

1 Слайд

Добрый день, уважаемые члены комиссии! Вашему вниманию предлагается курсовая работа на тему: «Моделирование памятника архитектуры “Спасская башня”».

2 Слайд

Целью курсового проекта является разработка программы, моделирующей памятник архитектуры «Спасская башня» и создания собственных сооружений, состоящих из сфер, цилиндров, конусов, параллелепипедов, треугольных и четырехугольных пирамид.

3 Слайд

В рамках реализации проекта должны быть решены следующие задачи:

- Разработка программного обеспечения, которое позволит отобразить трехмерную сцену и визуализировать памятник архитектуры.
- Проектирование архитектуры программного обеспечения.
- Изучение и анализ алгоритмов компьютерной графики, использующихся для создания реалистичной модели взаимно перекрывающихся объектов, и выбор наиболее подходящего для решения поставленной задачи.
- Проведение исследования зависимости времени рендеринга программы от количества объектов на сцене.

4 Слайд

Сцена состоит из следующих объектов.

- Сооружения, состоящие из сфер, цилиндров, конусов, параллелепипедов, треугольных и четырехугольных пирамид.
- Точечных источников света – представляют собой фиксированную точку в пространстве, называемой его позицией, из которой свет

испускается равномерно во всех направлениях. Точечный источник полностью характеризуется его позицией и яркостью.

- Направленный источник света – представляет собой вектор направления света, предполагается, что источник расположен в бесконечности. Направленный источник света является аппроксимацией солнца.
- Плоскости основания.

5 Слайд

Критериями выбора алгоритма служат возможность работы с телами вращения и демонстрацией зеркального отражения.

Алгоритм Робертса и Варнока не могут быть применимы для решения поставленной задачи, т.к. работают с выпуклыми телами, т.е. с многогранниками, и не моделируют оптические свойства объекта.

Благодаря модификации алгоритма z-буфера, можно учитывать тени и прозрачность. Однако алгоритм не может визуализировать эффект зеркального отражения, которое попадает на другие поверхности.

Поэтому алгоритм трассировки лучей выбран для удаления невидимых линий и поверхностей. Он позволяет получить изображение высокого качества, учитывая эффект отражения, а также предоставляет возможность работы с телами вращения.

6 Слайд

Работа алгоритма трассировки лучей:

Из камеры испускаются лучи, проходящие через каждый пиксель вглубь сцены, затем идет поиск пересечений первичного луча с объектами сцены, как показано на слайде, в случае обнаружения пересечения, рассчитывается интенсивность пикселя, в зависимости от положения

источника света, при отсутствии пересечения, пиксель закрашивается цветом фона.

Серьёзным недостатком алгоритма трассирования является производительность. Для получения изображения необходимо создавать большое количество лучей, проходящих через сцену и отражаемых от объекта. Это приводит к существенному снижению скорости работы программы. Однако для ускорения работы программы лучи можно трассировать одновременно, поскольку каждый луч, исходящий из камеры, независим от всех остальных.

7 Слайд

Пересечение луча с объектами сцены

Наилучшим способом представления лучей является использование параметрического уравнения. Пусть O – начало луча, \vec{D} – вектор, показывающий направление луча. Любую точку P луча можно представить как $P = O + t\vec{D}$, где t — произвольное действительное число.

Необходимо рассмотреть объекты сцены, с которыми лучи сталкиваются. В сцене присутствуют сферы, цилиндры, конусы, параллелепипеды, треугольные и четырехугольные пирамиды, для представления которых используется такой геометрический примитив как треугольник, и плоскости. Для нахождения точки пересечения луча с произвольной поверхностью необходимо знать аналитические уравнения, определяющие оба эти объекта в трехмерном пространстве. Точка пересечения удовлетворяет всем уравнениям, так как принадлежит и лучу, и поверхности. Поэтому, сводя уравнения в систему и находя ее решения, можно получить координаты этой точки.

Для нахождения пересечения с объектами сцены используется скалярное произведение векторов, для которого принято следующее обозначение:

$$\text{dot}(\vec{a}, \vec{b}) - \text{скалярное произведение векторов } \vec{a} \text{ и } \vec{b}.$$

Более подробно останавлиюсь на поиске пересечения луча со сферой и плоскостью.

