



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 \_\_\_\_\_ Расколотов Дмитрий Юрьевич \_\_\_\_\_

*фамилия, имя, отчество*

Группа \_\_\_\_\_ ИУ7-54Б \_\_\_\_\_

Тип практики \_\_\_\_\_ стационарная \_\_\_\_\_

Название предприятия \_\_\_\_\_ МГТУ им. Н. Э. Баумана, каф. ИУ7 \_\_\_\_\_

Студент \_\_\_\_\_ Расколотов Д.Ю. \_\_\_\_\_

*подпись, дата*

*фамилия, и.о.*

Руководитель практики \_\_\_\_\_ Куров А.В. \_\_\_\_\_

*подпись, дата*

*фамилия, и.о.*

Оценка \_\_\_\_\_

Москва, 2020 г.

**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)**

---

УТВЕРЖДАЮ

Заведующий кафедрой ИУ7  
(Индекс)

И.В.Рудаков  
(И.О.Фамилия)

« \_\_\_\_ » \_\_\_\_\_ 2020 г.

## **ЗАДАНИЕ**

### **на выполнение производственной практики**

по дисциплине \_\_\_\_\_ Компьютерная графика

Студент группы ИУ7-54Б

\_\_\_\_\_ Расколотов Дмитрий Юрьевич  
(Фамилия, имя, отчество)

Тема производственной практики Трёхмерный планировщик прожекторов в фотостудии

Направленность практики (учебный, исследовательский, практический, производственный)  
\_\_\_\_\_ производственный

Источник тематики (кафедра, предприятие, НИР) \_\_\_\_\_ кафедра

**Техническое задание** Спроектировать программное обеспечение для визуализации и редактирования тени, создаваемой от прожекторного света в фотостудии на определённый предмет. Разработать интерфейс, который позволит выбирать из предложенного набора различные месторасположения источников света (прожекторы) для отображения тени в нужную сторону. Проектируемый программный продукт должен предоставлять возможность размещения источников света, а также возможность просмотра сцены для разных положений наблюдателя.

#### **Оформление производственной практики:**

Расчетно-пояснительная записка на 20-25 листах формата А4.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)

Дата выдачи задания « \_\_\_\_ » \_\_\_\_\_ 2020г.

**Руководитель производственной практики** \_\_\_\_\_ А. В. Куров  
(Подпись, дата) (И.О.Фамилия)

**Студент** \_\_\_\_\_ Д.Ю.Расколотов  
(Подпись, дата) (И.О.Фамилия)

# Содержание

Введение .....	4
1. Аналитический раздел.....	5
1.1 Формализация объектов создаваемой сцены.....	5
1.2 Анализ способов задания трехмерных моделей.....	7
1.3 Анализ способов задания поверхностных моделей.....	8
1.4 Алгоритмы удаления невидимых рёбер и поверхностей.....	12
1.5 Модели освещения.....	13
1.6 Алгоритмы закраски.....	15
1.7 Отображение теней.....	16
2. Конструкторский раздел.....	18
2.1 Алгоритм визуализации сцены.....	18
2.2 Алгоритм, использующий Z-буфер.....	18
2.3 Алгоритм, использующий Z-буфер для построения теней.....	19
2.4 Модель освещения Ламберта.....	19
2.5 Преобразования сцены.....	20
Вывод .....	21
Список использованной литературы.....	22

## Введение

В современном мире компьютерная графика является неотъемлемой частью человеческой жизни. Она используется повсеместно: начиная отображением данных на дисплее до все возможных эффектов в киноиндустрии.

Люди всё чаще стали обращаться к графическому моделированию перед реализацией проектов. Такой подход позволяет как можно чётче обозначить детали проекта, и реже допускать ситуации, в которых становится известно, что исполнитель неправильно воспринял мысль и идею заказчика, когда заказ уже находится на стадии завершения проектирования. Некоторую реалистичность предоставляемого изображения важно учитывать при реализации средств предварительного показа. Так, при отправке на фотостудию, лучше уже заранее представлять, какое оборудование пригодится для выполнения той или иной задачи. При этом не стоит вносить в рассмотрение, например, рассеивание, интерференцию, дифракцию, отражения света или передачу цвета объектов. Связано это с тем, что на поздних этапах согласования проекта более качественное изображение, в котором больше уделено внимания деталям, синтезируют в специализированном программном обеспечении.

