

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное автономное образовательное учреждение высшего  
образования  
Национальный исследовательский нижегородский государственный университет им. Н.И.  
Лобачевского  
Институт информационных технологий, математики и механики  
Кафедра математического обеспечения и суперкомпьютерных технологий

## Отчёт по практике

Тема :

Методы трассировки лучей

**Выполнил:**

студент гр. 381706-1

Власов Андрей Сергеевич

**Научный руководитель:**

Профессор, доктор технических наук

Турлапов Вадим Евгеньевич

Нижний Новгород 2019

# Содержание

<b>1</b>	<b>Введение</b>	<b>3</b>
<b>2</b>	<b>Необходимый теоретический минимум</b>	<b>4</b>
<b>3</b>	<b>Обзор источников</b>	<b>7</b>
<b>4</b>	<b>Разбор статей</b>	<b>8</b>
4.1	Статья 1 : Двухуровневая трассировка лучей . . . . .	8
4.2	Статья 2 : Двухнаправленная трассировка лучей . . . . .	10
<b>5</b>	<b>Результаты практики</b>	<b>12</b>
<b>6</b>	<b>Заключение</b>	<b>13</b>
<b>7</b>	<b>Ссылки</b>	<b>14</b>

# 1 Введение

Одним из методов визуализации в компьютерной графике является трассировка лучей, или рейтрейсинг (ray tracing). Он позволяет добиться синтеза реалистичных изображений, так как процедура трассировки лучей имитирует поведение фотона. Данный метод требует большой вычислительной мощности, поэтому ранее он применялся редко и в ограниченном объеме. Но сейчас с развитием технологий становится все популярнее и набирает обороты.

Впервые масштабно трассировку лучей начали использовать в киноиндустрии. В таком случае просчет лучей необходимо было делать всего лишь один раз на стадии монтажа. В игровую же индустрию рейтрейсинг пришел совсем недавно, так как в играх необходимо рассчитывать поведение источников света в реальном времени, что намного затрудняет задачу. Несмотря на развитие видеокарт, на сегодняшний день существует немного игр, где используется трассировка лучей. Но даже и там она реализована не в полной мере, а только частично. Например, в Battlefield V реализованы только отражения и эффекты преломления, а в Metro Exodus – глобальное освещение и затенение. Данная технология выводит графику в играх на новый уровень, и в будущем мы сможем видеть все более и более реалистичную картину.

В данном отчете будут

- рассмотрены методы трассировки лучей
- разобраны основные составляющие части программной реализации этих методов
- описаны основные положения из теории, лежащие в основе выбранных методов
- предъявлены результаты проведенных экспериментов

## **Постановка задачи :**

1. Исследовать различные публикации на поставленную тему.
2. Изучить методы трассировки лучей.
3. Начать проведение практических экспериментов.

## 2 Необходимый теоретический минимум

### 1. Рендеринг [1]

Рендеринг или отрисовка (англ. rendering — «визуализация») — термин в компьютерной графике, обозначающий процесс получения изображения по модели с помощью компьютерной программы. Здесь модель — это описание любых объектов или явлений на строго определённом языке или в виде структуры данных. Такое описание может содержать геометрические данные, положение точки наблюдателя, информацию об освещении, степени наличия какого-то вещества, напряжённость физического поля и пр.

Часто в компьютерной графике (художественной и технической) под рендерингом (3D-рендерингом) понимают создание плоской картинки — цифрового растрового изображения — по разработанной 3D-сцене. Синонимом в данном контексте является визуализация. Визуализация — один из наиболее важных разделов в компьютерной графике, и на практике он тесным образом связан с остальными. Обычно программные пакеты трёхмерного моделирования и анимации включают в себя также и функцию рендеринга. Существуют отдельные программные продукты, выполняющие рендеринг.

В зависимости от цели, различают рендеринг, как достаточно медленный процесс визуализации, применяющийся в основном при создании видео, например в Vegas Pro, и рендеринг в режиме реального времени, например, в компьютерных играх. Последний часто использует 3D-ускорители.

