

Оглавление

Введение	3
1 Аналитический раздел	5
1.1 Описание объектов сцены	5
1.2 Обоснование выбора формы задания трехмерных моделей .	5
1.3 Задание объемных моделей	6
1.4 Выбор алгоритма удаления невидимых ребер и поверхностей	7
1.4.1 Алгоритм, использующий Z-буфер	7
1.4.2 Алгоритм обратной трассировки лучей	8
1.4.3 Алгоритм Робертса	9
1.4.4 Алгоритм художника	11
1.4.5 Алгоритм Варнока	11
1.4.6 Вывод	13
1.5 Анализ и выбор модели освещения	13
1.5.1 Модель Ламберта	13
1.5.2 Модель Фонга	14
1.5.3 Вывод	14
1.6 Вывод	15
2 Конструкторский раздел	16
3 Технологический раздел	17
4 Исследовательская часть	18
Заключение	19
Список использованных источников	20

Введение

На сегодняшний день компьютерная графика является одним из самых быстрорастущих сегментов в области информационных технологий.

Область ее применения очень широка и не ограничивается художественными эффектами: в отраслях техники, науки, медицины и архитектуры трехмерные графические объекты используются для наглядного отображения разнообразной информации и презентации различных проектов.

Алгоритмы создания реалистичных изображений требуют особого внимания, поскольку они связаны с внушительным объемом вычислений, требуют больших компьютерных ресурсов и затратны по времени. Для создания качественного изображения объекта следует учесть не только оптические законы, но и расположение источника света, фактуры поверхностей.

Основным направлением в развитии компьютерной графики является ускорение вычислений и создание более качественных изображений.

Цель рабочей пояснительной записки — разработка, реализация, описание программного обеспечения, генерирующее изображение стержня помещенного в цилиндр с жидкостью.

Задачи рабочей пояснительной записки:

- описать структуру трехмерной сцены, включая объекты, из которых она состоит;
- проанализировать существующие алгоритмы построения изображения и обосновать выбор тех из них, которые в наибольшей степени подходят для решения поставленной задачи;
- проанализировать и выбрать варианты оптимизации ранее выбранного алгоритма удаления невидимых линий;
- реализовать выбранные алгоритмы;
- разработать программное обеспечение для отображения сцены и визуализации стержня в цилиндре, наполненного жидкостью;

1 Аналитический раздел

В этом разделе будут представлены описание объектов, а также обоснован выбор алгоритмов, которые будут использован для ее визуализации.

1.1 Описание объектов сцены

Сцена состоит из источника света, цилиндра, жидкости, стержня и плоскости.

Источник света представляет собой материальную точку, пускающую лучи света во все стороны (если источник расположен в бесконечности, то лучи идут параллельно). Источником света в программе является вектор.

Цилиндр — это тонкостенный прозрачный объект, в котором располагается два других объекта - жидкость, стержень.

Жидкость — это объект, который тоже является прозрачным тонкостенным цилиндром.

Стержень — это непрозрачный прямоугольный параллелепипед. Служит для того чтобы отобразить на экране преломление твердого тела в жидкости.

Плоскость — это некая ограничивающая плоскость. Предполагается, что под такой плоскостью не расположено никаких объектов. Располагается на минимальной координате по оси Y .

1.2 Обоснование выбора формы задания трехмерных моделей

Отображением формы и размеров объектов являются модели. Обычно используются три формы задания моделей.

1. Каркасная (проволочная) модель.

Одна из простейших форм задания модели, так как мы храним информацию только о вершинах и ребрах нашего объекта. Недостаток

данной модели состоит в том, что она не всегда точно передает представление о форме объекта.

2. Поверхностная модель.

Поверхностная модель объекта — это оболочка объекта, пустая внутри. Такая информационная модель содержит данные только о внешних геометрических параметрах объекта. Такой тип модели часто используется в компьютерной графике. При этом могут использоваться различные типы поверхностей, ограничивающих объект, такие как полигональные модели, поверхности второго порядка и др.

3. Объемная (твердотельная) модель.

При твердотельном моделировании учитывается еще материал, из которого изготовлен объект. То есть у нас имеется информация о том, с какой стороны поверхности расположен материал. Это делается с помощью указания направления внутренней нормали.

При решении данной задачи будет использоваться объемная модель. Этот выбор обусловлен тем, что каркасные модели могут привести к неправильному восприятию формы объекта, а поверхностные модели не подходят, так как важен материал из которого сделаны объекты сцены.

1.3 Задание объемных моделей

После выбора модели, необходимо выбрать лучший способ представления объемной модели.

