



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»

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К КУРСОВОЙ РАБОТЕ
НА ТЕМУ:**

**«Визуализатор распространения звуковых волн в
замкнутом пространстве»**

Студент _____ Постнов С. А.
подпись, дата *фамилия, и.о.*

Руководитель курсовой работы _____ Куров А. В.
подпись, дата *фамилия, и.о.*

Оценка _____

Оглавление

Введение	4
1. Аналитическая часть.....	5
1.1 Описание требований к алгоритмам	5
1.2 Описание и формализация объектов сцены.....	5
1.3 Методы описания моделей на сцене	6
1.4 Методы задания поверхностных моделей на сцене	7
1.5 Анализ алгоритмов удаления невидимых линий и поверхностей	7
1.6 Выбранный алгоритм удаления невидимых линий и поверхностей	12
1.7 Анализ алгоритмов закрашки	12
1.8 Выбранный алгоритм закрашки.....	14
2. Конструкторская часть.....	15
2.1 Требования к программному обеспечению	15
2.2 Общий алгоритм построения сцены.....	15
2.3 Алгоритм изображения звуковой волны в текущий момент времени	16
2.4 Алгоритм простой закрашки изображения	18
2.5 Алгоритм, использующий Z-буфер.....	18
2.6 Используемые типы и структуры данных.....	21
2.7 Вывод	21
3. Технологическая часть.....	22
3.1 Выбор средств для разработки	22
Заключение.....	24
Список использованных источников	25

Введение

Компьютерная графика играет важную роль в современной технологической среде, находя применение в различных сферах, от разработки видеоигр до создания визуальных эффектов в фильмах и инженерных моделирований. Однако, помимо визуальных аспектов, компьютерная графика имеет потенциал для решения более глубоких и многогранных задач.

Целью данной практической работы является исследование и анализ существующих алгоритмов визуализации процессов распространения звуковых волн в замкнутых пространствах. Звук, как важный аспект нашей жизни, имеет широкое применение, и его понимание и контроль в различных средах и помещениях являются ключевыми задачами в инженерии и акустике.

Для достижения поставленной цели необходимо решить некоторые подзадачи:

1. описать и формализовать объекты сцены
2. определиться с методами построения реалистичного изображения
3. определиться с языком программирования и инструментами для работы
4. реализовать выбранные алгоритмы и структуры данных

1. Аналитическая часть

1.1 Описание требований к алгоритмам

- 1) Реализованные алгоритмы должны быть быстродействующими и эффективными
- 2) Реализованные алгоритмы должны предоставлять полноценное представление о сцене и обеспечивать корректную визуализацию

1.2 Описание и формализация объектов сцены

- 1) **Моделируемое замкнутое пространство** – помещение прямоугольной формы, окруженное четырьмя виртуальными стенами и имеющее константный размер
- 2) **Объекты сцены** – элементы трехмерной виртуальной среды, которые представляют собой геометрические объекты
- 3) **Точечный источник звуковых волн** – материальная точка в пространстве, характеризующаяся следующими свойствами:
 - a. *Излучение*. Источник звука излучает звуковые волны во все стороны равномерно.
 - b. *Однородность*. Источник не имеет предпочтительного направления для распространения звука.
 - c. *Сферическая волна*. Звук распространяется от источника радиально, образуя сферическую волну, в которой интенсивность звука уменьшается с увеличением расстояния от источника.
 - d. *Интенсивность*. Интенсивность звука, испускаемого точечным источником, остается постоянной на всех расстояниях от источника.
- 4) **Камера** – материальная точка в пространстве, которая определяет точку обзора и параметры визуализации для создания двухмерного изображения трехмерной сцены. Камера задает, каким образом трехмерные объекты и сцена будут проецироваться на двумерный экран или изображение.

1.3 Методы описания моделей на сцене

1) *Каркасный* метод моделирования заключается в составлении модели из точек и линий, что делает ее самой простой и примитивной. Однако такое отображение обладает некоторыми серьезными недостатками:

- a. Сложно отличить видимые ребра от невидимых
- b. Отображение мнимых ребер (цилиндр)
- c. Отсутствие средств выполнения тоновых изображений

Несмотря на эти недостатки, каркасная модель является экономичной в плане памяти и подходит для решения простых задач.

2) *Поверхностный* метод моделирования описывает форму объекта с помощью поверхностей, а не традиционного каркаса из вершин, ребер и граней. Этот метод позволяет создавать более гладкие и реалистичные модели, так как он лучше передает свойства материалов и позволяет работать с объектами произвольной формы. Кроме того, поверхностное моделирование позволяет создавать объекты с меньшими вычислительными затратами, что может быть полезно при работе с большими моделями или при рендеринге в реальном времени.

3) *Твердотельный* метод моделирования представляет объекты в виде набора геометрических примитивов, таких как точки, линии и поверхности. Эти примитивы затем объединяются в более сложные объекты, что позволяет создавать модели с высокой точностью и детализацией.

При поверхностном моделировании форма описывается с помощью поверхностей, что позволяет создавать более реалистичные и гладкие модели. В отличие от каркасного метода, данный способ лучше передает свойства материалов, а по сравнению с твердотельными объектами требует меньше вычислительных ресурсов и дополнительных параметров (например, материал). Таким образом, для дальнейшей работы будут использоваться *поверхностные модели*.