### 2. Трассировка лучей [2]

Основной идеей трассировки лучей является симулирование физического поведения света. Используя ее, видеокарта в точности моделирует прохождения лучей от источников освещения и их взаимодействие с объектами. При этом, учитываются свойства поверхностей объектов, на основании чего вычисляются точки начала рассеивания, особенности отражения света, возникновения теней и многое другое.

По сути, это симуляция модели человеческого зрения, которая вплотную приближает компьютерную графику к кинематографическому уровню.

**Прямой метод :** Метод прямой трассировки предполагает построение траекторий лучей от всех источников освещения ко всем точкам всех объектов сцены. Это, так называемые, первичные лучи. Точки, лежащие на противоположной от источника света стороне, исключаются из расчета. Для всех остальных точек вычисляется освещенность с помощью локальной модели освещения. Если объект не является отражающим или прозрачным, то траектория луча на этой точке обрывается. Если же поверхность объекта обладает свойством отражения или преломления то из точки строятся новые лучи, направления которых совершенно точно определяются законами отражения и преломления. Траектории новых лучей также отслеживаются. Построение новых траекторий и расчеты ведутся до тех пор, пока все лучи либо попадут в камеру, либо выйдут за пределы видимой области. Очевидно, что при прямой трассировке лучей мы вынуждены выполнять расчеты для лучей, которые не попадут в камеру, то есть проделывать бесполезную работу. По некоторым оценочным данным, доля таких «слепых» лучей довольно велика. Эта главная, хотя и далеко не единственная причина того, что метод прямой трассировки лучей считается неэффективным и на практике не используется.

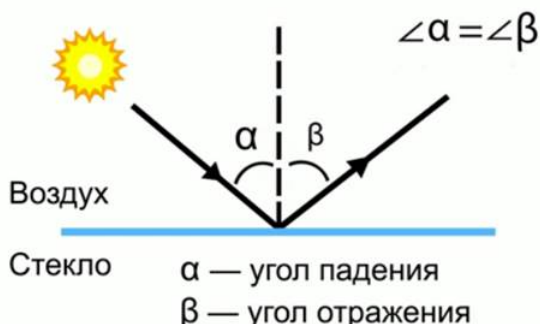
**Обратный метод :** В отличие от прямого в данной методе лучи выпускаются как бы из глаза к источнику. Обратная трассировка более эффективна, так как гарантирует, что число лучей, достигших глаза, будет точно таким же, как число пикселей изображения. Для того, чтобы представить этот процесс, считайте, что ваш экран монитора - это окно, через которое вы можете видеть трехмерную модель, нарисованную вашим компьютером. Мнимые линии, называемые лучами, проводятся из вашего глаза через каждый пиксел экрана и проецируются на модель. Каждый раз, когда луч пересекает некоторую поверхность в некоторой точке, из этой точки испускаются дополнительные лучи. Если поверхность отражающая, то

генерируется отраженный луч. Если поверхность пропускает свет, то генерируется пропущенный луч, причем учитывается тот факт, что луч меняет направление, когда проходит из одной среды в другую. Это явление называется преломлением света. Некоторые поверхности одновременно и отражают, и пропускают лучи, и тогда испускаются оба вида лучей. Пути этих лучей отслеживаются по всей модели, и если лучи пересекают другие поверхности, то снова испускаются лучи. В каждой точке, где луч пересекает поверхность, рисуется луч тени из точки пересечения к каждому источнику света. Если этот луч пересекает другую поверхность перед тем, как достигнуть источника света, то на ту поверхность, с которой был послан луч, падает тень с поверхности, блокирующей свет. Математически, все эти лучи вместе с данными о физических характеристиках объектов модели (цвет, прозрачность, зеркальность и т.д.) позволяют компьютеру определить цвет и его интенсивность для каждой точки изображения. Рассмотрим все варианты взаимодействия луча с объектами сцены:

1. Луч не сталкивается ни с одним объектом. В этом случае пиксель закрашивается цветом фона и процесс обрывается.
2. В остальных случаях луч соударяется с некоторым объектом. Необходимо рассчитать прямое освещение и вторичное освещение, а также выяснить, находится ли точка в тени.
3. Чтобы выяснить, находится ли точка в тени, следует выпустить вторичный теневой луч из точки к источнику света. Если теневой луч столкнется с объектом сцены, то точка находится в тени.

