



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 «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К КУРСОВОЙ РАБОТЕ

НА ТЕМУ:

*«Разработка программы построения 3Д сцен
помещений различной планировки»*

Студент ИУ7-55Б
(Группа)

(Подпись, дата)

О.Н. Талышева
(И.О.Фамилия)

Руководитель курсовой работы

(Подпись, дата)

Н.Н. Мартынюк
(И.О.Фамилия)

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)

УТВЕРЖДАЮ

Заведующий кафедрой

ИУ7

(индекс)

И.В.Рудаков

(И.О. Фамилия)

(подпись)

(дата)

ЗАДАНИЕ

на выполнение курсовой работы

по дисциплине Компьютерная графика

Студент группы ИУ7-45Б Талышева Олеся Николаевна

(Фамилия, имя, отчество)

Тема курсовой работы

Разработка программы построения 3Д сцен помещений различной планировки.

Направленность КР (учебная, исследовательская, практическая, производственная, др.)
практическая

Источник тематики (кафедра,
предприятие, НИР)

предприятие

Задание Разработать программу построения 3Д сцен помещений различной

планировки. Создать объекты (стена, окно, дверь), которые можно добавлять
на сцену, удалять со сцены, перемещать, поворачивать и масштабировать на сцене.

Предоставить возможность рассмотреть сцену из разных точек с помощью камеры.

Обеспечить сохранение модели в файл и загрузку существующей для последующего
редактирования.

Оформление курсовой работы:

2.1. Расчетно-пояснительная записка на 25–30 листах формата А4.

Расчетно-пояснительная записка должна содержать постановку введение,
аналитическую часть, конструкторскую часть, технологическую часть,
экспериментально-исследовательский раздел, заключение, список литературы,
приложения.

2.2. Перечень графического материала (плакаты, схемы, чертежи и т. п.) На защиту
проекта должна быть представлена презентация, состоящая из 15–20 слайдов.

Дата выдачи задания « ____ » _____ 20__ г.

Руководитель курсовой работы

(подпись, дата)

Мартынюк Н.Н.

(И.О. Фамилия)

Студент

(подпись, дата)

Талышева О.Н.

(И.О. Фамилия)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится
на кафедре.

Содержание

Введение	3
1 Аналитическая часть	6
1.1 Удаление невидимых линий	6
1.1.1 Введение	6
1.1.2 Алгоритмы предварительного удаления невидимых линий и по- верхностей	10
1.2 Алгоритмы отрисовки теней	10
1.3 Алгоритмы освещения	10
Список использованных источников	11

ВВЕДЕНИЕ

В настоящее время введение технологий 3D моделирования активно развивается в самых разных сферах: архитектуре, дизайне, инженерии, безопасности и оперативном реагировании. Программное обеспечение для построения 3D сцен позволяет визуализировать объекты и пространства, что существенно облегчает их анализ, планирование и эксплуатацию. В особенности, технологии моделирования помещений с гибкой настройкой планировок и объектов находят важное применение для задач охраны и контроля доступа на стратегически важные объекты.

Одной из ключевых сфер, в которой данная тема является особенно актуальной, выступает обеспечение безопасности и оперативного реагирования силовых структур на проникновение на охраняемые объекты. В условиях возросших угроз безопасности и повышенных требований к защите территорий и объектов, необходимость быстрого создания точных моделей помещений с целью их анализа является приоритетной задачей. Использование 3D моделей позволяет оперативно прорабатывать сценарии вторжений, рассчитывать оптимальные пути пресечения проникновений и создавать план эвакуации или нейтрализации угрозы.

Для силовых структур важно иметь инструменты, которые позволяют в режиме реального времени моделировать различные планировки и адаптировать стратегические планы по обеспечению безопасности. Программа, позволяющая моделировать помещения с возможностью добавления объектов (стен, окон, дверей), их перемещения, масштабирования и управления камерой, может стать незаменимым инструментом для подготовки к операциям, тренировок и планирования реагирования в критических ситуациях.

