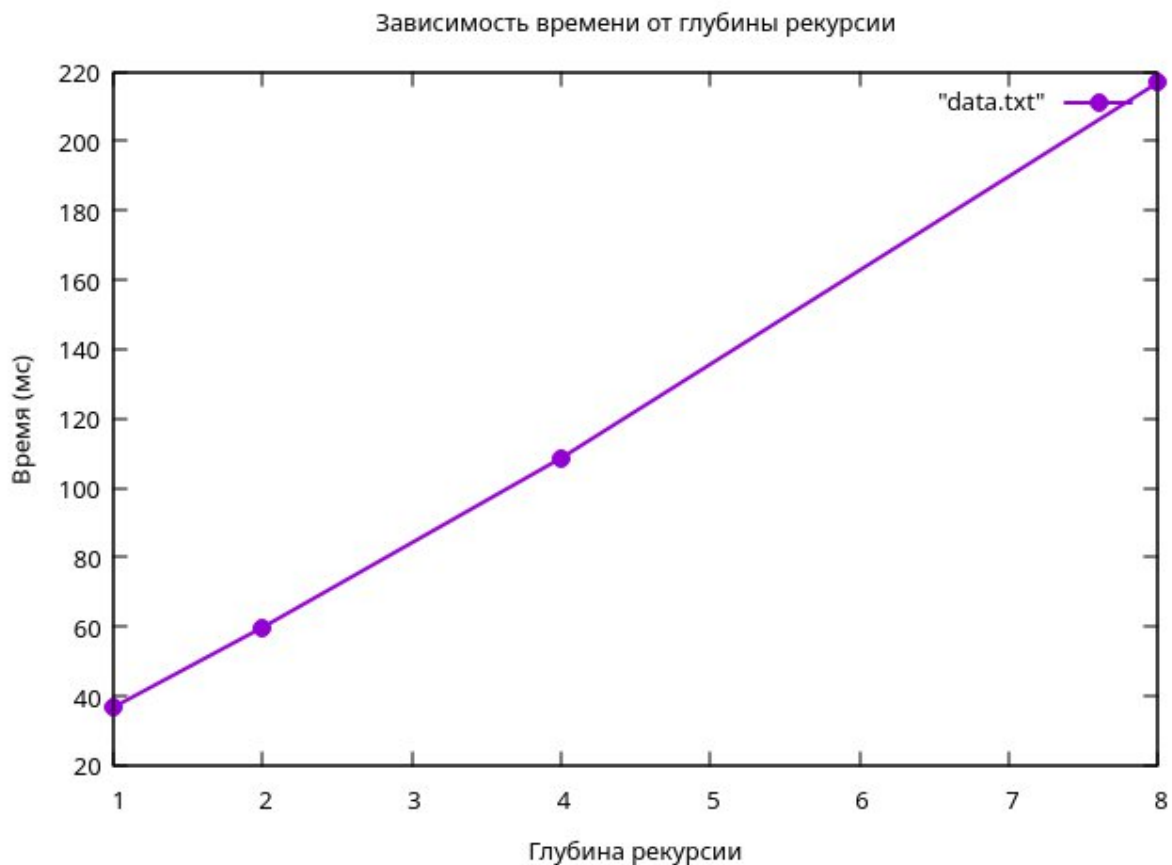
Трассировка лучей реализована в ядре `gpuTrace`, в котором каждый луч пересекается с каждым полигоном на сцене. Также есть вспомогательные ядра: `gpuClearData`, `gpuInitRays`, `gpuWriteData`.

Исследовательская часть и результаты

Таблица замеров времени для различных конфигураций, в том числе на CPU, и для различной глубины рекурсии. Разрешение кадра — 640x480 пикселей. Время в миллисекундах.

Конфигурация	Глубина рекурсии			
	1	2	4	8
<<<1, 32>>>	3738.984	6125.233	11729.407	24115.496
<<<32, 32>>>	126.727	212.541	387.929	791.759
<<<256, 32>>>	34.499	58.535	111.653	226.201
<<<256, 256>>>	36.731	59.733	108.560	216.931
CPU	933.004	1266.042	2295.060	4852.641

Можно заметить, что зависимость времени от глубины рекурсии для одной конфигурации линейная. Например, построим график для конфигурации <<<256, 256>>>.