### 3. Закон отражения света. [3]

Представьте, что вы направили тонкий луч света на отражающую поверхность, — например, посветили лазерной указкой на зеркало или полированную металлическую поверхность. Луч отразится от такой поверхности и будет распространяться дальше в определенном направлении. Угол между перпендикуляром к поверхности (нормалью) и исходным лучом называется углом падения, а угол между нормалью и отраженным лучом — углом отражения. Закон отражения гласит, что угол падения равен углу отражения.

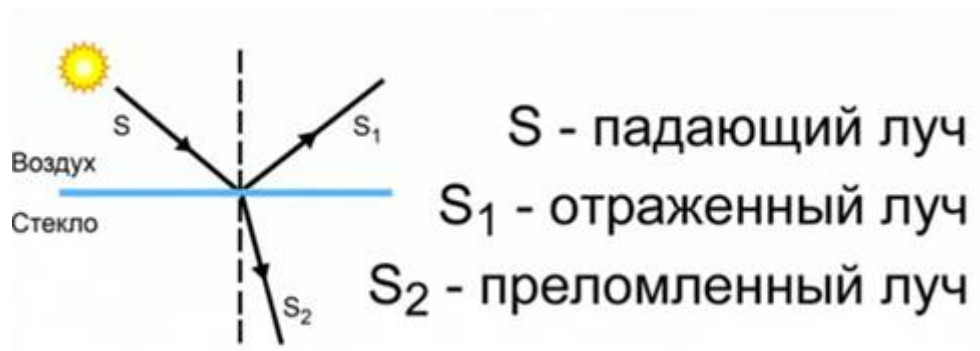


\*Рис. 1 - Иллюстрация закона отражения света.

Этот закон является следствием применения принципа Ферма к отражающей поверхности и, как и все законы геометрической оптики, выводится из волновой оптики. Закон справедлив не только для идеально отражающих поверхностей, но и для границы двух сред, частично отражающей свет. В этом случае, равно как и закон преломления света, он ничего не утверждает об интенсивности отражённого света.

### 4. Закон преломления света. [3]

Рассмотрим второй закон геометрической оптики, который связан с отражением света. Если направить тонкий луч света на прозрачную поверхность, то произойдет преломление света. Это явление, при котором луч света, переходя из одной среды в другую, изменяет направление на границе этих сред. Часть луча, которая прошла внутрь второй среды, будет называться преломленным лучом.



\*Рис. 2 - Иллюстрация закона преломления света.

Отношение синуса угла падения к синусу угла преломления есть величина постоянная. Эта величина называется показателем преломления среды.

### 3 Обзор источников

Трассировка лучей является центральным алгоритмом, вокруг которого построены практически все методы расчета освещенности.

Поэтому каждая статья отличается своим подходом к решению данной проблемы, и авторы предлагают модифицированные трассировки лучей.

Каждый из существующих методов дает отличные от других результаты. Модификации можно разделить на две группы: первые добиваются еще более реалистичного изображения, а вторые увеличивают производительность стандартной трассировки.

Примеры работ, представляющих подобные алгоритмы :

1. Двухнаправленная трассировка лучей.

