

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 Аналитический раздел	5
1.1 Формализация объектов сцены	5
1.2 Анализ способов описания трехмерных моделей	6
1.3 Анализ алгоритмов удаления невидимых поверхностей	7
1.4 Анализ алгоритмов закраски	9
1.5 Анализ алгоритмов освещения	11
1.6 Анализ алгоритмов морфинга	12
2 Конструкторский раздел	15
3 Технологический раздел	16
4 Исследовательский раздел	17
ЗАКЛЮЧЕНИЕ	18
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	19
ПРИЛОЖЕНИЕ А	20

ВВЕДЕНИЕ

Морфинг трехмерных объектов представляет собой технику анимации, при котором объект плавно переходит в другой так, что человеческий глаз не распознает момент, когда заканчивается первый и начинается второй. С помощью морфинга создают плавные анимационные эффекты в различных областях, включая кинопроизводство, рекламу, игры и дизайн.

Целью данной работы является разработка приложения для морфинга объектов. Для достижения поставленной цели требуется решить следующие задачи:

1. Проанализировать алгоритмы морфинга трехмерных моделей;
2. Выбрать наиболее подходящий для достижения поставленной цели;
3. Выбрать средства реализации приложения.
4. Разработать приложение и реализовать выбранные алгоритмы.
5. Реализовать графический интерфейс.
6. Исследовать временные характеристики выбранных алгоритмов на основе созданного приложения.

1 Аналитический раздел

В данном разделе проводится анализ существующих алгоритмов построения изображений и выбор подходящих алгоритмов для решения задачи.

1.1 Формализация объектов сцены

Сцена состоит из следующих объектов:

- объект (стартовый или получаемый) — трехмерная модель, представляющая собой каркасное тело;
- источник света — вектор направления света;
- камера — характеризуется своим положением и направлением просмотра.

1.2 Анализ способов описания трехмерных моделей

В компьютерной графике в основном используют три типа представления трехмерных объектов: каркасная, поверхностная и твердотельная модели. Они предоставляют различные способы представления объектов и позволяют достичь правильного отображения их формы и размеров на сцене.

Каркасная модель это простейший вид моделей. В этой модели задается информация о вершинах и рёбрах объектов. Этим видам модели присущ весьма существенный недостаток: не всегда модель правильно передает представление об объекте. К преимуществам же можно отнести сравнительно низкое число затрачиваемой памяти.

Поверхностные модели несут информацию обо всех точках пространства, принадлежащих поверхности объекта, а внутренние точки в них не учитываются. К недостаткам данной модели относится отсутствие информации о том, с какой стороны поверхности находится материал.

Твердотельная модель отличается от поверхностной наличием информации о том, где расположен материал. Проще всего это можно сделать путём указания направления внутренней нормали.[1]

Для решения поставленной задачи наилучшим образом подойдет каркасная модель, так как для осуществления морфинга необходимо знать информацию только о вершинах и рёбрах объекта. Хранить остальную информацию об объекте избыточно.

1.3 Анализ алгоритмов удаления невидимых поверхностей

Алгоритмы удаления невидимых линий и поверхностей определяют, какие линии, поверхности или объемы видимы или невидимы для наблюдателя, находящегося в определенной точке пространства.

Алгоритм Робертса удаляет из каждого тела те ребра или грани, которые скрываются самим телом. Затем каждое из видимых ребер каждого тела сравнивается с каждым из оставшихся тел для определения того, какая его часть или части, если таковые есть, скрываются этими телами. При этом вычислительная трудоемкость алгоритма Робертса растет теоретически, как квадрат числа объектов. При этом математические методы, используемые в этом алгоритме, просты, мощны и точны.

Алгоритм плавающего горизонта чаще всего используется для удаления невидимых линий трехмерного представления функций, описывающих поверхность в виде $F(x, y, z) = 0$.

Идея метода заключается в сведении трехмерной задачи к двумерной путем пересечения исходной поверхности последовательностью параллельных секущих плоскостей, имеющих постоянные значения координаты z .

Алгоритм сначала упорядочивает плоскости $z = \text{const}$ по возрастанию расстояния до них от точки наблюдения. Затем для каждой плоскости, начиная с ближайшей к точке наблюдения, строится кривая, лежащая на ней, т.е. для каждого значения координаты x в пространстве изображения определяется соответствующее значение y .

