ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

**Факультет информатики, математики и компьютерных наук**

**Программа подготовки бакалавров по направлению   
09.03.04 Программная инженерия**

*Канделов Дамир Русланович*

**КУРСОВАЯ РАБОТА**

Сравнение различных стратегий выборки в рендеринге трассировкой путей

|  |  |
| --- | --- |
|  | Научный руководитель  Старший научный сотрудник  программной инженерии  факультета компьютерных наук  канд. физ. мат. наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  И.С. Бычков |

Нижний Новгород, 2024

**Содержание**

[Введение 3](#_Toc165849438)

[Постановка задачи 5](#_Toc165849439)

[1. Исследование предметной области 6](#_Toc165849440)

[1.1. Что такое Path tracing? 6](#_Toc165849441)

[1.2. Отличие Path tracing от Ray tracing 7](#_Toc165849442)

[1.3. Алгоритм работы Path tracing 8](#_Toc165849443)

[2. Описание решения 11](#_Toc165849444)

[3. Библиографический список 12](#_Toc165849445)

# Введение

В настоящее время компьютерная графика, а именно методы рендеринга изображений развиваются очень быстро. Рендеринг – это процесс генерации фотореалистичных или нереалистичных изображений по заданным 2D или 3D моделям, с помощью компьютерных программ. Он используется в киноиндустрии при создании визуальных эффектов (VFX) в фильмах, в разработке компьютерных игр и симуляторов для отрисовки сцен, которые пользователь будет видеть на экране, а также в сферах дизайна и архитектуры.

Существует несколько алгоритмов рендеринга для создания фотореалистичных изображений, наиболее известными являются:

Rasterization (Растеризация) – это процесс перевода изображения, описанного в векторном формате, в растровое изображение. По сравнению с другими техниками рендеринга является одной из наиболее быстрых, из-за чего используется в большом количестве 3D движков, отрисовывающих изображение в реальном времени.

Ray tracing (Трассировка лучей) – этот метод способен симулировать большое количество оптических эффектов, таких как отражения, преломления и дисперсию света, мягкие тени, глубину резкости, размытие в движении, каустику.

Path tracing (Трассировка путей) – это метод, наиболее близкий к реальности по поведению глобального освещения. Алгоритм высчитывает освещенность в точке, исходя из освещения от каждого источника света. Если метод используется вместе с физически корректными материалами и объектами, то в результате получаются изображения, неотличимые от фотографий.

Основными потребителями алгоритмов рендеринга являются кино- и гейм-дев индустрии. Последние тенденции в обоих индустриях – создание реалистичной графики, как можно заметить, алгоритм Path tracing подходит для решения этой задачи. Однако, основным недостатком является большая вычислительная сложность данного алгоритма. Было замечено, что от метода выбора случайных чисел зависит качество и время получения изображений хорошего качества.

Таким образом, востребованность и актуальность темы данной работы обусловлена потребностями сферы разработки игр и сферы создания визуальных эффектов в кино, нацеленных на получение реалистичной графики в своих проектах за сравнительно малое время работы.

# Постановка задачи

В данной работе будет подробно рассмотрен алгоритм Path tracing. На примере его работы мы разберем и сравним несколько наиболее популярных подходов к генерации случайных чисел.

Цель данного проекта: реализовать несколько подходов к генерации случайных чисел и добавить эту реализацию в программу для рендеринга изображений с возможностью выбора алгоритма генерации случайных чисел и выделить методы, которые выдают наилучший результат за наименьшее время.

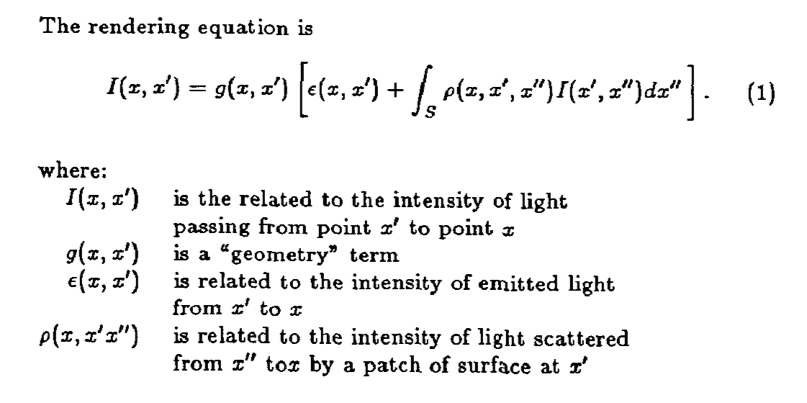
Задачами данного проекта являются:

1. Разобраться в алгоритме работы Path tracing и написать его с самого начала.
2. Узнать, где в алгоритме используются случайные числа и почему метод их выбора влияет на результирующие изображения.
3. Реализовать несколько подходов к генерации случайных чисел и имплементировать их в собственном проекте.
4. Сгенерировать и сравнить изображения, полученные с помощью новых методов выбора случайных чисел.

# Исследование предметной области

## Что такое Path tracing?

Path tracing – это метод рендеринга изображений по 3D сценам в компьютерной графике, который стремится воссоздать изображение максимально близкое к реальности. Данный метод реализует все визуальные эффекты.



