



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 ИУ «Информатика и системы управления»

КАФЕДРА ИУ-7 «Программное обеспечение ЭВМ и информационные технологии»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К КУРСОВОЙ РАБОТЕ
НА ТЕМУ:
«Визуализация озера с растительностью и
фламинго»

Студент ИУ7-54Б
(Группа)

(Подпись, дата)

Пронина Л.Ю.
(Фамилия И.О.)

Руководитель курсовой работы

(Подпись, дата)

Кострицкий А.С.
(Фамилия И.О.)

2023 г.

Оглавление

Введение	4
1. Аналитическая часть	5
1.1. Описание объектов сцены	5
1.2. Обзор способов задания трехмерной модели	5
1.3. Обзор способов задания поверхностных моделей	6
1.4. Анализ алгоритмов удаления невидимых линий и поверхностей	7
1.4.1. Алгоритм, использующий Z-буфер	8
1.4.2. Алгоритм Робертса	8
1.4.3. Алгоритм обратной трассировки лучей	10
1.4.4. Алгоритм Варнока	10
1.4.5. Вывод	11
1.5. Анализ алгоритмов закраски	13
1.5.1. Закраска Гуро	13
1.5.2. Закраска Фонга	13
1.5.3. Вывод	14
1.6. Анализ алгоритмов построения теней	14
1.7. Анализ моделей освещения	15
1.7.1. Модель Ламберта	15
1.7.2. Модель Фонга	15
1.7.3. Вывод	17
1.8. Итоговый выбор алгоритмов	17
Список использованных источников	18

Введение

Современная компьютерная графика широко применяется в таких областях, как кинематография и компьютерные игры. Однако, алгоритмы, используемые для создания реалистичных изображений, часто требуют больших вычислительных ресурсов. Чем более высокое качество изображения необходимо получить, тем больше времени и памяти требуется для его создания. Это становится особенно проблематичным при работе с динамическими сценами, где каждый кадр требует повторных вычислений.

Цель данной работы — реализовать построение трехмерной сцены и визуализацию озера с растительностью и фламинго.

Для достижения поставленной цели, требуется выполнить следующие задачи:

- 1) описать структуру трехмерной сцены, определить объекты, из которых состоит сцена, такие как озеро, растительность и фламинго;
- 2) провести анализ существующих алгоритмов построения реалистичных трехмерных изображений, исследовать преимущества и ограничения каждого алгоритма;
- 3) выбрать и, если необходимо, модифицировать существующие алгоритмы трехмерной графики для создания реалистичных изображений;
- 4) реализовать выбранные или модифицированные алгоритмы;
- 5) создать модель трехмерного объекта фламинго;
- 6) разработать программное обеспечение, которое позволит отобразить трехмерную сцену и визуализировать озеро с растительностью и фламинго, а также перемещаться по этой сцене;
- 7) протестировать и оптимизировать производительность трехмерной сцены, чтобы обеспечить плавную визуализацию;
- 8) провести тестирование, оценить визуальные эффекты, реалистичность и функциональность созданной трехмерной сцены озера с растительностью и фламинго.

1. Аналитическая часть

В данной части проводится анализ объектов сцены и существующих алгоритмов построения изображений, а также выбор более подходящих из них для дальнейшего использования.

1.1. Описание объектов сцены

Сцена состоит из следующих объектов:

- 1) Источник света — невидимый точечный объект, который описан тремя координатами положения и коэффициентом освещенности.
- 2) Камера характеризуется своим пространственным положением и направлением просмотра.
- 3) Объекты сцены — тела, которые представляются множеством точек в пространстве и полигонов. Каждое тело задается определенными характеристиками, такими как цвет, и коэффициент отражения.

1.2. Обзор способов задания трехмерной модели

Модель – отображение форм и размеров объекта. Используются три вида модели: каркасная, поверхностная и твердотельная.

- 1) Каркасная модель.

В этой модели хранится информация только о вершинах и рёбрах объектов. Недостатком данной модели является то, что она может неправильно передавать форму объекта.

