МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра программной инженерии

УТВЕРЖДАЮ
Проректор по учебной работе
_____О. Г. Локтионова
«10» 02 2022 г.

АЛГОРИТМЫ ОТСЕЧЕНИЯ ОТРЕЗКОВ

Методические указания по выполнению лабораторной работы
по дисциплине «Компьютерная графика»
для студентов всех форм обучения направления подготовки
09.03.04 «Программная инженерия»

Курск 2022

УДК 004.92
Составитель Е.А. Петрик

Рецензент
Кандидат технических наук, доцент Т.И.Лапина

Алгоритмы отсечения отрезков: методические указания по выполнению лабораторной работы / Юго-Зап. гос. ун-т; сост.: Е. А. Петрик. Курск, 2022. 10 с.: ил.2. Библиогр.: с.10.

Содержат краткие теоретические сведения об алгоритмах отсечения отрезков, а также приведены примеры и задания для лабораторной работы.

Методические указания соответствуют требованиям программы по направлению подготовки бакалавров: 09.03.04 «Программная инженерия»

Предназначены для студентов всех форм обучения направления подготовки бакалавров 09.03.04 «Программная инженерия»

Текст печатается в авторской редакции

Подписано в печать Формат 60х84 1/16.
Усл. печ. л. Уч. – изд. л. .Тираж экз. Заказ . Бесплатно.
Юго - Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

Цель работы

Изучение алгоритмов отсечения, создание программы для визуализации работы алгоритмов.

Основные понятия

Алгоритмы отсечения

Отсечение – это процесс выделения некоторой части данных, описывающих изображение, выполняется для визуализации конкретной сцены.

Отсечение выполняется достаточно просто в случае окна прямоугольного вида со сторонами, параллельными координатным осям. Как и ранее, будем считать прямоугольную область заданной, если определены координаты ее диагонали: (x1,y1)-координаты верхнего левого угла, (x2,y2)-координаты нижнего правого угла. Зная координаты элементов исходного изображения (пусть они хранятся в массивах XS и YS) и координаты окна, надо определить координаты сохраняемых элементов (для хранения которых можно отвести массивы XN и YN). Отдельная точка сохраняется в процессе отсечения, если она находится внутри окна, т.е. если одновременно выполняются два условия x1<=x<=x2 и y1<=y<=y2, где (x,y) - координаты точки. Отрезок прямой сохраняется полностью, если оба его конца лежат внутри окна. Отрезок полностью удаляется, если все его точки лежат вне окна.

Если часть отрезка лежит вне окна, то она отсекается, а сохраняется только та часть, которая находится внутри окна. Для выполнения отсечения:

Во-первых, можно использовать процедуры определения поля вывода и очистки его. В качестве поля вывода в этом случае берется вся область экрана, за исключением окна, при этом одной прямоугольной области поля вывода мы не получим, поэтому описанную процедуру придется повторить несколько раз (от двух, когда окно находится на краю экрана, до четырех, когда окно находится в центре экрана).

Во-вторых, можно запомнить в памяти значения всех пикселей, расположенных в окне, затем, стерев содержимое экрана, вывести запомненную информацию в любом месте экрана, для чего можно

воспользоваться уже готовыми процедурами сохранения в памяти и выдачи изображения.

Алгоритм Сазерленда-Коэна

Рассмотрим плоскую сцену и отсекающее окно регулярной формы (рис. 1)