Пусть C – центр сферы, r – ее радиус, P – точка пересечения луча со сферой

Тогда есть два уравнения, одно из которых описывает точки сферы, а другое – точки луча:

$$\text{dot}(P - C, P - C) = r^2 \quad (1.1)$$

$$P = O + t\vec{D} \quad (1.2)$$

Точка P , в которой луч падает на сферу, является одновременно и точкой луча, и точкой на поверхности сферы, поэтому она должна удовлетворять обоим уравнениям одновременно. Получим:

$$t^2 \text{dot}(\vec{D}, \vec{D}) + 2 * t * \text{dot}(\vec{CO}, \vec{D}) + \text{dot}(\vec{CO}, \vec{CO}) - r^2 = 0 \quad (1.4)$$

Следовательно, для нахождения точки пересечения луча со сферой, необходимо решить квадратное уравнение, в котором отсутствие корней означает отсутствие пересечения, два корня – луч проходит через сферу, один корень – луч касается сферы. При получении двух корней выбирается меньший из них, он и будет расстоянием от начала луча до первого пересечения.

Пусть \vec{V} – вектор нормали, P – точка пересечения луча с плоскостью, C – начальная точка (см. рис 2.4).

Уравнение плоскости:

$$Ax + By + Cz = 0 \quad (2.1)$$

Данное уравнение можно записать следующим образом:

$$\text{dot}(\vec{V}, P - C) = 0 \quad (2.2)$$

Тогда пересечение уравнения, описывающего точки плоскости с уравнением, описывающим точки луча, находится следующим образом:

$$\text{dot}(\vec{V}, O + t\vec{D}) = 0 \quad (2.3)$$

$$\text{dot}(\vec{V}, O) + t * \text{dot}(\vec{V}, \vec{D}) - \text{dot}(\vec{V}, \vec{C}) = 0 \quad (2.4)$$

Выразим отсюда t:

$$t = (\text{dot}(\vec{V}, \vec{C}) - \text{dot}(\vec{V}, O)) / \text{dot}(\vec{V}, \vec{D}) \quad (2.5)$$

8 Слайд

На данном слайде представлены вспомогательные рисунки, демонстрирующие поиск пересечения луча с такими объектами как конус, цилиндр, параллелепипеда и треугольника.

9 Слайд

Для расчета интенсивности в точке было принято решение использовать модель освещения Фонга, поскольку она позволяет учитывать матовые и блестящие поверхности. Основная идея этой модели заключается

в предположении, что освещенность каждой точки тела разлагается на 3 компоненты:

1. фоновое освещение (ambient);
2. рассеянный свет (diffuse);
3. зеркальная составляющая (specular).

Свойства источника определяют мощность излучения для каждой из этих компонент, а свойства материала поверхности определяют ее способность воспринимать каждый вид освещения.

10 Слайд

На слайде представлена схема алгоритма трассировки лучей.

11 Слайд

На слайде представлена схема алгоритма расчета освещенности в соответствии с моделью Фонга.

12 Слайд

На слайде приведена диаграмма классов.

Разработанная программа состоит из следующих классов:

- Классы объектов
 - Primitive – базовый класс примитивов;
 - Sphere – класс сферы;
 - Cylinder – класс цилиндра;
 - Cone – класс конуса.
 - Parallelepiped – класс параллелепипеда.
 - TrianglePyramid – класс треугольной пирамиды.
 - QuadPyramid – класс четырехугольной пирамиды.
- Вспомогательные классы сцены

- Camera – класс камеры с возможностью перемещения по сцене;
- Light – класс источника освещения с возможностью перемещения и изменения интенсивности.
- Классы интерфейса
 - Facade – класс, который предоставляет интерфейс работы системы.
 - Controller – класс для взаимодействия управляющих классов с классами интерфейса.
- Класс визуализации сцены: Raytracer.

13 Слайд

На слайде показан интерфейс программы.

14 Слайд

При исследовании временных характеристик разработанной программы использовался компьютер на базе 4-х ядерного процессора Intel Core i3-8100.

На слайде изображены графики зависимости времени рендеринга программы от количества объектов на сцене и числа потоков для параллельной версии алгоритма трассировки лучей.

Как видно из графика, время рендеринга сцены линейно зависит от количества объектов. Максимальная скорость работы достигается при количестве потоков равным 4. Дальнейшее увеличение их числа не дает прироста в скорости. Это связано с тем, что число потоков, которые работают параллельно, равно числу логических процессоров.

Заключение

Цель работы достигнута. Было спроектировано и реализовано программное обеспечение, моделирующее памятник архитектуры «Спаская

башня» и создание собственных сооружений, состоящих из сфер, цилиндров, конусов, параллелепипедов, треугольных и четырехугольных пирамид.

В ходе работы были проанализированы существующие алгоритмы удаления невидимых линий и поверхностей, модели освещения, указаны их преимущества и недостатки.

Разработаны собственные и адаптированы существующие структуры данных и алгоритмы, необходимые для решения поставленной задачи.

15 Слайд

Спасибо за внимание!