Таким образом, разработка программы построения 3D сцен помещений имеет важное практическое значение, поскольку позволяет оперативно создавать модели для анализа и разработки решений по обеспечению безопасности.

Целью данной курсовой работы является разработка программного обеспечения для создания и редактирования 3D сцен помещений с возможностью интерактивного добавления объектов (стен, окон, дверей), их перемещения, масштабирования, поворота, а также обеспечения сохранения и загрузки моделей.

В рамках работы были поставлены следующие задачи:

1. Анализ требований к программе и исследование существующих решений:
 - (a) Изучить программные продукты для 3D моделирования помещений, чтобы понять их функциональные особенности и интерфейсные решения.
 - (b) Оценить, какие элементы и функции наиболее важны для конечного пользователя.
2. Изучить алгоритмы реализации технических решений и выбрать наиболее подходящие для работы с 3D сценами.

3. Разработка архитектуры программы:

- (a) Спроектировать структуру программы, определив основные компоненты: объекты сцены (стены, окна, двери), камера, управление сценой.
- (b) Разработать систему хранения данных о 3D моделях, обеспечивающую сохранение и загрузку сцены.

4. Реализация объектов для создания сцены:

- (a) Создать базовые 3D объекты (стена, окно, дверь) с параметрами (размеры, позиции, углы поворота, текстуры).
- (b) Обеспечить возможность добавления, перемещения, удаления, масштабирования и поворота объектов на сцене.

5. Разработка системы управления камерой:

- (a) Предоставить пользователю возможность управления камерой для осмотра сцены под разными углами.
- (b) Реализовать функции перемещения камеры, вращения вокруг объектов, изменения масштаба.

6. Реализация пользовательского интерфейса (UI):

- (a) Разработать удобный интерфейс для добавления и редактирования объектов сцены.
- (b) Включить панели инструментов для выбора объектов, изменения их параметров, управления сценой и камерой.

7. Сохранение и загрузка 3D сцен:

- (a) Реализовать функционал сохранения текущей сцены в файл в специальном формате, чтобы пользователи могли продолжить работу позже.
- (b) Предусмотреть возможность загрузки ранее сохранённых сцен для редактирования.

8. Тестирование программы:

- (a) Провести тестирование работы программы для различных вариантов планировки помещений.
- (b) Проверить корректность работы с сохранением и загрузкой сцен, взаимодействие с объектами и камерой.

9. Оценка производительности программы:

- (a) Провести анализ производительности программы при увеличении количества объектов на сцене.
- (b) Оптимизировать работу с 3D объектами для плавного взаимодействия даже при больших сценах.

10. Документирование и подготовка отчётной документации:

- (a) Описать процесс разработки, результаты тестирования, а также подготовить руководство пользователя для программы.

1. Аналитическая часть

1.1. Удаление невидимых линий

1.1.1 Введение

Одной из основных задач компьютерной графики является визуализация трёхмерных сцен. Подобные задачи возникают в системах автоматизированного проектирования, пакетах моделирования физических процессов, средствах компьютерной анимации и виртуальной реальности.

При отображении трёхмерной сцены на экране некоторые из объектов сцены могут заслонить другие объекты. Заслонённые части объектов невидимы и не должны рисоваться, или должны рисоваться иначе, чем видимые части, например, пунктиром. Если этого не делать, то изображение будет выглядеть неправильно.

Такие задачи решают с помощью алгоритмов удаления невидимых линий и поверхностей. Если сцена отображается в каркасном виде, линиями, то нужно удалять невидимые линии. Каркасное изображение обычно строится из отрезков – рёбер, и алгоритм должен выделить части отрезков, заслонённых объектами сцены. Если объекты сцены отображаются в виде закрашенных поверхностей, то нужно удалять невидимые части этих поверхностей. Обычно в качестве поверхностей используются выпуклые многоугольники, чаще всего – треугольники. В курсовой работе необходимо реализовать один из алгоритмов удаления невидимых линий и поверхностей и произвести анализ его производительности. В соответствующих разделах приведены описания основных алгоритмов: Робертса, Варнака, Z-буфера, художника, трассировки лучей, построчного сканирования.

