

Курсовая работа*

Обратная трассировка лучей (Ray Tracing) на GPU.

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

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. В цилиндрических координатах (r, φ, z) , положение и точка направления камеры в момент времени t определяется следующим образом:

$$r_c(t) = r_c^0 + A_c^r \sin(\omega_c^r \cdot t + p_c^r)$$

$$z_c(t) = z_c^0 + A_c^z \sin(\omega_c^z \cdot t + p_c^z)$$

$$\varphi_c(t) = \varphi_c^0 + \omega_c^\varphi t$$

$$r_n(t) = r_n^0 + A_n^r \sin(\omega_n^r \cdot t + p_n^r)$$

$$z_n(t) = z_n^0 + A_n^z \sin(\omega_n^z \cdot t + p_n^z)$$

$$\varphi_n(t) = \varphi_n^0 + \omega_n^\varphi t$$

где

$$t \in [0, 2\pi]$$

Требуется реализовать алгоритм обратной трассировки лучей (<http://www.ray-tracing.ru/>) с использованием технологии CUDA. Выполнить покадровый рендеринг сцены. Для устранения эффекта «зубчатости», выполнить сглаживание (например с помощью алгоритма SSAA). Полученный набор кадров склеить в видеоролик любым доступным программным обеспечением. Подобрать параметры сцены, камеры и освещения таким образом, чтобы получить наиболее красочный результат. Провести сравнение производительности гри и сри (т.е. дополнительно нужно реализовать алгоритм без использования CUDA).

Варианты заданий: на сцене должны располагаться три тела:

1. Тетраэдр, Гексаэдр, Октаэдр
2. Тетраэдр, Гексаэдр, Додекаэдр
3. Тетраэдр, Гексаэдр, Икосаэдр
4. Тетраэдр, Октаэдр, Додекаэдр
5. Тетраэдр, Октаэдр, Икосаэдр
6. Тетраэдр, Додекаэдр, Икосаэдр
7. Гексаэдр, Октаэдр, Додекаэдр
8. Гексаэдр, Октаэдр, Икосаэдр
9. Гексаэдр, Додекаэдр, Икосаэдр
10. Октаэдр, Додекаэдр, Икосаэдр

Программа должна принимать на вход следующие параметры:

1. Количество кадров.
2. Путь к выходным изображениям. В строке содержится спецификатор %d, на место которого должен подставляться номер кадра. Формат изображений соответствует формату описанному в лабораторной работе 2.
3. Разрешение кадра и угол обзора в градусах по горизонтали.
4. Параметры движения камеры $r_c^0, z_c^0, \varphi_c^0, A_c^r, A_c^z, \omega_c^r, \omega_c^z, \omega_c^\varphi, p_c^r, p_c^z$, и $r_n^0, z_n^0, \varphi_n^0, A_n^r, A_n^z, \omega_n^r, \omega_n^z, \omega_n^\varphi, p_n^r, p_n^z$.
5. Параметры тел: центр тела, цвет (нормированный), радиус (подразумевается радиус сферы в которую можно было бы вписать тело), коэффициент отражения, коэффициент прозрачности, количество точечных источников света на ребре.
6. Параметры пола: четыре точки, путь к текстуре, оттенок цвета и коэффициент отражения.
7. Количество (не более четырех) и параметры источников света: положение и цвет.
8. Максимальная глубина рекурсии и квадратный корень из количества лучей на один пиксель (для SSAA).

Пример входных данных:

1024

/home/checker/pgp/ivanov/kp/out/img_%d.data

640 480 120

7.0 3.0 0.0 2.0 1.0 2.0 6.0 1.0 0.0 0.0

2.0 0.0 0.0 0.5 0.1 1.0 4.0 1.0 0.0 0.0

2.0 0.0 0.0 1.0 0.0 0.0 1.0 0.9 0.1 10

0.0 2.0 0.0 0.0 1.0 0.0 0.75 0.8 0.2 5

0.0 0.0 0.0 0.0 0.7 0.7 0.5 0.7 0.3 2

-5.0 -5.0 -1.0 -5.0 5.0 -1.0 5.0 5.0 -1.0 5.0 -5.0 -1.0 ~/floor.data 0.0 1.0 0.0 0.5

2

-10.0 0.0 10.0 1.0 1.0 1.0

1.0 0.0 10.0 0.0 0.0 1.0

10 16

Программа должна поддерживать следующие ключи запуска:

--cpu Для расчетов используется только центральный процессор

--gpu Для расчетов задействуется видеокарта

--default В stdout выводится конфигурация входных данных (в формате описанном ранее) при которой получается наиболее красочный результат, после чего программа завершает свою работу.

