

## СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ</b>	<b>3</b>
<b>1 Аналитическая часть</b>	<b>4</b>
1.1 Формализация модели . . . . .	4
1.2 Выбор модели представления объекта . . . . .	4
1.3 Выбор модели освещения . . . . .	5
1.3.1 Локальная модель освещения . . . . .	5
1.3.2 Глобальная модель освещения . . . . .	5
1.3.3 Вывод . . . . .	5
1.4 Выбор алгоритма удаления невидимых ребер и поверхностей . .	5
1.4.1 Алгоритм обратной трассировки лучей . . . . .	6
1.4.2 Алгоритм испускания лучей (raycasting) . . . . .	6
1.4.3 Алгоритм развития лучей (raymarching) . . . . .	7
1.4.4 Вывод . . . . .	7
1.5 Выбор метода закраски . . . . .	8
1.5.1 Метод Гуро . . . . .	8
1.5.2 Метод Фонга . . . . .	8
1.5.3 Метод обратной трассировки лучей . . . . .	8
1.5.4 Вывод . . . . .	8
1.6 Вывод . . . . .	8
<b>ЗАКЛЮЧЕНИЕ</b>	<b>9</b>
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</b>	<b>10</b>

## ВВЕДЕНИЕ

Целью данной работы является получение набора методов и алгоритмов, позволяющих наиболее качественно реализовать программное обеспечение с пользовательским интерфейсом для моделирования зеркала, поверхность которого может иметь определенные оптические свойства.

Для достижения поставленной цели требуется решить следующие задачи:

- формализовать разрабатываемую модель;
- проанализировать модели представления объектов сцены, выбрать наиболее подходящую для достижения цели и обосновать выбор;
- проанализировать методы и алгоритмы удаления невидимых ребер и поверхностей, выбрать наиболее подходящий для достижения цели и обосновать выбор;
- проанализировать методы и алгоритмы закраски, модели освещения, выбрать наиболее подходящие для достижения цели и обосновать выбор.

# 1 Аналитическая часть

## 1.1 Формализация модели

Сцена состоит из:

- источника света,
- трехмерного объекта,
- наблюдателя,
- зеркала,
- ограничивающей плоскости.

*Источник света* является материальной точкой, из которой во все стороны исходят лучи света. Он не может отражать или преломлять свет. В частном случае, когда источник расположен в бесконечности, он имеет направленность. Положение источника света задается трехмерными координатами, цвет света описывается через RGB-параметры. Количество источников света не может меняться.

*Трехмерный объект* отображается с помощью модели из стандартного набора: пирамида, призма, шар. Основные параметры выбранной модели (например, количество углов основания у призмы) могут меняться.

Положение *наблюдателя* и *трехмерного объекта* задается координатами  $x$ ,  $y$ ,  $z$ . Они могут меняться посредством поворота, переноса.

*Зеркало* представляет собой трехмерный объект, свойства поверхности которого могут задаваться пользователем (степень полировки, цвет, радиус кривизны, размеры зеркала). Цвет поверхности описывается через RGB-параметры. Степень полировки и радиус кривизны задаются численно. Размеры зеркала представляются длиной и шириной, задаются численно. Глубина зеркала будет определяться программистом и не может меняться пользователем.

*Ограничивающая плоскость* представляет собой бесконечную плоскость, которая ограничивает нерабочую область зеркала и содержит его рабочую плоскость. Предполагается, что трехмерный объект и наблюдатель находятся со стороны рабочей поверхности зеркала.

## 1.2 Выбор модели представления объекта

Модели бывают: каркасные, поверхностные, объемные. [1]

Недостатком использования каркасной модели является неоднозначность получаемого результата, а объемной – избыточность для нашей задачи

информации о материале поверхности объекта.

Таким образом, будет использоваться *поверхностная* модель представления объекта, в которой его поверхность будет описываться, например, аналитически. Это позволит получать однозначный результат с необходимым минимумом информации об объекте.

### **1.3 Выбор модели освещения**

Предназначением модели освещения является расчет интенсивности отраженного от объектов света. Существует два вида таких моделей: локальная и глобальная. [2, 3]

#### **1.3.1 Локальная модель освещения**

Особенностью локальной модели освещения является то, что в них не учитывается влияние предметов друг на друга. Другими словами, в них на закраску объекта влияют лишь источники света.

#### **1.3.2 Глобальная модель освещения**

Глобальная же модель освещения учитывает также отражение света от других объектов или его прохождение сквозь них. Таким образом, возможно добиться изображения отражений.

