



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н. Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н. Э. Баумана)

---

ФАКУЛЬТЕТ «Информатика, искусственный интеллект и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

# РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

## *К КУРСОВОЙ РАБОТЕ*

### *НА ТЕМУ:*

*«Визуализация воды с использованием вокселей»*

*2023 г.*

# СОДЕРЖАНИЕ

<b>1</b>	<b>Введение</b>	<b>3</b>
<b>2</b>	<b>Аналитический раздел</b>	<b>5</b>
2.1	Выбор оборудования . . . . .	5
2.2	Выбор алгоритма отрисовки . . . . .	5
2.2.1	Алгоритм, использующий Z-буффер . . . . .	6
2.2.2	Трассировка лучей . . . . .	6
2.3	Выбор алгоритма обхода воксельной сцены . . . . .	6
2.3.1	Алгоритм быстрого обхода вокселей для трассировки лучей	7
2.3.2	KD-дерево . . . . .	7
2.3.3	Разреженное воксельное октодерево . . . . .	7
2.4	Выбор модели освещения . . . . .	7
2.4.1	Простая закраска . . . . .	7
2.4.2	Модель освещения Фонга . . . . .	7
2.4.3	Физически-корректная модель . . . . .	7
2.5	Вывод . . . . .	8
	<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</b>	<b>9</b>

# 1 Введение

Термин "Компьютерная графика" обозначает любое использование компьютера для создания и манипуляции изображениями. Задачи и области применения компьютерной графики сложно охарактеризовать в силу их большого количества, однако в общем случае можно выделить следующие направления:

- **Моделирование** – занимается математическим описанием формы и внешнего вида в формате, пригодным для использования в компьютере;
- **Рендеринг** или **Отрисовка** – термин, заимствованный из изобразительного искусства, обозначающий создание изображений из компьютерных моделей;
- **Анимация** – это техника создания иллюзии движения используя последовательность изображений [1].

Рендеринг – это основная составляющая компьютерной графики. На самом высоком уровне абстракции рендеринг является процессом преобразования описания трехмерной сцены в изображение. Алгоритмы для создания анимации, геометрического моделирования, нанесения текстур и других областей компьютерной графики должны проходить через некий процесс рендеринга, чтобы их результаты могли быть видны на изображении. Рендеринг стал повсеместным: от кино до игр и далее, он открыл новые горизонты для творческого выражения, развлечения и визуализации.

В первые годы развития этой области исследования в графическом рендеринге сосредоточивались на решении фундаментальных проблем, таких как определение, какие объекты видны из определенной точки обзора. По мере нахождения эффективных решений для этих проблем и доступности более богатых и реалистичных описаний сцен благодаря продолжающемуся прогрессу в других областях графики, современный рендеринг начал включать идеи из широкого спектра дисциплин, включая физику и астрофизику, астрономию, биологию, психологию и изучение восприятия, а также чистую и прикладную математику. Междисциплинарный характер рендеринга является одной из причин, почему это такая увлекательная область исследований.

В последние годы активно стала развиваться фотореалистичная компьютерная графика. Это дисциплина находится на стыке физики и классической

компьютерной графики. Высокий реализм изображений требует больших вычислительных мощностей для их получения, что заставляет разработчиков постоянно искать новые, более эффективные способы рендеринга.

Фотореалистичная компьютерная графика теперь повсеместно используется, включая такие области как развлечения, в частности, кино и видеоигры, дизайн продуктов и архитектура. За последнее десятилетие широкое распространение получили физически ориентированные методы визуализации, где точное моделирование физики рассеяния света является основой синтеза изображений. Эти подходы обеспечивают как визуальный реализм, так и предсказуемость [2].

Цель данной работы – реализовать программу для построения реалистичных изображений трехмерных воксельных сцен в реальном времени, с возможностями физически-корректной визуализации воды.

Чтобы достигнуть поставленной цели, требуется решить следующие задачи.

1. Описать структуру трехмерной сцены, включая объекты, из которых состоит сцена и их материалы, и определить способ задания исходных данных;
2. Выбрать или разработать алгоритмы компьютерной графики, позволяющие визуализировать трехмерную воксельную сцену в реальном времени;
3. Реализовать выбранные и адаптированные алгоритмы построения трехмерной сцены;
4. Исследовать возможности улучшения производительности программы и повышения сложности задаваемых сцен.

## **2 Аналитический раздел**

В данном разделе будут рассмотрены возможности отрисовки трехмерных изображений в реальном времени, проанализированы особенности трехмерной воксельной графики; будет выбран метод решения поставленной задачи.

### **2.1 Выбор оборудования**

Построение трехмерных изображений – вычислительно сложная, и в то же время, чрезвычайно параллельная задача [3]. Такие особенности стали причиной появления специального оборудования для выполнения популярных задач компьютерной графики. Такие вычислительные устройства называются Графическими процессорами (GPU), хотя их применение не ограничивается вычислениями графических данных. Появление таких устройств вызвано ограничениями в дизайне центральных процессоров. Они используются для последовательного выполнения кода с большим числом условных переходов, на небольшом числе ядер при симметричной многопроцессорности (SMP). И несмотря на развитие SIMD парадигмы, позволяющей обрабатывать большее число данных, чем позволяет последовательное исполнение, ЦПУ не смогли стать универсальными устройствами для рендеринга. Особенностью графических сопроцессоров является большое число ядер, способных эффективно выполнять тысячи вычислений параллельно [4].