Входные данные, на которых получается наиболее красочный результат:

```
200
./output/%d.data
640 480 120
```

```

4.5 0.0 0.0 1.0 0.5 1.0 1.0 1.0 0.0 0.0
2.0 2.0 -3.0 0.5 0.1 1.0 1.0 1.0 0.0 0.0

2.0 0.0 0.25 1.0 0.0 0.0 1.5 0.65 0.25 10
-1.41 -1.41 0.0 0.0 1.0 0.0 1.25 0.25 0.65 4
0.0 2.0 0.25 0.0 0.0 1.0 1.75 0.45 0.45 8

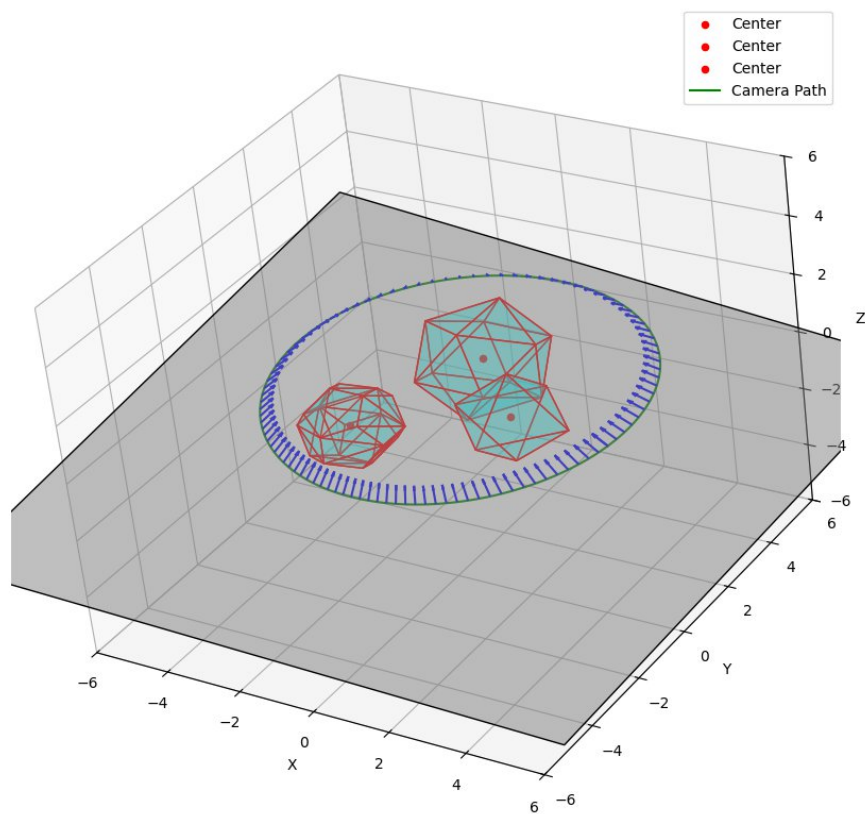
-8.0 -8.0 -2.0 -8.0 8.0 0.0 8.0 8.0 -2.0 8.0 -8.0 -2.5 ./textures/floor_1.data 0.1 0.1
0.1 0.7

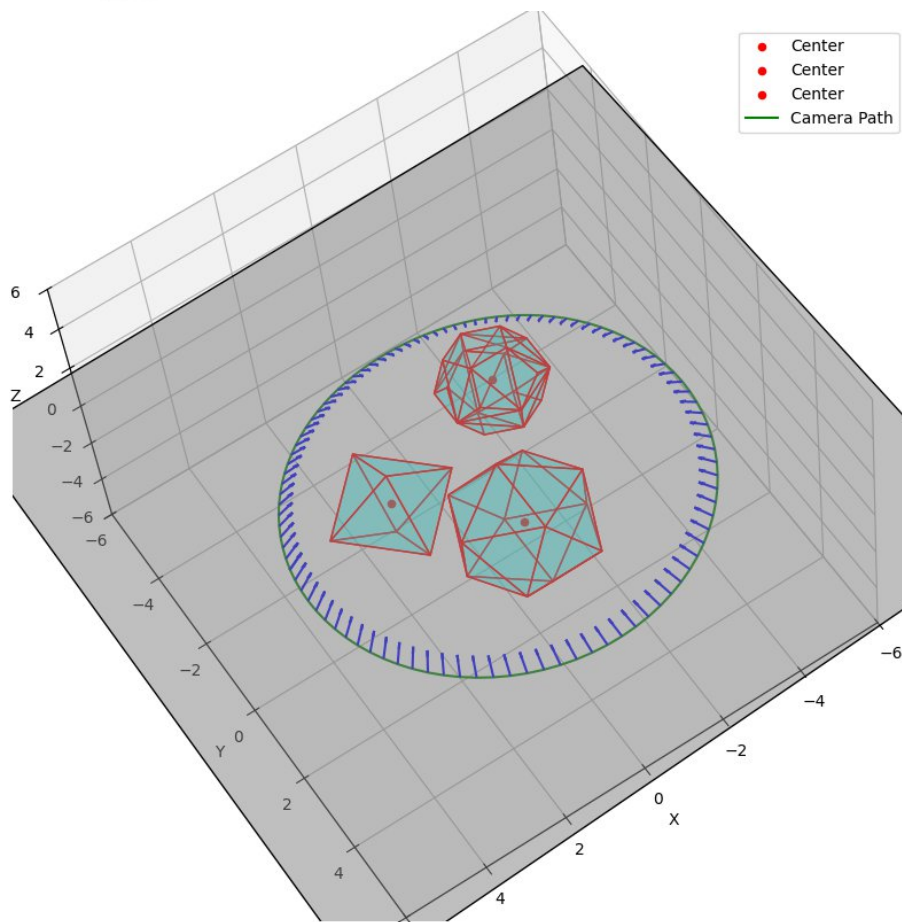
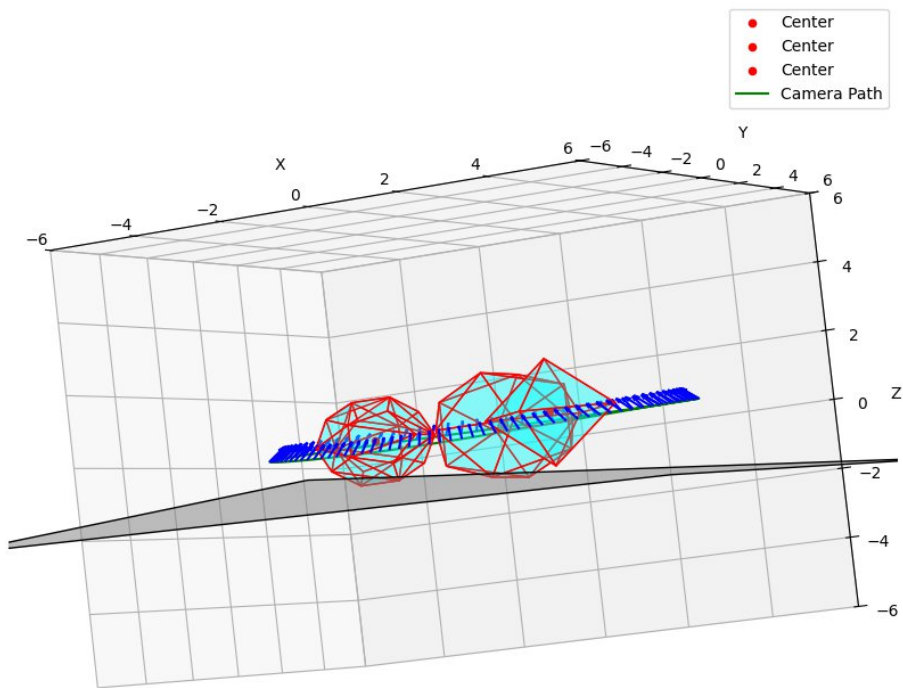
3
3 0 5 1.0 0.0 0.0
-2.12 -2.12 5 0.0 1.0 0.0
0 3 5 0.0 1.0 1.0

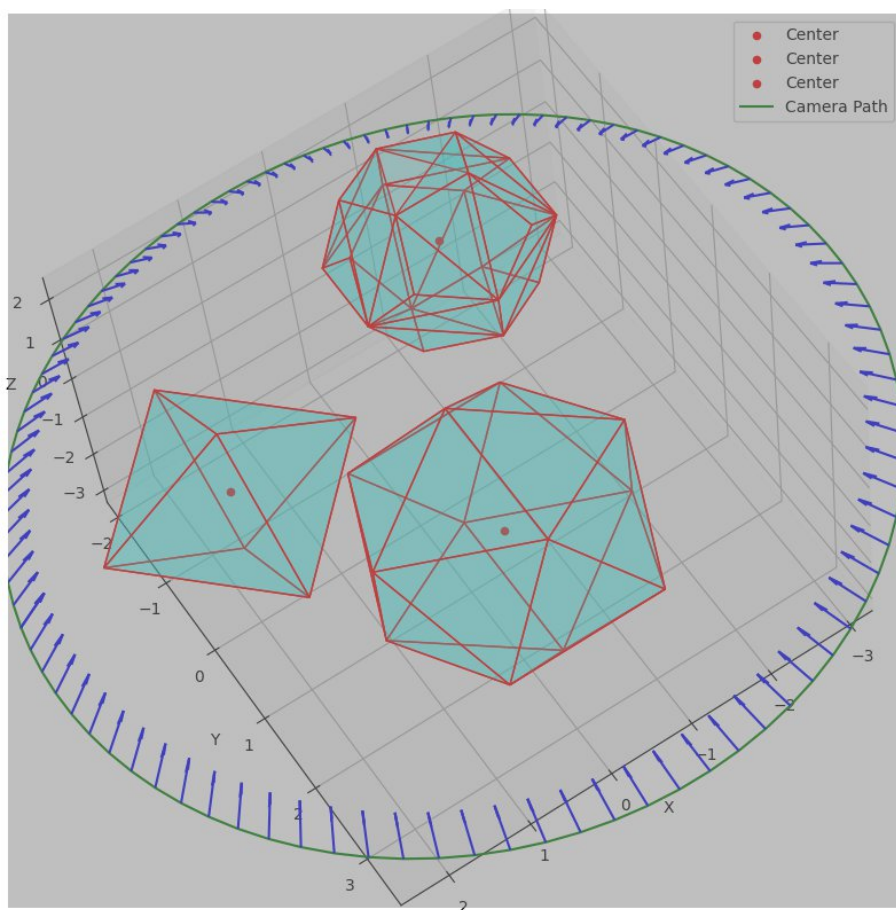
9 1