Целью практики является анализ и выбор наиболее оптимального из существующих алгоритмов, который позволит визуализировать правильную постановку тени у объекта, используя трёхмерные источники света.

Необходимо предусмотреть в работе алгоритмы, связанные с удалением невидимых граней, изменением положения и интенсивности источника света, визуализацией теней.

# **1. Аналитический раздел**

## **1.1 Формализация объектов создаваемой сцены**

Сцена состоит из следующих объектов:

- Плоскость выставочного стенда – плоскость с заданной сеткой, на которой расставляются модели интерьера. Объекты располагаются только на одной из сторон плоскости. При инициализации сцены левый верхний угол расположен у начала экранных координат, также она параллельна плоскости XY. Размеры границ плоскости задаются количеством квадратов (ячеек) по ширине и длине плоскости, причём размер ячеек – константная величина, определяемая в программном обеспечении; •
- Объекты интерьера – модели, которые занимают ячейки сетки или их часть. Каждая модель представляет собой набор граней, описываемых точками в пространстве, которые соединены рёбрами. Все доступные модели поставляются вместе с программным обеспечением, внесение новых моделей в базу не предусмотрено. Имеется возможность изменить положение модели на плоскости: переместить, повернуть. Также можно выбрать длину, ширину и высоту модели, при этом единица измерения – занимаемые ячейки (или их частей) плоскости выставочного стенда; •
- Источники света – материальные точки, испускающие параллельные лучи света. В зависимости от расположения источника и вектора направления распространения лучей света, определяется тень от объектов, расположенных на сцене. Положение источника задаётся трёхмерными координатами, а направление распространения света – вектором.

## **1.2 Анализ способов задания трехмерных моделей**

Модель является отображением формы и размеров объекта. Модель может быть задана одним из нескольких способов:

1. Каркасная (проволочная) модель. В данной модели задаётся информация о вершинах и рёбрах объекта. Это одна из простейших форм

задания модели, но она имеет один существенный недостаток: модель не всегда однозначно передаёт представление о форме объекта.

2. Поверхностные модели. Этот тип модели часто используется в компьютерной графике. Поверхность может описываться аналитически, либо задаваться другим способом (например, отдельными участками поверхности, задаваемыми в качестве участков поверхности того или иного вида). При этом вложенные криволинейные поверхности можно представлять в упрощённом виде, выполняя, например, полигональную аппроксимацию: такая поверхность будет задаваться в виде поверхности многогранника.

Данная форма также имеет свой недостаток: отсутствует информация о том, с какой стороны поверхности находится материал.

3. Твёрдотельные (объёмные) модели. Отличие данной формы задания модели от поверхностной формы состоит в том, что в объёмных моделях к информации о поверхностях добавляется информация о том, где расположен материал. Это можно сделать путём указания направления внутренней нормали.

Таким образом, можно сделать вывод, что для решения задачи не подойдёт каркасная форма, так как такое представление будет приводить к неправильному восприятию заказчиком форм моделей, а также не подойдут и объёмные модели, так как на этапе проектирования обстановки исполнителю совершенно не важно, из какого материала будет выполнен тот или иной объект сцены. Методом исключения приходим к выбору поверхностной формы модели.

### **1.3 Анализ способов задания поверхностных моделей**

Поверхностные модели задаются:

- Аналитическим способом. Этот способ задания модели характеризуется описанием модели объекта, которое доступно в неявной форме, то есть для получения визуальных характеристик необходимо дополнительно вычислять некоторую функцию, которая зависит от параметра;

- Полигональной сеткой. Данный способ характеризуется совокупностью вершин, рёбер и граней, определяющих форму объекта в трёхмерном пространстве.

Также существует множество различных способов хранения информации о сетке:

- ❖ Список граней. Объект – это множество граней и множество вершин. В каждую грань входят как минимум 3 вершины;
- ❖ «Крылатое» представление. Каждая точка ребра указывает на две вершины, две грани и четыре ребра, которые её касаются;
- ❖ Полурёберные сетки. То же «крылатое» представление, но информация обхода хранится для половины грани;
- ❖ Таблица углов. Таблица, хранящая вершины. Обход заданной таблицы неявно задаёт полигоны. Такое представление более компактно и более производительное для нахождения полигонов, но, в связи с тем, что вершины присутствуют в описании нескольких углов, операции по их изменению медленны.
- ❖ Вершинное представление. Хранятся лишь вершины, которые указывают на другие вершины. Простота представления даёт возможность проводить над сеткой множество операций.