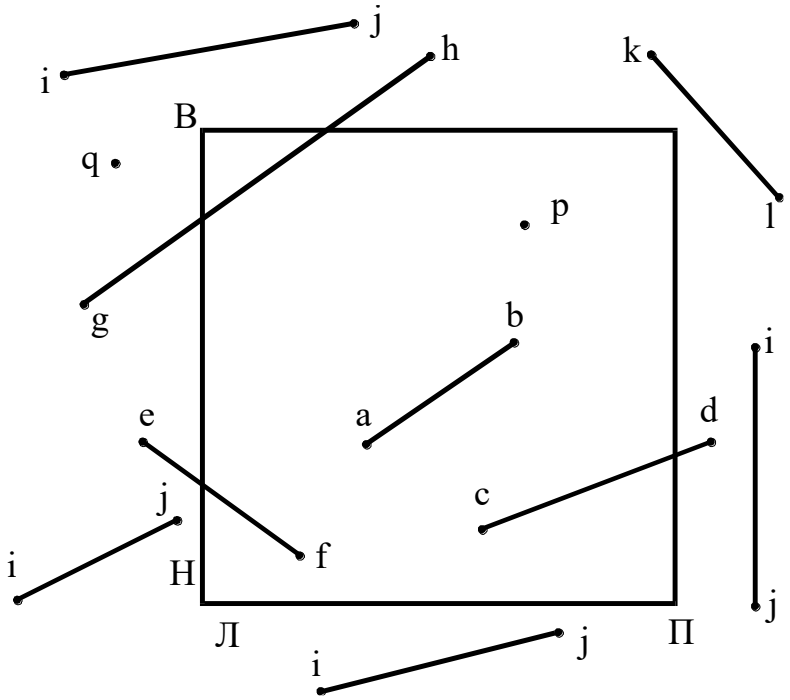

Рисунок 1 – Двумерное отсекающее окно

Окно задается левым (Л), правым (П), верхним (В) и нижним (Н) двумерными ребрами. Регулярным отсекающим окном является прямоугольник, стороны которого параллельны осям координат объектного пространства или осям координат экрана.

Поскольку в обычных сценах необходимо отсекать большое число отрезков или точек, то повышение эффективности (быстродействия) алгоритмов отсечения является актуальной задачей. Важно уметь быстро отбирать отрезки и точки, которые лежат целиком внутри или вне отсекающего окна (на рис..1 - ab, p, ij, q, kl).

Точки, лежащие внутри отсекающего окна, удовлетворяют условию: хЛ<=х<=хП и уН<=у<=уВ. Точки, лежащие на границе окна, считаются находящимися внутри него.

Отрезок является видимым, если обе его концевые точки лежат внутри окна (ab на рис. 1). Однако если оба конца отрезка лежат вне окна, то этот отрезок не обязательно лежит целиком вне окна (gh на рис..1). Если же оба конца отрезка лежат справа, слева, выше или

ниже окна, то этот отрезок целиком лежит вне окна, а значит, невидим (это условие не касается отрезков gh и kl).

Для идентификации положения отрезков используются специальные коды Сазерленда-Коэна для концевых точек отрезка:

Для двумерного случая используется 4-битовый код:

1 в первом бите - если конец отрезка левее окна,

1 во втором бите - если конец отрезка выше окна,

1 в третьем бите - если конец отрезка ниже окна,

1 в четвертом бите - если конец отрезка правее окна.

Таким образом, в целом простой алгоритм двумерного отсечения можно представить следующими основными шагами:

1. Вычисление кодов концевых точек отрезка. Занесение этих кодов в два массива M1 и M2 размерностью 1*4 каждый.

2. Инициализация флажка видимости.

3. Инициализация наклона.

4. Проверка полной видимости отрезка (коды обоих концов равны нулю): суммирование элементов M1 и M2. Если сумма элементов M1 и сумма элементов M2 равны нулю, то вычерчивание отрезка.

5. Проверка случая тривиальной невидимости. Если логическое произведение соответствующих элементов матриц не равно нулю, то флажок видимости устанавливается в "невидимость" и осуществляется переход к следующему отрезку.

7. Процедура отсечения отрезка каждой из сторон окна и получения точек пересечения с помощью проверки их принадлежности внутренней области окна. Эта последняя процедура реализуется по следующей схеме: сначала она применяется к отрезку P1P2 и получается отрезок P1'P2, а затем - к отрезку P1'P2 и получается результирующий отрезок P1'P2'.

Замечания:

Для эффективной реализации данного алгоритма необходимо иметь в виду следующее:

1). Если наклон бесконечен, то отрезок параллелен левой и правой сторонам окна и надо искать его пересечение только с верхней и нижней сторонами.

2). Если наклон равен нулю, то отрезок параллелен верхней и нижней сторонам окна, а искать его пересечения надо только с левой и правой сторонами.

3). Если код одного из концов отрезка равен нулю, то этот конец лежит внутри окна, и поэтому отрезок может пересечь только одну сторону окна.