Запуск программы без аргументов подразумевает запуск с ключом --gpu.

В процессе работы программа должна выводить в stdout статистику в формате:

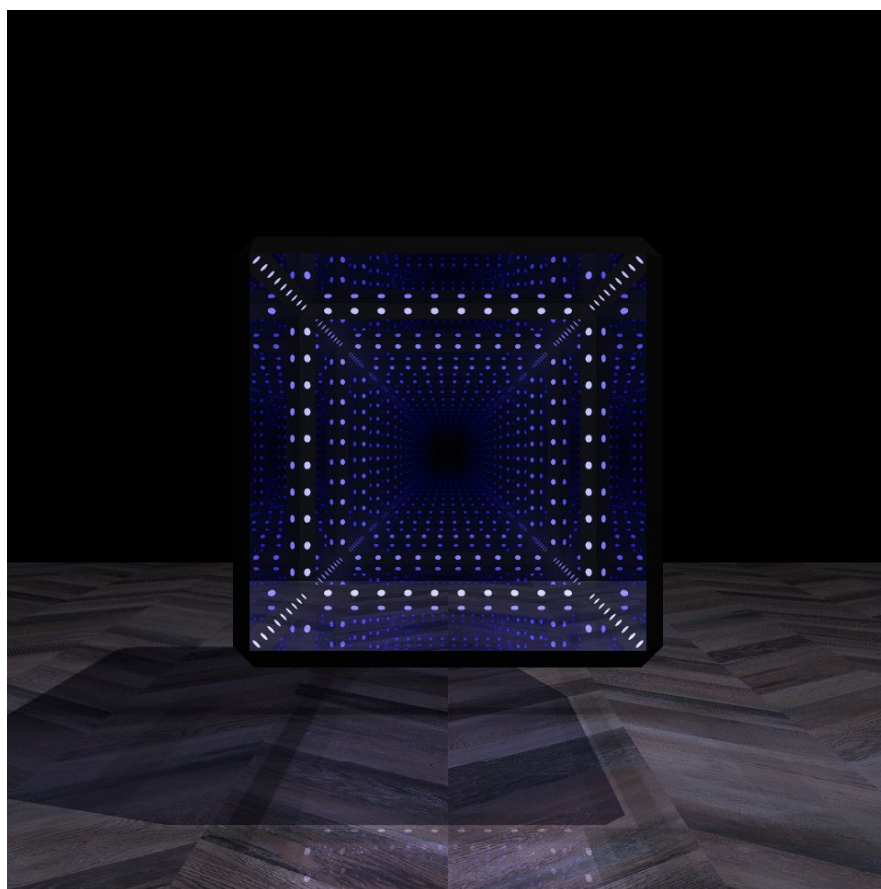
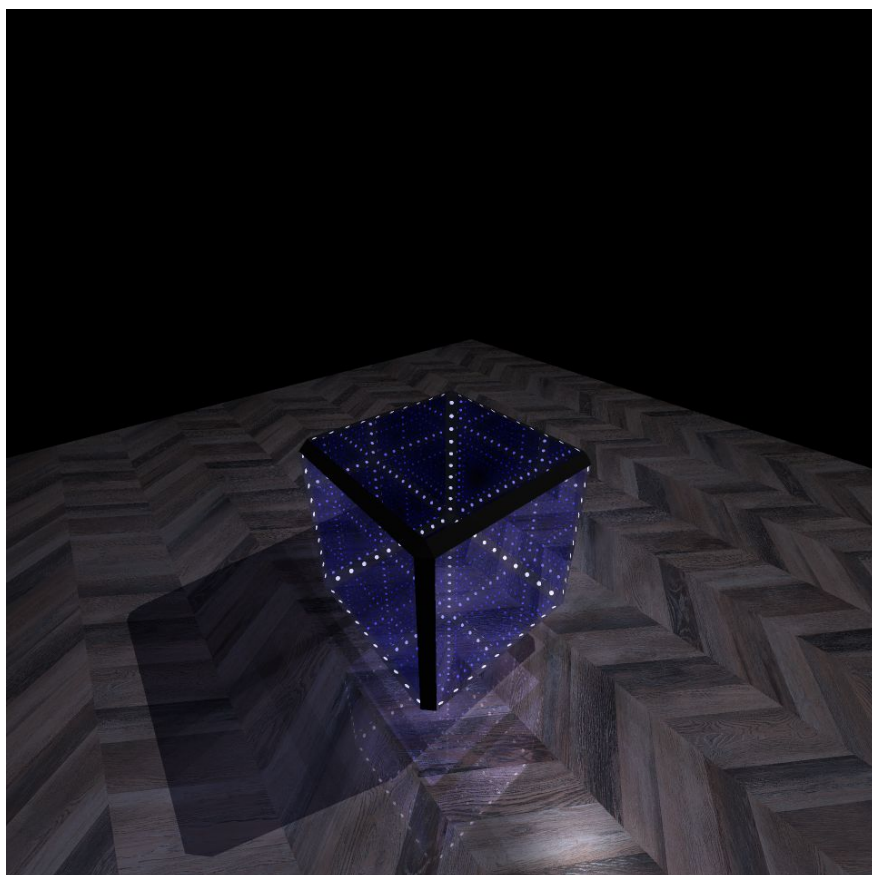
{номер кадра}\t{время на обработку кадра в миллисекундах}\t{общее количество лучей}\n

