|  |  |
| --- | --- |
|  | **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» |

**ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ**

|  |  |
| --- | --- |
| Студент | Лагутин Даниил Валерьевич |
|  | фамилия, имя, отчество |
| Группа | ИУ7-43Б |
| Тип практики | технологическая |
| Название предприятия | МГТУ им. Н. Э. Баумана |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент |  |  |  | Лагутин Д. В. |
|  |  | подпись, дата |  | фамилия, и.о. |
| Руководитель практики |  |  |  | Куров А. В. |
|  |  | подпись, дата |  | фамилия, и.о. |

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оглавление

[Введение 3](#__RefHeading___Toc1314_1001217329)

[1 Аналитическая часть 4](#__RefHeading___Toc1316_1001217329)

[1.1 Анализ необходимых качеств реалистического изображения 4](#__RefHeading___Toc1780_1001217329)

[1.2 Формализация объектов синтезируемой сцены 5](#__RefHeading___Toc1318_1001217329)

[1.3 Анализ алгоритмов удаления невидимых линий и ребер поверхностей 6](#__RefHeading___Toc273_797087708)

[1.4 Анализ алгоритмов построения теней 8](#__RefHeading___Toc277_797087708)

[1.5 Анализ алгоритмов построения отражений и преломления 8](#__RefHeading___Toc279_797087708)

[1.6 Анализ алгоритмов реализации глубины резкости 8](#__RefHeading___Toc391_797087708)

[1.7 Анализ алгоритмов реализации подповерхностного рассеивания 9](#__RefHeading___Toc1782_1001217329)

[Вывод 9](#__RefHeading___Toc393_797087708)

[2 Конструкторская часть 10](#__RefHeading___Toc1320_1001217329)

[3 Технологическая часть 11](#__RefHeading___Toc1322_1001217329)

[Заключение 12](#__RefHeading___Toc1324_1001217329)

[Список использованной литературы 13](#__RefHeading___Toc1326_1001217329)

# Введение

Компьютерная графика является неотъемлемой частью современного мира в особенности она востребована в сфере кинопроизводства и игровой промышленности. Именно поэтому синтез реалистического изображения является одной из важнейших задач.

Подобные условия требуют создания средств и методик, учитывающих такие оптические явления как преломление, отражение и рассеивание света, зависящие от свойств визуализируемых объектов. Но рост точности и детальности разрабатываемых алгоритмов зачастую приводит к более высоким затратам по времени и памяти.

Целью данной работы является создание программы, позволяющей создать реалистическое изображение с учетом подповерхностного рассеивания, на основе трехмерной сцены наполненной объектами (сфера, куб, конус, цилиндр, полигональная модель) и источниками света, положение, количество и свойства которых задается пользователем.

Для достижения поставленной цели, требуется решить следующие задачи:

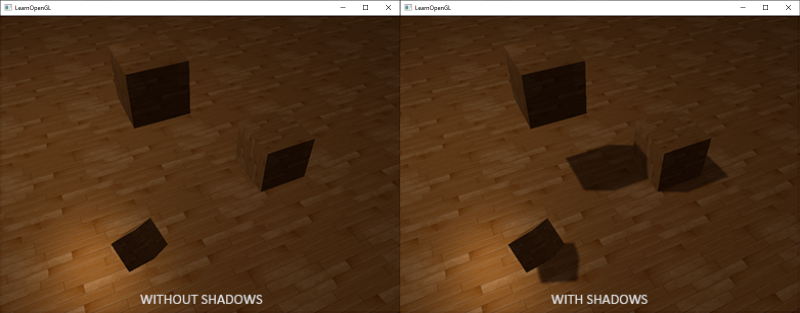
1. Выявить неотъемлемые качества реалистического изображения;
2. Провести анализ существующих алгоритмов и выбрать оптимальные пути для решения основной задачи;
3. Создать программный продукт для решения задачи, реализовать выбранные алгоритмы.
4. 1 Аналитическая часть
5. В этом разделе будут рассмотрены модели представления трехмерных тел, формализованы объекты, наполняющие сцену, и требования, предъявляемые к работе программы и конечному результату. Будут рассмотрены алгоритмы построения трехмерного изображения, методы закраски, модели освещения.

