



**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 и системы управления»**

**КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»**

## **Лабораторная работа № 4 по дисциплине «Анализ Алгоритмов»**

**Тема Параллельные вычисления на основе нативных потоков**

**Студент Пермякова Е. Д.**

**Группа ИУ7-52Б**

**Преподаватели Строганов Д. В., Волкова Л. Л**

Москва, 2024

# СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ</b>	<b>4</b>
<b>1 Входные и выходные данные</b>	<b>5</b>
<b>2 Преобразование входных данных в выходные</b>	<b>5</b>
<b>3 Примеры работы программы</b>	<b>5</b>
<b>4 Тестирование</b>	<b>6</b>
<b>5 Описание исследования</b>	<b>7</b>
<b>ЗАКЛЮЧЕНИЕ</b>	<b>8</b>
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</b>	<b>9</b>

# ВВЕДЕНИЕ

Целью работы является исследование зависимости времени отрисовки сцены от числа полигонов сцены для варьируемого числа рабочих потоков.

Задачи:

- 1) реализовать алгоритма трассировки лучей для визуализации трёхмерной сцены, состоящей из моделей, заданных треугольными полигонами;
- 2) исследовать зависимость времени отрисовки сцены от числа полигонов сцены для варьируемого числа рабочих потоков. Изменять количество дополнительных потоков от 0 (вычисление в основном потоке), до  $4 \cdot k$ , где  $k$  — количество логических ядер используемой ЭВМ, по степеням числа 2;

# 1 Входные и выходные данные

Входными данными программы является информация о сцене (об объектах которые на ней находятся). Модели на сцене задаются множеством треугольных полигонов. Выходными данными является растровое изображение.

## 2 Преобразование входных данных в выходные

Программа определяет цвет каждого пикселя растрового изображения методом трассировки лучей.

## 3 Примеры работы программы

На рисунке 3.1-3.2 представлены примеры работы программы.

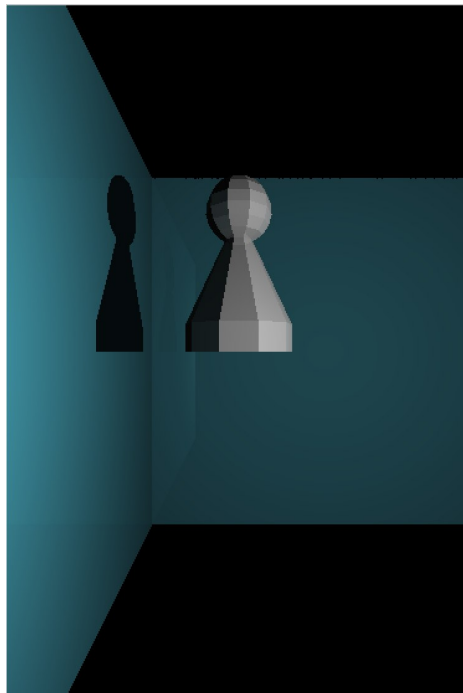


Рисунок 3.1 — Пример работы программы

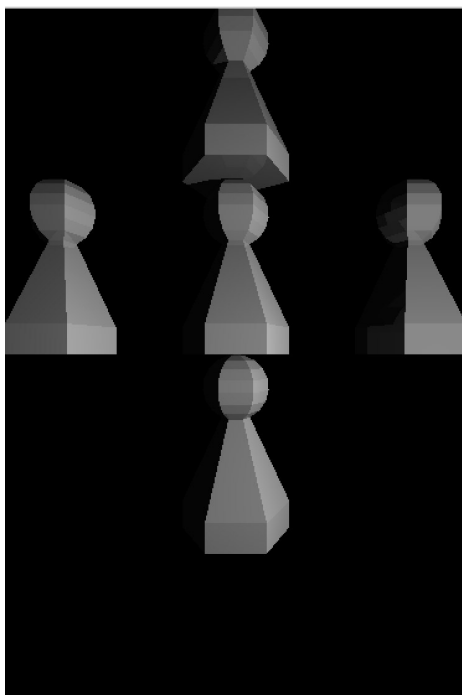


Рисунок 3.2 — Пример работы программы

## 4 Тестирование

Для тестирования были выделены следующие классы тестирования:

- 1) луч не пересекается ни с одним объектом;
- 2) луч пересекается с объектом один раз;
- 3) луч и пересекается с объектом и отразившись пересекается с другим объектом;
- 4) луч пересекается с объектом в тени другого объекта;

В таблице 4.1 представлены функциональные тесты алгоритма трассировки лучей. Все тесты пройдены успешно.

