**Лабораторная работа N8**

**Реализация алгоритмов растровой развертки линий**

Краткие теоретические сведения.

Поскольку используемые дисплеи - растровые, для представления геометрического объекта на экране необходимо уметь получать его дискретное или растровое представление. Это означает, что заданной геометрической фигуре следует поставить в соответствие множество точек растра, которое в некотором смысле является приближением этой фигуры. Такое представление неоднозначно, т.к. существуют разные понятия и способы приближения непрерывного объекта. Например, растровым приближением круга на плоскости может служить множество внутренних точек круга, а прямой - множество точек - центров ячеек растра, с которыми эта прямая пересекается. Важным является эффективность того или иного представления.

Существуют стандартные процедуры программной или аппаратной реализации алгоритмов генерации линий и закраски многоугольников в различных графических пакетах, но для студентов необходимо представлять и уметь программно реализовывать такие алгоритмы при работе с устройствами типа дисплея, принтера, плоттера. Необходимость использовать свою версию алгоритма генерации возникает, например, когда атрибуты инициируемого пиксела зависят от каких-либо условий (положения пиксела на прямой или внутри закрашиваемого многоугольника) или когда Вас не устраивает скорость работы "стандартного" алгоритма. Такие ситуации типичны при реализации алгоритмов загораживания (например, растровой реализации одной из версий алгоритма плавающего горизонта или метода буфера глубины).

***Разложение отрезка в растр.***

Процесс последовательной инициализации множества пикселов экрана составляющих отрезок называется его растровой разверткой, а само это множество - растровым представлением отрезка. Далее будем рассматривать два таких представления: восьмисвязное и четырехсвязное. Восьмисвязная развертка отрезка предполагает различие у соседних точек на отрезке только значения одной координаты. У четырехсвязной - могут отличаться обе координаты.

***Простой пошаговый алгоритм.***

Если увеличивать с определенным шагом координату Х, а затем находить координату Y используя уравнение прямой Y=mX+b и подкрашивать пиксел с координатами (Х,ROUND(Y)), то потребуется много времени (на нецелочисленные операции).

Если шаг по Х принять равным единице, то m=dY/dX сводится к m=dY, т.е. изменение Х на единицу приведет к изменению углового коэффициента на m. Т.о., если Х(I+1)=X(I)+1, то Y(I+1)=Y(I)+m. Алгоритм корректно работает только для отрезков в первом и восьмом октантах. В остальных случаях требует модификации, что предлагается сделать студентам самостоятельно в ходе выполнения данной лабораторной работы. При модификации следует учесть, что при m>1,единичный шаг по Х приведет к такому увеличению Y, при котором две соседние точки на прямой расположатся далеко друг от друга. Поэтому X и Y следует поменять, чтобы увеличивать на единицу Y, а Х – на dX = dY/m = 1/m.

Описание алгоритма для работы в первом и восьмом октантах:

Дано:

m - угловой коэффициент(-1<m<1);

(x1,y1) (x2,y2) - кооpдинаты концов отрезка;

col- цвет для подкрашивания точек отрезка;

Нарисовать\_точку (x,y,col) – процедура для подкрашивания цветом col пиксела с координатами(x,y); Round(x) - функция округления х до ближайшего целого.

Требуется: разложить отрезок в pастp.

1.Если х1 # x2 идти к 2 иначе к 4.

2.m=(y2-y1)/(x2-x1);y=y1;

3.Для x от х1 до х2 выполнить:

начало

Нарисовать\_точку(x,Round(y),col);

y=y+m;

конец

4.Если y1=y2 идти к 5 иначе к 6.

5.Нарисовать\_точку(x1,y1,col).Идти к 7.

6.Вывести сообщение об ошибке.

7.Закончить.

***Целочисленные алгоритмы Брезенхема.***

Основной недостаток простейшего алгоритма – использование вещественных вычислений. В 1965 году Брезенхемом был предложен целочисленный алгоритм растрового построения отрезка, первоначально предназначенный для графопостроителей.