«The Rendering Equation» Jim Kajiya, 1986

В 1986 году Jim Kajiya выпустил статью под названием «The Rendering Equation», которая объединяла компьютерную графику и физику. В ней было представлено уравнение, описывающие поведение света и позволяющее точно представить, как свет движется по 3D сцене. В этой же статье было и название нового метода рендеринга: **Path tracing.**

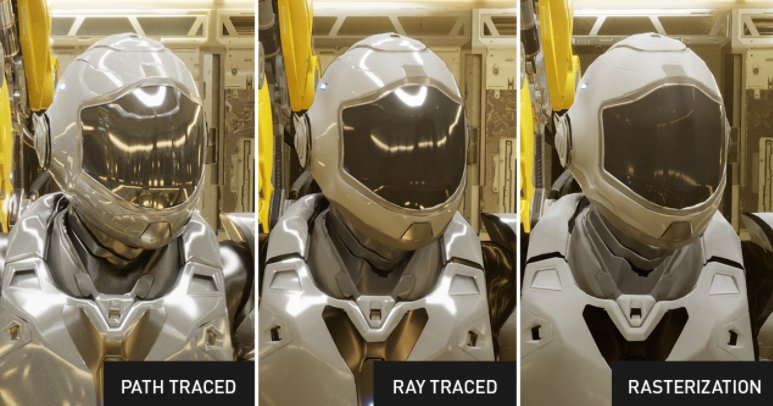
Полученное уравнение рендеринга было довольно небольшим и понятным, однако решить его не тек просто из-за того, что сцены в компьютерной графике очень сложны и описываются миллионами или даже миллиардами треугольников в современном мире. Нельзя полностью решить это уравнение.

Jim Kajiya предложил решение и этой проблемы: не требуется решать уравнение напрямую, его можно решить по путям для каждого отдельного луча (например, с помощью алгоритма Ray tracing). Соответственно, получение фотореалистичных изображений возможно, если решить это уравнение для достаточного количество лучей.

## Отличие Path tracing от Ray tracing

Одним из наиболее частых вопросов при знакомстве с методами рендеринга является вопрос отличия Path tracing и Ray tracing. Как уже было сказано ранее, в алгоритме Path tracing используется Ray tracing, однако следует разделять эти два метода рендеринга.

Основное отличие Path tracing от Ray tracing в том, что в трассировке путей мы следуем за лучом до тех пор, пока он не достигнет источника, чтобы получить цветовую и световую характеристику. В результате чего мы получаем более физически-корректное и реалистичное изображение.



Сравнение Path tracing и Ray tracing [1]

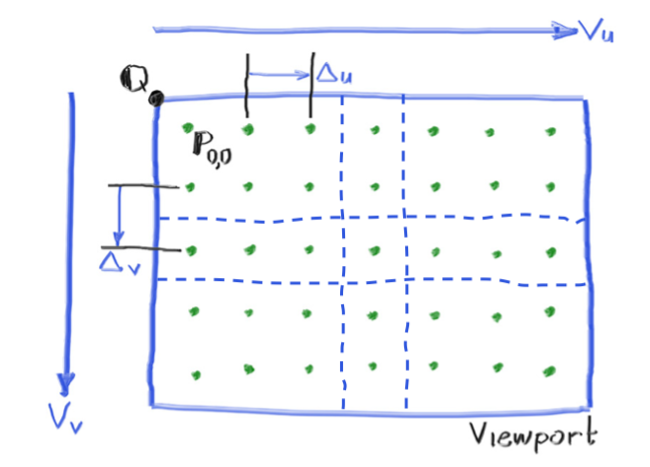
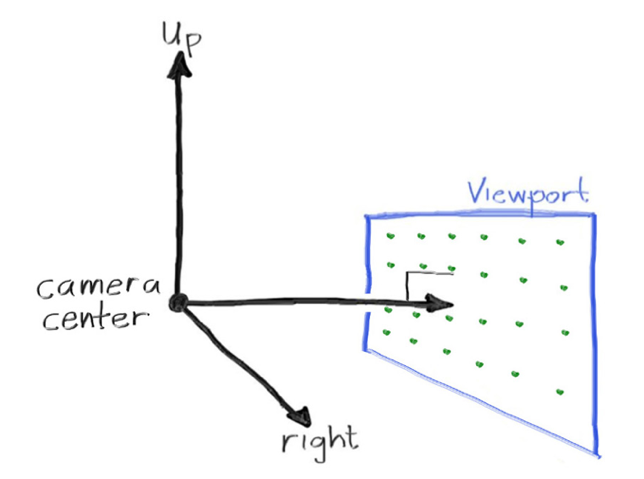
Как мы видим, в данном примере явно видно, что Path tracing симулирует большее количество света, что гораздо ближе к реальности: в отражениях мы должны хорошо видеть источники света.

## Алгоритм работы Path tracing

Давайте разберемся, как же это работает. Как было сказано выше, алгоритм максимально приближен к тому, как это происходит в реальном мире. Но с одним исключением.