1.4 Методы задания поверхностных моделей на сцене

1) *Параметрический* способ задания поверхностных моделей на сцене предполагает использование математических уравнений для описания поверхностей объектов. Примерами таких уравнений могут быть:

- a. NURBS-поверхности (Non-Uniform Rational B-Splines) - позволяют создавать гладкие поверхности с высоким уровнем контроля над формой.
- b. Т-сплайны (T-splines) - представляют собой адаптацию NURBS для работы с трехмерными объектами.
- c. Бикубические рациональные В-сплайновые поверхности (Rational Bicubic Splines).
- d. В-сплайновые кривые и поверхности (Bézier curves and surfaces) - используются для создания гладких кривых и поверхностей.

2) *Полигональная сетка* является еще одним способом задания поверхностных моделей. Это метод, при котором поверхность объекта разбивается на множество вершин, ребер и граней.

Исходя из потребности в простоте и скорости, был выбран метод задания модели при помощи полигональной сетки с использованием хранения в виде списка граней, что также позволит наиболее удобным образом представлять и описывать объекты сцены.

1.5 Анализ алгоритмов удаления невидимых линий и поверхностей

Для корректного и реалистичного отображения объектов сцены необходимо выбрать быстродействующий алгоритм. Рассмотрим два подхода для решения поставленной задачи:

1. *Алгоритмы, которые работают в пространстве изображения.*
Принцип действия данного алгоритма основан на удалении невидимых линий и поверхностей непосредственно после проецирования объектов

на плоскость экрана. Примерами являются: алгоритм Z-буфера, алгоритм Варнака.

2. *Алгоритмы, которые работают в объектном пространстве.* Принцип действия данного алгоритма заключается в определении того, какие части трехмерных объектов видны с определенной точки наблюдения, а какие скрыты за другими объектами. Они позволяют оптимизировать рендеринг трехмерных сцен, исключая изображение невидимых элементов, что повышает производительность и создает правдоподобное визуальное представление сцены для наблюдателя. Примерами являются: алгоритм Робертса, алгоритм обратной трассировки лучей.

Алгоритм, использующий Z-буфер

Алгоритм, использующий Z-буфер (или буфер глубины), основан на идее определения видимости пикселей в трехмерной сцене относительно точки наблюдения (камеры) и удаления невидимых поверхностей. Он базируется на следующих основных принципах:

- 1) *Буфер глубины (Z-буфер):* для каждого пикселя на экране создается буфер глубины, где хранится значение глубины (Z-координата) этого пикселя относительно камеры. Исходно значения Z-буфера устанавливаются на максимальные значения (чаще всего, бесконечность).
- 2) *Процесс рендеринга:* каждый объект в трехмерной сцене визуализируется на экране. В ходе этого процесса:
 - а. Происходит преобразование объекта, чтобы учесть его положение, ориентацию и масштаб относительно камеры.
 - б. Объект проецируется на двумерную плоскость экрана с учетом перспективы.

с. Для каждого пикселя, который принадлежит объекту, проверяется его текущее значение глубины (Z-координата) относительно значения в Z-буфере. Если текущий пиксель находится ближе к камере, чем пиксель, уже находящийся в Z-буфере, то текущий пиксель заменяет старый, и его информация о цвете и текстуре сохраняется, а значение глубины в Z-буфере обновляется.

3) *Завершение рендеринга*: По завершении рендеринга на экране отображаются только те пиксели, которые представляют собой ближайшие видимые объекты в каждой точке экрана. Это гарантирует, что на экране будут видны только те части объектов, которые находятся ближе к наблюдателю, а скрытые части будут автоматически удалены.

В итоге алгоритм с Z-буфером обеспечивает правильное отображение глубины и обнаружение невидимых поверхностей в трехмерной сцене. Использование данного алгоритма позволяет создавать сложные трехмерные сцены с реалистичным отображением глубины и обеспечивает автоматическое удаление скрытых поверхностей. Этот метод широко применяется в компьютерной графике для рендеринга трехмерных сцен в видеоиграх, анимации, визуализации данных и других приложениях, так как является довольно быстродействующим и простым в реализации.

Алгоритм Варнака

Алгоритм Варнака основан на физических законах оптики и является одним из методов рендеринга трехмерных сцен в компьютерной графике. Он основан на принципе трассировки лучей света от камеры через каждый пиксель экрана в сцене. По мере перемещения лучей, они взаимодействуют с объектами в сцене, что позволяет определить цвет и освещение каждого пикселя на экране. Алгоритм использует физические модели, такие как отражение, преломление и тени. Он основан на следующих принципах:

- 1) Для каждого пикселя на экране определяется луч, который исходит из точки наблюдения (камеры) и проходит через этот пиксель.
- 2) Луч отслеживается в сцену, и для каждой встреченной поверхности определяется, как свет взаимодействует с материалом этой поверхности, включая отражение, преломление и тени.
- 3) Рекурсивно проводится трассировка дополнительных лучей, например, отраженных и преломленных лучей, для учета бликов, отражений и преломлений света.
- 4) По завершении трассировки лучей для каждого пикселя собирается информация о цвете и освещении, что позволяет создать финальное изображение.

Алгоритм Варнака широко используется в компьютерной графике и визуализации для создания фотореалистичных изображений и эффектов. Его применение находит в киноиндустрии, анимации, визуализации данных, играх и других областях, где требуется высококачественная графика и реалистичное моделирование освещения [2].

