

# Оглавление

<b>Введение</b>	<b>2</b>
<b>1 Аналитическая часть</b>	<b>3</b>
1.1 Способы определения моделей . . . . .	3
1.1.1 Каркасная модель . . . . .	3
1.1.2 Поверхностная модель . . . . .	4
1.1.3 Твёрдотельная модель . . . . .	5
1.2 Способы представления трёхмерных поверхностей . . . . .	6
1.2.1 Способы описания полигональных сеток . . . . .	7
1.3 Формализация объектов сцены . . . . .	9
1.4 Алгоритмы удаления невидимых линий и поверхностей . . .	10
1.4.1 Вывод . . . . .	13
1.5 Алгоритм построения теней . . . . .	14
<b>Литература</b>	<b>16</b>

# Введение

Программное моделирование загородной местности с каждым годом набирает все большее значение в области ландшафтного дизайна, архитектуры и планирования. Применение современных технологий дает возможность разработки детализированных трехмерных моделей, обеспечивая тем самым более глубокое понимание и визуализацию географических и ландшафтных особенностей, а также предстоящих изменений. Использование метода 3D-визуализации при планировании местности преобразует сложные концепции в ясные визуальные образы, что значительно упрощает процесс коммуникации и минимизирует риск возникновения недопонимания между заказчиком и исполнителем [1].

Цель работы - разработка программного обеспечения для визуализации и макетирования загородной местности. ?

Чтобы достигнуть поставленной цели, требуется решить следующие задачи:

- Формально описать структуру моделей.
- Выбрать алгоритмы трехмерной графики для визуализации сцены и объектов.
- Выбрать язык программирования и среду разработки.
- Реализовать выбранные алгоритмы.
- Реализовать программное обеспечения для визуализации и редактирования загородной местности. ?

# 1 Аналитическая часть

## 1.1 Способы определения моделей

В системах трехмерного моделирования используются широко используется, каркасные, поверхностные и объемные твердотельные модели. Правильный выбор метода определения моделей на сцене является ключевым фактором, определяющим размер и визуализацию модели в сцене, что в свою очередь способствует точному представлению их формы и размеров.

### 1.1.1 Каркасная модель

Каркасные модели представляют объекты, созданные из соединенных ребер, похожие на объекты, сделанные из проволоки. В таких моделях грани объекта не определены, но их границы представлены ребрами. Каркасная модель не имеет поверхности, которые бы скрывали ребра, поэтому она выглядит прозрачной [2].

Этот тип моделирования относится к категории наиболее примитивных и имеет ряд значительных ограничений. Большинство из них связаны с отсутствием информации о гранях, образованных линиями, а также с невозможностью разграничить внутреннюю и внешнюю зоны в изображении твердого тела. Несмотря на ограничения, каркасная модель занимает меньше памяти и является достаточно эффективной для выполнения простых задач. Этот тип представления часто применяется не столько для моделирования, сколько для отображения моделей в качестве одного из методов визуализации. Наиболее широко каркасное моделирование используется для имитации траектории движения инструмента, выполняющего несложные операции [3].

**Недостатки каркасной модели:**

1. Неоднозначность — визуализация всех ребер может привести к неясности и непредсказуемым результатам. Видимые грани и невидимые трудно отличить.
2. Затруднения в идентификации криволинейных граней: каркасная модель не способна адекватно представить поверхности с непостоянной кривизной, например, боковые поверхности цилиндров, что может вводить в заблуждение.
3. Ограниченность в обнаружении взаимодействий между компонентами - из-за отсутствия информации о поверхностях, модель не может предсказать возможные конфликты между гранями объекта. ?(Ограничения в выявлении взаимодействий между элементами: отсутствие информации о поверхностях в каркасной модели препятствует анализу возможных взаимных влияний между гранями объекта.)
4. Трудности в вычислении физических характеристик из-за простоты модели.
5. Невозможность создания градационных изображений: каркасная модель не предполагает создания тоновых изображений, которые требуют затенения граней, а не ребер [3]. ??

