Computer Graphics

Line and Circle

1-Line Drawing algorithms

1-1 : A straight line may be defined by two endpoints and an equation

If the two endpoints used to specify a line are (X1,Y1) and (X2,Y2) , then the equation of the line is used to describe the X , Y coordinates of all the pointes that lie between these two endpoints

The equation of the straight line is :

$$Y = M * X + B$$

Where (M) is the slope of the line ∴ $M = \frac{\Delta Y}{\Delta X}$

and (B) is the Y intercept.

The Y values of the pointes of the line can be calculated using the above equation, by incrementing X values from X1 to X2 and substitute it in the line equation.

Note : The slope between any point (X,Y) on the line and (X1,Y1) is the same as the slope between (X2,Y2) and (X1,X2)

$$\frac{Y-Y1}{X-X1} = \frac{Y2-Y1}{X2-X1}$$

1-2 : DDA (Digital Differential Analyzer) algorithm

The DDA algorithm generates lines from their differential equations.

We calculate the length of the line in the X direction (number of pointes) by the equation :

ABS (X2-X1)

and calculate the length of the line in the Y direction (number of pointes) by the Equation :

ABS (Y2-Y1)

Where *ABS* is a function takes the positive of the arguments.

The Length estimates is equal to the larger of the magnitudes of the above two equations.

The increment steps (dX and dY) are used to increment the X and Y coordinates for the next pointes to be plotted

$$dX = \frac{X2 - X1}{\text{Larger Length}} \qquad dY = \frac{Y2 - Y1}{\text{Larger Length}}$$

Algorithm DDA

Start

If ABS(X2-X1) > ABS (Y2-Y1) Then
Length=ABS (X2-X1)
Else
Length=ABS (Y2-Y1)

dX = (X2-X1) / Length
dY = (Y2-Y1) / Length

*X=X1+ 0.5 * Sign(ΔX)*
*Y=Y1+ 0.5 * Sign(ΔY)*

For I=1 to Length
Begin
Plot(Int(X) , Int (Y))
X=X+dX
Y=Y+dY
End

Finish

Note :

1- Sign function returns : -1 if its argument is < 0
: 0 if its arguments is = 0
: +1 if its arguments is > 0

Ex. Sign(-10) = -1 Sign (5) = 1

Using the Sign function makes the algorithm work in all quadrants.

2- Int function works as follow :
Int (8.5) = 8
Int (-8.5) = -9

Example 1 : Consider the line from (0,0) to (5,5)
Use DDA to rasterize the line.

Sol 1 :

X1=0 ; Y1=0 ; X2=5 ; Y2=5 ; Length=5

dX=1 ; dY=1 ; X=0.5 ; Y=0.5

I	Plot	X	Y
		0.5	0.5
1	(0,0)	1.5	1.5
2	(1,1)	2.5	2.5
3	(2,2)	3.5	3.5
4	(3,3)	4.5	4.5
5	(4,4)	5.5	5.5

Note : the integer part of X and Y are used in plotting the line.

This would normally have the effect of truncating rather then rounding so we initialize the DDA with the value 0.5 in each of the fractional parts to achieve true rounding. One advantage of this arrangement is that it allows us to detect changes in X and Y and hence to avoid plotting the same point twice

Example 2 : Consider the line from (0,0) to (-8,-4) ; evaluate the DDA algorithm

Sol 2 :

X1=0 ; Y1=0 ; X2=-8 ; Y2=-4 ; Length =8

dX=-1 ; dY=-0.5 ; X=-0.5 ; Y=-0.5

i	plot	X	Y
		-0.5	-0.5
1	(-1,-1)	-1.5	-1
2	(-2,-1)	-2.5	-1.5
3	(-3,-2)	-3.5	-2
4	(-4,-2)	-4.5	-2.5
5	(-5,-3)	-5.5	-3
6	(-6,-3)	-6.5	-3.5
7	(-7,-4)	-7.5	-4
8	(-8,-4)	-8.5	-4.5

Features of DDA

1- The algorithm is orientation dependent
2- The end point accuracy deteriorates
3- The algorithm suffer from the fact that it must be performed using floating point arithmetic