#### **1.3.3 Вывод**

Исходя из того, что перед нами стоит цель моделирования поверхности зеркала, выбор падает на *глобальную модель освещения*. Таким образом, отпадает необходимость рассматривать конкретные примеры локальных моделей освещения.

### **1.4 Выбор алгоритма удаления невидимых ребер и поверхностей**

В зависимости от системы координат алгоритмы делятся на:

- работающие в пространстве экрана (изображения),
- работающие в пространстве сцены (объектов). [4]

Основной идеей алгоритмов, работающих в пространстве экрана, является нахождение видимых точек для каждого объекта и их отображение на экране в виде пикселей. В ходе их работы пиксель рисуется один раз, а отношения между объектами анализируются более одного раза. Объем вычислений для таких алгоритмов растет, как  $C * N$ , где  $C$  - количество пикселей,  $N$  - количество граней объекта.

Идеей же алгоритмов, работающих в пространстве сцены, является опе-

рирование геометрическими примитивами, проверка их пересечений и получение списка видимых объектов и их частей. В ходе их работы пиксель может перерисовываться более одного раза, а объект анализируется один раз. Объем вычислений для таких алгоритмов растет, как  $N^2$ , где  $N$  - количество граней объекта.

В рамках данной практической работы основными критериями выбора алгоритма удаления невидимых ребер и поверхностей является:

- точность результата работы алгоритма,
- возможность использования глобальной модели освещения.

Сведем описание алгоритмов к общему виду:

- пространство, в котором работает алгоритм (сцены, экранное);
- обрабатываемые объекты;
- суть алгоритма;
- возможность использования глобальной модели освещения;
- точность результатов.

#### **1.4.1 Алгоритм обратной трассировки лучей**

Характеристики алгоритма:

- пространство алгоритма – экранное;
- обрабатываются объекты, представленные поверхностными или объемными моделями;
- суть алгоритма заключается в отслеживании взаимодействия лучей, испускаемых из точки наблюдения, с описывающими объекты сцены сферами и/или параллелепипедами, что существенно сокращает количество вычислений по сравнению с прямой трассировкой, однако алгоритм все еще является ресурсозатратным [5, 6];
- имеется возможность использования глобальной модели освещения;
- высокая точность результатов, так как работа идет непосредственно с точками, принадлежащими объектам.

#### **1.4.2 Алгоритм испускания лучей (raycasting)**

Характеристики алгоритма:

- пространство алгоритма – экранное;
- обрабатываются объекты простой формы, представленные поверхностными моделями;
- суть алгоритма заключается в том, что для каждого луча, испуска-

емого из точки наблюдения, рассчитывается и запоминается расстояние, которое он прошел до столкновения с ближайшим объектом сцены, и в дальнейшем объекты визуализируются в зависимости от этого расстояния;

- отсутствует возможность использования глобальной модели освещения;
- высокая точность результатов, так как работа идет непосредственно с точками, принадлежащими объектам. [7–9]

### 1.4.3 Алгоритм развития лучей (raymarching)

Характеристики алгоритма:

- пространство алгоритма – экранное;
- обрабатываются объекты, представленные поверхностными и объемными моделями;
- суть алгоритма заключается в том, что для каждого луча, испускаемого из точки наблюдения, происходит его развитие вдоль заданного направления и нахождение пересечения с объектом, при этом для каждой точки развития луча определяется радиус, на который можно пустить луч, на основе которого и значения вспомогательной функции поля расстояний со знаком (SDF) делается вывод о пересечении;
- имеется возможность использования глобальной модели освещения;
- низкая точность результатов, так как работа идет не с точками объекта, а с вычисляемыми точками пересечения. [10]

### 1.4.4 Вывод

Для наглядности составим таблицу, содержащую информацию о каждом рассмотренном алгоритме (таблица 1.1)

Таблица 1.1 — Анализ алгоритмов удаления невидимых ребер и поверхностей

	Обратная трассировка лучей	Испускание лучей	Развитие лучей
Пространство, в котором работает алгоритм	Экранное	Экранное	Экранное
Сложность алгоритма	$O(C * N)$	$O(C * N)$	$O(C * N)$

Наличие возможности использования глобальной модели освещения	+	–	+
Высокая точность результатов	+	+	–

Исходя из формализации модели, описания алгоритмов и составленной таблицы был сделан выбор в сторону *алгоритма обратной трассировки лучей*.