### 1.1.2 Поверхностная модель

Поверхностные модели включают как ребра, так и поверхности, что позволяет более точно представить объект, чем каркасные модели. В поверхностных моделях грани, расположенные спереди, перекрывают грани, находящиеся на заднем плане. Когда изображение выводится на монитор, можно получить более реалистичное представление трехмерного объекта. Поверхностные модели имеют объем, но не имеют массы [2]

Поверхностная модель имеет следующие преимущества по сравнению с каркасным:

1. Способность распознавания и изображения сложных криволинейных граней.
2. Изображение грани для получения тоновых изображений.
3. Особые построения на поверхности (отверстия).
4. Возможность получения качественного изображения [3].

В основу поверхностной модели положены два основных математических положения:

1. Любую поверхность можно аппроксимировать многогранником, каждая грань которого является простейшим плоским многоугольником;
2. Наряду с плоскими многоугольниками в модели допускаются поверхности второго порядка и аналитически не описываемые поверхности, форму которых можно определить с помощью различных методов аппроксимации и интерполяции. В отличие от каркасного моделирования каждый объект имеет внутреннюю и внешнюю часть [3].

Недостаток поверхностной модели — отсутствует информация, о том, с какой стороны поверхности находится материал, а с какой пустота.

### **1.1.3 Твердотельная модель**

Твердотельное моделирование представляет собой наиболее всесторонний и точный метод создания виртуального аналога реального объекта. В результате применения данного метода формируется монолитная модель будущего изделия, включающая в себя такие элементы как линии, грани и, что особенно важно, образует зону поверхности в рамках геометрической формы объекта с такими ключевыми характеристиками как масса и объем [3].

**Преимущества твердотельных моделей:**

1. Полное определение объемной формы с возможностью разграничивать внутренний и внешние области объекта.
2. Обеспечение автоматического удаления скрытых линий.
3. Автоматическое построение 3D разрезов компонентов, что особенно важно при анализе сложных сборочных изделий.
4. Применение методов анализа с автоматическим получением изображения точных весовых характеристик методом конечных элементов.
5. Получение тоновых эффектов, манипуляции с источниками света [3].

## Выбор определения моделей

В контексте представленной задачи наиболее оптимальным выбором являются поверхностные модели объектов. Поверхностные модели позволяют детализировать геометрическую форму объекта, минуя углубленное рассмотрение его внутренней структуры и свойств материалов. Данный подход находит свое применение в ситуациях, когда специфика материала объекта не влияет на проводимый анализ. Так, не требуется тратить ресурсы на моделирование внутренних особенностей и характеристик материала, что делает использование поверхностных моделей более рациональным. Каркасные модели в данном контексте не подходят из-за неполноты представления формы объекта, в то время как твердотельные модели избыточны из-за детальности, не требуемой для задачи.

### 1.2 Способы представления трехмерных поверхностей

Для представления трехмерных поверхностей существуют два широко используемых метода:

1. Метод полигональных сеток: Этот метод представляет объект в виде связанной между собой сетки плоских многоугольников. Несмотря на его простоту и удобство для описания некоторых типов объектов, например, архитектурных конструкций, метод имеет ограниченную точность, особенно в случаях сложных или криволинейных форм.
2. Метод параметрических бикубических кусков: Этот подход использует математические формулы, описывающие координаты поверхностей. Эти уравнения имеют два параметра и варьируются по степеням не выше третьей. Этот подход обеспечивает высокую точность при описании поверхности и требует меньше элементов для описания сложных форм, в сравнении с полигональными сетками [2].

Исходя из требований к простоте моделирования, экономии ресурсов и приемлемой детализации для задачи макетирования загородной среды, метод полигональных сеток является оптимальным выбором. Благодаря геометрической простоте, он подходит для представления основных элементов и обеспечивает рациональное использование вычислительных ресурсов. При этом, уровень детализации соответствует требованиям проекта, не делая необходимым применение более сложных методов моделирования.