При реализации программного продукта наиболее удобным представлением является модель, заданная полигональной сеткой – это поможет избежать проблем при описании сложных моделей. Способ хранения полигональной сетки

при этом – список граней, так как он предоставляет явное описание граней, что поможет при реализации алгоритма удаления невидимых рёбер и поверхностей. Также этот способ позволит эффективно преобразовывать модели, так как структура будет включать в себя список вершин. При этом недостаточная эффективность преобразований геометрии объектов, сопутствующая такой форме представления, на программный продукт не повлияет, так как задание геометрии тел предусмотрены только на этапе выбора специфичных моделей, для которых можно определить длину, ширину или высоту. То есть начальная геометрия тел будет задаваться определённым алгоритмом, что разрешит описанную проблему.

#### **1.4 Алгоритмы удаления невидимых линий и поверхностей**

Решать поставленную задачу удаления можно как в объектном пространстве (в мировой системе координат), так и в пространстве изображения (в экранных координатах).

Обозначим свойства, которыми должен обладать алгоритм, для оптимальной работы реализуемого программного обеспечения:

- Алгоритм должен быть достаточно быстрым при работе с множеством объектов сцены, чтобы пользователь не ожидал долгой загрузки изображения;
- Алгоритм может работать в любом пространстве (скорость важнее точности) Рассмотрим алгоритмы для решения поставленной задачи.

Рассмотрим алгоритмы для решения поставленной задачи.

#### **Алгоритм Робертса**

Алгоритм Робертса Данный алгоритм работает в объектном пространстве, решая задачу только с выпуклыми телами.

Алгоритм выполняется в 3 этапа:



1. Этап подготовки исходных данных. На данном этапе должна быть задана информация о телах. Для каждого тела сцены должна быть сформирована матрица тела  $V$ . Размерность матрицы -  $4 * n$ , где  $n$  – количество граней тела.

Каждый столбец матрицы представляет собой четыре коэффициента уравнения плоскости  $ax + by + cz + d = 0$ , проходящей через очередную грань.

Таким образом, матрица тела будет представлена в следующем виде:

$$V = ( a_1 \ b_1 \ c_1 \ d_1 \ a_2 \ b_2 \ c_2 \ d_2 \ \dots \ a_n \ b_n \ c_n \ d_n )$$

Матрица тела должна быть сформирована корректно, то есть любая точка, расположенная внутри тела, должна располагаться по положительную сторону от каждой грани тела. В случае, если для очередной грани условие не выполняется, соответствующий столбец матрицы надо умножить на  $-1$ . Для проведения проверки следует взять точку, расположенную внутри тела. Координаты такой точки можно получить путём усреднения координат всех вершин тела.

2. Этап удаления рёбер, экранируемых самим телом.

На данном этапе рассматривается вектор взгляда  $E = \{0 \ 0 \ -1 \ 0\}$ .

Для определения невидимых граней достаточно умножить вектор  $E$  на матрицу тела  $V$ . Отрицательные компоненты полученного вектора будут соответствовать невидимым граням.

3. Этап удаления невидимых рёбер, экранируемых другими телами сцены.

На данном этапе для определения невидимых точек ребра требуется построить луч, соединяющий точку наблюдения с точкой на ребре. Точка будет невидимой, если луч на своём пути встречает в качестве преграды рассматриваемое тело. Если тело является преградой, то луч должен пройти

через тело. Если луч проходит через тело, то он находится по положительную сторону от каждой грани тела.

Для того, чтобы определить, подходит ли данный алгоритм для решения поставленной задачи, рассмотрим его преимущества и недостатки.

Преимущество:

- Алгоритм работает в объектном пространстве, точность вычислений высокая.