Все алгоритмы удаления невидимых линий и поверхностей можно разделить на две группы:

1. В одних, сначала определяется видимость для участков линий или поверхностей, а затем рисуются видимые части. К таким алгоритмам относится, например, алгоритм Робертса. В подобных алгоритмах необходимо сравнить каждый объект сцены (линию или поверхность) с остальными объектами, способными заслонить его.
2. В других, для каждого пиксела изображения определяется, какой из объектов сцены в нём виден. К таким алгоритмам относятся, например, алгоритмы трассировки лучей или Z-буфера. В этом случае нужно для каждого пиксела выбрать ближайший к наблюдателю объект сцены.

В обоих случаях требуется много вычислений, из-за того, что необходимо перебирать все объекты сцены. Для сокращения объёма вычислений используется свойство когерентности (англ. coherence – связность) расположенных рядом объектов. Можно выделить три вида когерентности:

1. Когерентность в картинной плоскости (плоскости экрана) – расположенные рядом пиксели, скорее всего, имеют одинаковые свойства, например, принадлежат одному и тому же объекту сцены, или видимы или, наоборот, заслонены одним и тем же объектом сцены.
2. Когерентность в объектном пространстве (пространстве сцены) – расположенные рядом объекты сцены, скорее всего, имеют одинаковые свойства, например, видимы или, наоборот, заслонены одним и тем же объектом сцены.
3. Когерентность во времени – при перемещении наблюдателя, на соседних кадрах будут видимы примерно одни и те же объекты сцены

В некоторых алгоритмах когерентность используется явно. Например, в алгоритме Варнака картинная плоскость рекурсивно делится на области расположенных рядом пикселей, и, если вся область видима или, наоборот, заслонена, то дальнейшее её разделение не требуется. Другие алгоритмы можно модифицировать, используя свойство когерентности, существенно повышая их производительность. [3]

Существуют два различных способа изображения трехмерных тел — каркасное (wireframe — рисуются только ребра) и сплошное (рисуются закрашенные грани). Тем самым возникают два типа задач — удаление невидимых линий (ребер для каркасных изображений) и удаление невидимых поверхностей (граней для сплошных изображений).

Анализ видимости объектов можно производить как в исходном трехмерном пространстве, так и на картинной плоскости. Это приводит к разделению методов на два класса:

1. методы, работающие непосредственно в пространстве самих объектов;
2. методы, работающие в пространстве картинной плоскости, т. е. работающие с проекциями объектов.

Получаемый результат представляет собой либо набор видимых областей или отрезков, заданных с машинной точностью (имеет непрерывный вид), либо информацию о ближайшем объекте для каждого пикселя экрана (имеет дискретный вид).

Методы первого класса дают точное решение задачи удаления невидимых линий и поверхностей, никак не привязанное к растровым свойствам картинной плоскости.

Они могут работать как с самими объектами, выделяя те их части, которые видны, так и с их проекциями на картинную плоскость, выделяя на ней области, соответствующие проекциям видимых частей объектов, и, как правило, практически не привязаны к растровой решетке и свободны от погрешностей дискретизации. Так как эти методы работают с непрерывными исходными данными и получающиеся результаты не зависят от растровых свойств, то их иногда называют непрерывными методами (continuous methods).

Простейший вариант непрерывного подхода заключается в сравнении каждого объекта со всеми остальными, что дает временные затраты, пропорциональные n^2 , где n — количество объектов в сцене.

Однако следует иметь в виду, что непрерывные методы, как правило, достаточно сложны.