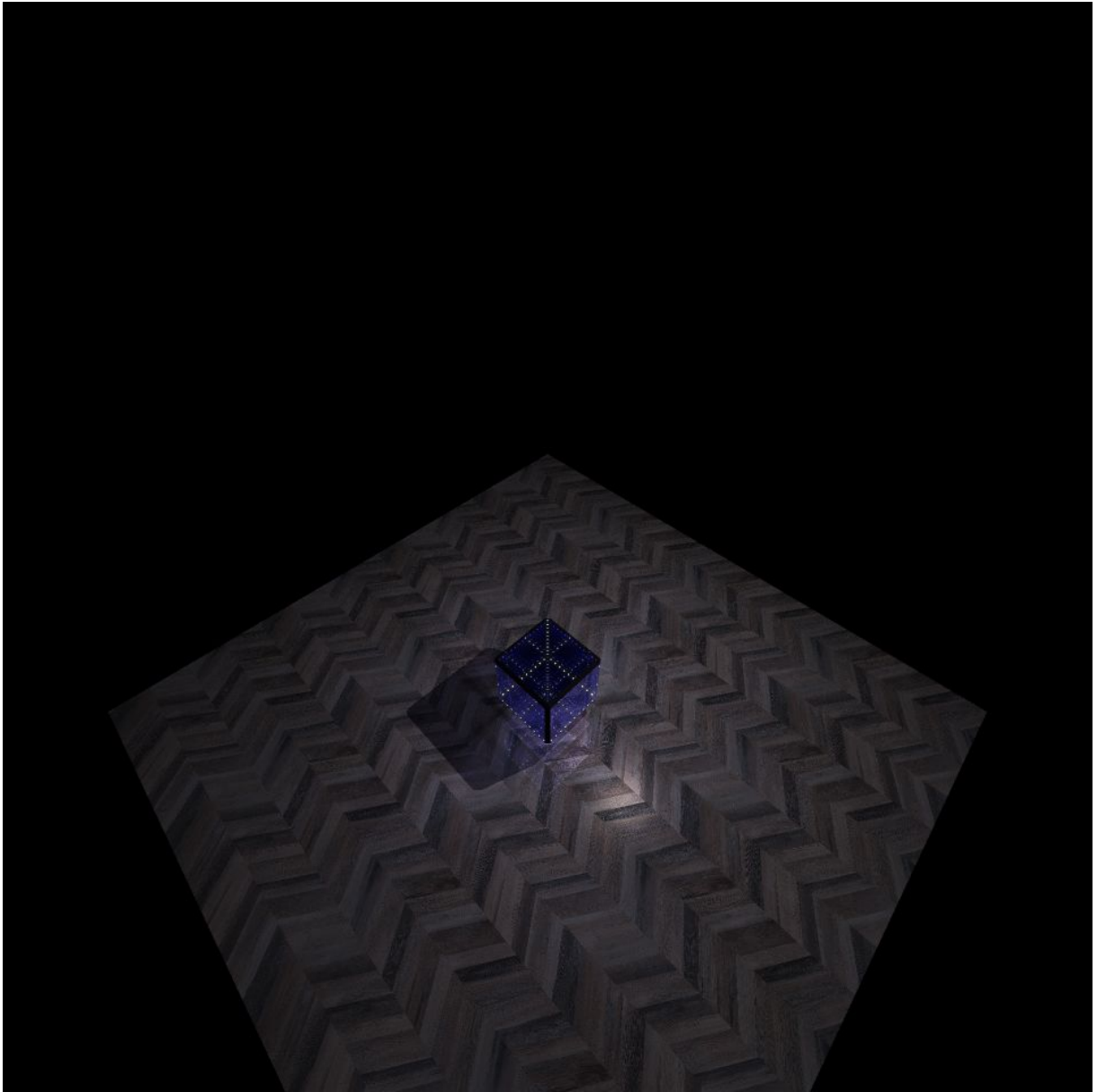
Программу необходимо отправить на чекер с темой письма rgr:kp