### **1.2.1 Способы описания полигональных сеток**

Существует несколько способов описания полигональных сеток, каждый из которых имеет свои преимущества и недостатки в зависимости от конкретных требований и ограничений приложения.

**Наиболее распространенные методы представления полигональных сеток:**

- **Список граней** — Один из наиболее распространенных подходов к представлению трехмерных моделей, представляет объект как мно-

жество граней и множество вершин. В каждую грань входят как минимум 3 вершины [4].

#### **Преимущества использования списка граней:**

- Простота поиска вершин грани: Благодаря тому, что список всех граней содержит информацию обо всех связанных вершинах, поиск вершин грани облегчается.
- Возможность динамического обновления формы: Используя графический процессор, форма может быть обновлена динамически без обновления связности граней [4].

#### **Недостатки использования списка граней:**

- Трудности при выполнении операций разрыва и объединения граней: Недостаток явной информации о связях между гранями может усложнить выполнение таких операций.
  - Проблемы с поиском граней: В силу отсутствия явно заданных ребер, поиск всех граней может стать затруднительным [4].
- **вершинное представление** — Это метод представления модели через коллекцию вершин, которые связаны между собой [4].

#### **Преимущества использования вершинного представления:**

- Простота: Вершинное представление является наиболее базовым и простым способом представления трехмерной модели [4].

#### **Недостатки использования вершинного представления:**

- Отсутствие явного выражения информации о гранях и рёбрах: Для генерации списка граней для визуализации, требуется пройти по всем данным.
- Редкое использование: Из-за ограниченности функционала, вершинное представление редко используется в современных системах визуализации [4].



- **«Крылатое» представление** — Это метод представления модели, который решает проблему перехода от ребра к ребру, путем упорядочивания множества граней вокруг каждого ребра [4].

**Преимущества использования «крылатого» представления:**

- Поддержка сложных операций: "Крылатое" представление подходит для операций подразделения поверхностей и интерактивного моделирования, благодаря своей уникальной структуре [4].

**Недостатки использования «крылатого» представления:**

- Высокие требования к памяти: Увеличивающаяся сложность структуры приводит к увеличению объема занимаемой памяти при использовании данного представления [4].

## **Выбор способа описания полигональной сетки**

В своей работе я выбрал подход к представлению моделей с использованием списка граней, поскольку он предлагает ясное описание граней и удобный доступ к элементам сетки. Список граней позволяет легко и эффективно модифицировать модели, включая добавление, удаление и изменение граней, вершин и ребер. Благодаря этому методу, я могу выполнять различные операции, такие как вычисление нормалей и текстурирование, с высокой эффективностью.

### **1.3 Формализация объектов сцены**

После изучения и выбора наиболее эффективных методов и технологий, а именно : использование поверхностных моделей для описания трех-

мерных объектов, применение полигональных сеток для визуализации поверхностей и выбор списка граней для их детализации, можем перейти к конкретизации объектов сцены.

на сцене могут размещаться следующие объекты:

- **Дома** —
- **Тротуары** —
- **Дороги** —
- **Машины** —
- **Деревья** —
- **Кусты** —

Для моделирования освещения в компьютерной графике обычно используются три основных типа источников света: точечные, направленные и общий свет (Ambient Light) [5]. В данной работе выбор сделан в пользу точечного источника света.

Точечный источник света излучает свет равномерно во все стороны из определенной точки в 3D-пространстве, что позволяет эффективно управлять освещением и тенями на сцене, учитывая положение объектов относительно источника света. Важно подчеркнуть, что положение источника света устанавливается относительно текущей точки наблюдения с помощью последовательных поворотов по осям  $X$  и  $Y$ .