Суть алгоритма в следующем: в процессе работы одна из координат либо х, либо y(в зависимости от углового коэффициента) изменяется на единицу. Изменение другой координаты (на 0 или 1) зависит от расстояния - (е) между действительным положением отрезка и ближайшими координатами растра (е назовем управляющей переменной). Алгоритм построен так, что на каждом шаге проверяется лишь знак е и корректируется ее значение после каждого изменения очередной координаты. Значение исходной управляющей переменной: е=2\*(y2-y1)-(x2-x1), где x1,y1,x2,y2 - координаты начальной и конечной точек отрезка.

В каждом шаге при e>=0 значение y от предыдущего увеличивается на единицу, а е уменьшается на 2\*(x2-x1), в противном случае - y не меняется, а значение e увеличивается на 2\*(y2-y1). В обоих случаях координата х следующего пиксела увеличивается на единицу от предыдущего значения.

Описание общего алгоритма Брезенхема:

Для восьмисвязной развертки:

Дано:

(x1,y1,x2,y2)-координаты начальной и конечной точек отрезка;

col - цвет пикселов отрезка;

нарисовать\_точку(x,y,col) - процедура для подкрашивания цветом col пиксела с координатами(x,y);

sign(x)-функция,возвращающая 1,если х>=0 и -1,если х<0;

round(x)- функция округления х до ближайшего целого;

abs(x) - функция,возвращающая целое от х.

Требуется : Pазложить отрезок в pастp.

1.Присвоить: x=x1; y=y1; dx=abs(x2-x1); dy=abs(y2-y1);

s1=sign(x2-x1); s2=sign(y2-y1);

2.Если dy>dx идти к 3, иначе к 4.

3.Поменять ролями dx и dy.Переменной l присвоить значение "истина".

Идти к 5.

4.l присвоить значение "ложь".Идти к 5.

5.Присвоить е исходное значение 2\*dy-dx.

6.Для i от 1 до dx выполнить: начало

Нарисовать\_точку(x,y,col); Пока e>=0 выполнить:

начало если l="истина" то x=x+s1 иначе y=y+s2;

e=e-2\*dx

конец

Если l="истина" то y=y+s2 иначе x=x+s1; e=e+2\*dy

конец

7.Нарисовать\_точку(x,y,col).

8.Закончить.

Для четырехсвязной развертки:

Дано:

(x1,y1,x2,y2)-координаты начальной и конечной точек отрезка;

сol - цвет пикселов отрезка;

Нарисовать\_точку(x,y,col) - процедура для подкрашивания цветом col пиксела с координатами(x,y);

sign(x)- функция,возвращающая 1,если х>=0 и -1,если х<0;

round(x) - функция округления х до ближайшего целого;

abs(x) - функция,возвращающая целое от х;

Требуется : Pазложить отрезок в pастp.

1.Присвоить: x=x1; y=y1; dx=abs(x2-x1); dy=abs(y2-y1);

s1=sign(x2-x1); s2=sign(y2-y1);

2.Если dy<dx то присвоить переменной l значение "ложь" и идти к

4 иначе идти к 3.

3.Поменять ролями dx и dy.Переменной l присвоить значение "исти-

на".Идти к 4.

4.Присвоить е исходное значение 2\*dy-dx.

5.Для i от 1 до dx+dy выполнить: начало

Нарисовать\_точку(x,y,col);

Если e<dx выполнить: начало

если l="истина" то y=y+s2 иначе x=x+s1;

e=e+2\*dy

конец иначе выполнить:

начало

если l="истина" то х=х+s1 иначе y=y+s2;

e=e-2\*dx

конец

конец

6.Нарисовать\_точку(x,y,col).

7.Закончить.