В алгоритме Сазерленда-Коэна отрезок тоже разбивается сторонами окна. Отличие состоит в том, что здесь не производится проверки попадания точки пересечения внутрь окна, вместо этого каждая из пары получающихся частей отрезка сохраняется или отбрасывается в результате анализа кодов ее концевых точек. Рассмотрим отрезок P1P2, разбитый левой стороной окна на P1P1' и P1'P2. Для алгоритма Сазерленда-Коэна это особый случай, поскольку коды концевых точек каждого из отрезков таковы, что оба они могут быть частично видимы. Ни один из них нельзя отвергнуть. Ключом к алгоритму Сазерленда-Коэна является информация о том, что одна из концевых точек лежит вне окна. Поэтому тот отрезок, который заключен между этой точкой и точкой пересечения, можно отвергнуть как невидимый. Фактически это означает замену исходной концевой точки на точку пересечения.

Алгоритм Сазерленда-Коэна можно сформулировать следующим образом:

Для каждой стороны окна выполнить:

Для каждого отрезка P1P2 определить, не является ли он полностью видимым или может быть тривиально отвергнут как невидимый.

Если P1 вне окна, то продолжить выполнение, иначе поменять местами и P2.

Заменить P1 на точку пересечения P1P2 со стороной окна.

Рассмотрим пример.

Пример. Алгоритм отсечения Сазерленда-Коэна.

Рассмотрим вновь P1 P2.

Коды концевых точек P1(-3/2,1/6) и P2(1/2,3/2) равны (0001) и (1000) соответственно. Отрезок не является ни полностью видимым, ни тривиально невидимым.

P1P2 пересекает левую сторону окна. P1 - вне окна.

Пересечение с левой стороной (x=-1) окна происходит в т.P1'(-1,1/2). Замена P1 на P1' дает новый отрезок - P1(-1,1/2) P2(1/2,3/2). Коды: P1 - (0000), P2 - (1000). Отрезок не является ни полностью видимым, ни тривиально невидимым.

Отрезок не пересекается с правой стороной окна. Перейти к нижней стороне. Коды по-прежнему: (0000), (1000).

Отрезок не пересекается с нижней стороной окна. Перейти к верхней стороне. Коды по-прежнему: (0000), (1000).

Отрезок пересекается с верхней стороной окна. P1 - не снаружи окна. Поменяв P1 и P2 местами, получаем: P1(1/2,3/2) P2(-1,1/2). Точка пересечения P1'(-1/4,1). Заменив P1 на P1', получим P1(-1/4,1) P2(-1,1/2).

Теперь P1 - (0000), P2 - (0000). Отрезок полностью видим.

Начертить отрезок

В простом алгоритме и в алгоритме Сазерленда-Коэна требуется вычислить пересечение отрезка со стороной окна.

Алгоритм разбиения средней точкой

Алгоритм разбиения средней точкой позволяет избежать непосредственного вычисления. Этот алгоритм отсечения реализует двоичный поиск пересечения путем деления отрезка его средней точкой. Идея этого алгоритма была предложена Спруллом и Сазерлендом в 1968г. для аппаратной реализации. Программная реализация этого алгоритма медленнее, чем реализация предыдущего алгоритма, аппаратная - быстрее и эффективнее, поскольку можно использовать параллельную архитектуру. Кроме того, аппаратные сложения и деления на 2 очень быстры (деление на 2 аппаратно эквивалентно сдвигу каждого бита вправо).

В алгоритме используются коды концевых точек отрезка и проверки, выявляющие полную видимость отрезков, например отрезка *a*, и тривиальную невидимость отрезков, например *b*. Отрезки *c, d, e, f, g* разбиваются на две равные части. Затем те же проверки применяются к каждой из половин до тех пор, пока не будет обнаружено пересечение со стороной окна или длина разделяемого отрезка не станет пренебрежимо малой, т.е. пока он не выродится в точку. После вырождения определяется видимость полученной точки. Максимальное число разбиений пропорционально точности задания координат концевых точек отрезка.