```

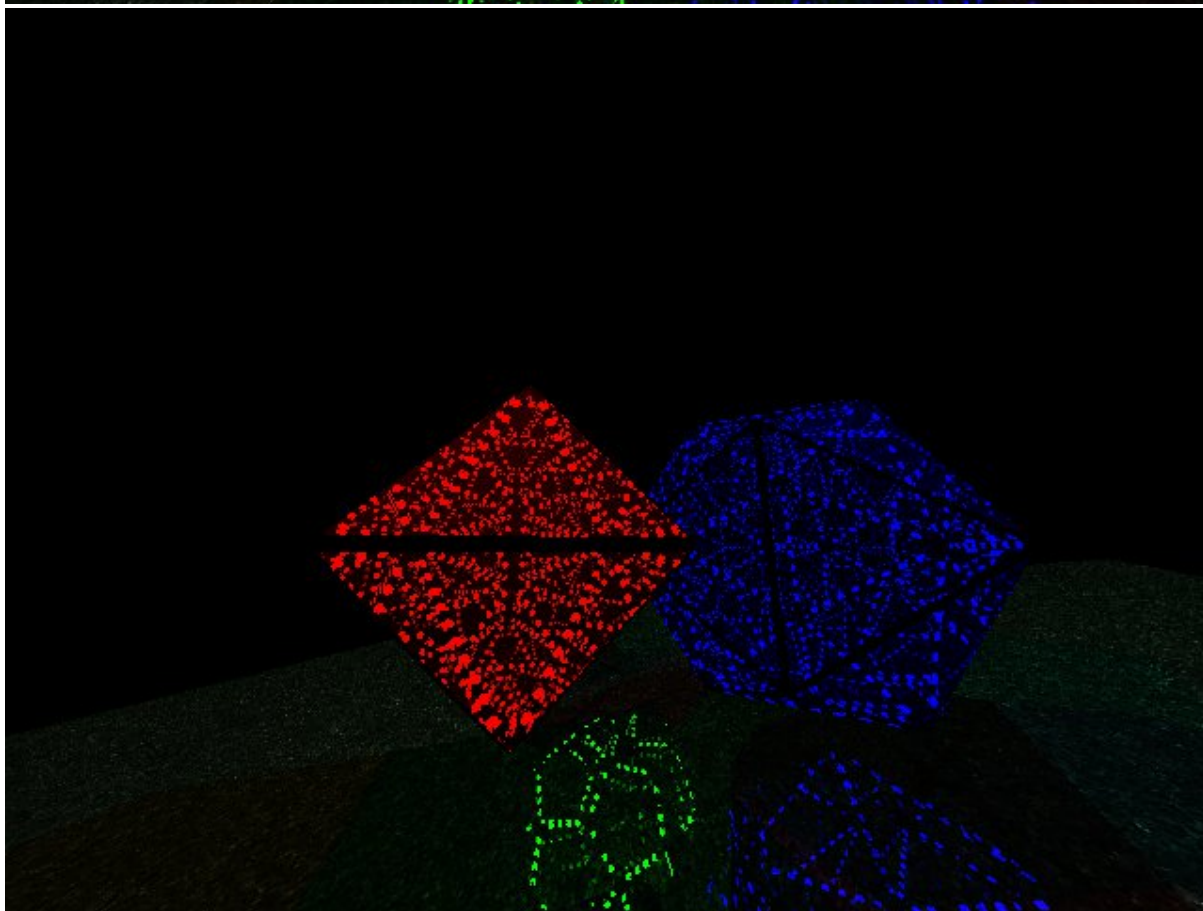
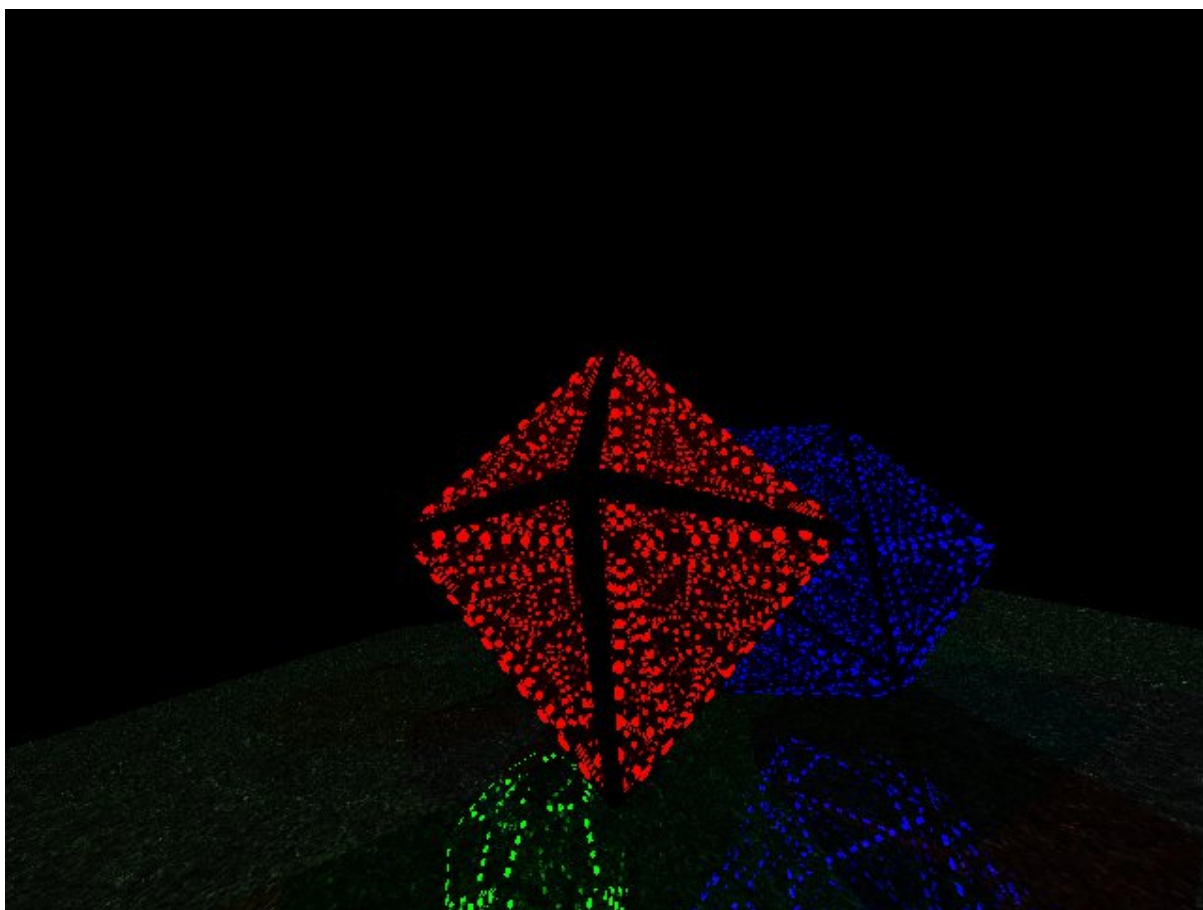
Трёхмерные графики сцены:

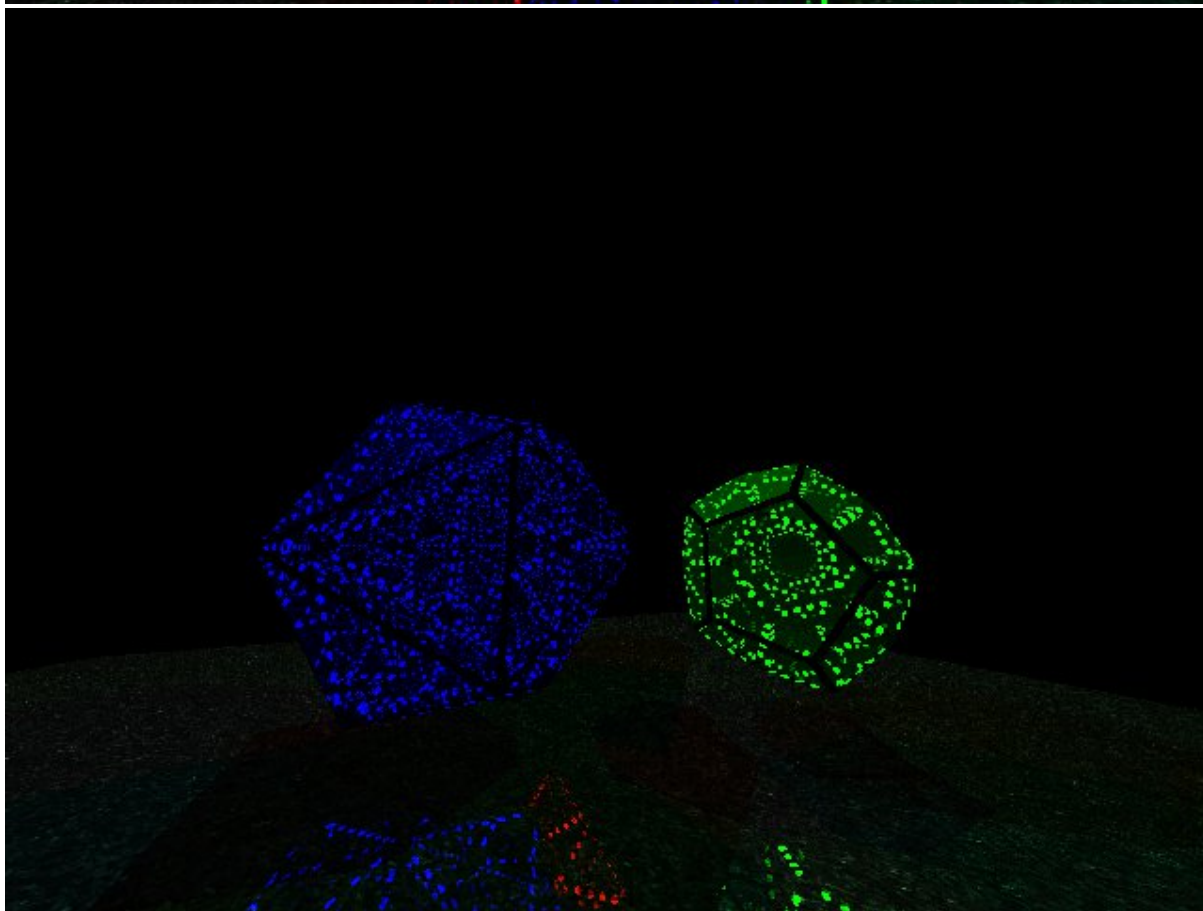
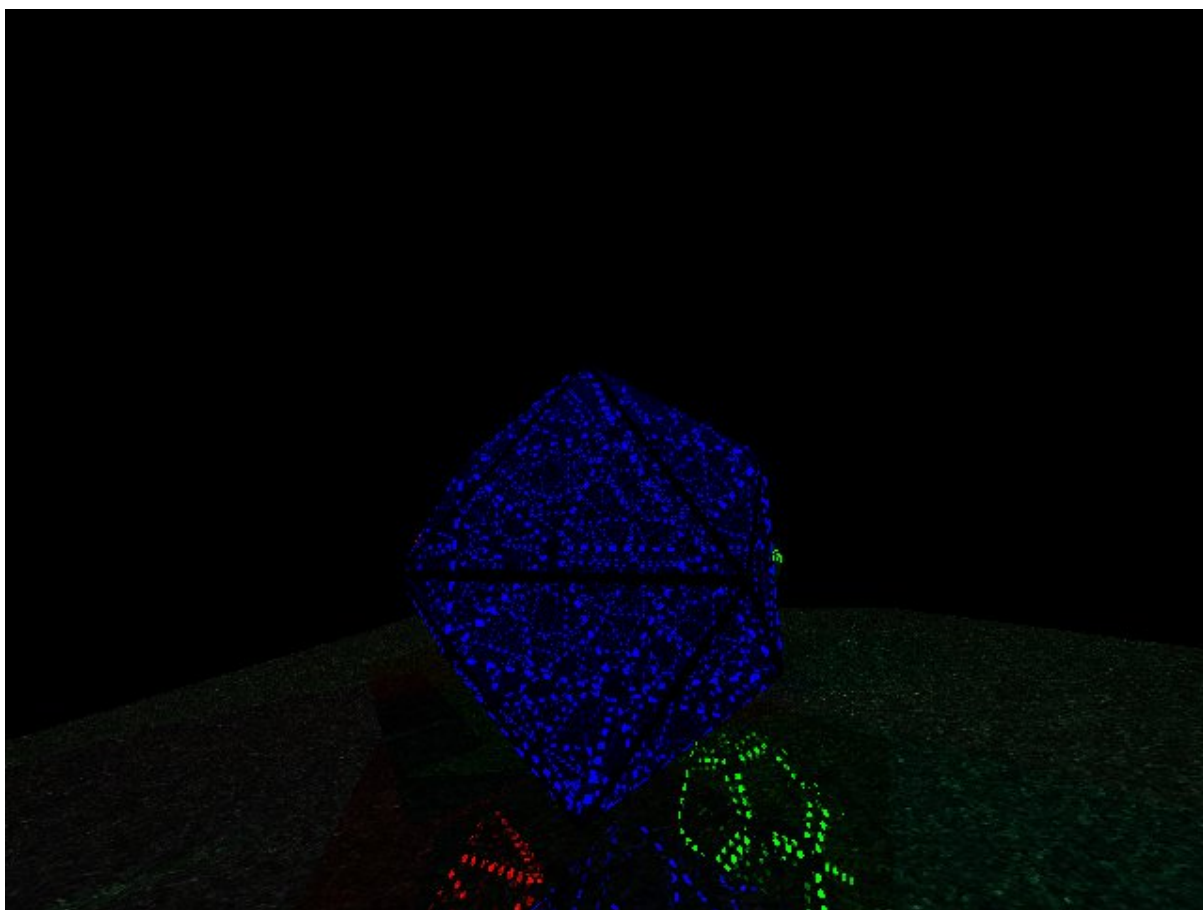