### 1.5 Выбор метода закраски

Рассмотрим несколько методов закраски поверхностей: метод Гуро, метод Фонга, метод обратной трассировки лучей. [11]

#### 1.5.1 Метод Гуро

*Метод Гуро* основывается на билинейной интерполяции интенсивностей для создания иллюзии гладкой криволинейной поверхности.

#### 1.5.2 Метод Фонга

*Метод Фонга* заключается в билинейной интерполяции векторов нормалей граней. Это является еще более трудоемким процессом.

Стоит отметить, что оба эти метода применяются только к поверхностям представления объектов.

#### 1.5.3 Метод обратной трассировки лучей

*Метод обратной трассировки лучей* по сути уже был описан в предыдущем пункте. Для закраски объектов используется рекурсия – добавляются отраженные и преломленные лучи и так далее, пока не будет достигнут предел рекурсии.

#### 1.5.4 Вывод

Наиболее удобным и точным методом закраски для достижения поставленной цели будет являться *метод обратной трассировки лучей*.

### 1.6 Вывод

В ходе проведения анализа были выбраны: поверхностные модели представления объектов, глобальная модель освещения, метод обратной трассировки лучей (как для удаления невидимых ребер и поверхностей, так и для закраски).

## ЗАКЛЮЧЕНИЕ

В ходе практической работы был получен набор методов и алгоритмов, позволяющих наиболее качественно реализовать программное обеспечение с пользовательским интерфейсом для моделирования зеркала, поверхность которого может иметь определенные оптические свойства.

Для достижения цели были решены поставленные задачи, то есть:

- была формализована разрабатываемая модель;
- были проанализированы модели представления объектов сцены и выбрана наиболее подходящая для достижения цели с обоснованием выбора;
- были проанализированы методы и алгоритмы удаления невидимых ребер и поверхностей и выбран наиболее подходящий для достижения цели с обоснованием выбора;
- были проанализированы методы и алгоритмы закраски, модели освещения и выбраны наиболее подходящие для достижения цели с обоснованием выбора.



## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] Лисяк В.В. Основы компьютерной графики: 3D-моделирование и 3D-печать – Издательство Litres, 2021. – 111 с.
- [2] Модели освещения и алгоритмы затенения в компьютерной графике: учебное пособие / Задорожный А.Г. – Новосибирск: Изд-во НГТУ, 2020. – 80 с.
- [3] Глобальная модель освещения с трассировкой лучей [Электронный ресурс]. Режим доступа: (дата обращения - 23.07.23) свободный. URL: <http://www.mari-el.ru/mmlab/home/kg/Lecture12/3.html>
- [4] Головнин А.А. Базовые алгоритмы компьютерной графики / Головнин А.А. // Проблемы качества графической подготовки студентов в техническом ВУЗе: традиции и инновации – 2016. – № 1. – С. 13-30.
- [5] Порев В.Н. Компьютерная графика – СПб.: БХВ-Петербург, 2002. – 432 с.
- [6] Янова Р.Ю. Метод прямой и обратной трассировки / Янова Р.Ю. // Вестник науки и образования. – 2016. – № 5(17) – С. 29-30.
- [7] Евстратов В.В. Создание программы визуализации псевдотрехмерного изображения с помощью рейкастинга / В. В. Евстратов. – Текст : непосредственный // Молодой ученый. – 2020. – № 50 (340). – С. 12-15. – URL: <https://moluch.ru/archive/340/76504/> (дата обращения: 23.07.2023).
- [8] A ray casting method for the computation of the area of feasible solutions for multicomponent systems: Theory, applications and FASPACK-implementation [Электронный ресурс]. Режим доступа: (дата обращения - 23.07.23) свободный. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0003267016314209>
- [9] Обучение технологии ray-casting, часть 1 [Электронный ресурс]. Режим доступа: (дата обращения - 23.07.23) свободный. URL: <https://habr.com/ru/articles/515256/>

- [10] Biagioli A. Raymarching Distance Fields: Concepts and Implementation in Unity [Электронный ресурс]. Режим доступа: (дата обращения - 15.07.2023) свободный. URL: <https://adrianb.io/2016/10/01/raymarching.html>
- [11] Методы закрашки [Электронный ресурс]. Режим доступа: (дата обращения - 21.07.23) свободный. URL: [https://portal.tpu.ru/SHARED/j/JBOLOTOVA/academic/ComputerGraphics/8. Методы закрашки\\_2019.pdf](https://portal.tpu.ru/SHARED/j/JBOLOTOVA/academic/ComputerGraphics/8.Методы%20закраски_2019.pdf)