*Статья Двухнаправленная трассировка лучей для интегрирования освещенности методом Квази-Монте Карло, А.Г. Волобой, В.А. Галактионов, К.А. Дмитриев, Э.А. Копылов [4].*

В статье рассматривается метод комбинирования прямой и обратной трассировки лучей для того, чтобы исключить недостатки обоих алгоритмов. Эффективная реализация двухнаправленных методов осуществляется перераспределением трассируемых лучей, при котором трассируются лишь лучи существенно влияющие на формируемое изображение. Реализация двухнаправленной трассировки лучей подразумевает решение оптимизационной задачи, минимизирующей количество трассируемых лучей, не приводящих, в конечном счете, к переносу энергии от источника света до камеры наблюдения.

2. Двухуровневая трассировка лучей

*Статья The two level ray tracing algorithm and its elaboration toward Optical elements Preprint, Inst. Appl. Math., the Russian Academy of Science, B.H.Barladyan, A.G.Voloboy, V.A.Galaktionov, L.Z.Shapiro [5].*

В статье рассматривается метод ускорения трассировки лучей путем использования специальных алгоритмов, основанных на разбиении всего пространства сцены на множество подпространств, образуя из них линейное множество или некоторую иерархию. В итоге разбиения пространства строятся вспомогательные структуры, хранящие индексы объектов сцены, принадлежащих каждому подпространству. Эти структуры позволяют существенно ускорить процесс трассировки сцены. Существует большое множество различных видов вспомогательных структур: равномерное разбиение, октарные деревья, BSP и kd деревья. Время, необходимое для построения таких структур, может быть существенным и иногда значительно превышать время генерации изображения для одного положения камеры.

3. Стохастическая трассировка лучей.

*Диссертация Методы решения проблемы глобальной освещенности на графических процессорах, Фролов Владимир Александрович [6].*

В работе рассматривается стохастическая трассировка лучей (также называемая Монте-Карло трассировкой). Из виртуальной камеры испускается луч, который трассируется в сцену. В точке пересечения с поверхностью выпускается некоторое число лучей по полусфере и для каждого луча процедура выполняется рекурсивно. Полученные значения яркости на каждом уровне рекурсии складываются с учетом значения интеграла освещенности. В дальнейшем рассматривается модификация стохастической трассировки лучей, при которой отраженный луч всегда один. Такое упрощение немного замедляет сходимость метода, но позволяет упростить реализацию и при желании избавиться от рекурсии в реализации алгоритма.

Далее приведен более подробный анализ двух конкретных статей :

- The two level ray tracing algorithm and its elaboration toward Optical elements Preprint, Inst. Appl. Math., the Russian Academy of Science, B.H.Barladyan, A.G.Voloboy, V.A.Galaktionov, L.Z.Shapiro

- Двухнаправленная трассировка лучей для интегрирования освещенности методом Квази-Монте Карло, А.Г. Волобой, В.А. Галактионов, К.А. Дмитриев, Э.А. Копылов





разных параллелепипедов, если они содержат объекты различных типов. Каждый объект имеет свою ускоряющую структуру, наиболее оптимальную для него. При этом все копии одного объекта в сцене (т.е. все объекты сцены, которые можно получить один из другого применением аффинного преобразования) являются одним типом объекта, имеют одинаковые ускоряющие структуры для трассировки нижнего уровня, но являются разными включениями в структуры трассировки верхнего уровня.



\*Рис. 4 - Организация сцены для двухуровневой трассировки лучей. Контурами выделены параллелепипеды, ограничивающие объекты сцены.

При изменениях сцены ускоряющие структуры модифицируются следующим образом:

- при добавлении копии объекта в сцену модифицируется только ускоряющая структура трассировки верхнего уровня;
- при удалении объекта из сцены модифицируется только ускоряющая структура трассировки верхнего уровня;
- при движении объекта, его поворотах или масштабировании модифицируется только ускоряющая структура трассировки верхнего уровня;
- при добавлении в сцену нового, уникального объекта создается его ускоряющая структура и модифицируется ускоряющая структура трассировки верхнего уровня.