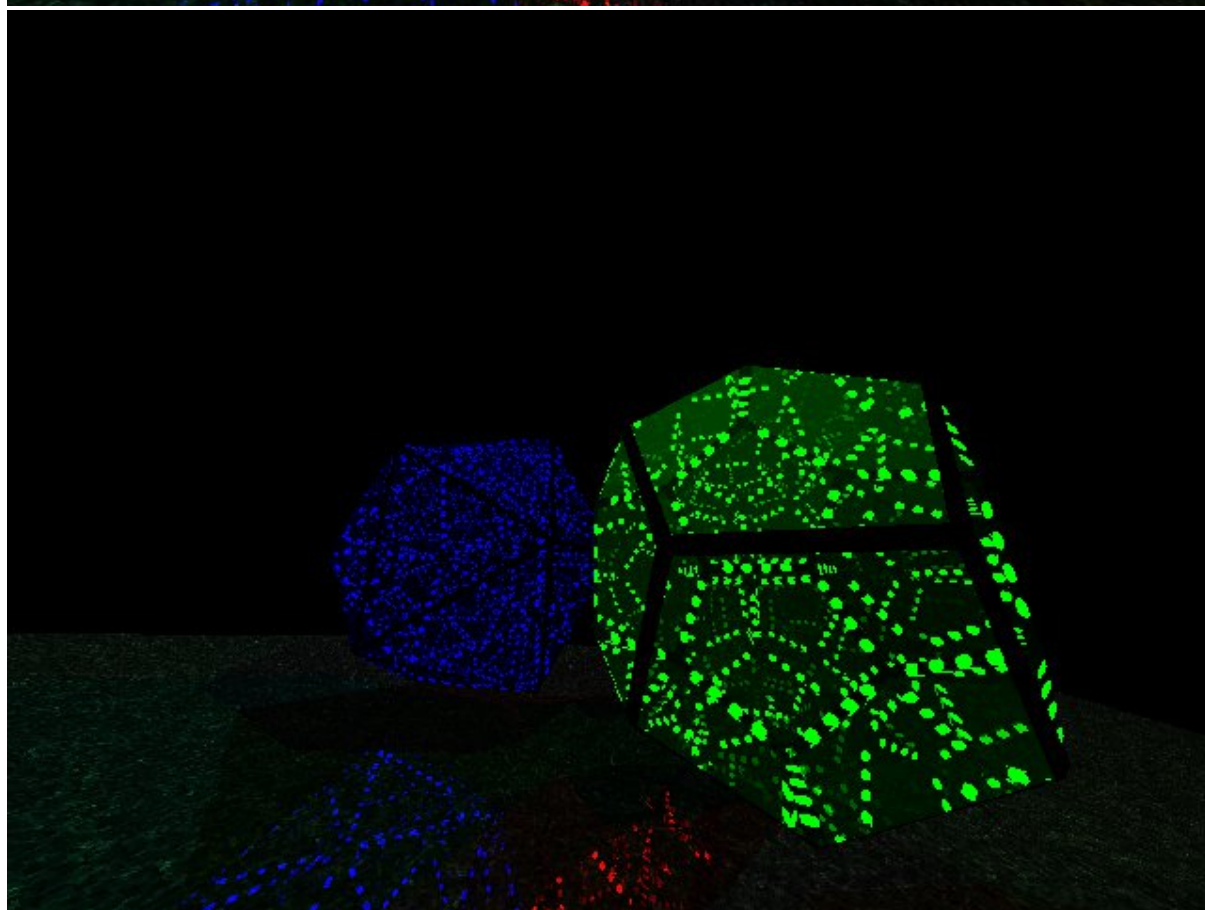
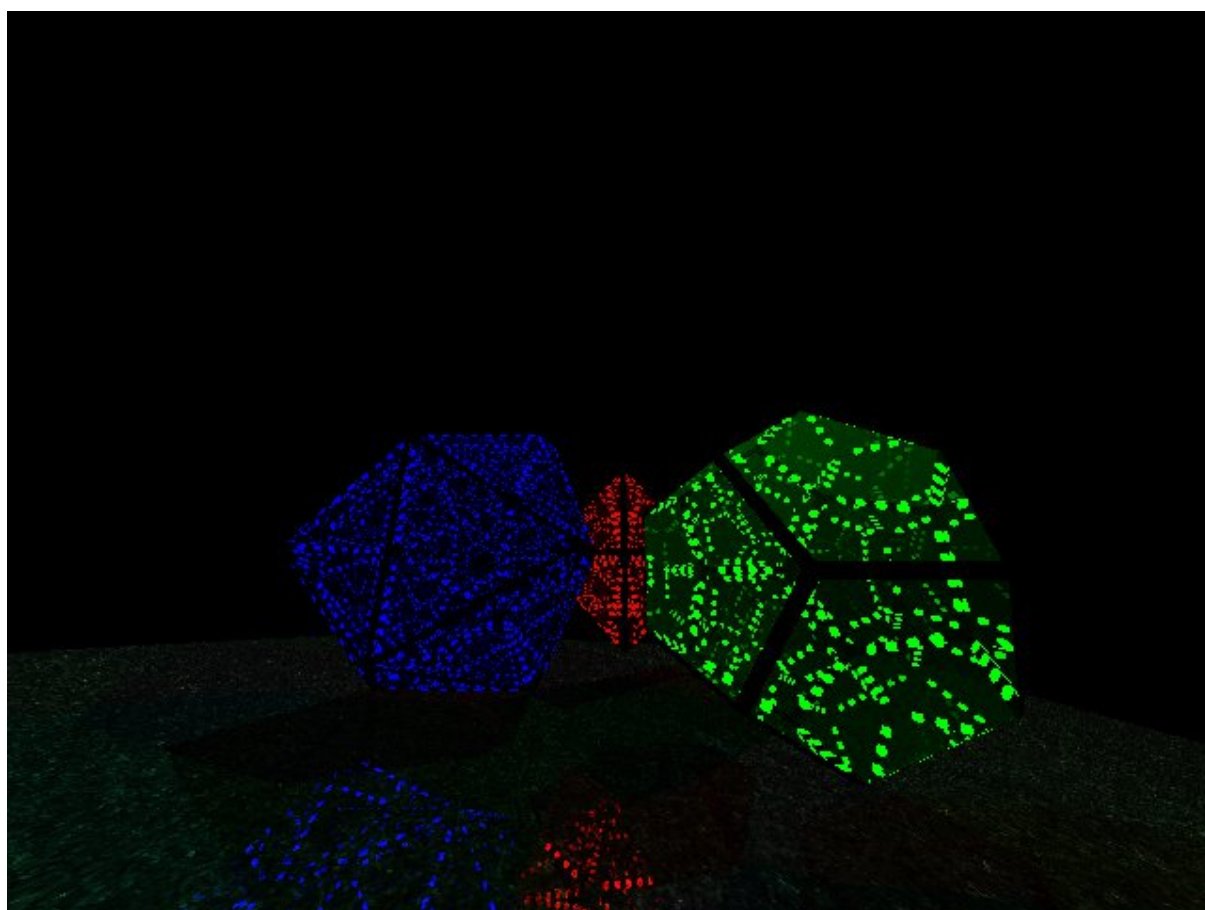