Недостатки:

- Теоретический рост сложности алгоритма – квадрат числа объектов. Для решения данной проблемы достаточно воспользоваться модифицированными реализациями, например, с использованием габаритных тестов или сортировки по оси z;
- Все тела сцены должны быть выпуклыми. Данная проблема также приводит к усложнению алгоритма, так как потребуются прибегнуть к проверке объектов на выпуклость и их разбиению на выпуклые многоугольники.

Вывод: Алгоритм Робертса не подходит для решения поставленной задачи по следующим причинам:

- От программного обеспечения не требуется той точности визуализации объектов, которую предоставляет алгоритм;
- На сцене может находиться множество объектов, что замедлит скорость работы алгоритма. Это не удовлетворяет предъявленным требованиям к скорости выполнения алгоритма. Особенные проблемы также могут наблюдаться при наличии множества невыпуклых тел на сцене;
- Реализация модификаций, позволяющих приблизить рост сложности алгоритма к линейной, очень трудозатратна.

## Алгоритм, использующий z-буфер

Данный алгоритм работает в пространстве изображения. Используется два буфера: буфер кадра, в котором хранятся атрибуты каждого пикселя в пространстве изображения, и z-буфер, куда помещается информация о координате  $z$  для каждого пикселя.

Первоначально в z-буфере находятся минимально возможные значения  $z$ , а в буфере кадра располагаются пиксели, описывающие фон. Каждый многоугольник преобразуется в растровую форму и записывается в буфер кадра.

В процессе подсчета глубины нового пикселя, он сравнивается с тем значением, которое уже лежит в z-буфере. Если новый пиксель расположен ближе к наблюдателю, чем предыдущий, то он заносится в буфер кадра и происходит корректировка z-буфера.

Для решения задачи вычисления глубины  $z$  каждый многоугольник описывается уравнением  $ax + by + cz + d = 0$ . При  $c = 0$  многоугольник для наблюдателя вырождается в линию.

Для некоторой сканирующей строки  $y = \text{const}$ , поэтому имеется возможность рекуррентно высчитывать  $z'$  для каждого  $x' = x + dx$ :

$$z' - z = - \underbrace{ax' + d}_c + \underbrace{ax + d}_c = a \underbrace{(x - x')}_c$$

$$\text{Получим: } z' = z - a \underbrace{dx}_c, \text{ так как } x - x' = dx = 1$$

При этом стоит отметить, что для невыпуклых многогранников предварительно потребуется удалить нелицевые грани.

Рассмотрим преимущества и недостатки описанного алгоритма.

Преимущества:

- Простота реализации;
- Оценка вычислительной трудоемкости алгоритма линейна;
- Экономия вычислительного времени, так как элементы сцены не сортируются.

Недостатки:

- Большой объём требуемой памяти.
- Реализация эффектов прозрачности сложна.

Вывод:

Алгоритм отвечает главному требованию - скорости работы с множеством объектов. Также простота алгоритма позволит достаточно быстро реализовать данный алгоритм и, что важнее, в полной мере отладить его.

При этом следует отметить, что само изображение будет относительно малых размеров, что приведёт к некритично большим затратам памяти для выполнения данного алгоритма.

### **Алгоритм обратной трассировки лучей**

Наблюдатель видит объект посредством испускаемого источником света, который падает на этот объект и согласно законам оптики некоторым путём доходит до глаза наблюдателя. Отслеживать пути лучей от источника к наблюдателю неэффективно с точки зрения вычислений, поэтому наилучшим способом будет отслеживание путей в обратном направлении, то есть от наблюдателя к объекту.

Предполагается, что сцена уже преобразована в пространство изображения, а точка, в которой находится наблюдатель, находится в бесконечности на положительной полуоси  $z$ , и поэтому световые лучи параллельны этой же оси. При этом каждый луч проходит через центр пикселя растра до сцены. Траектория каждого луча отслеживается для определения факта пересечения определённых объектов сцены с этими лучами. При этом необходимо проверить пересечение каждого объекта сцены с каждым лучом, а пересечение с  $z_{min}$  представляет видимую поверхность для данного пикселя.

Если же точка наблюдателя находится не в бесконечности, то есть в рассмотрении фигурирует перспективная проекция, то предполагается, что сам наблюдатель по-прежнему находится на положительной полуоси  $z$ , а сам растр при этом перпендикуляром оси  $z$ . Задача будет состоять в том, чтобы построить одноточечную центральную проекцию на картинную плоскость.