**Растровая развертка окружностей.**

Существуют несколько простых способов преобразования окружности в растровую форму. Например, по формуле X\*X+Y\*Y=R\*R для окружности с центром в начале координат. Чтобы изобразить четверть такой окружности на каждом шаге меняем Х от 0 до R на 1 и вычисляем Y как SQRT(R\*R-X\*X). Остальные четверти изображаем симметрично. Этот метод содержит операции умножения и извлечения корня, потому не эффективен. Кроме того, при Х близких к R в окружности появляются заметные промежутки, так как при таких Х тангенс угла наклона касательной к окружности стремится к бесконечности. Процесс можно улучшить, если вычислять одну восьмую часть окружности, а остальные семь частей отображать симметрично (в предыдущем случае Х менять от 0 до R/SQRT(2)). Но необходимый эффект можно получить только при работе с целыми числами(алгоритмами, подобными рассмотренным выше). Для генерации окружностей Брезенхем предложил следующий алгоритм:

Дано: rad - радиус окружности

col - цвет пикселов окружности

e - управляющаяпеременная

Нарисовать\_точку(x,y,col) - процедура для подкрашивания цветом col пиксела с координатами(x,y)

Требуется: Разложить окружность в растр.

1.Присвоить:x=0;y=rad;e=3-2\*rad;

2.Пока x<y выполнить:

Нарисовать\_точку(x,y,col);

Нарисовать\_точку(y,x,col);

Нарисовать\_точку(y,-x,col);

Нарисовать\_точку(x,-y,col);

Нарисовать\_точку(-x,-y,col);

Нарисовать\_точку(-y,-x,col);

Нарисовать\_точку(-y,x,col);

Нарисовать\_точку(-x,y,col)

3.Если е<0 то e=e+4\*x+6 и идти к 5 иначе идти к 4.

4.Выполнить:

e=e+4\*(x-y)+10; y=y-1

5.x=x+1 идти к 2 пока x<y иначе к 6.

6.Если x=y то идти к 7 иначе к 8.

7.Выполнить:

Нарисовать\_точку(x,y,col);

Нарисовать\_точку(y,x,col);

Нарисовать\_точку(y,-x,col);

Нарисовать\_точку(x,-y,col);

Нарисовать\_точку(-x,-y,col);

Нарисовать\_точку(-y,-x,col);

Нарисовать\_точку(-y,x,col);

Нарисовать\_точку(-x,y,col).

8.Закончить.

**Задание на лабораторную работу:**

1. Написать на языке PASCAL программу, реализующую алгоритмы построения прямой: простой пошаговый алгоритм и алгоритмы Брезенхема для четырех- и восьмисвязной развертки.

2. Проверить правильность работы программы, нарисовав, например, каждым алгоритмом семейство радиальных прямых, выходящих из одной точки с шагом 15 градусов.

3. Написать и отладить программу, реализующую два алгоритма построения окружности: по формуле Y=+-SQRT(r\*r-x\*x) и Брезенхема. В обоих случаях использовать свойство симметрии окружности (в первом - найдя точки четверти окружности, остальные - отразив симметрично; во втором - свойство симметрии использовать полностью).

**Дополнительное задание:**

Используя программы работы с мышью и программы, написанные при выполнении пунктов 1-3 задания, вывести на экран простой рисунок из линий и окружностей.

***Вопросы для самоподготовки.***

1. Каков визуальный эффект применения простого пошагового алгоритма к отрезкам, угловой коэффициент которых больше 1 или меньше -1.

2. Как избавиться от этого эффекта.

3. Почему простой пошаговый алгоритм не используют для генерации отрезков в современных графических пакетах.

4. Каков визуальный эффект при нахождении координаты Y всех точек или одной четверти окружности по формуле Х\*Х+Y\*Y=R\*R. Как избавиться от этого эффекта.

5. Каким образом в алгоритмах Брезенхема удается избежать нежелательных эффектов при построении точек линий.

Литература:

1.Фоли Дж., Ван Дэм А. Основы интерактивной машинной графики. В 2-х книгах.,кн.2 - М.:Мир,1985

2.Роджерс Д. Алгоритмические основы машинной графики.-М.:Мир,1989

3.ШикинЕ.В., Боресков А.В., Зайцев А.А. Начала компьютерной графики.-М.:ДИАЛОГ-МИФИ,1993

4.Шикин Е.В., Боресков А.В. Компьютерная графика. -М.: ДИАЛОГ - МИФИ, 1995