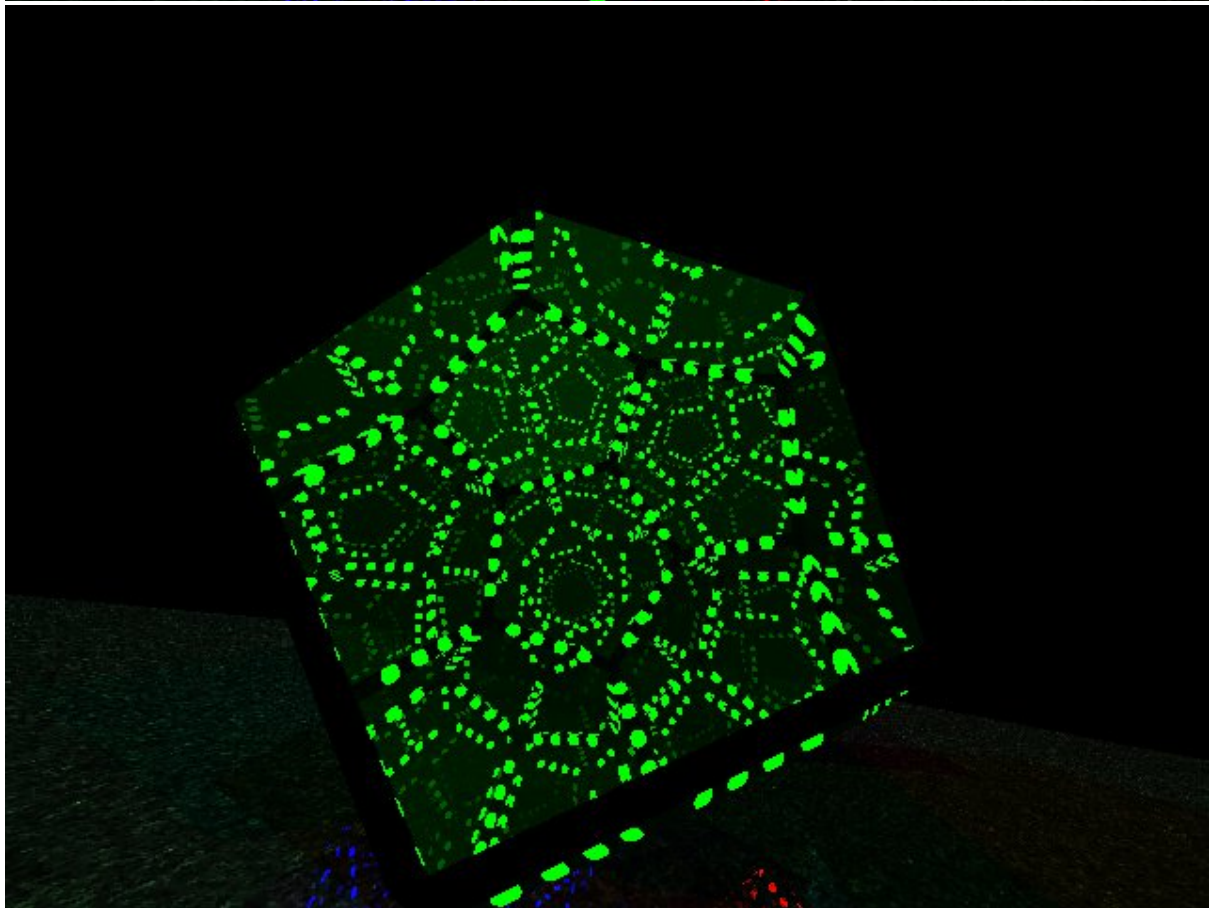
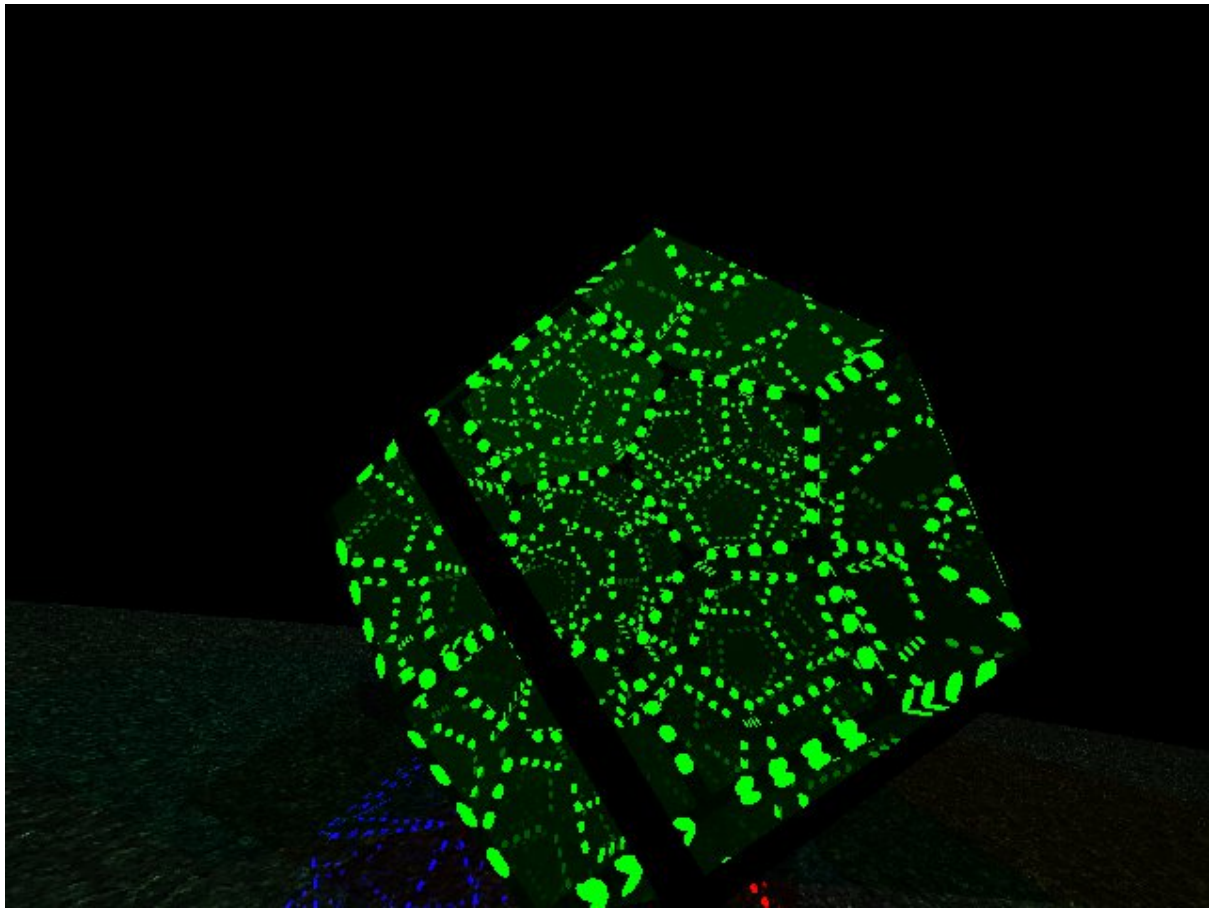