Определения пересечений происходит с помощью погружения объектов в некоторую выпуклую оболочку – например, сферическую. Поиск пересечения с такой оболочкой происходит проще: достаточно проверить превосходит ли радиус сферы-оболочки расстояние от центра этой сферы до луча.

Предположим, что некоторая прямая проходит через две точки

$P_1(x_1, y_1, z_1)$  и  $P_2(x_2, y_2, z_2)$ :

$$P(t) = P_1 + (P_2 - P_1)t$$

Компоненты при этом:

$$x = x_1 + (x_2 - x_1)t = x_1 + at$$

$$y = y_1 + bt$$

$$z = z_1 + ct$$

Таким образом минимальное расстояние от этой прямой до некоторой точки  $P_0(x_0, y_0, z_0)$ :

$$d^2 = (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2$$

Где параметр  $t$ , который определяет ближайшую точку  $P(t)$ :

$$t = -\frac{a(x_1 - x_0) + b(y_1 - y_0) + c(z_1 - z_0)}{a^2 + b^2 + c^2}$$

Если  $d^2 > R^2$ , где  $R$  – радиус сферической оболочки, то луч не может пересечься с объектом.

Рассмотрим преимущества и недостатки описанного алгоритма.

Преимущества:

- Высокая реалистичность синтезируемого изображения;
- Работа с поверхностями в математической форме;
- Вычислительная сложность слабо зависит от сложности сцены.

Недостаток:

- Производительность.

Вывод:

Алгоритм не отвечает главному требованию – скорости работы. Также от реализуемого продукта не требуется высокой реалистичности синтезируемого изображения и возможности работы с поверхностями, заданными в

математической форме. Указанные факты говорят о том, что обратная трассировка лучей не подходит для решения поставленной задачи.

Таким образом, в качестве алгоритма удаления невидимых рёбер и поверхностей был выбран алгоритм с использованием z-буфера.

## **1.5 Модели освещения**

Модели освещения разделяются на глобальные и локальные. Самая простая из них - локальная модель освещенности. Эта модель не рассматривает процессы светового взаимодействия объектов сцены между собой, а только расчет освещенности самих объектов. В рамках локальной модели освещения рассматривается свет только от явных точечных источников света, а взаимодействие ограничивается только однократным отражением света от непрозрачной поверхности. Глобальная модель освещенности, рассматривает трехмерную сцену как единую систему и пытается описывать освещение с учетом взаимного влияния объектов. В рамках этой модели рассматриваются такие вопросы, как многократное отражение и преломление света (трассировка лучей), рассеянное освещение и другие.

### **Модель освещения Ламберта**

Данная модель позволяет реализовать диффузное (имитируется воздействие на объект направленного источника света) освещение объектов. Свет, падая на поверхность, рассеивается равномерно во все стороны. При расчете такого освещения учитывается только ориентация поверхности и направление на источник света. Интенсивность отраженного света не зависит от расположения наблюдателя. Наблюдаемый цвет матовой поверхности определяется комбинацией собственного цвета поверхности и цвета излучения источника света. Интенсивность отражения пропорциональна косинусу угла между внешней нормалью к поверхности и направлением к источнику света [5].

### **Модель освещения Фонга**

Данная модель позволяет реализовать комбинацию диффузной (имитируется воздействие на объект направленного источника света) и зеркальной (имитирует блик света, который появляется на блестящих объектах) составляющих. Падающий и отраженный лучи лежат в одной плоскости с нормалью к отражающей поверхности в точке падения, и эта нормаль делит угол между лучами на две части. Отраженная составляющая освещенности в точке зависит от того, насколько близки направления вектора, направленного на наблюдателя, и отраженного луча.

## **Вывод**

Объекты на карте не имеют зеркальных элементов и являются матовыми, поэтому выбор пал на модель освещения Ламберта, локальную модель освещения.

## **1.6 Алгоритмы закраски**

### **Простая закраска**

Вся грань закрашивается одним уровнем интенсивности, который высчитывается по закону Ламберта. Этот метод крайне прост в реализации и совершенно не требователен к ресурсам. Однако плохо подходит для тел вращения, плохо учитывает отраженный свет.