Одно из часто используемых действий при построении сцены - это добавление объектов в сцену из библиотеки. Библиотеки содержат объекты, подготовленные заранее и содержащие также и ускоряющие структуры для трассировки объекта. В результате отпадает необходимость создавать вспомогательные структуры в динамическом режиме, и при таком действии требуется только модификация структур трассировки верхнего уровня.

Вторым важным преимуществом двухуровневой трассировки является то, что она позволяет обеспечить эффективную трассировку объектов различной природы. Для объектов разных типов хранятся разные вспомогательные структуры наиболее эффективные для каждого типа. На первом этапе разработки системы визуализации рассматривались только геометрические объекты, состоящие из массивов треугольников.

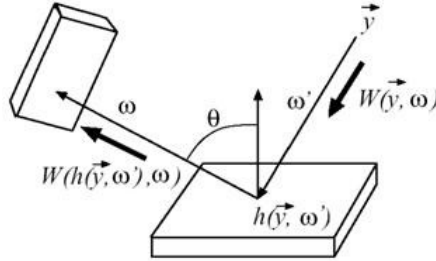
Двухуровневая трассировка лучей используется для построения фотореалистичных изображений, для оптического моделирования и ряда вспомогательных операций, необходимых в процессе назначения атрибутов, выбора объектов, анализа сцены и т.д. Время перестроения ускоряющих структур для сцен, состоящих из нескольких десятков объектов и сотен тысяч треугольников, составляет миллисекунды. Оно становится заметным только для сцен, состоящих из тысяч объектов.

## 4.2 Статья 2 : Двухнаправленная трассировка лучей

Рассмотрим статью "Двухнаправленная трассировка лучей для интегрирования освещенности методом Квази-Монте Карло" А.Г. Волобой, В.А. Галактионов, К.А. Дмитриев, Э.А. Копылов. [4]

Алгоритмы синтеза реалистических изображений с учетом полного моделирования освещенности направлены на решение уравнения рендеринга. Описывая излучательность  $L(\vec{x}, v)$  в точке поверхности  $\vec{x}$  по направлению  $v$ , уравнение рендеринга имеет рекуррентный вид:  $L(\vec{x}, v) = L^e(\vec{x}, v) + \Pi L(\vec{x}, v)$ , где  $L^e$  - излучательность точки поверхности от прямого освещения, а интегральный оператор  $\Pi$ , с учетом функции видимости точек сцены  $h(\vec{x}, v)$ , описывает взаимодействие света пришедшего от других освещенных поверхностей:

$$\Pi L(\vec{x}, v) = \int_{\Omega} L(h(\vec{x}, -v'), v') f_r(v', \vec{x}, v) \cos \theta' dv'$$



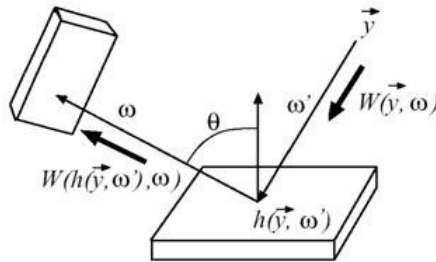
\*Рис. 5 - Компоненты уравнения излучательности.

Оптические свойства поверхности в уравнении заданы функцией двухнаправленного отражения (преломления)  $f_r$  представляющей отношение яркости поверхности к ее освещенности:  $f_r(v', \vec{x}, v) = dL(\vec{x}, v) / dL(\vec{x}, v')$

Наряду с энергетическими формами уравнения рендеринга рассматривают его сопряженный вид, описывающий функцию потенциальной видимости  $W(\vec{y}, v')$  из положения наблюдения  $\vec{y}$  в направлении  $v'$ :  $W(\vec{y}, v') = W^e(\vec{y}, v') + \Pi^* W(\vec{y}, v')$

где  $W^e$  - прямая видимость в направлении наблюдения, а интегральный оператор  $\Pi^*$  сопряжен с оператором  $\Pi$ :

$$\Pi^* W(\vec{y}, v') = \int_{\Omega} W(h(\vec{y}, v'), v) f_r^*(v, h(\vec{y}, v'), v') \cos \theta dv$$



\*Рис. 6 - Компоненты уравнения потенциальной видимости.