Алгоритм Робертса

Алгоритм Робертса предназначен для сцен, содержащих выпуклые тела с поверхностями в виде плоских граней. Метод хорошо описан математически и позволяет решать несколько задач, связанных с визуализацией:

- 1) Удаление невидимых линий
- 2) Определение задних граней
- 3) Определение интенсивности закраски
- 4) Определение сопряжения тел
- 5) Экранирование одних тел другими.

Данный алгоритм является переборным, и требует, чтобы грани были выпуклыми многоугольниками. Он работает с ребрами, т.е. больше подходит для

каркасных изображений. Ребра, смыкающие две невидимые грани, не рассматриваются, т.к. они заведомо невидимы. Каждое оставшееся ребро объекта анализируется на взаимоположение со всеми гранями [1, 2].

Алгоритм обратной трассировки лучей

Трассировка лучей – метод машинной графики, позволяющий создавать фотореалистические изображения любых трехмерных сцен. Трассировка лучей моделирует прохождение лучей света через изображаемую сцену. Фотореализм достигается путем математического моделирования оптических свойств света и его взаимодействия с объектами. Суть алгоритма можно описать следующим образом:

- 1) Луч испускается из начальной точки камеры через каждый пиксель изображения, и осуществляется поиск пересечения луча с объектом
- 2) Если луч попадает в объект, то из точки пересечения луча с объектом испускается новый луч в сторону источника света
- 3) В зависимости от того, сможет ли луч добраться до источника света, выбирается та или иная интенсивность цвета пикселя

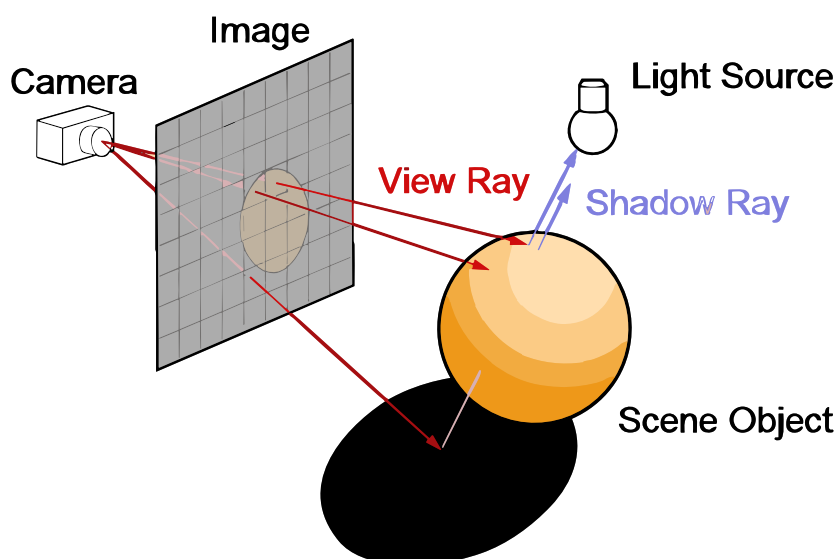


Рисунок 1.1 – Метод обратной трассировки лучей

С учетом материалов объектов, которые при отражении света могут последний рассеивать, приходится обрабатывать не один, а множество лучей. Отображаемая сцена также может быть сложной и состоять из миллионов треугольников. В такой ситуации простейший перебор треугольников в поисках пересечения с каждым лучом будет занимать огромное количество времени [3].

1.6 Выбранный алгоритм удаления невидимых линий и поверхностей

Исходя из потребностей в быстродействии и простоте алгоритма, в качестве основного способа для дальнейшего отсекаемого был выбран алгоритм, использующий Z-буфер.

1.7 Анализ алгоритмов закрашки

Для решения задачи изображения более реалистичных и удобных для восприятия объектов, необходимо выбрать алгоритм закрашки. Рассмотрим некоторые из них:

- 1) Простая закрашка
- 2) Закрашка по Гуро
- 3) Закрашка по Фонгу

Простая закрашка

Идея данного алгоритма заключается в том, чтобы закрасить всю грань одним цветом (монотонно). Для этого используется интенсивность, вычисляемая по закону Ламберта.

Данный метод является довольно простым в реализации, а также достаточно быстродействующим. Однако, по этим причинам, он является самым нереалистичным.

Закраска по Гуро

Метод Гуро основывается на идее закрашивания каждой плоской грани не одним цветом, а плавно изменяющимися оттенками, вычисляемыми путем билинейной интерполяции цветов примыкающих граней.

Данный способ используется для создания более реалистичных изображений и плавного перехода между вершинами грани. Существенными недостатками метода Гуро является невозможность получения качественных бликов на блестящих поверхностях и более высокая ресурсо-затратность, в сравнении с простой закраской [6].