Несмотря на большие возможности параллельного выполнения вычислительных задач, графические сопроцессоры имеют органичные возможности условного выполнения программ, в частности, рекурсивных, малую скорость памяти, и возможности взаимодействия с ней.

Такие особенности породили целый ряд алгоритмов, оптимизированных для выполнения на графических сопроцессорах, которые отличаются от аналогичных для процессоров общего назначения. В частности, графические сопроцессоры могут иметь специальный аппаратные блоки для ускоренного выполнения алгоритмов трассировки лучей. [5]

### **2.2 Выбор алгоритма отрисовки**

### **2.2.1 Алгоритм, использующий Z-буффер**

123 [6]

### **2.2.2 Трассировка лучей**

132 [2]

**Метод Монте-Карло**

123 [2]

**Алгоритм кеширования обратной репроекции**

123 [7]

## **2.3 Выбор алгоритма обхода воксельной сцены**

Воксель (аналогично Пиксель) – способ представления геометрии, альтернативный типичному полигональному. Воксель представляет собой некоторое значение на регулярной решетке в трехмерном пространстве. Воксели часто используются при анализе медицинских и научных данных.

В интерактивных графических приложениях, выполняющих отрисовку в реальном времени, воксели редко применялись в качестве основного геометрического примитива. Это связано с тем, что графические сопроцессоры были специально оптимизированы для работы с полигональными данными, и они были более простым способом достичь требуемого качества изображения.

В последние годы замечается тенденция повышения популярности вокселей в интерактивных приложениях. Это связано с повышением мощности вычислительной техники, позволяющей выполнять ранее невозможные вычисления на пользовательских компьютерах. Воксельная графика позволяет достигать большей детализации, чем возможно при использовании полигонов.

Главная проблема воксельной графики – большое число затрачиваемой памяти и медленный доступ к ней. Поэтому все воксельные алгоритмы фокусируются на ускорении доступа к индивидуальным вокселям, для использования в алгоритмах отрисовки. Рассмотрим и выберем алгоритм обхода воксельной сцены и структуру данных хранения вокселей.

### **2.3.1 Алгоритм быстрого обхода вокселей для трассировки лучей**

123 [8]

### **2.3.2 KD-дерево**

123 [9]

### **2.3.3 Разреженное воксельное октодерево**

123 [10]

## **2.4 Выбор модели освещения**

123

### **2.4.1 Простая закраска**

123 [6]

### **2.4.2 Модель освещения Фонга**

123 [11]

### **2.4.3 Физически-корректная модель**

123 [2]

**Микрогранные модели**

123 [2] [MMfRrRS]

## 2.5 Вывод

Было принято решение выполнять разработку для выполнения на графическом сопроцессоре, поскольку это позволяет выполнять отрисовку значительно более эффективно, чем на ЦПУ.

В качестве алгоритма трассировки была выбрана обратная трассировка лучей с помощью метода Монте-Карло, поскольку он имеет лучшие возможности для получения физически-корректных изображений, чем алгоритм, использующий Z-буфер. Для оптимизации отрисовки был выбран способ кеширования обратной репроекции.

В качестве модели освещения была выбрана физически-корректная модель с использованием микрограней с моделью GGX, поскольку такой вариант предоставляет лучший вариант получения реалистичных изображений, чем другие модели освещения и модель микрограней Бекерманна [12].



## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Shirley P., Marschner S.* Fundamentals of Computer Graphics. — 3rd. — USA : A. K. Peters, Ltd., 2009. — ISBN 1568814690.
2. *Pharr M., Jakob W., Humphreys G.* Physically Based Rendering: From Theory to Implementation (3rd ed.) — 3rd. — San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 10.2016. — С. 1266. — ISBN 9780128006450.
3. *Foster I.* Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering. — 75 Arlington Street, Suite 300 Boston, MA United States : Addison-Wesley Longman Publishing Co., 01.1995. — ISBN 9780201575941.
4. *Fatahalian K., Houston M.* A Closer Look at GPUs. — 2008.
5. Ray Tracing In Vulkan / D. Koch [и др.]. — 2020. — Дек.
6. *Роджерс Д.* Алгоритмические основы машинной графики. — 1989.
7. Accelerating Real-Time Shading with Reverse Reprojection Caching / D. Nehab [и др.]. — 2007. — АВГ.
8. *Amanatides J., Woo A.* A Fast Voxel Traversal Algorithm for Ray Tracing // Proceedings of EuroGraphics. — 1987. — АВГ. — Т. 87.
9. *Foley T., Sugerman J.* KD-Tree Acceleration Structures for a GPU Raytracer. — 2005.
10. *Laine S., Karras T.* Efficient Sparse Voxel Octrees – Analysis, Extensions, and Implementation. — 2010.
11. *Phong B. T.* Illumination for Computer Generated Pictures. — 1975.
12. Microfacet Models for Refraction through Rough Surfaces / B. Walter [и др.]. — 2007.