Аналитический способ — 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, граней и ребер, которые определяют форму многогранного объекта в трехмерной компьютерной графике.

Стоит отметить, что одним из решающих факторов в выборе способа задания модели в данном проекте является скорость выполнения преобразований над объектами сцены.

При реализации программного продукта представлением является аналитический способ, так как все объекты в сцене являются простыми геометрическими фигурами.

1.4 Выбор алгоритма удаления невидимых ребер и поверхностей

Перед выбором алгоритма удаления невидимых ребер необходимо выделить несколько свойств, которыми должен обладать выбранный алгоритм, чтобы обеспечить оптимальную работу и реалистичное изображение, а именно:

- алгоритм должен использовать как можно меньше памяти;
- алгоритм должен обладать небольшой трудоемкостью;
- алгоритм должен выдавать реалистичность изображения.

1.4.1 Алгоритм, использующий Z-буфер

Суть данного алгоритма – это использование двух буферов: буфера кадра, в котором хранятся атрибуты каждого пикселя, и Z-буфера, в котором хранится 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 $x' = x + dx$: $z' - z = -\frac{ax'+d}{c} + \frac{ax+d}{c} = \frac{a(x-x')}{c}$.

Получим $z' = z - \frac{a}{c}$, так как $x - x' = dx = 1$.

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

Преимущества

- простота реализации;
- оценка трудоемкости линейна.

Недостатки

- сложная реализация прозрачности;
- большой объем требуемой памяти.

Данный алгоритм не подходит для решения поставленной задачи, так как требует большого объема памяти, что не удовлетворяет требованиям.

1.4.2 Алгоритм обратной трассировки лучей

Алгоритмы трассировки лучей на сегодняшний день считаются наиболее мощными при создании реалистичных изображений.

Изображение формируется из-за того, что свет попадает в камеру. Выпустим из источников света множество лучей (первичные лучи). Часть этих лучей “улетит” в свободное пространство, а часть попадет на объекты. На них лучи могут преломляться и отражаться. При этом часть энергии луча поглотится. Преломленные и отраженные лучи образуют новое поколение лучей. Далее эти лучи опять же преломятся, отразятся и образуют новое

поколение лучей. В конечном итоге часть лучей попадет в камеру и сформирует изображение. Это описывает работу прямой трассировки лучей.

Метод обратной трассировки лучей позволяет значительно сократить перебор световых лучей. В этом методе отслеживаются лучи не от источников, а из камеры. Таким образом, трассируется определенное число лучей, равное разрешению картинки [?].

Преимущества

- высокая реалистичность синтезируемого изображения;
- работа с поверхностями в математической форме;
- вычислительная сложность слабо зависит от сложности сцены.

Недостатки

- производительность.

Данный алгоритм не отвечает главному требованию – скорости работы, но при некоторой адаптации скорость работы алгоритма можно повысить.

1.4.3 Алгоритм Робертса

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

Алгоритм выполняется в 3 этапа.

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

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

Матрица тела должна быть сформирована корректно, то есть любая точка, 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.

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.

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 [?].

Преимущества

- работа в объектном пространстве;
- высокая точность вычисления.

Недостатки

- рост сложности алгоритма – квадрат числа объектов;
- тела сцены должны быть выпуклыми (усложнение алгоритма, так как нужна будет проверка на выпуклость);
- сложность реализации.

Данный алгоритм не подходит для решения поставленной задачи из-за высокой сложности реализации как самого алгоритма, так и его модификаций, отсюда низкая производительность.

1.4.4 Алгоритм художника

Данный алгоритм работает аналогично тому, как художник рисует картину – то есть сначала рисуются дальние объекты, а затем более близкие. Наиболее распространенная реализация алгоритма – сортировка по глубине, которая заключается в том, что произвольное множество граней сортируется по ближнему расстоянию от наблюдателя, а затем отсортированные грани выводятся на экран в порядке от самой дальней до самой ближней. Данный метод работает лучше для построения сцен, в которых отсутствуют пересекающиеся грани [?].

Преимущества

- требование меньшей памяти, чем, например, алгоритм Z-буфера.

Недостатки

- недостаточно высокая реалистичность изображения;
- сложность реализации при пересечении граней на сцене.

Данный алгоритм не отвечает главному требованию – реалистичности изображения. Также алгоритм художника отрисовывает все грани (в том числе и невидимые), на что тратится большая часть времени.

1.4.5 Алгоритм Варнока

Алгоритм Варнока [?] является одним из примеров алгоритма, основанного на разбиении картинной плоскости на части, для каждой из которых исходная задача может быть решена достаточно просто.