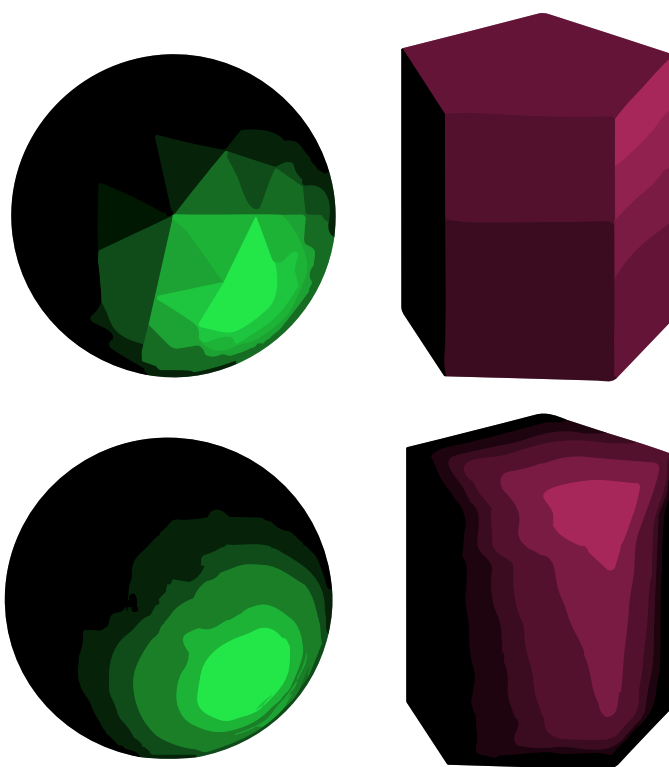


Рисунок 1.2 – Однотонная закразка (вверху) и закразка методом Гуро (внизу)

Закраска по Фонгу

В закраске Фонга сохраняется сглаживание рёбер между гранями, как и в закраске Гуро, однако добавляется формирование зеркальных бликов, которые придают объекту блестящий или глянцевый вид. Эта модель хорошо подходит для раскрашивания любых типов пластиковых или блестящих поверхностей.

Данный способ требует больших вычислительных затрат, однако он позволяет разрешить многие проблемы метода Гуро. При закрашке Гуро вдоль сканирующей строки интерполируется значение интенсивности, а при закрашке Фонга – вектор нормали. Затем он используется в модели освещения для вычисления интенсивности пикселя. При этом достигается лучшая локальная аппроксимация кривизны поверхности, следовательно, получается более реалистичное изображение [8].

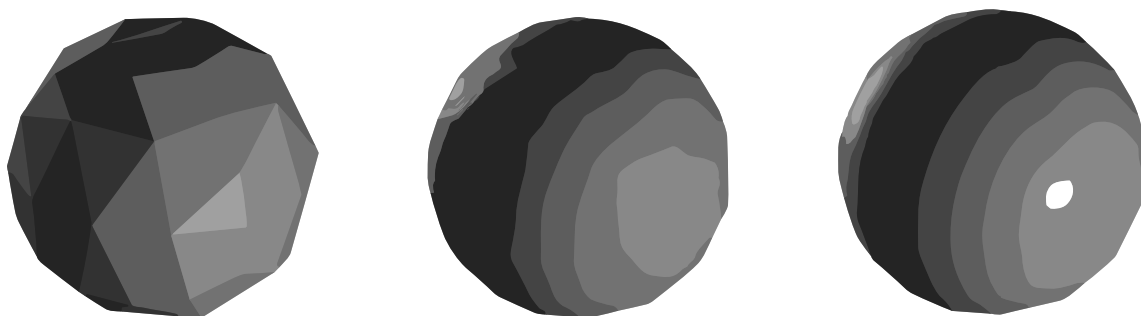


Рисунок 1.3 – Простая закрашка, методом Гуро и методом Фонга соответственно

1.8 Выбранный алгоритм закрашки

Исходя из потребностей в быстродействии и простоте алгоритма, в качестве основного способа для закрашки моделей был выбран простой алгоритм закрашки.

2. Конструкторская часть

2.1 Требования к программному обеспечению

Программа должна предоставлять графический интерфейс, обладающий следующим функционалом:

- 1) Перемещение, поворот и масштабирование сцены
- 2) Добавление, удаление и перемещение объектов сцены
- 3) Добавление и удаление источников звуковых волн, а также задание их параметров

2.2 Общий алгоритм построения сцены

Общий алгоритм построения сцены представлен на рисунке 2.1. На вход подаются параметры моделей, камеры, источника звуковых волн и самой волны, если она есть. На выходе – сцена в текущий момент времени.