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

После того, как вершины прошли все этапы геометрических преобразований и процедуру отсечения, «на конвейере» остались только геомет-

рические объекты, которые потенциально могут попасть в формируемое изображение. Но перед тем, как приступить к их преобразованию в растр, нужно решить еще одну задачу – удалить объекты, перекрываемые с точки зрения наблюдателя другими объектами

### **Алгоритм Робертса**

Алгоритм Робертса (в пространстве объектов) Первый из алгоритмов удаления невидимых линий. Требуется, чтобы каждая грань была выпуклым многоугольником:

шаг 1: Отбросить ребра, обе инцидентные грани которых являются нелицевыми;

шаг 2: Проверка закрывания каждого оставшегося ребра лицевыми гранями

2.1. грань ребра не закрывает;

2.2. грань полностью закрывает ребро;->раннее отсечение

2.3. частично закрывает – ребро разбивается на части и остаются только видимые части,  $\leq 2$

### **Алгоритм, использующий Z буфер.**

- Это один из простейших алгоритмов удаления невидимых поверхностей.
- Работает в пространстве изображений.
- В этом алгоритме используется идея о буфере кадра.

Буфер кадра используется для заполнения атрибутов (интенсивности) каждого пикселя в пространстве изображения. В данном алгоритме используется два буфера: буфер регенерации и собственно сам z-буфер, куда можно помещать информацию о координате z для каждого пикселя. В начале z-буфер кладут минимально возможные значения z, а в буфер регенерации кладут пиксели, описывающие фон. Затем каждый мно-

гоугольник приводят к растровому виду, и записывают в буффер регенерации (без упорядочивания) В процессе подсчета глубины нового пикселя (который надо занести в буффер кадра), сравнивается с тем значением, что лежит в z-буффере. Если новый пиксель расположен ближе, то он заносится в буффер кадра, при этом происходит корректировка z-буффера: в него заносится глубина нового пикселя. В сущности, алгоритм для каждой точки ищет максимальное значение  $z$  для каждой точки  $(x, y)$ . Вычисление глубины  $z$  Многоугольник описывается уравнением:  $Ax + By + Cz + D = 0$ , отсюда получаем:  $z = -(Ax + By + D) / C$  При  $c = 0$  - многоугольник для наблюдателя вырождается в линию. Ясно, что для сканирующей строки  $y = \text{const}$ . Поэтому, можем рекурентно считать  $z'$  для каждого  $x' = x + dx$ .  $z' - z = -(ax' + d) / c + (ax + d) / c = a(x - x') / c$ , откуда  $z' = z - (a / c)(x - x')$ , потому что  $(x - x') = dx = 1$  (единичный шаг раstra) Глубина пикселя, являющегося пересечением сканирующей строки с ребром многоугольника Сначала определяют ребра грани, вершины которых лежат по разные стороны от сканирующей строки, так как только в этом случае сканирующая строка пересекает ребро. Затем из найденных точек пересечения выбирают ближайшую к наблюдателю. Глубину определяют по соотношению:

$$z_3 = z_2 + (y_3 - y_2) / (y_2 - y_1) * (z_2 - z_1), \text{ где}$$

- $(y_1, z_1), (y_2, z_2)$  - координаты вершин проекции ребра на плоскость YOZ.
- $(x_3, z_3)$  - координаты проекции точки пересечения на ту же плоскость.

Оценка эффективности

Плюсы:

- Сцены могут быть произвольной сложности.
- Не нужна сортировка, как в других алгоритмах.
- Трудоемкость линейно зависит от числа рассматриваемых поверхностей

Недостатки:

- Большой объем памяти (под буфферы).
- Трудоемкость устранения лестничного эффекта.
- Трудность реализации эффектов прозрачности.

Псевдокод (алгоритм)

1. Инициализация буфера кадра фоновым значением.
2. Инициализация z-буфера минимальным значением  $Z$ .
3. Растровая развертка каждого многоугольника (в произвольном порядке).
4. Вычисление глубины  $z = (x, y)$  для каждого пикселя, принадлежащего многоугольнику.
5. Сравнение полученной глубины  $z$  со значением  $z$ , лежащей в буфере (для пикселя  $(x, y)$ ).