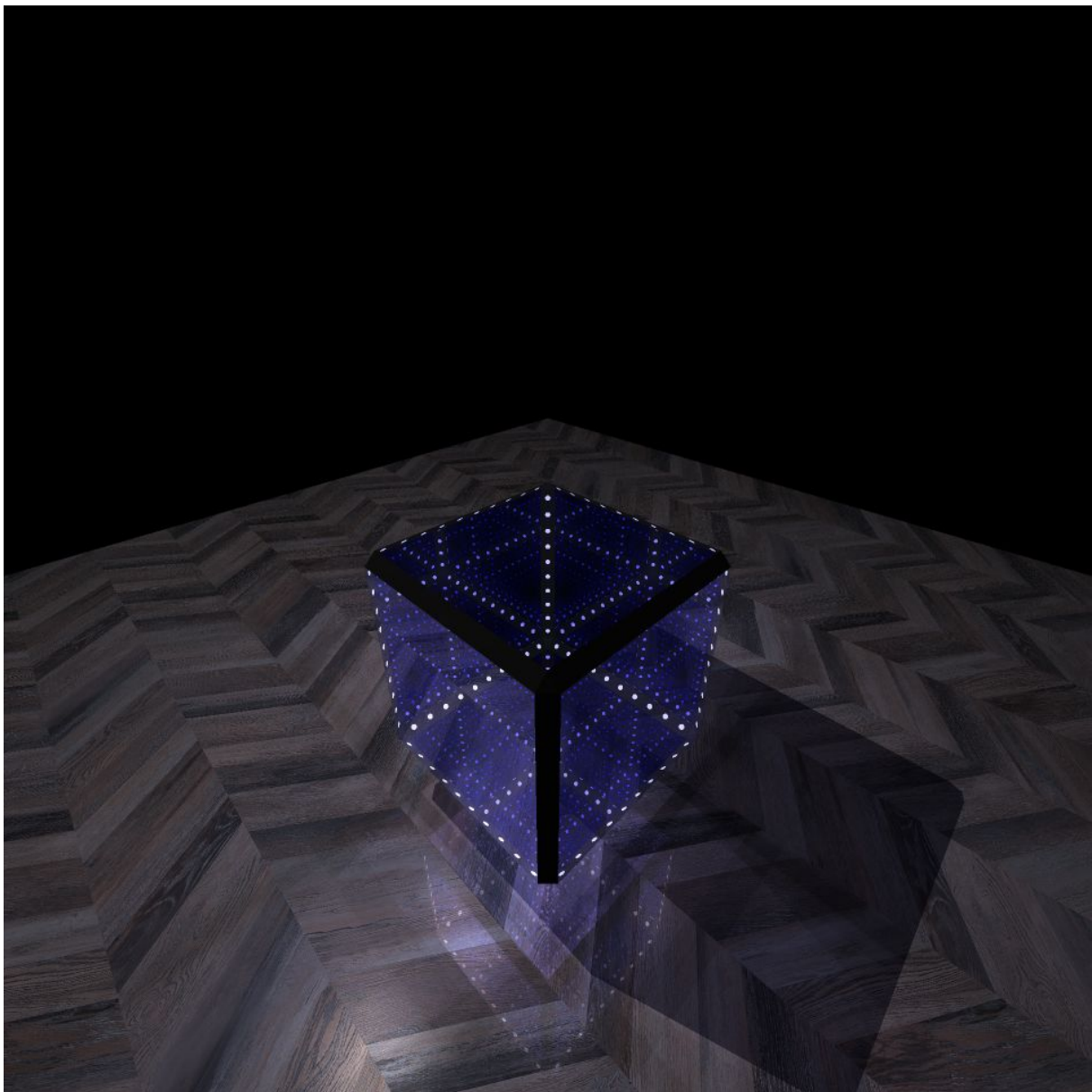
*Альтернатива КП: личное участие в международной конференции с публикацией тезисов в РИНЦе.

- 1) <http://aik.mai.ru/> 23-27 ноября 2020 г., МАИ, прием тезисов до 1 октября
- 2) <https://gagarin.mai.ru/> апрель 2021 г., МАИ, прием тезисов до февраля
- 3) ...

Примеры кадров:







Оценки

На три балла: без рекурсии, без текстур, без отражений (без эффекта бесконечности), простые модели без отдельных ребер с источниками света, один источник света.

На четыре балла: без рекурсии, с отражениями, упрощенная обработка прозрачности (луч далее не отражается), упрощенное наложение текстур (пол всегда квадратный и параллелен плоскости XY).

На пять баллов: все (развернутая рекурсия, произвольное наложение текстур, и т.д.).