- 2) Поверхностная модель.

Поверхность может описываться аналитически или полигональной сеткой. Такая информационная модель содержит данные только о

внешних геометрических параметрах объекта. Недостатком модели является отсутствие информации о том, с какой стороны поверхности находится материал.

3) Твердотельная модель.

При твердотельном моделировании учитывается еще и материал, из которого изготовлен объект. К информации о поверхности добавляется информация о том, с какой стороны поверхности расположен материал. Это делается с помощью указания направления внутренней нормали.

Для достижения поставленной цели лучше всего подходит поверхностная модель. Каркасная модель не предоставляет достаточной информации о форме объекта и может привести к искаженному восприятию. А твердотельная модель, которая учитывает материал и его внутреннюю нормаль, не является необходимой для поставленной задачи и требует дополнительных ресурсов для реализации. Поэтому использование поверхностной модели позволит достичь требуемой визуализации озера с растительностью и фламинго, сохраняя ресурсоэффективность и достаточную точность отображения формы объектов.

1.3. Обзор способов задания поверхностных моделей

Поверхностную модель можно задать следующими способами:

- 1) Аналитический способ — характеризуется тем, что для получения поверхности нужно дополнительно вычислять функцию, зависящую от параметра.
- 2) Полигональная сетка — характеризуется тем, что информация о модели хранится в виде совокупности вершин, ребер и граней. Гранями обычно являются простые выпуклые многоугольники (полигоны).

Из двух представленных вариантов наиболее оптимальным является использование полигональной сетки, так как такой вариант задания поверхности позволит быстро выполнять операции над объектами.

Существует несколько способов хранения полигональной сетки:

- Список граней — характеризуется множеством граней и вершин. В каждую грань входят как минимум три вершины.
- Вершинное представление — характеризуется хранением информации о вершинах, которые указывают на другие вершины, с которыми они соединены.
- Таблица углов — это таблица, хранящая вершины. Ее обход неявно задаёт полигоны. Такое представление занимает меньше места и более производительное для нахождения полигонов, но операции по их изменению медленные.
- «Крылатое» представление — характеризуется заданием точек, каждая из которых указывает на 2 вершины, 2 грани и 4 ребра, которые ее касаются, благодаря чему обход поверхностей выполняется за постоянное время. Однако метод требует много памяти и изменения геометрических характеристик приводит к формированию списка индексов граней. Данный способ полезен для определения столкновений объектов.

Для хранения полигональной сетки будет использоваться список граней.

1.4. Анализ алгоритмов удаления невидимых линий и поверхностей

Для выбора оптимального алгоритма построения изображения необходимо провести обзор существующих алгоритмов и выбрать наиболее подходящий для решения данной задачи.

1.4.1. Алгоритм, использующий Z -буфер

Данный алгоритм работает в пространстве изображения [1].

Используется два буфера:

- буфер кадра, в котором хранятся атрибуты каждого пикселя в пространстве изображения;
- Z -буфер, куда помещается информация о координате z для каждого пикселя.

Алгоритм с Z -буфером начинается с инициализации Z -буфера минимальными значениями z , а буфер кадра заполняется фоновым значением интенсивности или цвета. Затем каждый многоугольник преобразуется в растровую форму и записывается в буфер кадра.

При вычислении глубины нового пикселя, ее значение сравнивается с Z -буфером. Если новый пиксель находится ближе к наблюдателю, он записывается в буфер кадра, и Z -буфер корректируется.

Для определения глубины z каждого пикселя в текущей сканирующей строке используется уравнение поверхности многоугольника вида $ax + by + cz + d = 0$. При $c = 0$ многоугольник вырождается в линию для наблюдателя. Иначе значение глубины z' вычисляется рекуррентно как $z' = z - \frac{a}{c}$.

Также необходимо помнить, что для многогранников с невыпуклыми частями, требуется предварительное удаление нелицевых граней.

Главной особенностью алгоритма является его простота реализации и высокая скорость обработки объектов. При этом использование памяти для буферов изображения для современных компьютеров является незначительным и не вызывает проблем.