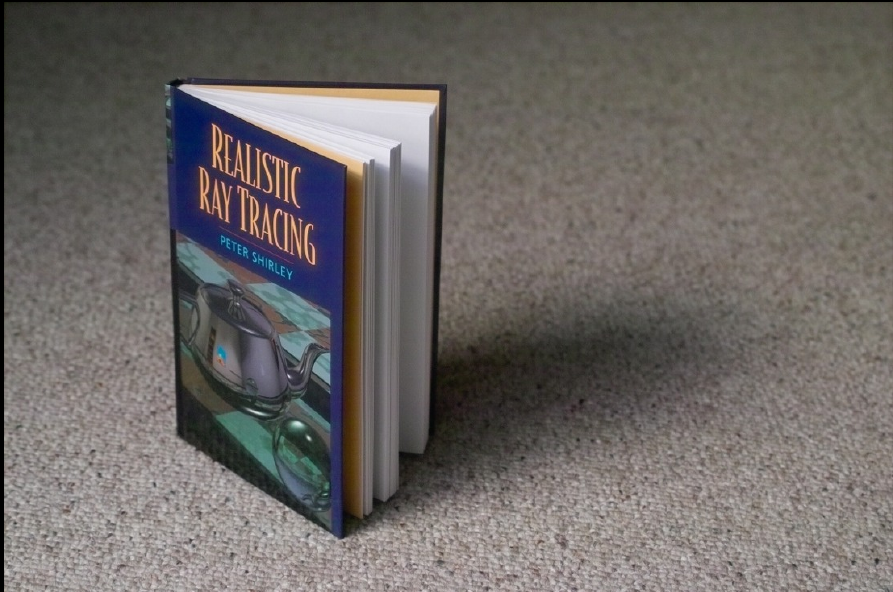
## 1.1 Анализ необходимых качеств реалистического изображения

Большинство алгоритмов, изученных в курсе компьютерной графики нацелены на корректное отображение геометрии объектов и их положения в сцене, однако для построения реалистического изображения этих аспектов недостаточно.

Первым аспектом, наполняющим сцену объемом можно считать тень.

Однако большинство моделей подразумевает источник света материальной точкой, из-за чего все тени получаются четкими и однородными, что в общем случае неверно.

Рисунок 1: Пример сцены с тенью и без тени

Рисунок 2: Пример мягкой тени от светящейся плоскости

Подобное образом, для построения изображения стоит учитывать такую важную составляющую оптических систем, как линза. Изображения предметов на фотографии или у нас перед глазами получаются четкими только в случае, если они находятся в фокальной плоскости, иначе из представление размывает, что также необходимо учитывать.

Не стоит забывать, что объекты в сцене состоят из различных материалов, что вносит на рассмотрение блестящие и прозрачные предметы, с присущими явлениями отражения, преломления и рассеивания.

## 1.2 Формализация объектов синтезируемой сцены

Сцена состоит из следующего набора объектов:

1. Геометрический объект: непосредственное представление объектов реального маира внутри сцены. В число рассматриваемых тел входят параметрические модели, такие как куб, сфера, конус, цилиндр, а также тела производной формы, заданные полигональной моделью. Данная модель предназначена исключительно для хранения геометрического представления и иерархии составного тела.

Для описания каждого объекта используются:

* + Куб: координаты двух диагональных вершин;
  + Сфера: координаты центра и радиус;
  + Конус: координаты центра основания, радиус основания, высота;
  + Цилиндр: координаты центра нижнего основания, радиус, высота;
  + Полигональная модель: набор полигонов с указанием нормали к поверхности, определяющей наружную часть объекта. В качестве основной единицы такой модели был выбран треугольник, так как с его помощью можно представить полигон любой формы.

1. Свойство объекта: отражение качеств, присущих реальному объекту, в частности свечение, прозрачность, фактура;
2. Источник света: геометрическое тело, обладающее свойством свечения. Для источника света присущи такие вырожденные случаи геометрического тела, как точка (точечный источник света, направленный свет);
3. Карта окружения: куб, опоясывающий все объекты сцены и бесконечно удаленный от них. Служит для определения фонового изображения и создания дополнительных отражений.
4. Камера: геометрическое тело, определяющее видимую часть сцены. Положение задается координатой центра, направление взгляда совпадает с осью Z тела.

## 1.3 Анализ алгоритмов удаления невидимых линий и ребер поверхностей

***Алгоритм, использующий Z-буфер***

Смысл алгоритма заключается в использовании двух буферов: буфера кадра и Z-буфера, хранящих атрибуты и информацию о координате Z каждого пикселя соответственно.

Z-буфер инициализируется минимальным значением координаты, а буфер кадра — информацией о пикселе, описывающем фон. Глубина каждого нового пикселя при подсчете сравнивается со значением, которое уже есть в Z-буфере. В случае, если новый пиксель расположен ближе к наблюдателю, то информация о нем заносится в буфер кадра и происходит редактирование Z-буфера.

**Вывод:** данный алгоритм достаточно прост в реализации и, в силу отсутствия сортировок, не требует больших затрат по времени и памяти (для современных компьютеров), однако требуются дополнительные техники для учета отражений, прозрачности и теней.

***Алгоритм обратной трассировки лучей***

Алгоритм берет свое начало из законов оптики: наблюдатель видит объект, только если до него доходят лучи испускаемого света, прошедшие некоторый путь от источника.

Однако, вопреки физической модели, трассировка лучей происходит в обратном направлении — от наблюдателя к источнику. В противном случае приходится рассматривать множество лучей, которые никогда не попадут в наблюдаемую область.

**Вывод:** алгоритм построен на физически верной модели, что дает широкий простор для реализации различных визуальных явлений, но такая гибкость и обобщенность требует большого количества вычислений (возможность усовершенствования засчет параллельных вычислений, так как каждый пиксель в данном методе не зависим от окружающих).

***Алгоритм Робертса***

Данный алгоритм решает задачу удаления невидимых ребер и граней только с выпуклыми телами, работает только в объектном пространстве.