Если на текущей плоскости при некотором заданном значении x соответствующее значение y на кривой больше значения y для всех предыдущих кривых при этом значении x , то текущая кривая видима в этой точке; в противном случае она невидима.

Алгоритм Коэна — Сазерленда разделяет плоскость на девять частей прямыми, которые образуют стороны прямоугольника. Каждой из девяти частей присваивается четырехбитный код.

Окну присваивается код 0000. Конечным точкам отрезка приписывается

4-битный код "вне/внутри" в зависимости от нахождения отрезка в соответствующей подобласти. Каждому биту присваивается значение 1 в соответствии со следующим правилом.

- Бит 1 — точка находится выше окна;
- Бит 2 — точка находится ниже окна;
- Бит 3 — точка находится справа от окна;
- Бит 4 — точка находится слева от окна;
- Иначе биту присваивается нулевое значение.

1001	1000	1010
0001	0000	0010
0101	0100	0110

Рисунок 1.1 – Разбиение на подобласти в методе Козна—Сазерленда

Алгоритм определяет код конечных точек отрезка. Если оба кода равны нулю, то отрезок полностью находится в прямоугольнике. Если битовое И кодов не равно нулю, то отрезок не пересекает прямоугольник (так как это значит, что обе конечные точки отрезка находятся с одной стороны прямоугольника). В прочих случаях алгоритм выбирает конечную точку, находящуюся вне прямоугольника, находит ближайшую к ней точку пересечения отрезка с одной из линий, образующей стороны прямоугольника, и использует эту точку пересечения как новую конечную точку отрезка. Укороченный отрезок снова пропускается через алгоритм. [2]

Проанализировав алгоритмы удаления невидимых линий и поверхностей, заметим, что для поставленной задачи больше всего подходит алгоритм Z-буфера, так как он может работать со сценами любой сложности и не требует больших вычислительных мощностей для сцен с множеством объектов, что необходимо при реализации морфинга.

1.4 Анализ алгоритмов закраски

Для придания реалистичности изображениям кроме удаления невидимых линий и поверхностей необходимо правильно отразить свет от присутствующих на сцене объектов. Существует несколько основных методов закрашивания изображаемых поверхностей. Среди них можно выделить:

- метод однотонной закраски;
- метод, основанный на интерполяции значений интенсивности освещенности поверхности — метод Гуро;
- метод, основанный на интерполяции векторов нормалей к граням многогранника — метод Фонга.

При однотонной закраски для изображаемого объекта вычисляется один уровень интенсивности освещения, который и используется для закраски всего объекта. Следует заметить, что, если каждая плоская грань имеет один постоянный цвет, то различные цвета соседних граней очень заметны, и поверхность выглядит как многогранник. Этот эффект можно сгладить, если увеличить число рассматриваемых граней. Так как зрение человека имеет способность подчеркивать перепады яркости на границах смежных граней (эффект Маха), для обеспечения иллюзии гладкой поверхности изображения нужно намного увеличить количество таких граней, а это приведет к снижению эффективности всей вычислительной процедуры.

Метод Гуро основывается на идее закрашивания каждой грани не одним цветом, а плавно изменяющимися оттенками, вычисляемыми путем интерполяции цветов примыкающих граней. Закрашивание граней по методу Гуро осуществляется в четыре этапа:

1. Вычисляются нормали к поверхности;
2. Определяются нормали в вершинах многогранника путем усреднения нормали по всем полигональным граням, которым принадлежит вершина;

3. Используя нормали в вершинах и применяя произвольный метод закраски, вычисляют значения интенсивности освещения в вершинах;
4. Каждый многоугольник закрашивают путем линейной интерполяции значений интенсивности в вершинах, сначала вдоль каждого ребра, а затем и между ребрами вдоль каждой сканирующей строки.