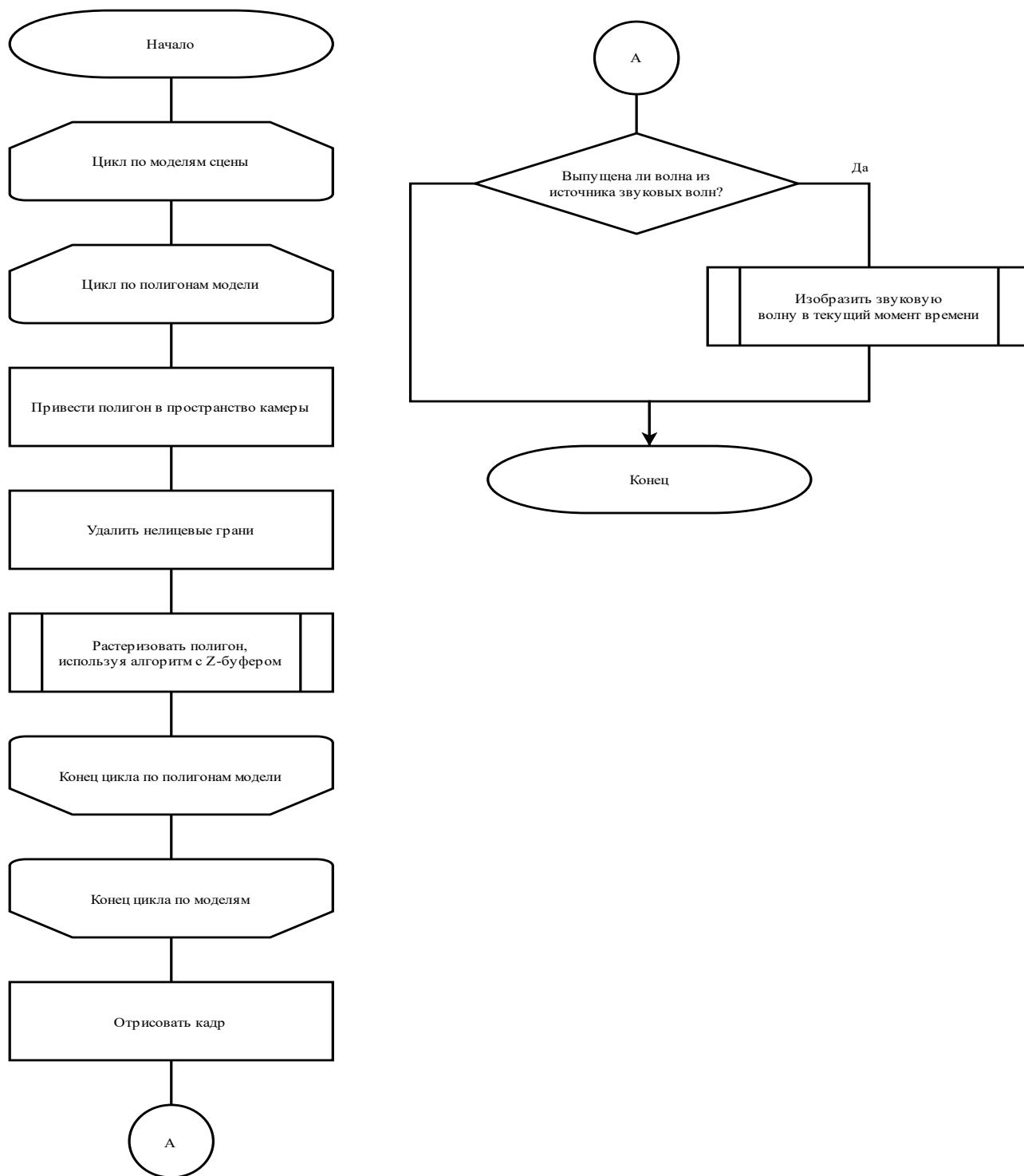


Рисунок 2.1 — Общая схема алгоритма построения сцены

2.3 Алгоритм изображения звуковой волны в текущий момент времени

Схема алгоритма изображения звуковой волны в текущий момент времени представлена на рисунке 2.2. На вход подаются параметры волны. На выходе — сцена с изображенной звуковой волной.