### **Закраска Гуро**

Этот метод предназначен для создания иллюзии гладкой криволинейной поверхности, описанной в виде многогранников или полигональной сетки с плоскими гранями. Если каждая плоская грань имеет один постоянный цвет, определенный с учетом отражения, то различные цвета соседних граней очень заметны, и поверхность выглядит именно как многогранник. Казалось бы, этот дефект можно замаскировать с помощью увеличения количества граней при аппроксимации поверхности, Но зрение человека имеет способность подчеркивать перепады яркости на границах смежных граней – такой эффект называется эффектом полос Маха. Поэтому для создания иллюзии гладкости нужно намного увеличить количество граней, что приводит к существенному



замедлению визуализации. Метод Гуро основывается на идее закрашивания плоской грани не одним цветом, а плавно изменяющимися оттенками, вычисляемыми путем билинейной интерполяции цветов примыкающих граней. Определение интерполированных значений интенсивности отраженного света в каждой точке грани удобно выполнять во время цикла заполнения полигона [1].

### **Закраска Фонга**

Этот метод закрашки аналогичен методу Гуро, но при использовании метода Фонга для определения цвета в каждой точке интерполируются не интенсивности отраженного света, а векторы нормалей. При использовании данного алгоритма изображение получается более качественным, повышается правдоподобность изображения зеркальных бликов. Но данный алгоритм требует гораздо больше вычислительных ресурсов.

### **Вывод**

Объекты сцены являются матовыми, на сцене присутствуют лишь грани объектов и местности, если ребра объектов будут сглажены, то объект будет хуже восприниматься, поэтому простая закрашка будет наиболее оптимальная. Данный алгоритм позволит сохранить большое количество вычислительных ресурсов.

### **1.7 Отображение теней**

Тень состоит из двух частей: полутени и полной тени. Полная тень – это центральная, темная, резко очерченная часть, а полутень – это окружающая ее более светлая часть. В целях оптимизации вычислительных затрат рассматривается создание только полных теней [1]. Для построения теней требуется дважды удалить невидимые поверхности: для положения источника освещения и для положения точки наблюдения. Так как для удаления невидимых поверхностей применяется метод, использующий z-буфер, то нужно применить его дважды. В результате построения теневого z-буфера для источника света сохраняются значения глубин для этого вида. Затем сцена строится из точки

наблюдения, при этом для определения затенённости глубина каждого пиксела сравнивается с глубиной соответствующего пиксела в теневом буфере.

## **2. Конструкторский раздел**

### **2.1 Общий алгоритм решения поставленной задачи**

1. Задать размеры области размещения объектов
2. Разместить объекты сцены и источники света
3. С помощью модифицированного алгоритма, использующего z-буфер, определить падающие от объектов сцены тени и визуализировать обстановку, основываясь на текущем положении наблюдателя.

### **2.2 Алгоритм, использующий Z-буфер**

- 1) заполнить буфер кадра фоновым значением интенсивности или цвета;
- 2) заполнить Z-буфер минимальными значениями  $z$ ;
- 3) для каждого пиксела в многоугольнике вычислить его глубину  $z(x,y)$ ;
- 4) сравнить глубину  $z(x,y)$  со значением  $Z$  буфер в той же точке, хранящимся в  $Z$  буфере в этой же позиции.
- 5) если  $Z(x,y) > Z \text{ буфер } (x,y)$ , то записать атрибут этого многоугольника в буфер кадра и заменить  $Z \text{ буфер } (x,y)$  на  $z(x,y)$ .

### **2.3 Алгоритм, использующий Z-буфер для построения теней**

При построении теней с использованием алгоритма z-буфера выполняется два прохода: один - относительно источника света, другой – относительно наблюдателя. Для этого выделяется отдельный теневой z-буфер. Первый проход необходим для того, чтобы определить, какие точки видны со стороны источника света. При втором проходе сцена визуализируется из положения наблюдателя с учетом того, что точки, которые оказались невидимыми со стороны источника света, находятся в тени. Таким образом, алгоритм работает в два этапа. При первом проходе сцена рассчитывается при совмещении точки наблюдения с положением