1.4.2. Алгоритм Робертса

Данный алгоритм работает в объектном пространстве и для него требуется, чтобы все тела были выпуклыми. Невыпуклые тела должны быть разбиты на выпуклые части [1].

Он состоит из 3 этапов.

На первом этапе происходит подготовка исходной матрицы V , которая содержит информацию о каждом теле. Размерность матрицы составляет $4 * n$, где n - количество граней тела. Каждый столбец матрицы представляет собой четыре коэффициента уравнения плоскости, проходящей через грань. Таким образом, матрица тела имеет следующий вид:

$$V = \begin{pmatrix} a_1 & a_2 & \dots & a_n \\ b_1 & b_2 & \dots & b_n \\ c_1 & c_2 & \dots & c_n \\ d_1 & d_2 & \dots & d_n \end{pmatrix} \quad (1.1)$$

Матрица тела должна быть сформирована корректно, то есть любая точка, находящаяся внутри тела, должна находиться по положительную сторону от каждой грани. Если условие не выполняется для очередной грани, соответствующий столбец матрицы умножается на -1 . Для проверки условия необходимо взять точку, находящуюся внутри тела, координаты которой могут быть получены путем усреднения координат всех вершин тела.

На втором этапе происходит удаление ребер, закрытых самим телом, с помощью рассмотрения вектора взгляда $E = \{0, 0, -1, 0\}$. Чтобы определить невидимые грани, достаточно умножить вектор E на матрицу тела V . Отрицательные компоненты полученного вектора соответствуют невидимым граням.

Третий этап заключается в удалении невидимых ребер, закрытых другими телами сцены. Для определения невидимых точек ребра необходимо построить луч, соединяющий точку наблюдения с точкой на ребре. Точка будет невидимой, если на пути луча встречается рассматриваемое тело. Если тело оказывается преградой, луч должен проходить сквозь него. Если луч проходит сквозь тело, то он находится по положительную сторону от каждой грани тела.

Главным недостатком данного алгоритма является его вычислительная сложность, которая составляет $O(n^2)$, где n - количество объектов на сцене. Также все тела на сцене должны быть выпуклыми, что требует дополнительных проверок. Однако, работа в объектном пространстве и высокая сложность вычислений обеспечивают высокую точность результата.

1.4.3. Алгоритм обратной трассировки лучей

В настоящее время методы трассировки лучей считаются наиболее мощными в создании реалистичных изображений.

Алгоритмы прямой и обратной трассировки лучей отслеживают путь лучей от источника света до камеры и вычисляют взаимодействия лучей с объектами, которые они пересекают на своем пути. Луч может быть поглощен, диффузно или зеркально отражен, а при прозрачности объектов может быть преломлен [1].

Правила обратной трассировки требуют рассмотрения каждой частицы в осадках как отдельного объекта сцены, где могут происходить явления дисперсии, преломления и внутреннего отражения. Однако данный алгоритм требует значительного объема вычислений, так как предполагает поиск пересечений всех объектов сцены со всеми лучами. Поэтому время генерации изображения может быть очень большим.

Таким образом, данный алгоритм не подходит для моей задачи, так как не обеспечивает быструю отрисовку динамических сцен. Однако возможно повышение скорости алгоритма путем использования параллельных вычислений.

1.4.4. Алгоритм Варнока

Алгоритм Варнока оперирует в пространстве изображения и позволяет определить, какие грани или части граней объектов сцены видимы, а какие скрыты другими объектами [1].

Основная идея алгоритма заключается в разделении области изображения на более мелкие окна. Для каждого окна определяются связанные многоугольники, а затем определяется их видимость на сцене.

В алгоритме обычно используются выпуклые многоугольники в качестве граней, так как эффективность работы с ними выше, чем с произвольными многоугольниками.

Окно, в котором требуется отобразить сцену, должно быть прямоугольным. Алгоритм работает рекурсивно: на каждом шаге происходит

анализ видимости граней, и если невозможно "легко" определить видимость, окно делится на 4 части, и анализ повторяется отдельно для каждой части.