Методы построения изображений на основе решения уравнения рендеринга для излучательности носят название прямых методов, в то время как обратные методы для тех же целей используют уравнение потенциальной видимости. Общая классификация способов решения интегральных уравнений для любого из методов приведена в и сводится к трем типам: обращение, итерирование и распространение. Наиболее гибкий и универсальный механизм взаимодействия света с точкой поверхности обеспечивает метод распространения, анализу преимуществ которого посвящена работа. Применительно к уравнениям рендеринга для излучательности и потенциальной видимости метод распространения сводит решение к нахождению интегральных сумм:

для ур-я излучательности  
(прямая трассировка)

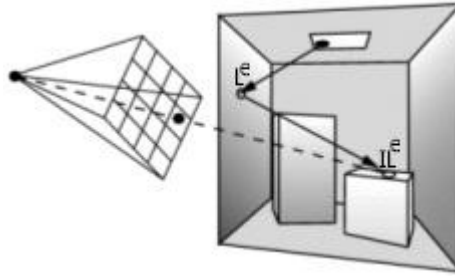
$$\begin{aligned}
 L &= L^e + IL \Rightarrow \\
 L &= L^e + I(L^e + IL) \Rightarrow \\
 L &= L^e + IL^e + I^2L \Rightarrow \\
 L &= \sum_{i=0}^n I^i L^e + I^{n+1}L \Rightarrow \\
 L &\xrightarrow{\lim_{n \rightarrow \infty} I^{n+1}L=0} \sum_{i=0}^{\infty} I^i L^e
 \end{aligned}$$

для ур-я видимости  
(обратная трассировка)

$$\begin{aligned}
 W &= W^e + I^*W \Rightarrow \\
 W &= W^e + I^*(W^e + I^*W) \Rightarrow \\
 W &= W^e + I^*W^e + I^{*2}W \Rightarrow \\
 W &= \sum_{i=0}^n I^{*i} W^e + I^{*n+1}W \Rightarrow \\
 W &\xrightarrow{\lim_{n \rightarrow \infty} I^{*n+1}W=0} \sum_{i=0}^{\infty} I^{*i} W^e
 \end{aligned}$$

\*Рис. 7 - Метод распространения.

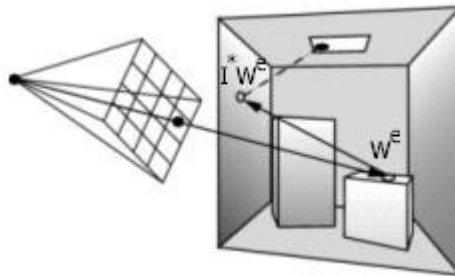
Прямая трассировка лучей, использующая метод распространения для решения уравнения излучательности, носит название трассировки фотонов. Распространение фотонов в качестве бесконечного ряда  $\sum I^i L^e$  имеет простой интуитивный смысл:



\*Рис. 8 - Прямая трассировка лучей.

$L^e$  вносит вклад прямого освещения,  $I^1 L^e$  - вклад света, пришедшего из источника после первого отражения,  $I^2 L^e$  - вклад после двух переотражений и т.д. Нетрудно заметить, что член  $I^d L^e$  имеет размерность  $2d+2$ , что естественно сразу же приводит к высоким размерностям интегрирования.

Обратную трассировку лучей, дающую решение уравнения потенциальной видимости, называют трассировкой путей. Каждый трассируемый путь берет начало от наблюдателя и вносит вклад в соответствующий пиксель экрана, непосредственно формируя изображение на экране. Как сопряженный метод, трассировка путей, очевидно, имеет ту же размерность, что и трассировка фотонов.



\*Рис. 9 - Обратная трассировка лучей.