В методе закраски, разработанном Фонгом, используется интерполяция вектора нормали к поверхности вдоль видимого интервала на сканирующей строке внутри многоугольника, а не интерполяция интенсивности. Интерполяция выполняется между начальной и конечной нормалью, которые сами тоже являются результатами интерполяции вдоль ребер многоугольника между нормалью в вершинах. Нормали в вершинах, в свою очередь, вычисляются так же, как в методе закраски, построенном на основе интерполяции интенсивности. [3]

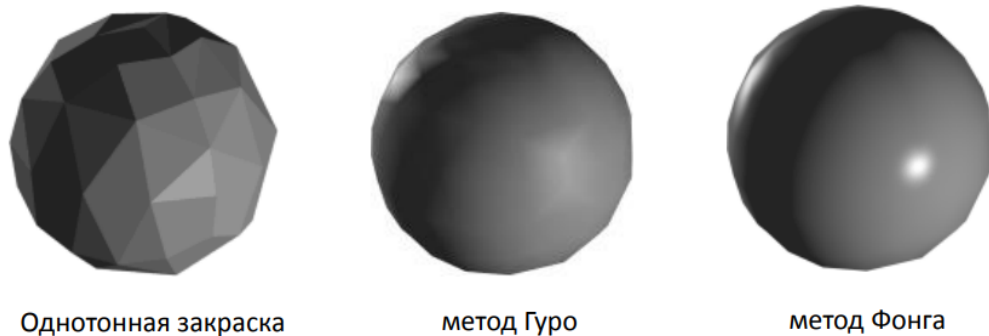


Рисунок 1.2 – Визуальное сравнение различных алгоритмов закраски

Наиболее подходящей закраской для поставленной задачи является закраска Гуро, так как программа подразумевает обработку моделей с большим количеством примитивов.

1.5 Анализ алгоритмов освещения

К основным моделям относятся модели Ламберта и Фонга, построенные на основе трехкомпонентного освещения.

В общем виде модель освещения Фонга состоит из суммы фоновой, диффузной и зеркальной составляющей и имеет следующий вид:

$$I = I_a + I_d + I_s = m_a \cdot L_a + m_d \cdot k_d \cdot L_d + m_s \cdot k_s \cdot L_s \quad (1.1)$$

В общем виде модель освещения Ламберта состоит из суммы фоновой и диффузной компонент:

$$I = I_a + I_d = m_a \cdot L_a + m_d \cdot k_d \cdot L_d \quad (1.2)$$

Модель Ламберта является одной из самых простых моделей освещения. Данная модель очень часто используется как часть других моделей, поскольку практически в любой другой модели освещения можно выделить диффузную составляющую. Более-менее равномерная часть освещения (без присутствия какого-либо всплеска), как правило, будет представляться моделью Ламберта с определенными характеристиками. Данная модель может быть очень удобна для анализа свойств других моделей (за счет того, что ее легко выделить из любой модели и анализировать оставшиеся составляющие).

В частности, модель Ламберта является существенной частью модели Фонга, которая представляет собой комбинацию диффузной составляющей (модели Ламберта) и зеркальной составляющей. [4]

Наилучшим решением для поставленной задачи будет остановиться на модели освещения Ламберта.

1.6 Анализ алгоритмов морфинга

Среди алгоритмов морфинга можно выделить три алгоритма: линейный, слиянием и д-морфинг.

Линейный морфинг основан на линейной интерполяции между начальными и конечными позициями каждой точки или вершины модели. Алгоритм состоит из следующих этапов:

1. Выравнивание вершин. Чтобы линейный морфинг работал корректно, у исходной и конечной моделей должны быть одинаковое количество вершин и морфологическая структура. Обычно это требует предварительного выравнивания вершин;
2. Интерполяция позиций вершин. Процесс начинается с вычисления позиций каждой вершины на промежуточных этапах между исходной и конечной моделями. Используется линейная интерполяция, чтобы постепенно изменять координаты вершин по оси x , y и z :

$$P_t = (1 - t) \cdot P_0 + t \cdot P_1 \quad (1.3)$$

где P_t — позиция вершины на промежуточном этапе, P_0 — позиция вершины в исходной модели, P_1 — позиция вершины в конечной модели, а t — параметры интерполяции, принимающие значения от 0 до 1;

3. Текстурирование и цвета. Наряду с изменением позиций вершин может также потребоваться интерполяция текстурных координат и цветов вершин, чтобы добиться качественного визуального эффекта;
4. Анимация и кадры. Для создания анимации между двух моделей процесс интерполяции выполняется для каждого промежуточного кадра. Значение t увеличивается от 0 до 1 по мере прогресса анимации, что создает плавный переход.

Данный алгоритм плохо справляется с задачей плавного перехода, если число вершин между исходной и итоговой моделями сильно разнится. К преимуществам линейного морфинга относится высокая скорость по сравнению с морфингом слиянием и д-морфингом.