Методы второго класса (point-sampling methods) дают приближенное решение задачи видимости, определяя видимость только в некотором наборе точек картинной плоскости — в точках растровой решетки. Они очень сильно привязаны к растровым свойствам картинной плоскости и фактически заключаются в определении для каждого пиксела той грани, которая является ближайшей к нему вдоль направления проектирования. Изменение разрешения приводит к необходимости полного перерасчета всего изображения.

Простейший вариант дискретного метода имеет временные затраты порядка Cn , где C — общее количество пикселей экрана, а n — количество объектов.

Всем методам второго класса традиционно свойственны ошибки дискретизации (aliasing artifacts). Однако, как правило, дискретные методы отличаются известной простотой.

Кроме этого существует довольно большое количество смешанных методов, использующих работу как в объектном пространстве, так и в картинной плоскости, методы, выполняющие часть работы с непрерывными данными, а часть — с дискретными.

Большинство алгоритмов удаления невидимых граней и поверхностей тесно связано с различными методами сортировки. Некоторые алгоритмы проводят сортировку явно, в некоторых она присутствует в скрытом виде. Приближенные методы отличаются друг от друга фактически только порядком и способом проведения сортировки.

Очень распространенной структурой данных в задачах удаления невидимых линий и поверхностей являются различные типы деревьев — двоичные (BSP-trees), четверичные (Quadtree), восьмеричные (Octree) и др.

Методы, практически применяющиеся в настоящее время, в большинстве являются комбинациями ряда простейших алгоритмов, неся в себе целый ряд разного рода оптимизаций. [4]

Основные методы оптимизации:

1. Метод отсечения нелицевых граней (culling)

Позволяет примерно вдвое сократить количество рассматриваемых граней.

Для определения того, является заданная грань лицевой или нет достаточно взять произвольную точку этой грани и проверить выполнение условия $(\mathbf{N}, \mathbf{L}) \leq 0$, где \mathbf{N} — нормаль к грани, \mathbf{L} — направление проецирования.

2. Метод оболочек (Bounding Volumes, Bounding-Volume-Hierarchy (BVH))

Если оболочки не пересекаются, то и содержащиеся в них объекты тоже пересекаться не будут. Однако, если оболочки пересекаются, то сами объекты пересе-

каться не обязаны. В качестве ограничивающих тел чаще всего используются прямоугольные ограничивающие параллелепипеды. Оболочка описывается числами ($X_{min}, Y_{min}, Z_{min}$) и ($X_{max}, Y_{max}, Z_{max}$) из координат точек исходного объекта (4 для 2D).

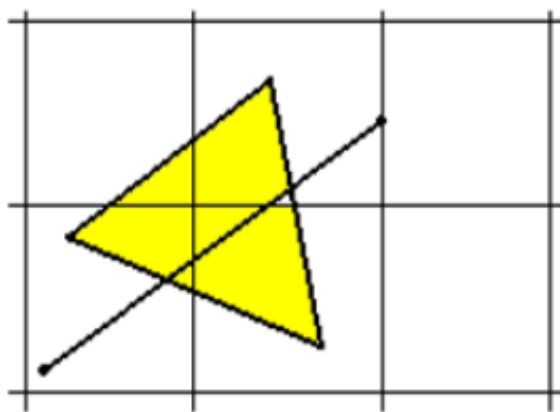
3. Разбиение пространства (картинной плоскости)

Еще один метод, облегчающий сравнение объектов, позволяющий использовать когерентность как в пространстве, так и в картинной плоскости.