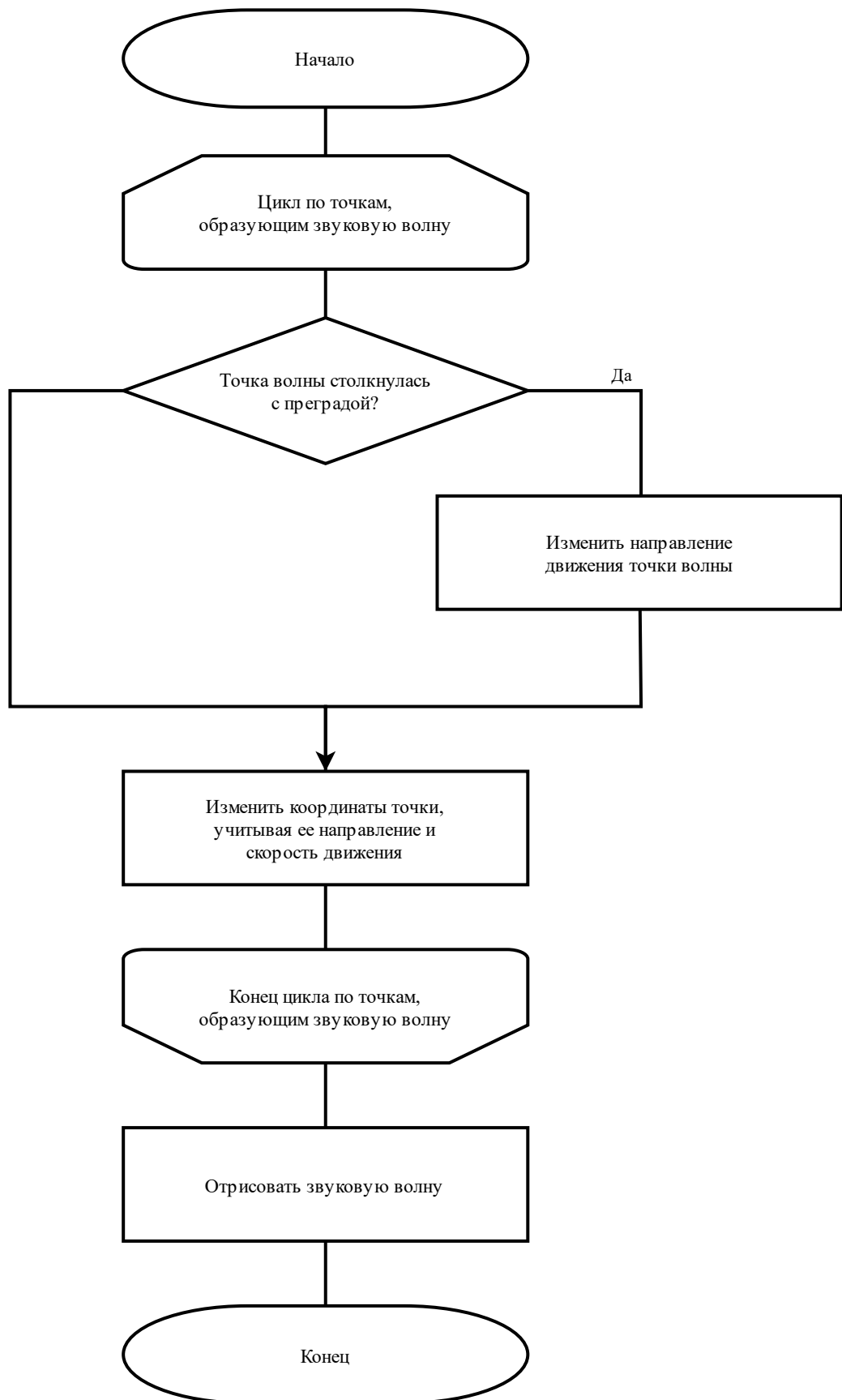


Рисунок 2.2 — Схема алгоритма изображения звуковой волны в текущий момент времени

2.4 Алгоритм простой закрашки изображения

Схема алгоритма простой закрашки представлена на рисунке 2.3. На вход подаются параметры модели, интенсивность и цвет. На выходе – закрашенная модель.

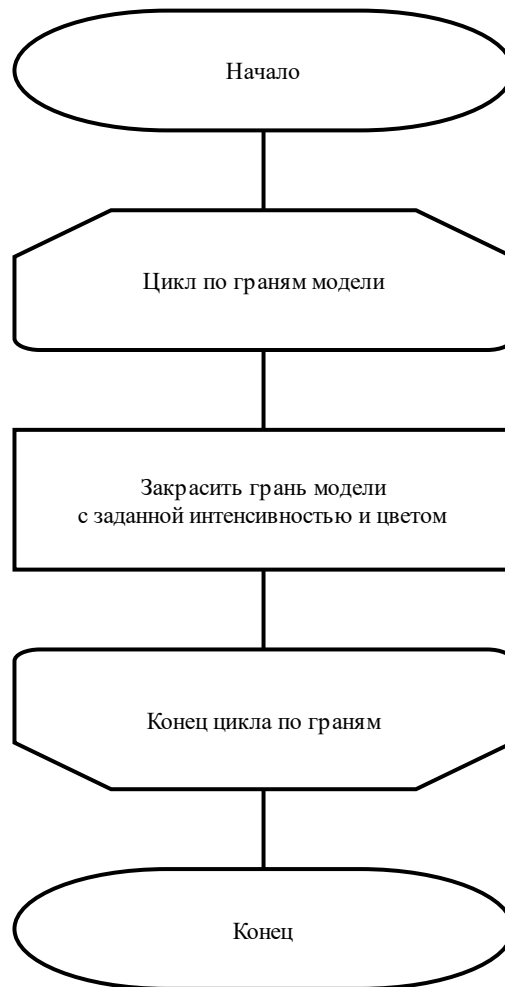


Рисунок 2.3 — Схема алгоритма простой закрашки

2.5 Алгоритм, использующий Z-буфер

Алгоритм, работающий с Z-буфером, обрабатывает сцену, представляющую собой набор объектов, каждый из которых характеризуется своим положением, ориентацией и свойствами поверхности. Затем для каждого объекта алгоритм выполняет растеризацию, т.е. преобразование его трёхмерных координат в двумерные. В ходе этого процесса также вычисляется глубина

объекта относительно камеры. Схема работы алгоритма представлена на рисунке 2.4 [10]:

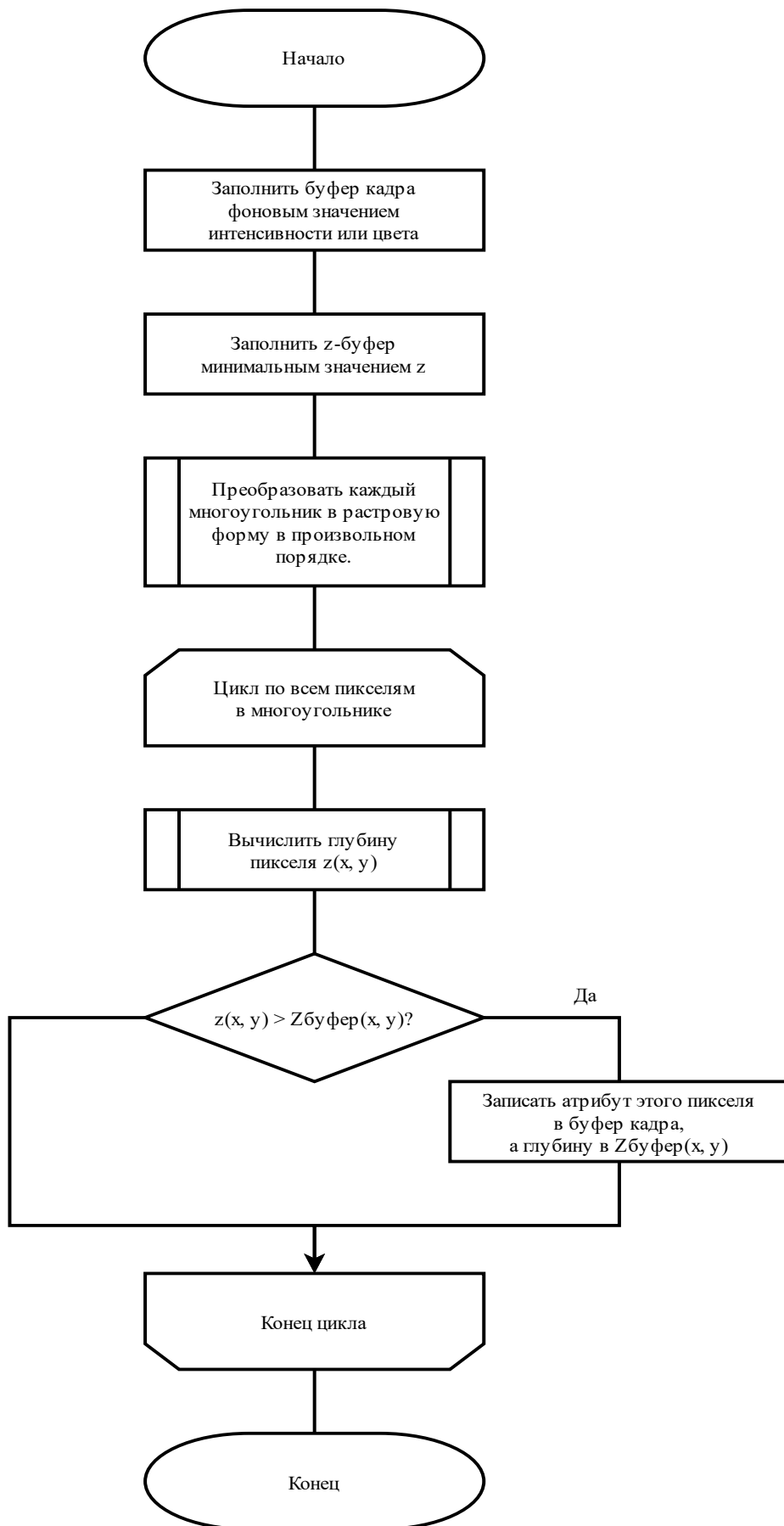


Рисунок 2.4 — Схема алгоритма, использующего Z-буфер

2.6 Используемые типы и структуры данных

В процессе разработки программного обеспечения, необходимо реализовать следующие типы и структуры данных:

- Точка в трехмерном пространстве – задание координатами x , y , z
- Грань – задается тремя точками в трехмерном пространстве
- Объект – задается списком граней и цветом
- Источник звуковых волн – задается точкой в трехмерном пространстве с заданными параметрами
- Камера – задается двумя точками в трехмерном пространстве

2.7 Вывод

Исходя из теоретических данных и знаний, полученных в аналитическом разделе, были сформированы основные требования к программному обеспечению. Также были описаны общие алгоритмы построения сцены, алгоритм, использующий Z-буфер, и алгоритм простой закрашки, что позволит решить поставленные задачи. Также, был сформулирован список используемых типов и структур данных.

3. Технологическая часть

3.1 Выбор средств для разработки

Для реализации программного обеспечения был выбран язык C++, исходя из следующих преимуществ:

- 1) *Производительность*: C++ позволяет создавать очень эффективные программы, благодаря прямому доступу к аппаратным ресурсам и отсутствию “дополнительного слоя” абстракции, как это происходит в других языках программирования
- 2) *Гибкость*: C++ является языком программирования с широкими возможностями, что делает его гибким выбором для разработки графических приложений. Он позволяет создавать программы любой сложности, от простых визуализаторов до сложных игр или симуляций
- 3) *Язык преподавался ранее*: C++ преподавался в рамках курса «Объектно-ориентированное программирование», что позволяет увеличить скорость разработки путем отбрасывания знакомства с возможностями языка
- 4) *Большое количество информации и пособий*: C++ является довольно популярным языком программирования, поэтому в сети есть большое количество методичек, учебных пособий и материалов, позволяющие качественно, удобно и быстро решать самые разные задачи

Для удобного и качественного написания графического интерфейса был выбран фреймворк Qt, обладающий следующими преимуществами:

- 1) *Простота использования*: Qt имеет простой и понятный API, который облегчает процесс разработки
- 2) *Кроссплатформенность*: Qt позволяет создавать приложения, которые работают на разных платформах, таких как Windows, macOS, Linux и другие. Это делает разработку более удобной и эффективной

3) *Богатая функциональность*: Qt предоставляет большое количество виджетов, инструментов для работы с графикой и других возможностей, которые могут быть полезны при создании приложений

Заключение

В рамках практики по компьютерной графике основной задачей были анализ и выбор существующих алгоритмов для создания реалистических и качественных трехмерных сцен.

В процессе достижения поставленной задачи, были рассмотрены методы удаления невидимых линий и поверхностей и способы закраски, по результатам изучения которых были выбраны наиболее подходящие алгоритмы: алгоритм, использующий Z-буфера и алгоритм простой закраски.

Таким образом, был выполнен подготовительный этап перед разработкой ПО, который также включал в себя выбор необходимых для реализации инструментов.