Широкое использование в алгоритмах распространения света для построения реалистичских изображений получили двунаправленные методы, комбинирующие прямую и обратную трассировку лучей. Эффективная реализация двунаправленных методов осуществляется перераспределением трассируемых лучей, при котором трассируются лишь лучи существенно влияющие на формируемое изображение.

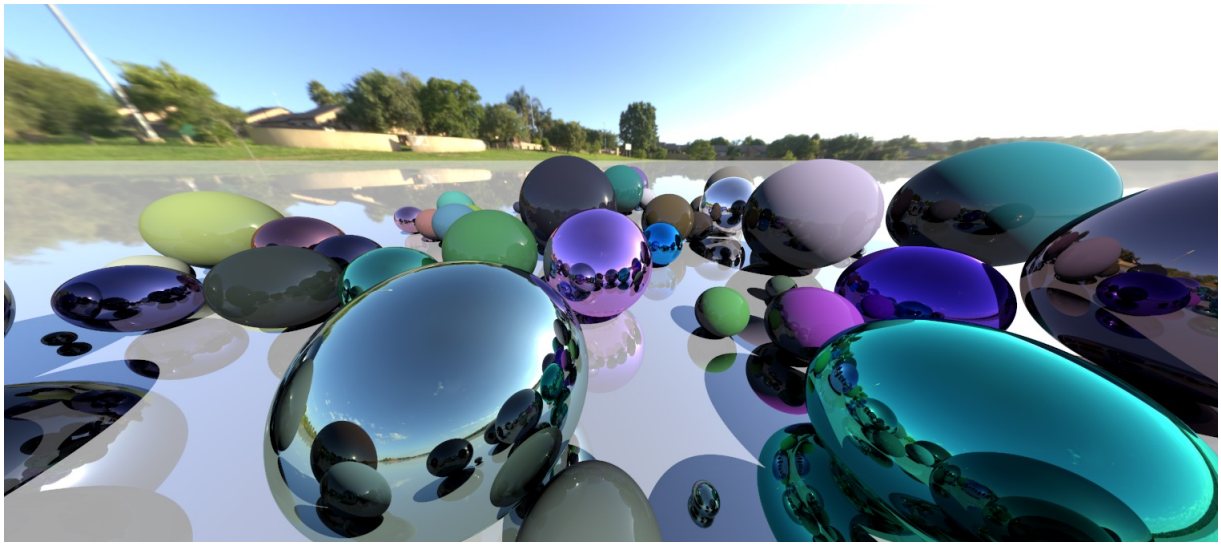
## 5 Результаты практики

Была предпринята попытка реализовать обратную трассировку лучей. Для этого была выбрана межплатформенная среда разработки компьютерных игр Unity.