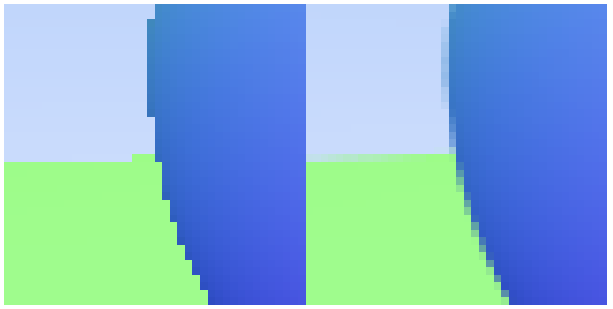
В реальном мире свет распространяется от источника света, отражается от объектов и попадает на сетчатку глаза. Однако реализация такого поведения имеет слишком большую вычислительную сложность, так как требуется запустить лучи во все стороны от источника света (так как свет распространяется во все стороны). Поэтому в компьютерной графике было принято решение развернуть эту последовательность действий: запускать лучи не из источников света, а из камеры (позиции, откуда пользователь смотрит на 3D сцену).

Далее, требуется понять, куда запускать лучи из камеры в пространстве.



Генерация лучей из камеры [2]

Заранее задаются параметры камеры: размер изображения, количество пикселей и т.д. На этом этапе мы создаем Viewport – матрица пикселей изображения, расположенная между камерой и сценой. На ней мы явно выделяем пиксели и, чтобы получить цвет пикселя требуется запустить луч через этот пиксель. Здесь мы и встречаемся с первым примером использования случайных чисел: если запускать лучи только через центры пикселей (зеленые точки на рисунке), то мы можем неправильно вывести цвет для данного пикселя. Самый простой пример такого явления – рендеринг краев объектов, например шара, когда мы запускаем один луч через центр, мы получаем только один цвет. Допустим луч попал в шар, то есть, грубо говоря, функция вернет цвет шара, а луч через следующий пиксель, например, уже не попадет в шар и вернет цвет фона. Получается грубая граница объекта (левая картинка). Это отличается от того, как мы видим это в жизни. Здесь на помощь приходит эффект сглаживания (Antialiasing): будем запускать не один луч через центр пикселя, а несколько через разные точки внутри пикселя. Чтобы получить эти точки нам нужно взять случайные координаты внутри пикселя, затем нужно просто взять среднее значение от получившихся цветов и записать в текущий пиксель.

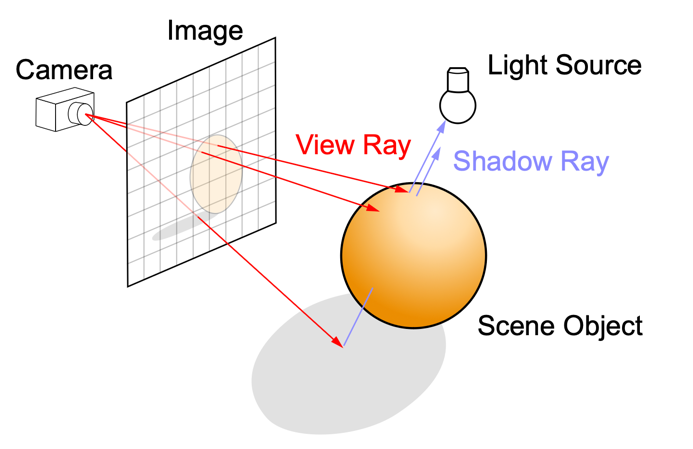


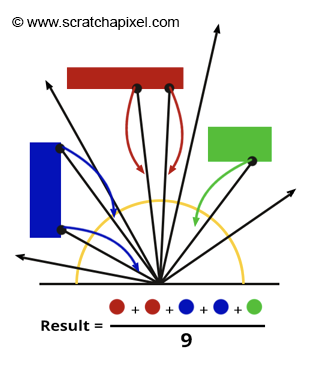
Эффект сглаживания [2]

(слева из камеры летит один луч через центры пикселей,

справа – несколько лучей, через случайные точки внутри пикселя)

Здесь мы впервые сталкиваемся





# Описание решения

# Библиографический список

<https://blogs.nvidia.com/blog/what-is-path-tracing/>

<https://raytracing.github.io/books/RayTracingInOneWeekend.html>

<https://www.techspot.com/article/2485-path-tracing-vs-ray-tracing/#:~:text=Traditional%20ray%20tracing%20involves%20calculating,off%20in%20a%20random%20direction>.

<https://dl.acm.org/doi/pdf/10.1145/15922.15902>

<https://raytracing.github.io/books/RayTracingTheNextWeek.html>

<https://raytracing.github.io/books/RayTracingTheRestOfYourLife.html>

<https://pbr-book.org/4ed/contents>

<https://jcgt.org/published/0009/03/02/>

<https://psychopath.io>

<https://www.scratchapixel.com/index.html>

<https://developer.nvidia.com/discover/ray-tracing#:~:text=Ray%20tracing%20generates%20computer%20graphics,back%20to%20the%20light%20sources>.

**Blue-noise Dithered Sampling** Iliyan Georgiev Marcos Fajardo