MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY
HARINGHATA, WEST BENGAL

MAULANA ABUL KALAM AZAD
UNIVERSITY OF TECHNOLOGY,
WEST BENGAL

Utech

In Pursuit Of Knowledge And Excellence

NAME – NIMA DORJEE LAMA

ROLL NO.- 10000220031

REGISTRATION NO. – 201000100210010

DEPT / YEAR - B. TECH (IT) 5th SEMESTER

SUBJECT – COMPUTER GRAPHICS

1.DDA stands for Digital Differential Analyzer. It is an incremental method of scan conversion of lines. In this method calculation is performed at each step but by using results of previous steps.

Suppose at step i, the pixels is (x_i,y_i)

The line of equation for step i

y_i=mx_{i+b}......................equation 1

Next value will be

y_{i+1}=m$_{xi+1}$+b................equation 2

m = $\frac{\Delta y}{\Delta x}$

y_{i+1}-y_i=Δy ....equation 3

y_{i+1}-x_i=Δx...................... equation 4

y_{i+1}=y_i+Δy

Δy=mΔx

y_{i+1}=y_i+mΔx

Δx=Δy/m

x_{i+1}=x_i+Δx

x_{i+1}=x_i+Δy/m

Case1: When |M|<1 then (assume that $x_1$2)

x= x_1,y=y_1 set Δx=1

y_{i+1}=y_{1+m},   x=x+1

Until x = x_2

Case2: When |M|<1 then (assume that $y_1$2)

x= x_1,y=y_1 set Δy=1

x_{i+1}=$\frac{1}{m}$,   y=y+1

Until y → y_2

Example: If a line is drawn from (2, 3) to (6, 15) with use of DDA. How many points will needed to generate such a line?

Solution: P1 (2,3)   P11 (6,15)

x1=2

y1=3

x2= 6

y2=15

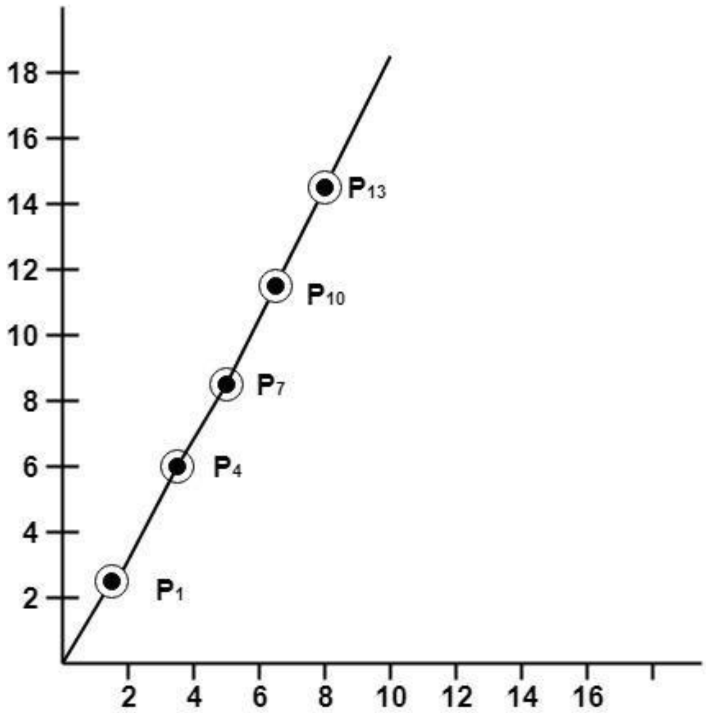
dx = 6 - 2 = 4

dy = 15 - 3 = 12

m = $\frac{dy}{dx} = \frac{12}{4}$

For calculating next value of x takes x = x + $\frac{1}{m}$

$P_1(2,3)$	point plotted
$P_2(2\frac{1}{3},4)$	point plotted
$P_3(2\frac{2}{3},5)$	point not plotted
$P_4(3,6)$	point plotted
$P_5(3\frac{1}{3},7)$	point not plotted
$P_6(3\frac{2}{3},8)$	point not plotted
$P_7(4, 9)$	point plotted
$P_8(4\frac{1}{3},10)$	point not plotted
$P_9(4\frac{2}{3},11)$	point not plotted
$P_{10}(5,12)$	point plotted
$P_{11}(5\frac{1}{3},13)$	point not plotted
$P_{12}(5\frac{2}{3},14)$	point not plotted
$P_{13}(6,15)$	point plotted