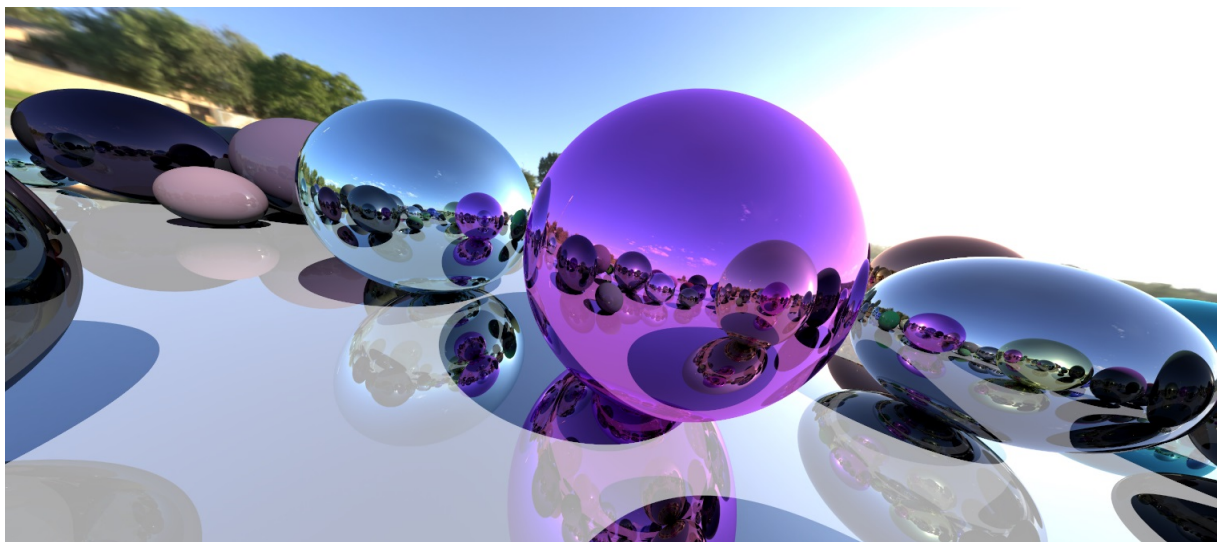
В результате работы был реализован простой трассировщик лучей, способный рендерить множество сфер с зеркальными отражениями, простым рассеянным освещением и резкими тенями. Также есть возможность настраивать количество сфер, их размер, расположение, цвет и коэффициенты зеркальности и альбедо.

Для реализации трассировщика был написан скрипт на C# и compute shader на HLSL, а также несколько скриптов на C# для управления камерой в реальном времени.

Примеры полученных изображений:



\*Рис. 10 - Полученное изображение №1.



\*Рис. 11 - Полученное изображение №2.

## 6 Заключение

В ходе проделанной работы были достигнуты поставленные цели.

### 1. Исследованы различные публикации на поставленную тему.

По рассмотренным двум статьям можно сделать вывод, что в зависимости от поставленной цели и необходимого результата можно применять разные методы трассировки лучей. Метод из первой статьи (двухуровневая трассировка лучей) стоит применять в сложных сценах, когда имеются объекты, которые можно представить в виде большого числа треугольников. Метод из второй статьи (двунаправленная трассировка лучей) подразумевает решение оптимизационной задачи, минимизирующей количество трассируемых лучей, не приводящих, в конечном счете, к переносу энергии от источника света до камеры наблюдения.

### 2. Изучены следующие методы трассировки лучей:

- (a) Прямая трассировка лучей.
- (b) Обратная трассировка лучей.
- (c) Двухуровневая трассировка лучей.
- (d) Двунаправленная трассировка лучей.

### 3. Проведены практические эксперименты.

В дальнейшем планирую добавить к моему трассировщику лучей рассеянное глобальное освещение, мягкие тени и применить вместо сфер треугольники.

## 7 Ссылки

1. Physically Based Rendering, Matt Pharr, Wenzel Jakob и Greg Humphreys
2. Методы глобального освещения для интерактивного синтеза изображений сложных сцен на графических процессорах, Боголепов Д. К.
3. Физика 8 класс, Перышкин А. В.
4. Двухнаправленная трассировка лучей для интегрирования освещенности методом Квази-Монте Карло, А.Г. Волобой, В.А. Галактионов, К.А. Дмитриев, Э.А. Копылов
5. The two level ray tracing algorithm and its elaboration toward Optical elements Preprint, Inst. Appl. Math., the Russian Academy of Science, B.N.Barladyan, A.G.Voloboy, V.A.Galaktionov, L.Z.Shapiro
6. Методы решения проблемы глобальной освещенности на графических процессорах, Фролов Владимир Александрович
7. Элементы глобального освещения сцены. Классическая трассировка лучей. Турлапов В.Е., Боголепов Д.К.
8. The Book of Shaders, Patricio Gonzalez Vivo, Jen Lowe
9. Unity в действии, Джозеф Хокинг
10. Introduction to Game Design, Prototyping, and Development: From Concept to Playable Game with Unity and C#, Jeremy Gibson Bond