источника света. Значения глубины пикселей для данного вида заносятся в теневой  $z$  - буфер, а значения интенсивности не учитываются. При втором проходе сцена строится из точки, в которой на самом деле находится наблюдатель. При сканировании каждой поверхности значение ее глубины каждого ее пиксела сравнивается со значением глубины в  $z$ -буфере. Если поверхность видима, то необходимо проверить, видима ли данная точка со стороны источника света. Для этого координаты точки  $x, y, z$  из вида наблюдателя линейно преобразуются в координаты  $x', y', z'$  на виде из источника света. Проверка на видимость осуществляется сравнением значения, которое хранится в теневом  $z$ -буфере для координат  $x', y'$ , и значения  $z'$ . Если точка невидима для источника света (значение в "теневом"  $z$ -буфере больше значения  $z'$ ), значит она находится в тени и ее свечение вычисляется с учетом затенения. Если же точка видима из положения источника света, то она изображается без изменений.

- 1) Заполнить теневой  $Z$ -буфер минимальными значениями глубины;
- 2) Выполнить алгоритм  $Z$ -буфера для источника освещения;
- 3) Выполнить алгоритм  $Z$ -буфера для точки зрения наблюдателя;
  - 3.1) Координаты точки  $x, y, z$  линейно преобразовать в координаты  $x', y', z'$  для вида из источника
  - 3.2) Сравнить глубину  $Z$  теневого буфера ( $x', y'$ ) со значением  $z'$  в той же точке ( $x', y'$ ). Если  $Z$  теневого буфера больше, пиксел отображается с учетом затемнения, иначе — без.

## 2.4 Модель освещения Ламберта

$$I = I_0 \cdot \cos(\alpha), \text{ где}$$

$I$  – результирующая интенсивность освещения;

$I_0$  – интенсивность источника освещения;

$\alpha$  – угол между нормалью к поверхности и вектором направления светового луча.

## 2.5 Преобразования сцены

Преобразование сцены осуществляется через умножение четырёхмерных вектор-столбцов  $(x,y,z,w)^T$  на соответствующие матрицы преобразований.

Матрицы поворота на угол  $A$  относительно осей

$$\text{OX: } \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos A & \sin A & 0 \\ 0 & -\sin A & \cos A & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}; \text{ OY: } \begin{pmatrix} \cos A & 0 & \sin A & 0 \\ 0 & 1 & 0 & 0 \\ -\sin A & 0 & \cos A & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}; \text{ OZ: } \begin{pmatrix} \cos A & \sin A & 0 & 0 \\ -\sin A & \cos A & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Матрица переноса:

$$\begin{pmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Матрица

масштабирования:

$$\begin{pmatrix} kx & 0 & 0 \\ 0 & ky & 0 \\ 0 & 0 & kz \\ 0 & 0 & 0 \end{pmatrix}$$

## **Вывод**

В ходе работы над поставленной задачей были проанализированы достоинства и недостатки существующих алгоритмов касающихся построения трехмерных изображений. Рассмотрены алгоритмы, связанные с удалением невидимых граней, изменением положения и интенсивности источника света, визуализацией теней.

Для удаления невидимых линий и поверхностей был выбран алгоритм, использующий Z-буфер, так как алгоритмы, позволяющие достичь более высокой реалистичности изображения, требуют больших вычислительных ресурсов, что негативно сказывается на скорости работы программы.

Объекты в студии не имеют зеркальных элементов и являются матовыми, поэтому была выбрана модель освещения Ламберта.

Простая закраска объектов будет наиболее оптимальная, потому что на сцене не будет тел вращения. Данный алгоритм позволит сохранить большое количество вычислительных ресурсов.

## **Список использованной литературы**

1. Роджерс Д. Математические основы машинной графики. / Д. Роджерс – М.: Мир, 1989. – 512 с.
2. Порев В.Н. Компьютерная графика. / В.Н. Порев – СПб.: БВХ-Петербург, 2002. – 432 с.
3. Польский С.В. Компьютерная графика: учебн.-методич. Пособие. / С.В. Польский – М.: ГОУ ВПО МГУЛ, 2008. – 38 с.
4. Компьютерная графика. Модели затенения [Электронный ресурс]. – Режим доступа: [https://compgraphics.info/3D/lighting/shading\\_model.php](https://compgraphics.info/3D/lighting/shading_model.php). – Дата доступа: 30.06.2020.
5. Основы освещения [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/333932/>. – Дата доступа: 01.07.2020.