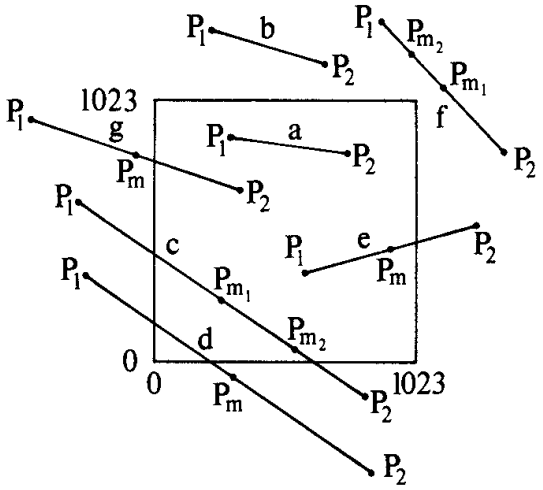

Рисунок 2. –Разбиение средней точкой

Рассмотрим отрезки c и f. Отрезок f не может быть тривиально отвергнут. Разбиение его средней точкой Pm1 позволяет тривиально отвергнуть половину Pm1P2. Однако половина Pm1P1 тоже пересекает диагональ окна, и ее нельзя отвергнуть тривиально. Далее проводится разбиение точкой Pm2, которое позволяет отвергнуть невидимый отрезок Pm2P1. Разбиение оставшегося куска Pm1Pm2, продолжается до тех пор, пока не будет найдено пересечение этого отрезка с прямой, несущей правую сторону окна, с наперед заданной точностью. Затем исследуется обнаруженная точка и она оказывается невидимой. Следовательно, и весь отрезок невидим.

Разбиение отрезка с его средней точкой Pm1 приводит к одинаковым результатам для обеих половин. Отложив отрезок Pm1P1 на потом, разобьем отрезок Pm1P2 точкой Pm2. Теперь отрезок Pm1Pm2 полностью видим, а отрезок Pm2P2 видим частично. Точка Pm2 запоминается как текущая видимая точка, которая наиболее удалена от P1. А разбиение отрезка Pm2P2 продолжается. Каждый раз, когда обнаруживается видимая средняя точка, она объявляется текущей наиболее удаленной от P1, до тех пор пока не будет обнаружено пересечение с нижней стороной окна с заранее заданной точностью. Это пересечение и будет объявлено самой удаленной от P1 видимой точкой. Затем точно так же обрабатывается отрезок Pm1P1. Наиболее удаленной от P2 видимой точкой будет точка его пересечения с левой стороной окна.

Для отрезков, подобных c и d, в алгоритме реализуется два двоичных поиска, которые при

Аппаратной реализации проводятся параллельно.

Трехмерное отсечение

Как и при двумерном отсечении, отрезки, которые полностью видимы или тривиально невидимы, можно идентифицировать с использованием обобщения кодов концевых точек Коэна-Сазерленда. В трехмерном случае используется 6-битовый код:

в первый бит - если конец ребра левее объема,
во второй бит - если конец ребра правее объема,
в третий бит - если конец ребра ниже объема,
в четвертый бит - если конец ребра выше объема,
в пятый бит - если конец ребра ближе объема,
в шестой бит - если конец ребра дальше объема.

Условия полной видимости отрезка и тривиальной невидимости аналогичны двумерному отсечению.

Поиск кодов точки относительно отсекающего прямоугольного параллелепипеда является прямым обобщением соответствующего двумерного алгоритма. Если отсекатель - усеченная пирамида, то один из методов заключается в преобразовании отсекателя в каноническую форму. В этом случае проверка кодов концевых точек заметно упрощается.

Задание

Написать программу (на языке высокого уровня), реализующую алгоритмы отсечения отрезков с их последующей отрисовкой. Количество отрезков и их координаты, а также координаты и размеры окна задаются пользователем.

Содержание отчета

Отчет должен содержать:

1. Титульный лист.
2. Задание.
3. Блок-схемы алгоритмов.
4. Листинг программы.
5. Пример работы программы.
6. Ответы на контрольные вопросы.

Контрольные вопросы

1. Что такое отсечение?
2. Для чего нужны алгоритмы отсечения?
3. Какие ещё алгоритмы отсечения существуют?

Список используемой литературы

1. Аммерал, Л. Принципы программирования в машинной графике / Л. Аммерал; пер. с англ. – Москва : Сол Систем, 1992. – 224 с. – ISBN 5-85316-001-X. – Текст : непосредственный.
2. Роджерс, Д. Алгоритмические основы машинной графики / Д. Роджерс; пер. с англ. – Москва : Мир, 1989. – 512 с. – ISBN 5-03-000476-9. – Текст : непосредственный.
3. Роджерс, Д. Математические основы машинной графики / Д. Роджерс, Дж. Адамс; пер. с англ. – Москва : Мир, 2001. – 604 с. – ISBN 5-03-002143-4. – Текст : непосредственный.
4. Шикин, Е. В. Начала компьютерной графики / Е. В. Шикин, А. В. Боресков, А. А. Зайцев. – Москва : ДИАЛОГ-МИФИ, 1993. – 138 с. – ISBN 5-86404-035-5. – Текст : непосредственный.