Список использованных источников

- [1] Емельянова Т. В., Аминов Л. А., Емельянов В. А. Реализация алгоритма удаления невидимых граней [Электронный ресурс]. – Режим доступа: https://elibrary.ru/download/elibrary_45049594_78874337.pdf (дата обращения: 16.07.23)
- [2] Витиска Н.И., Гуляев Н.А., Данилов И.Г., Селянкин В.В. Оптимизация прямой объемной визуализации с программируемым управлением качеством [Электронный ресурс]. – Режим доступа: <https://elibrary.ru/item.asp?id=44605108> (дата обращения: 16.07.23)
- [3] Королёв Д. А., Соломатин Д. И. Оптимизация метода обратной трассировки лучей с помощью bvh-дерева [Электронный ресурс]. – Режим доступа: <https://elibrary.ru/item.asp?id=35201148> (дата обращения: 18.07.23)
- [4] Пилипейко М. Н., Аксенова Н. А. Проект исследования метода обратной трассировки лучей [Электронный ресурс]. – Режим доступа: http://elib.gsu.by/bitstream/123456789/54736/1/%D0%9F%D0%B8%D0%BB%D0%B8%D0%BF%D0%B5%D0%B9%D0%BA%D0%BE_%D0%9F%D1%80%D0%BE%D0%B5%D1%82_%D0%98%D1%81%D1%81%D0%BB%D0%B5%D0%B4%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F.pdf (дата обращения: 18.07.23)
- [5] Алгоритмы трехмерной машинной графики: учеб. пособие по курсу "Машинная графика" / Авдеева С. М., Куров А. В.; МГТУ им. Н. Э. Баумана. - М.: Изд-во МГТУ им. Н. Э. Баумана, 1996. - 59 с.
- [6] Головнин А. А. Базовые алгоритмы компьютерной графики [Электронный ресурс]. – Режим доступа: https://elibrary.ru/download/elibrary_27220138_21752297.pdf (дата обращения: 20.07.23)
- [7] Исаев А. Л., Андросова Е. Е. Компьютерное моделирование комбинации из трехмерных объектов [Электронный ресурс]. – Режим доступа:

https://elibrary.ru/download/elibrary_27195686_13483712.pdf

(дата

обращения: 20.07.23)

- [8] Палташев Т. Т. Мультипроцессорные акселераторы в системах визуализации графических станций [Электронный доступ]. – Режим доступа:

<https://viewer.rsl.ru/ru/rsl01000333454?page=1&rotate=0&theme=white> (дата

обращения: 20.07.23)

- [9] Иванова Ю. А. Методы закрашки [Электронный ресурс]. – Режим доступа:

[https://portal.tpu.ru/SHARED/j/JBOLOTOVA/academic/ComputerGraphics/8.](https://portal.tpu.ru/SHARED/j/JBOLOTOVA/academic/ComputerGraphics/8.%20%D0%9C%D0%B5%D1%82%D0%BE%D0%B4%D1%8B%20%D0%B7%D0%B0%D0%BA%D1%80%D0%B0%D1%81%D0%BA%D0%B8_2019.pdf)

[%20%D0%9C%D0%B5%D1%82%D0%BE%D0%B4%D1%8B%20%D0%B7%D0%B0%D0%BA%D1%80%D0%B0%D1%81%D0%BA%D0%B8_2019.](https://portal.tpu.ru/SHARED/j/JBOLOTOVA/academic/ComputerGraphics/8.%20%D0%9C%D0%B5%D1%82%D0%BE%D0%B4%D1%8B%20%D0%B7%D0%B0%D0%BA%D1%80%D0%B0%D1%81%D0%BA%D0%B8_2019.pdf)

[pdf](https://portal.tpu.ru/SHARED/j/JBOLOTOVA/academic/ComputerGraphics/8.%20%D0%9C%D0%B5%D1%82%D0%BE%D0%B4%D1%8B%20%D0%B7%D0%B0%D0%BA%D1%80%D0%B0%D1%81%D0%BA%D0%B8_2019.pdf) (дата обращения: 24.07.23)

- [10] Электронный учебник по компьютерной графике [Электронный ресурс].

– Режим доступа: [https://ychebnikompgrafblog.wordpress.com/4-6-](https://ychebnikompgrafblog.wordpress.com/4-6-%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC-%D0%B8%D1%81%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D1%83%D1%8E%D1%89%D0%B8%D0%B9-z-%D0%B1%D1%83%D1%84%D0%B5%D1%80/)

[%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%](https://ychebnikompgrafblog.wordpress.com/4-6-%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC-%D0%B8%D1%81%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D1%83%D1%8E%D1%89%D0%B8%D0%B9-z-%D0%B1%D1%83%D1%84%D0%B5%D1%80/)

[BC-](https://ychebnikompgrafblog.wordpress.com/4-6-%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC-%D0%B8%D1%81%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D1%83%D1%8E%D1%89%D0%B8%D0%B9-z-%D0%B1%D1%83%D1%84%D0%B5%D1%80/)

[%D0%B8%D1%81%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D1%](https://ychebnikompgrafblog.wordpress.com/4-6-%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC-%D0%B8%D1%81%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D1%83%D1%8E%D1%89%D0%B8%D0%B9-z-%D0%B1%D1%83%D1%84%D0%B5%D1%80/)

[83%D1%8E%D1%89%D0%B8%D0%B9-z-](https://ychebnikompgrafblog.wordpress.com/4-6-%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC-%D0%B8%D1%81%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D1%83%D1%8E%D1%89%D0%B8%D0%B9-z-%D0%B1%D1%83%D1%84%D0%B5%D1%80/)

[%D0%B1%D1%83%D1%84%D0%B5%D1%80/](https://ychebnikompgrafblog.wordpress.com/4-6-%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC-%D0%B8%D1%81%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D1%83%D1%8E%D1%89%D0%B8%D0%B9-z-%D0%B1%D1%83%D1%84%D0%B5%D1%80/) (дата обращения: 26.07.23)