1. Bresenham's Algorithm

This algorithm is used for scan converting a line. It was developed by Bresenham. It is an efficient method because it involves only integer addition, subtractions, and multiplication operations. These operations can be performed very rapidly so lines can be generated quickly.

In this method, next pixel selected is that one who has the least distance from true line.

The method works as follows:

Assume a pixel $P_1'(x_1',y_1')$,then select subsequent pixels as we work our may to the night, one pixel position at a time in the horizontal direction toward $P_2'(x_2',y_2')$.

Once a pixel in choose at any step

The next pixel is

1. Either the one to its right (lower-bound for the line)
2. One top its right and up (upper-bound for the line)

The line is best approximated by those pixels that fall the least distance from the path between P_1',P_2'.

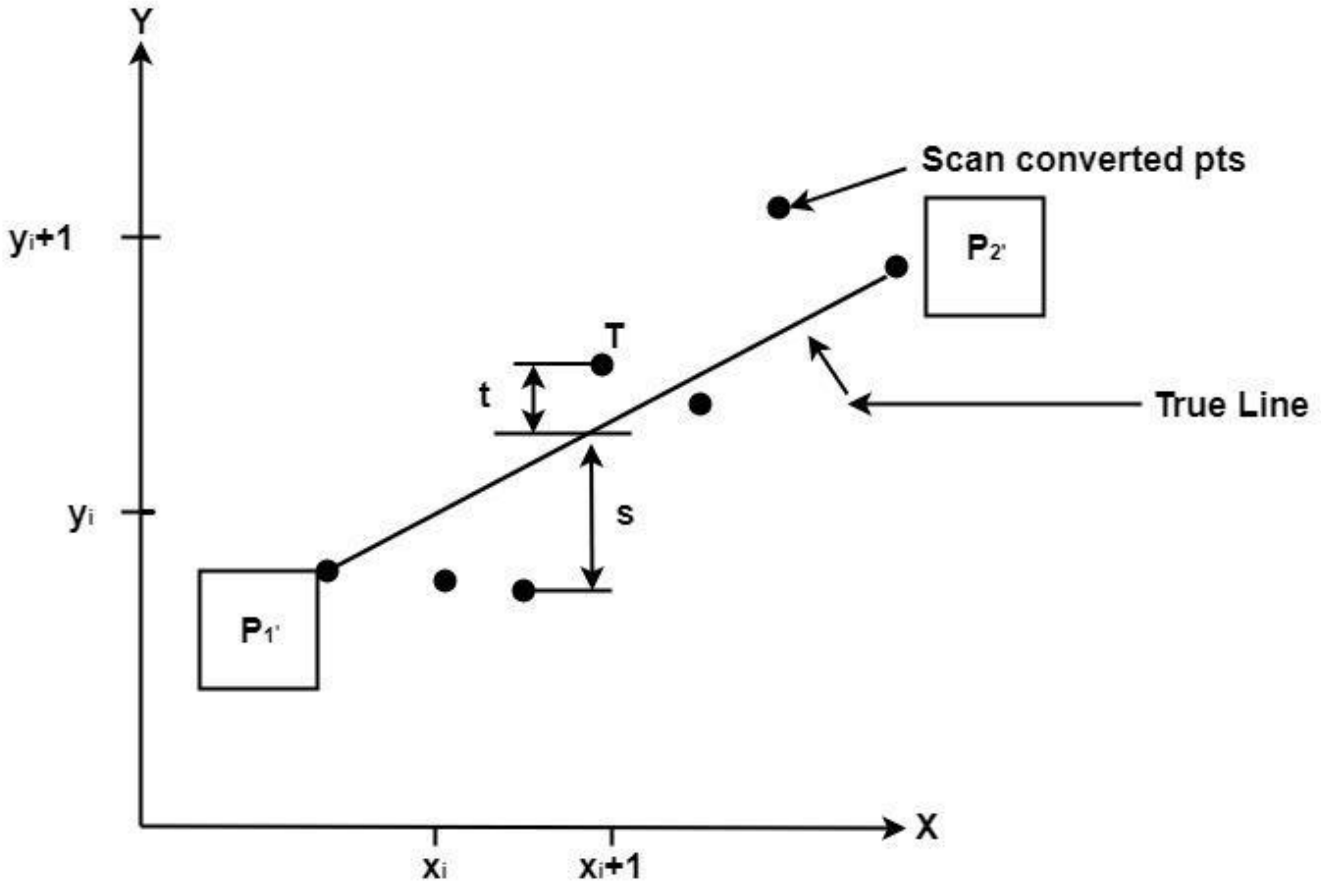


Fig: Scan Converting a line.

To chooses the next one between the bottom pixel S and top pixel T.

If S is chosen

We have $x_{i+1}=x_i+1$ and $y_{i+1}=y_i$

If T is chosen

We have $x_{i+1}=x_i+1$ and $y_{i+1}=y_i+1$

The actual y coordinates of the line at $x = x_{i+1}$ is

$y=mx_{i+1}+b$

$$\boxed{y = m(x_i + 1) + b}$$

The distance from S to the actual line in y direction

$s = y-y_i$

The distance from T to the actual line in y directiont

$= (y_i+1)-y$

Now consider the difference between these 2 distance values

s - t