Таблица 4.1 — Функциональные тесты

№	Координаты пикселя	Цвет пикселя (RGB)	Ожидаемый цвет пикселя (RGB)
1	(190, 443)	R=0, G=0, B=0	R=0, G=0, B=0
2	(95, 238)	R=5, G=10, B=11	R=5, G=10, B=11
3	(131, 351)	R=25, G=59, B=66	R=25, G=59, B=66
4	(46, 511)	R=33, G=77, B=86	R=33, G=77, B=86

## 5 Описание исследования

Было проведено исследование зависимости времени отрисовки сцены от числа полигонов на сцене для варьируемого числа рабочих потоков.

Был выбран язык программирования с++, так как на linux с его помощью можно создавать нативные потоки [1]. Проводились замеры реального времени работы программы, для этого использовалась функция `gettimeofday` [2]. Замеры проводились на виртуальной машине Linux с 8 логическими ядрами, и замерялось время для дополнительных рабочих потоков в размере от 0 (вычисление в основном потоке) до 32, по степеням числа 2, то есть было рассмотрено 0, 1, 2, 4, 8, 16, 32 дополнительных рабочих потока.

Был получен график зависимости времени отрисовки сцены от числа полигонов на сцене для варьируемого числа рабочих потоков 5.1.

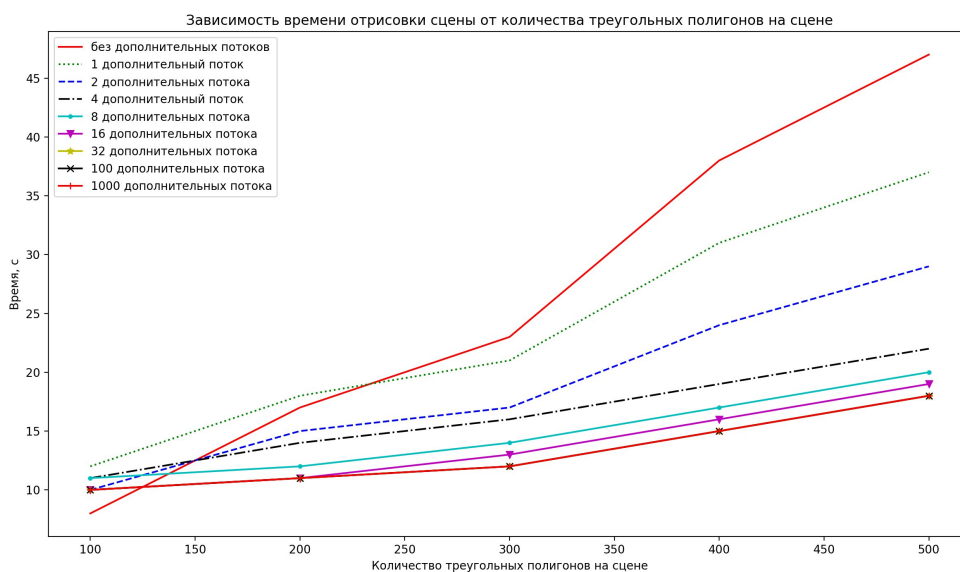


Рисунок 5.1 — Зависимость времени отрисовки сцены от числа полигонов на сцене для варьируемого числа рабочих потоков

Из проведённых замеров были сделаны следующие выводы:

- при увеличении количества потоков время обработки сцены уменьшается, если на сцене 200 и более полигонов;
- при небольшом заполнении сцены (количество полигонов = 100) алгоритм использующий последовательную обработку данных работает быстрее, так как не тратит время на создание потоков;

# ЗАКЛЮЧЕНИЕ

В ходе лабораторной работы была выполнена поставленная цель, которая заключалась в исследовании зависимости времени отрисовки сцены от числа полигонов сцены для варьируемого числа рабочих потоков.

Были выполнены следующие задачи:

- 1) реализовать алгоритма трассировки лучей для визуализации трёхмерной сцены, состоящей из моделей, заданных треугольными полигонами;
- 2) исследовать зависимость времени отрисовки сцены разработанным ПО от числа полигонов сцены для варьируемого числа рабочих потоков. Изменять количество дополнительных потоков от 0 (вычисление в основном потоке), до  $4 \cdot k$ , где  $k$  — количество логических ядер используемой ЭВМ, по степеням числа 2;

Основываясь на проведённом исследовании были сделаны следующие выводы.

- при увеличении количества потоков время обработки сцены уменьшается, если на сцене 200 и более полигонов;
- при небольшом заполнении сцены (количество полигонов = 100) алгоритм использующий последовательную обработку данных работает быстрее, так как не тратит время на создание потоков;

## **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. pthread\_create(3) — Linux manual page [Электронный ресурс]. Режим доступа: [https://man7.org/linux/man-pages/man3/pthread\\_create.3.html](https://man7.org/linux/man-pages/man3/pthread_create.3.html). (дата обращения: 20.10.2024).
2. gettimeofday(2) — Linux manual page [Электронный ресурс]. Режим доступа: <https://man7.org/linux/man-pages/man2/gettimeofday.2.html>. (дата обращения: 20.10.2024).