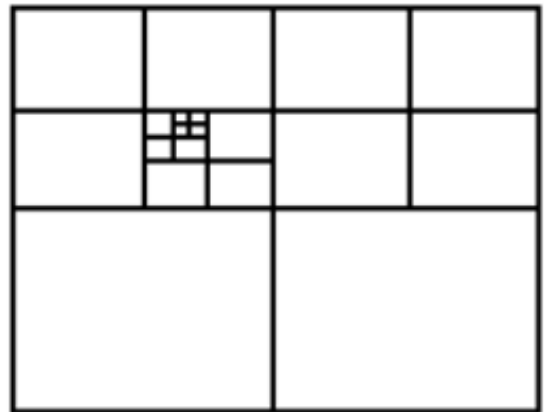
С этой целью разбиения стоятся уже на этапе пре-процессирования, и для каждой клетки разбиения составляется список всех объектов (граней), которые ее пересекают.

Простейшим вариантом разбиения является равномерное разбиение пространства на набор равных прямоугольных клеток (см. рисунок 1). Составляется список объектов, пересекающих клетку разбиения. Для отыскания всех объектов, которые закрывают рассматриваемый объект при проецировании, проверяются только объекты, попадающие в те же клетки разбиения картинной плоскости.

Для сцен с неравномерным распределением объектов имеет смысл использовать неравномерное (адаптивное) разбиение пространства или плоскости (см. рисунок 1).



равномерное разбиение



неравномерное разбиение

Рисунок 1 – Равномерное и неравномерное разбиения пространства объектов

4. Иерархические древовидные структуры

При работе с большими объемами используются различные древовидные (иерархические) структуры. Стандартными формами таких структур являются восьмеричные (Octrees, для 3D), тетрарные (Quadrees, для 2D), бинарные или BSP-деревья (Binary Space Partitioning Trees) и деревья ограничивающих тел. Иерархии позволяют упорядочивать грани объектов, производить быстрое и эффективное отсечение граней, не удовлетворяющих каким-либо из условий.

(a) Иерархия ограничивающих тел

Получается дерево, корнем которого является тело, описанное вокруг всей сцены, а потомками – тела, описанные вокруг первичных, вторичных и др. групп.

Отсечение основного количества объектов происходит уже на ранней стадии достаточно быстро, ценой всего лишь нескольких проверок.

(b) Иерархии разбиения

Каждая клетка исходного разбиения разбивается на части (которые, в свою очередь, так же могут быть разбиты и т.д. При этом каждая клетка разбиения соответствует узлу дерева).

Иерархии (как и разбиения) позволяют достаточно легко и просто производить частичное упорядочение граней. В результате получается список граней, практически полностью упорядоченный, что дает возможность применять специальные методы сортировки.

Специальные методы оптимизации:

1. Потенциально видимые множества граней (препроцессинг построения PVS)
2. Метод порталов (PVS "на ходу")
3. Метод иерархических подсцен (модификация метода порталов) [5]

1.1.2 Алгоритмы предварительного удаления невидимых линий и поверхностей

1.2. Алгоритмы отрисовки теней

1.3. Алгоритмы освещения

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] Д. Роджерс, Алгоритмические основы машинной графики. Москва: Мир, 1989. — 512 стр.
- [2] Компания ИТЕР, "Модель системы защиты". Дата обращения: 21 сентября 2024 г. [Электронный ресурс]. Доступно по адресу: https://iter.ru/model_sfz.html#section_9
- [3] Польский, С. В. П53 Компьютерная графика : учебн.-методич. пособие. – М. : ГОУ ВПО МГУЛ, 2008. – 38 с.
- [4] Шикин Е.В., Боресков А.В. Компьютерная графика. Полигональные модели. –М.: ДИАЛОГ-МИФИ, 2001.-464с.
- [5] В. Е. Турлапов, Удаление невидимых поверхностей. Оптимизация. Тени, Нижегородский государственный университет имени Н. И. Лобачевского, Национальный исследовательский университет, факультет вычислительной математики и кибернетики, кафедра программного обеспечения, курс: КГ 255. Компьютерная графика, 2013. Доступно по ссылке: http://www.graph.unn.ru/rus/materials/CG/CG13_HSROptimization.pdf.