When (s-t) <0 $\Longrightarrow$ s < t

The closest pixel is S

When (s-t) $\geq$0 $\Longrightarrow$ s < t

The closest pixel is T

This difference is

s-t = (y-yi)-[(yi+1)-y]

= 2y - 2yi -1

$$\boxed{s - t = 2m(x_i + 1) + 2b - 2y_i - 1}$$ [Putting the value of (1)]

Substituting m by $\frac{\triangle y}{\triangle x}$ and introducing decision variable

$d_i=\triangle x$ (s-t)

$d_i=\triangle x \, (2 \, \frac{\triangle y}{\triangle x} \, (x_i+1)+2b-2y_i-1)$

$=2\triangle xy_i-2\triangle y-1\triangle x.2b-2y_i\triangle x-\triangle x$

$d_i=2\triangle y.x_i-2\triangle x.y_i+c$

Where c= $2\triangle y+\triangle x$ (2b-1)

We can write the decision variable d_{i+1} for the next slip on

$d_{i+1}=2\triangle y.x_{i+1}-2\triangle x.y_{i+1}+c$

$d_{i+1}-d_i=2\triangle y.(x_{i+1}-x_i)- 2\triangle x(y_{i+1}-y_i)$

Since x_(i+1)=x_i+1,we have

$d_{i+1}+d_i=2\triangle y.(x_i+1-x_i)- 2\triangle x(y_{i+1}-y_i)$

Special Cases

If chosen pixel is at the top pixel T (i.e., $d_i \geq 0$) $\Longrightarrow$ $y_{i+1}=y_i+1$

$d_{i+1}=d_i+2\triangle y-2\triangle x$

If chosen pixel is at the bottom pixel T (i.e., $d_i<0$) $\Longrightarrow$ $y_{i+1}=y_i$

$d_{i+1}=d_i+2\triangle y$

Finally, we calculate d_1

$d_1=\triangle x[2m(x_1+1)+2b-2y_1-1]$

$d_1=\triangle x[2(mx_1+b-y_1)+2m-1]$

Since $mx_1+b-y_i=0$ and $m = \frac{\triangle y}{\triangle x}$, we have

$d_1=2\triangle y-\triangle x$

Example: Starting and Ending position of the line are (1, 1) and (8, 5). Find intermediate points.