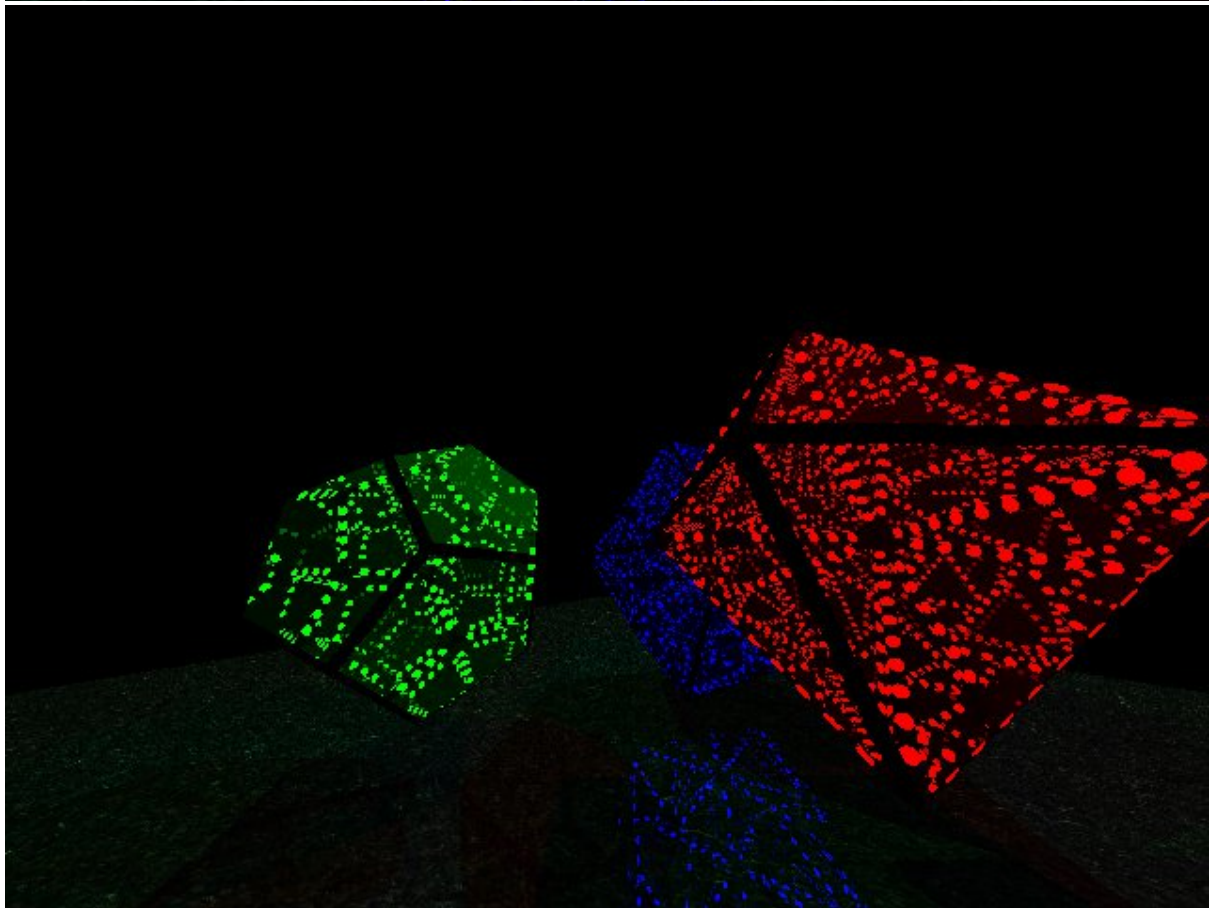
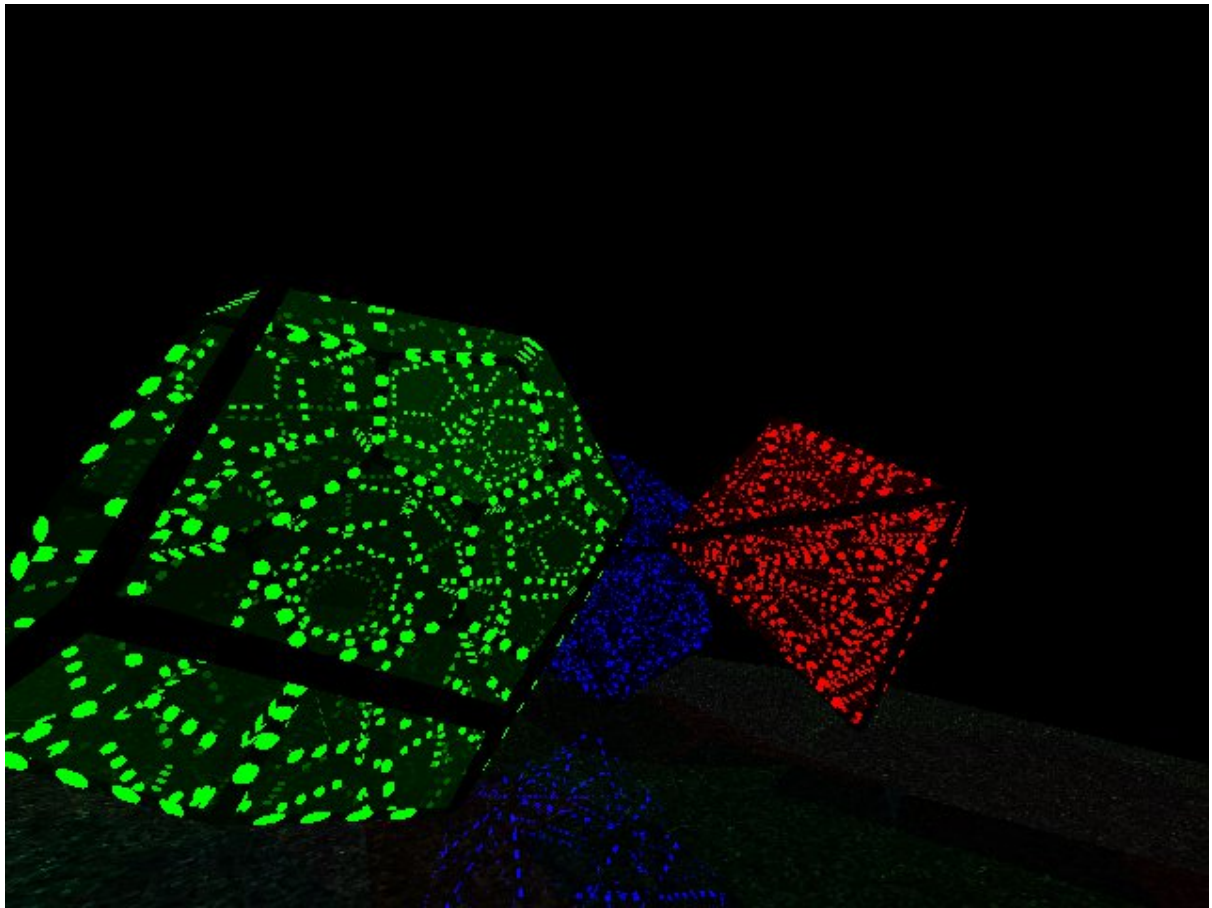
Скриншоты результатов работы программы:

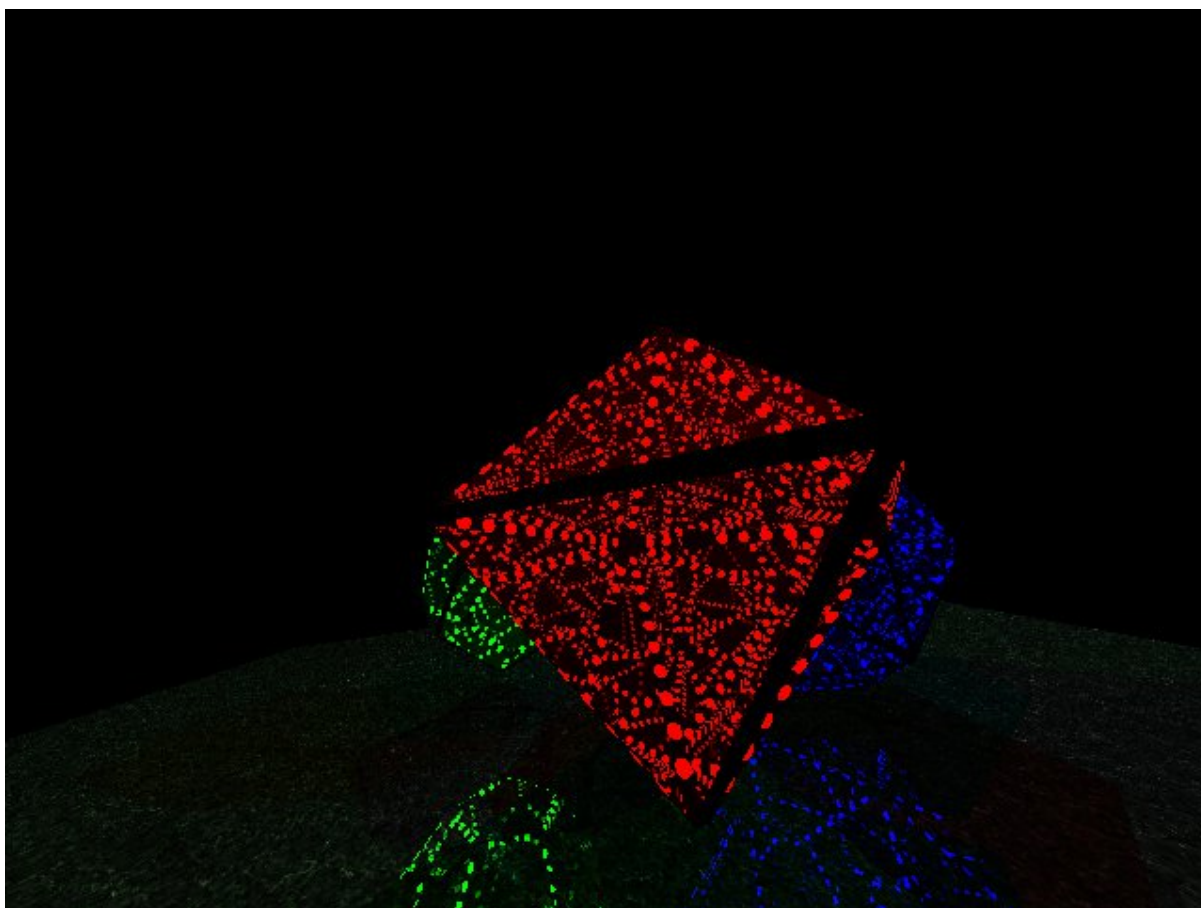












Выводы

Метод трассировки лучей и его расширения, такие как рекурсивная трассировка и алгоритмы сглаживания, обладают потенциалом для создания фотореалистичных изображений, за счет учета сложных взаимодействий света с объектами. Несмотря на значительные вычислительные затраты, современные технологии и методы оптимизации делают использование трассировки лучей возможным для различных приложений компьютерной графики, начиная от киноиндустрии и заканчивая видеоиграми.

Самым сложным оказалось построить сцену: было легко запутаться во всех этих полигонах, векторах и т.д. Ну и сам алгоритм трассировки оказался непростым в реализации.

Как видно из замеров времени работы программы, CPU выигрывает GPU только на самой маленькой конфигурации. Во всех остальных случаях графический процессор быстрее выполняет трассировку. Также стоит заключить, что время выполнения линейно зависит от глубины рекурсии.

Литература

1. 3d движок, трассировка лучей в реальном времени, интерактивная трассировка лучей // Ray-tracing.ru URL: <http://www.ray-tracing.ru/> (дата обращения: 05.01.2025).

2. Простые модели освещения // Компьютерная графика URL: <https://cgraph.ru/node/435> (дата обращения: 05.01.2025).
3. Немного о затенении по Фонгу // Хабр URL: <https://habr.com/ru/articles/441862/> (дата обращения: 05.01.2025).