Главным недостатком данного алгоритма является его рекурсивная природа. На каждом шаге анализируется видимость граней, и если это требует дополнительных вычислений, окно делится на 4 части, что потенциально может привести к большому количеству шагов рекурсии и значительной вычислительной сложности.

1.4.5. Вывод

В качестве алгоритма удаления невидимых рёбер и поверхностей был выбран алгоритм *Z*-буфера, так как он работает в пространстве изображения и быстро производит вычисления.

Таблица 1.1 – Сравнение алгоритмов удаления невидимых линий

Критерии	Алгоритмы			
	Использующий Z-буфер	Робертса	Обратной трассировки лучей	Варнока
Основная идея	использование буфера, хранящего глубину каждого пикселя	растеризация полигонов, попиксельно сравнивается глубина полигонов и видимые растеризуются	симуляция физического взаимодействия лучей света с объектами на сцене (отражаются, преломляются или поглощаются в зависимости от свойств материалов)	растеризация полигонов, распределяются непрозрачные полигоны в порядке их удаленности от камеры и выполняется затенение полигонов в порядке удаленности
Вычислительная трудоемкость (n - кол-во граней объектов, N - кол-во пикселей)	$O(N * n)$	$O(n^2)$	$O(N * n)$	$O(N * n)$
Работает с невыпуклыми	да	нет	да, с доп. проверками	да, с доп. обработкой сцены
Рабочее пространство	изображение	объект	изображение	изображение
Применение для сцен в реальном времени	может быть эффективным, но потребляет больше памяти	используется, но может быть не эффективным для сложных сцен	обеспечивает высокое качество изображений, но может быть более затратным по времени для рендеринга сложных сцен	используется и позволяет обрабатывать большие сцены за счет эффективной сортировки и однократного прохода по полигонам

1.5. Анализ алгоритмов закрашки

Существует два наиболее распространённых метода закрашки.

1.5.1. Закраска Гуро

Метод Гуро, основанный на интерполяции интенсивности, заключается в закрашивании разных точек грани с разными значениями интенсивности. Для этого вычисляется вектор нормали в каждой вершине грани, а затем значения интенсивности интерполируются по всем точкам примыкающих граней [1].

Закрашивание граней по методу Гуро выполняется в четыре этапа. Сначала вычисляются нормали к каждой грани. Затем определяются нормали в вершинах, которые вычисляются как усреднение нормалей примыкающих граней. На основе нормалей в вершинах вычисляются значения интенсивностей в вершинах с учетом выбранной модели отражения света. Затем полигоны граней закрашиваются цветом, соответствующим линейной интерполяции значений интенсивности в вершинах.

Однако метод Гуро применим только для небольших граней, которые находятся на значительном расстоянии от источника света. Если размер грани большой, расстояние от источника света до центра грани будет меньше, чем до вершин, что должно привести к более яркой освещенности центра грани по сравнению с ребрами. Однако из-за линейного закона интерполяции, используемого в методе, это не удастся достичь, что приводит к неестественной освещенности в некоторых участках грани.

1.5.2. Закраска Фонга

Закраска Фонга, подобно закрашке Гуро, осуществляет интерполяцию, но в отличие от метода Гуро, в методе Фонга интерполируются векторы нормалей, а затем используются для определения значения интенсивности для каждой точки.

Процесс закрашки в методе Фонга включает следующие этапы:

- 1) Вычисляются нормали к граням.
- 2) На основе нормалей к граням определяются нормали в вершинах грани. Используя эти нормали, для каждой точки закрашиваемой грани вычисляется интерполированный вектор нормали.
- 3) Направление векторов нормали используется для определения цвета точек грани в соответствии с выбранной моделью отражения света.

Метод Фонга требует больших вычислительных затрат по сравнению с методом Гуро, однако он обеспечивает более точное приближение кривизны поверхности и, следовательно, позволяет получить более реалистичное изображение.

