Миленко Н.В.

Вариант 1.

2. Алгоритм построчной заливки.

// подготовка данных для алгоритма

Для всех ребер:

Заносим в поле x минимальное значение x ребра

Заносим в поле n значение вершины ребра (наивысший y)

Заносим в поле dy модуль разности y-координат концов (кол-во строк)

Заносим в поле dx разность x-координат делёную на разность y-координат

Поле next = NULL

// сортировка

Сортируем ребра по n в порядке убывания

// алгоритм

Обнуляем сар

Помещаем в y наивысшую строку (figures[0][0].n)

Позииция = 0

Пока y > 0:

Если просмотрели все ребра и сар пуст, завершаем (позиция == кол-во ребер)

Для всех ребер после текущей позиции (for j in range(кол-во ребер):

Если поле n == y and x > 0 (вершина ребра в текущей сканирующей строке):

Вносим ребро в САР

Позиция = j + 1

заполняем необходимые пиксели (описано ниже)

y-- (переход на новую строку)

корректируем все рёбра (dy -= 1, x -= dx)

удаляем из САР рёбра, для которых поле dy обнулилось

// заполнение пикселей

Из САР заносим все текущие координаты x рёбер в массив/вектор

Сортируем по возрастанию

Высвечиваем все промежутки, номер которых (0 - n) не кратен 2 (каждый чётный промежуток)

3. Алгоритм Сазерленда-Коэна

Возможные значения флага:

F = 0 – соответствует отрезку общего положения

F = -1 – отрезок вертикальный

F = 1 – отрезок горизонтальный

Псевдокод по пунктам:

1. Ввод координат отсекателя (Xл, Xп, Yв, Yн)
2. Ввод координат концов отрезка (p1(x1, y1), p2(x2, y2))
3. Установка начального значения флага F = 0
4. Проверка вертикальности отрезка:

Если p1.x – p2.x == 0(вертикальный), то F = -1

Иначе вычислить тангенс угла наклона m:

M = (p2.y – p1.y) / (p2.x – p1.x)

1. Проверка горизонтальности отрезка:

Если m = 0, то F = 1

1. Начало цикла по i от 1 до 4 отсечения отрезка по всем четырем сторонам отсекателя.
2. Обращение к алгоритму (подпрограмме) определения видимости отрезка p1p2 относительно заданного окна.
3. Подпрограмма возвращает признак pr, принимающий следующие значения:

pr = 1 - отрезок видимый;

pr = -1 - отрезок полностью невидимый;

pr = 0 - отрезок может быть частично видимым.

1. Анализ полученного признака видимости:

если pr= -1, то переход к п.21;

если pr=1, то переход к п.20.

1. Проверка видимости обеих вершин отрезка относительно текущей i-ой стороны окна:

если T1i=T2i, то переход к п.19.

1. Проверка видимости первой вершины:

если T1i=0 (вершина видима), то обмен местами вершин:

R = p1;

p1 = p2;

p2 = R.

1. Проверка вертикальности отрезка:

если F = -1, то переход к п. 15.

1. Анализ номера шага отсечения: если i >= 3, то переход к п. 15.
2. Вычисление координат точки пересечения с i-ым ребром отсекателя (левым или правым):

p1.y = m(Oi-p1.x) + p1.y;

p1.x = Oi.

Переход к п. 19.

1. Проверка горизонтальности отрезка:

если F = 1, то переход к п. 19.

1. Проверка вертикальности отрезка:

если F = -1, то переход к п.18.

1. Вычисление абсциссы точки пересечения отрезка общего положения со стороной отсекателя(верхней или нижней):

p1.x = (Oi-p1.y) / m + p1.x .

1. Присвоение ординате вершины отрезка ординаты стороны отсекателя:

p1.y = Oi.

1. Конец цикла по i (вычисление нового значения параметра цикла i=i+1, анализ его значения и переход на повторное выполнение цикла или выход из цикла).
2. Вычерчивание отрезка p1p2.
3. Конец.

5. Определение изображения. Формализованная постановка задачи синтеза снимка плоскости с расположенными на ней выпуклыми многоугольниками. Декомпозиция первого уровня.