Алгоритм морфинга слиянием предполагает «надувание» исходного и итогового объектов до сферы. Оболочка каждого объекта проецируется на единичную сферу. Это отображение используется для идентификации соответствий между точками на двух исходных объектах путем связывания пар точек, которые отображаются в одно и то же место на сфере. Затем две топологии объединяются путем обрезки спроецированных граней одной модели спроецированными гранями другой. Объединенная топология затем отображается на поверхность обеих исходных моделей. Это создает две новые модели, которые имеют ту же форму, что и исходные две модели, но которые разделяют общую топологию. Это позволяет легко вычислить преобразование между двумя формами путем интерполяции координат каждой пары соответствующих вершин.

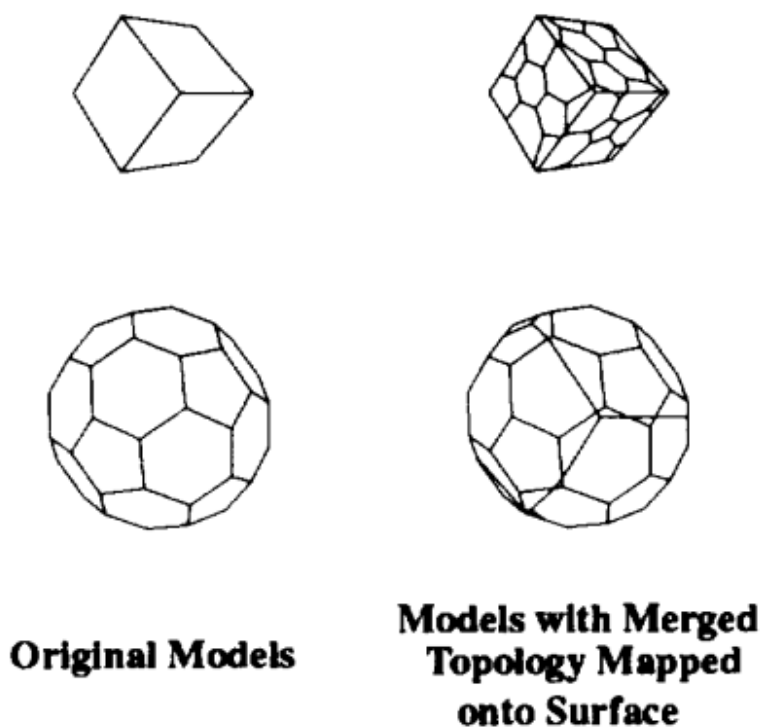


Рисунок 1.3 – Пара исходных моделей и та же пара с объединёнными топологиями

Д-морфинг разделяется на два этапа: построение описывающих фигур и генерация морфинга.

На первом этапе строятся два новых объекта: исходная фигура, изменённая с помощью параллельных переносов плоскостей, полностью включающая в себя итоговую, и наоборот. Чтобы получить морфинг, мы просто перемещаем все плоскости одновременно из их начальных положений в их конечные положения. Форма, образованная их пересечением, будет кадром морфинга. [5] [6]

После анализа популярных алгоритмов морфинга рациональным решением будет остановиться на морфинге слиянием. Такой выбор обусловлен тем, что линейный морфинг хуже работает с объектами, имеющими разное количество вершин, а д-морфинг позволяет обрабатывать только выпуклые объекты.

Вывод

В данном разделе проведен анализ существующих алгоритмов построения изображений и морфинга, и были выбраны подходящие для решения задачи.

2 Конструкторский раздел

3 Технологический раздел

4 Исследовательский раздел

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Косников Ю.* Поверхностные модели в системах трехмерной компьютерной графики. — Пензенский государственный университет, 2007.
2. Алгоритмы удаления невидимых линий и поверхностей [Электронный ресурс]. — URL: https://astro.tsu.ru/KGaG/text/5_6.html (дата обращения: 10.9.2024).
3. Методы закраски [Электронный ресурс]. — URL: 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: 11.9.2024).
4. *Задорожный А.* Модели освещения и алгоритмы затенения в компьютерной графике. — Новосибирский государственный технический университет, 2020.
5. DMorph [Электронный ресурс]. — URL: <https://www.graphicon.ru/oldgr/courses/cg/assigns/2004/hw4/g5078.pdf> (дата обращения: 11.9.2024).
6. *Кент Д., Карлсон У., Парент Р.* Shape Transformation for Polyhedral Objects. — 1992.

ПРИЛОЖЕНИЕ А