Solution: $x_1=1$

$y_1=1$

$x_2=8$

$y_2=5$

$dx= x_2-x_1=8-1=7$

$dy=y_2-y_1=5-1=4$

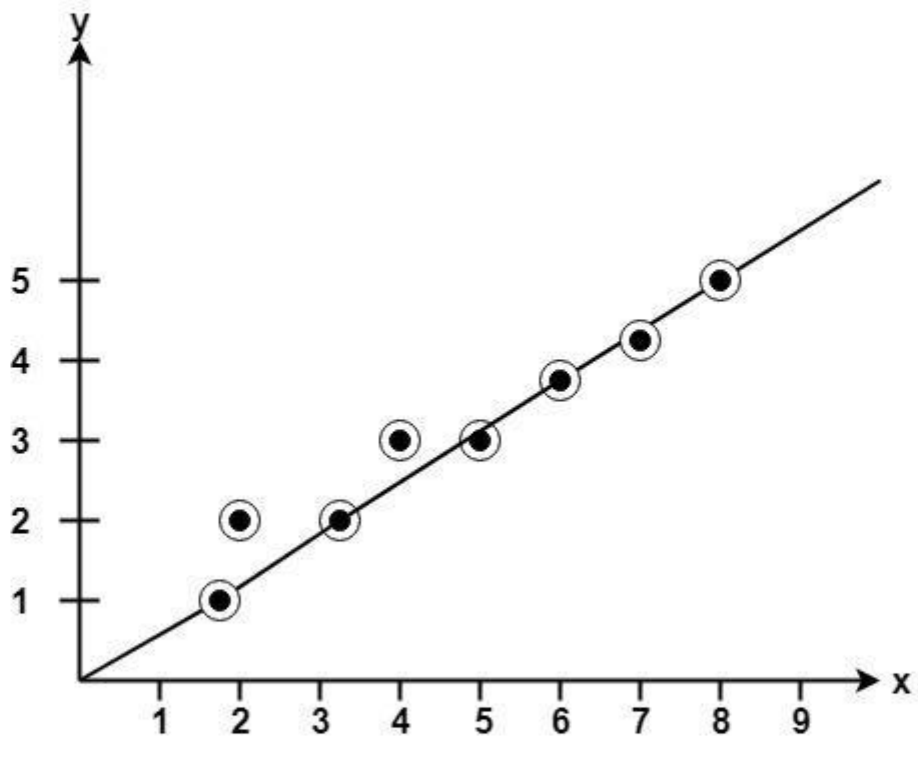
$I_1=2* \Delta y=2*4=8$

$I_2=2*(\Delta y-\Delta x)=2*(4-7)=-6d$

$= I_1-\Delta x=8-7=1$

x	**y**	**d=d+I₁ or I₂**
1	1	$d+I_2=1+(-6)=-5$

2	2	$d+I_1=-5+8=3$
3	2	$d+I_2=3+(-6)=-3$
4	3	$d+I_1=-3+8=5$
5	3	$d+I_2=5+(-6)=-1$
6	4	$d+I_1=-1+8=7$
7	4	$d+I_2=7+(-6)=1$
8	5	



3.

S.NO	DDA Line Algorithm	Bresenham line Algorithm
1.	DDA stands for Digital Differential Analyzer.	While it has no full form.
2.	DDA algorithm is less efficient than Bresenham line algorithm.	While it is more efficient than DDA algorithm.
3.	The calculation speed of DDA algorithm is less than Bresenham line algorithm.	While the calculation speed of Bresenham line algorithm is faster than DDA algorithm.
4.	DDA algorithm is costlier than Bresenham line algorithm.	While Bresenham line algorithm is cheaper than DDA algorithm.
5.	DDA algorithm has less precision or accuracy.	While it has more precision or accuracy.
6.	In DDA algorithm, the complexity of calculation is more complex.	While in this, the complexity of calculation is simple.
7.	In DDA algorithm, optimization is not provided.	While in this, optimization is provided.