1.5.3. Вывод

В данной работе будет использоваться метод закраски Фонга, так как он дает наиболее реалистичное изображение, в частности зеркальных бликов.

1.6. Анализ алгоритмов построения теней

В предыдущем рассмотренном алгоритме трассировки лучей, тени создаются в процессе выполнения алгоритма: пиксели затеняются, если луч пересекает объект, но не достигает источника света. При использовании алгоритма с z -буфером, можно внести некоторую модификацию. Для этого добавляется теневой z -буфер, основанный на точке наблюдения, которая совпадает с источником света. Такая модификация позволяет избежать усложнения структуры программы и, следовательно, сократить время отладки программного продукта.

1.7. Анализ моделей освещения

1.7.1. Модель Ламберта

Модель Ламберта моделирует идеальное диффузное освещение. Считается, что свет при попадании на поверхность рассеивается равномерно во все стороны [1].

Свет от точечного источника отражается по закону косинусов Ламберта: интенсивность отраженного света пропорциональна косинусу угла между направлением света и нормалью к поверхности (1.2). Значение этого угла определяет, в какую сторону отражается свет. Коэффициент диффузного отражения зависит от материала и длины волны света и обычно считается постоянным в простых моделях освещения.

$$I = I_i \cdot k_d \cdot \cos(\theta), \quad (1.2)$$

где

- I — интенсивность отраженного света;
- I_i — интенсивность источника света;
- k_d — коэффициент диффузного отражения;
- θ — угол между нормалью к поверхности и вектором направления света.

Модель Ламберта является простой в реализации моделью, однако основной недостаток — одинаковая интенсивность во всех точках, принадлежащих одной грани.

1.7.2. Модель Фонга

Модель Фонга является классической моделью освещения, которая комбинирует диффузную и зеркальную составляющие. Благодаря зеркаль-

ному отражению на блестящих предметах появляются световые блики [1].

Коэффициент зеркального отражения зависит от угла падения, однако даже при перпендикулярном падении зеркально отражается только часть света, а остальное либо поглощается, либо отражается диффузно. Эти соотношения определяются свойствами вещества и длиной волны. Объединяя эти результаты с формулой рассеянного света и диффузного отражения, получим модель освещения (1.3).

$$I = I_a \cdot K_a + \frac{I_l}{d + K} \cdot (K_d \cdot \cos \theta + K_s \cdot \cos^n \alpha), \quad (1.3)$$

где

- I_a — интенсивность рассеянного света;
- K_a — коэффициент диффузного отражения рассеянного света;
- d — расстояние от центра проекции до объекта;
- K — произвольная постоянная;
- n — степень, аппроксимирующая пространственное распределение зеркально отраженного света;
- k_s — константа, заменяющая функцию $w(i, \lambda)$ — кривую отражения, представляющую отношение зеркально отраженного света к падающему как функцию угла падения i и длины волны λ .

Если имеется несколько источников света, то их эффекты суммируются. В этом случае модель освещения определяется как (1.4).

$$I = I_a \cdot K_a + \sum_{j=1}^m \frac{I_{l_j}}{d + K} \cdot (K_d \cdot \cos \theta_j + K_s \cdot \cos^n \alpha_j), \quad (1.4)$$

где m — количество источников.

1.7.3. Вывод

Была выбрана модель освещения Фонга в сочетании с закраской Фонга, так как он позволяет получить реалистичную картинку с эффектами отражения и преломления лучей.

1.8. Итоговый выбор алгоритмов

Для задания трёхмерных моделей была выбрана поверхностная модель, она будет задаваться полигональной сеткой. В качестве алгоритма удаления невидимых линий был выбран алгоритм, использующий Z-буфер, построение теней будет выполняться с помощью теневых карт (модификация алгоритма z-буфера), освещение и закраска с помощью алгоритма Фонга.

Список использованных источников

1. Д., Роджерс. Алгоритмические основы машинной графики. / Роджерс Д. — М.Мир, 1989. — С. 512.