У следующие этапы выполнения:

1. Удаление ребер, экранируемых самим телом;
2. Удаление невидимых ребер, экранируемых другими телами сцены;
3. Удаление невидимых ребер, образованных в результате протыкания.

**Вывод:** алгоритм обладает высокой точностью, в силу того, что все вычисления происходят в объектном пространстве, однако тела должны быть выпуклыми, что не подходит для нашей задачи.

***Алгоритм художника***

Данный алгоритм работает аналогично тому, как художник рисует картину - то есть сначала рисуются дальние объекты, а затем более близкие. Наиболее распространенная реализация алгоритма - сортировка по глубине, которая заключается в том, что произвольное множество граней сортируется по ближнему расстоянию от наблюдателя, а затем отсортированные грани выводятся на экран в порядке от самой дальней до самой ближней.

**Вывод:** данный алгоритм требует меньше памяти в сравнении с остальными, однако так же, как и алгоритм Z-буфера требует дополнительных методов отображения визуальных эффектов.

***Вывод***

Так как цель работы — создание реалистического изображения, то лучшим выбором станет алгоритм трассировки лучей. Данный алгоритм предоставляет удобную базу для реализации оптических эффектов, а затраты по времени не окажут особого значения, так как объекты в сцене — статические, поэтому можно сконцентрироваться на качестве получаемого изображения.

## 1.4 Анализ алгоритмов построения теней

Выбранный в предыдущем пункте алгоритм трассировки лучей, позволяет легко реализовать отрисовку тени.

Корректное построение достигается засчет того, что после нахождения ближайшей точки пересечения очередного луча, из нее посылаются «теневые» лучи к каждому источнику света в сцене. Если теневой луч не достигает источника света (имеется более близкое пересечение с каким-либо объектом сцены), то точка не освещена, иначе — в точке рассчитывается интенсивность, согласно модели освещения.

Помимо этого, также решается проблема построения мягких теней, так как имеется возможность испускания множества теневых лучей к одному объекту, что позволит определить видимую часть и итоговую интенсивность.

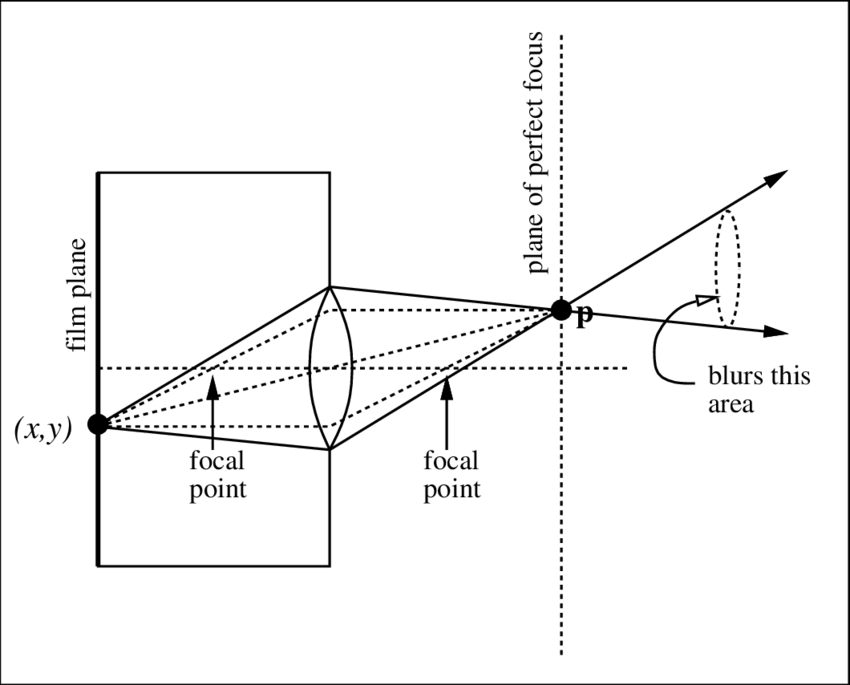
## 1.5 Анализ алгоритмов построения отражений и преломления

Нахождение отражений и преломления также является логическим продолжением алгоритма трассировки лучей. Используя нормаль к поверхности и направление отраженного луча, нетрудно найти падающий луч, для которого аналогично применяется трассировка.

## 1.6 Анализ алгоритмов реализации глубины резкости

Трассировка лучей также позволяет достаточно очевидно реализовать глубину резкости, засчет симуляции множества лучей проходящих через оптическую систему (линзу). К примеру можно реализовать тонкую линзу, пропуская луч из точки картинной плоскости в произвольные точки на поверхности диска, представляющего собой линзу. Если объект находятся в фокусе, то все лучи попадут в одну точку, в результате чего изображение будет четким, иначе — множество различных значений излучения создадут размытое изображение.

## 1.7 Анализ алгоритмов реализации подповерхностного рассеивания

Рисунок 3: Способ реализации глубины резкости

## Вывод

1. 2 Конструкторская часть
2. 3 Технологическая часть

# Заключение

# Список использованной литературы