Поскольку алгоритм Варнока нацелен на обработку картинки, он работает в пространстве изображения. В пространстве изображения рассматривается окно и решается вопрос о том, пусто ли оно, или его содержимое достаточно просто для визуализации. Если это не так, то окно разбивается на фрагменты до тех пор, пока содержимое фрагмента не станет достаточно простым для визуализации или его размер не достигнет требуемого предела разрешения.

Сравнивая область с проекциями всех граней, можно выделить случаи, когда изображение, получающееся в рассматриваемой области, определяется сразу:

- проекция ни одной грани не попадает в область;
- проекция только одной грани содержится в области или пересекает область, то в этом случае проекции грани разбивают всю область на две части, одна из которых соответствует этой проекции;
- существует грань, проекция которой полностью накрывает данную область, и эта грань расположена к картинной плоскости ближе, чем все остальные грани, проекции которых пересекают данную область, то в данном случае область соответствует этой грани.

Если ни один из рассмотренных трех случаев не имеет места, то снова разбиваем область на четыре равные части и проверяем выполнение этих условий для каждой из частей. Те части, для которых таким образом не удалось установить видимость, разбиваем снова и т. д.

Преимущества

- меньшие затраты по времени в случае области, содержащий мало информации.

Недостатки

- алгоритм работает только в пространстве изображений;
- большие затраты по времени в случае области с высоким информационным содержанием.

Данный алгоритм не отвечает требованию работы как в объектном пространстве, так и в пространстве изображений, а также возможны большие затраты по времени работы.

1.4.6 Вывод

Для удаления невидимых линий выбран алгоритм обратной трассировки лучей. Данный алгоритм позволит добиться максимальной реалистичности и даст возможность смоделировать распространение света в пространстве, учитывая законы геометрической оптики. Данный алгоритм можно модернизировать, добавив в него обработку новых световых явлений. Также этот алгоритм позволяет строить качественные тени с учетом большого числа источников. Стоит отметить тот факт, что алгоритм трассировки лучей не требователен к памяти, в отличие, например, от алгоритма Z-буфера.

1.5 Анализ и выбор модели освещения

Физические модели материалов стараются аппроксимировать свойства некоторого реального материала. Такие модели учитывают особенности поверхности материала или же поведение частиц материала.

Эмпирические модели материалов устроены иначе, чем физически обоснованные. Данные модели подразумевают некий набор параметров, которые не имеют физической интерпретации, но которые позволяют с помощью подбора получить нужный вид модели.

В данной работе следует делать выбор из эмпирических моделей, а конкретно из модели Ламберта и модели Фонга.

1.5.1 Модель Ламберта

Модель Ламберта [?] моделирует идеальное диффузное освещение, то есть свет при попадании на поверхность рассеивается равномерно во все

стороны. При такой модели освещения учитывается только ориентация поверхности (N) и направление источника света (L). Иллюстрация данной модели представлена на рисунке 1.1.

Эта модель является одной из самых простых моделей освещения и очень часто используется в комбинации с другими моделями. Она может быть очень удобна для анализа свойств других моделей, за счет того, что ее легко выделить из любой модели и анализировать оставшиеся составляющие.

1.5.2 Модель Фонга

Это классическая модель освещения. Модель представляет собой комбинацию диффузной и зеркальной составляющих. Работает модель таким образом, что кроме равномерного освещения на материале могут появляться блики. Местонахождение блика на объекте определяется из закона равенства углов падения и отражения. Чем ближе наблюдатель к углам отражения, тем выше яркость соответствующей точки [?].

Падающий и отраженный лучи лежат в одной плоскости с нормалью к отражающей поверхности в точке падения (рисунок 1.2). Нормаль делит угол между лучами на две равные части. L – направление источника света, R – направление отраженного луча, V – направление на наблюдателя.

1.5.3 Вывод

В качестве модели освещения в данной работе была выбрана модель Ламберта из-за своей простоты по сравнению с моделью Фонга. Для расчета данных модели Ламберта необходимо выполнять меньше вычислений, а значит ее реализация потребует меньшего количества времени.

1.6 Вывод

В данном разделе был проведен анализ алгоритмов удаления невидимых линий и модели освещения, которые возможно использовать для решения поставленных задач. В качестве ключевого алгоритма, который также можно оптимизировать, выбран алгоритм обратной трассировки лучей, который будет реализован в рамках данного курсового проекта.

2 Конструкторский раздел

3 Технологический раздел

В данном разделе будут приведены требования к программному обеспечению, средства реализации и листинга кода.

4 Исследовательская часть

Заключение

Список использованных источников