если полученная глубина больше значения в буфере, то записать атрибут многоугольника в буфер кадра и заменить значение в Z-буфере на полученное значение

Для невыпуклых многогранников, предварительно надо удалить нелицевые грани. Алгоритм так же можно применять для построения сечений поверхностей, в таком случае изменится только операция сравнения:

### 1.4.1 Вывод

На основании рассмотренных алгоритмов удаления невидимых поверхностей, я пришел к выводу, что использование алгоритма Z-буфера является предпочтительным для данной задачи. Размер изображения не является

очень большим, поэтому использование Z-буфера не вызовет проблем с памятью. Кроме того, данный алгоритм обеспечивает более эффективную обработку множества объектов в сцене.

Одним из главных преимуществ выбора алгоритма Z-буфера является его легкость в понимании и отладке. Алгоритм основан на простом принципе сравнения глубины пикселей с текущим значением в Z-буфере, что облегчает понимание его работы и возможность быстрой и эффективной отладки.

Таким образом, на основании этих факторов, я буду использовать алгоритм Z-буфера для удаления невидимых объектов в данном контексте.

## 1.5 Алгоритм построения теней

Учет теней в алгоритмах удаления невидимых поверхностей может быть осуществлен путем модификации Z-буфера. Z-буфер является буфером глубины, который хранит информацию о глубине каждого пикселя в сцене. При обработке геометрических объектов и определении их видимости, Z-буфер используется для сравнения текущей глубины пикселя с сохраненным значением в буфере. Однако, для учета теней необходимо внести изменения в этот процесс.

Одним из подходов к учету теней является модификация Z-буфера путем введения дополнительной информации о тени. Для каждого пикселя в Z-буфере помимо значения глубины сохраняется также значение, указывающее, находится ли данный пиксель в тени или нет. Это значение может быть булевым флагом или числовым значением, указывающим интенсивность тени.

При рендеринге сцены с учетом теней, перед сравнением глубины текущего пикселя с сохраненным значением в Z-буфере, также проверяется, находится ли данный пиксель в тени. Если пиксель находится в тени, то его глубина не обновляется, и он может быть отображен с учетом соответ-

ствующей интенсивности тени.

Важным аспектом при использовании модифицированного Z-буфера для учета теней является правильное определение, какие пиксели находятся в тени. Для этого необходимо провести предварительные вычисления и определить, какие объекты или грани находятся между источником света и точкой наблюдения, и соответственно, создают тень.

Таким образом, модификация Z-буфера позволяет эффективно учитывать тени при рендеринге сцены и одновременно выполнять удаление невидимых поверхностей. Этот подход требует дополнительных вычислений и хранения информации о тени в буфере, но позволяет достичь реалистичного отображения теней на изображении.

# Литература

- [1] A. Lovett K. Appleton B. Warren-Kretzschmar. Using 3D visualization methods in landscape planning: An evaluation of options and practical issues // Landscape and Urban Planning. 2015. С. 85–94.
- [2] Лисяк В.В. Основы геометрического моделирования. 2022. С. 6–7.
- [3] В. Ю. Донченко Е.Н. Черевань. Обзор и анализ методов построения геометрических моделей сложных конструкций. Режим доступа: [https://dspace.lgpu.org/xmlui/bitstream/handle/123456789/4116/700-19\\_%20sb.pdf?sequence=1&isAllowed=y#page=17](https://dspace.lgpu.org/xmlui/bitstream/handle/123456789/4116/700-19_%20sb.pdf?sequence=1&isAllowed=y#page=17) (дата обращения: 19.07.2023). 2014.
- [4] А.В. Киселёв Г.Н. Верхотурова. Способы представления и размещения трехмерных моделей для прототипирования ювелирных изделий // Материалы VI Международной научно-практической конференции (школы-семинара) молодых ученых. 2020. С. 840–842.
- [5] Gambetta G. Computer Graphics from Scratch. 2021. С. 47–50.