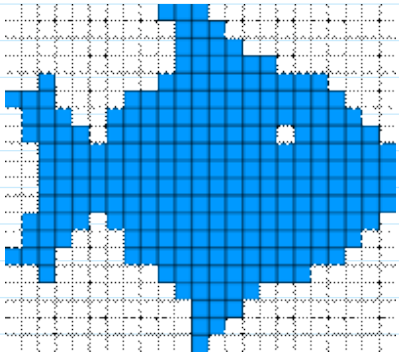
12. Scan conversion algorithms for Line

14 February 2024 12:34

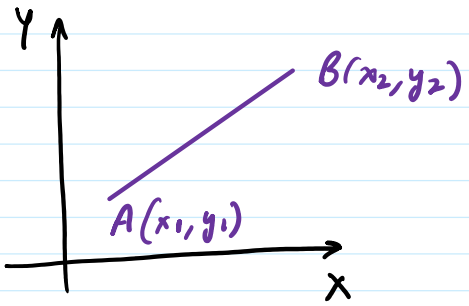
Rasterization / Scan Conversion :-

Representation of continuous graphics objects as a collection of discrete pixels.



Scan Conversion of a Line :-

Line :-



Equation of line,

$$y = mx + c$$

$$m = \Delta y / \Delta x$$

$$\Delta x = x_2 - x_1$$

$$\Delta y = y_2 - y_1$$

Using scan conversion algorithms, line is drawn by plotting the

pixels in sequence.

Scan conversion algorithms for Line Drawing

- Direct Method
- Digital Differential Analyzer (DDA)
- Bresenham's Algorithm

Direct Method :-

Two known end points
↓
Find other points lying on the line using $y = mx + c$

Algorithm :-

① Read $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$

② Calculate

$dx = x_2 - x_1$

$dy = y_2 - y_1$

③ Calculate slope

$m = dy/dx$

④ Set (x, y) to starting point

if $dx > 0$ then

$x = x_1$

$y = y_1$

$x_{end} = x_2$

if $dx < 0$ then

$x = x_2$

$y = y_2$

$x_{end} = x_2$

⑤ Now calculate

$c = y - mx$

⑥ Plot a point at current (x, y) coordinates.

⑦ Increment x, $x = x + 1$

⑧ Compute y, $y = mx + c$

⑨ If $x = x_{end}$ then stop

otherwise Go to ⑥

Q8- Draw a line using direct method between points $(0,0)$ & $(8, 16)$

① (x_1, y_1) (x_2, y_2)

(0,0) and (8, 16)

② $d_x = 8-0 = 8$, $d_y = 16-0 = 16$.

③ $m = dy/dx = 16/8 = 2$.

④ Set initial point (x, y)

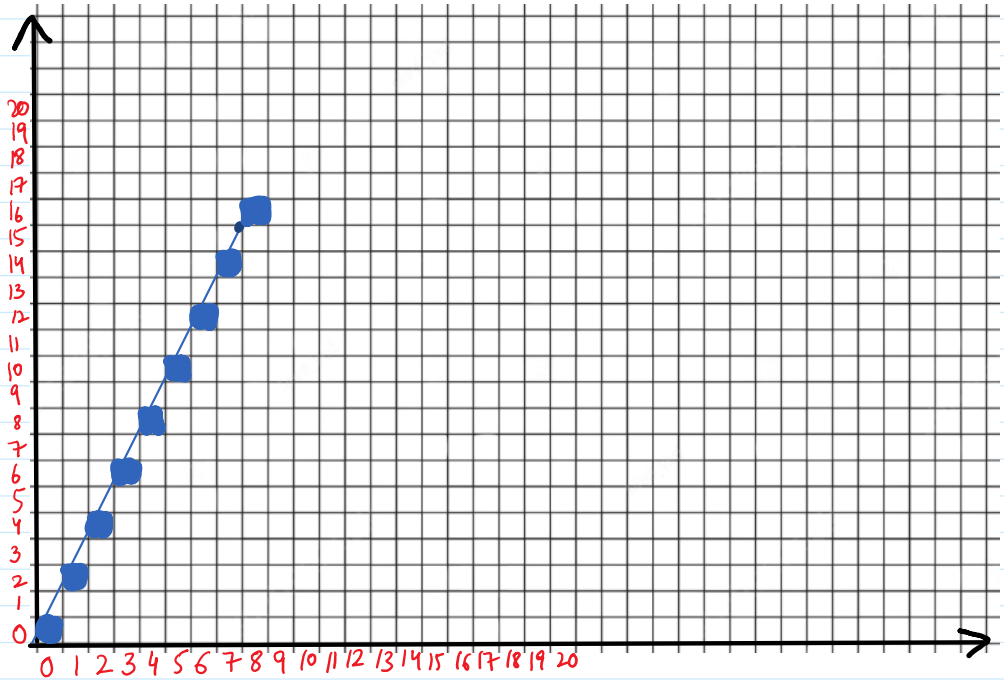
$dx > 0$

$\Rightarrow x = 0 \quad y = 0 \quad x_{end} = 8$

⑤ Calculate $c = y - mx \quad c = 0$

While $x = x_{end}$

$x = x+1$	$y = mx + 0$	Points
0	0	$P_1(0,0)$
1	2	$P_2(1, 2)$
2	4	$P_3(2, 4)$
3	6	$P_4(3, 6)$
4	8	$P_5(4,8)$
5	10	$P_6(5,10)$
6	12	$P_7(6, 12)$
7	14	$P_8(7, 14)$
8	16	$P_9(8, 16)$



② Digital Differential Analyzer (DDA)

① $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$

② Finding appropriate length of the line

if (abs $(x_2 - x_1)$ > abs $(y_2 - y_1)$) then

Length = abs $(x_2 - x_1)$

otherwise

Length = abs $(y_2 - y_1)$

③ Find raster unit

$dx = (x_2 - x_1) / \text{Length}$

$dy = (y_2 - y_1) / \text{Length}$

} floating point calculations

④ set $x = x_1$, $y = y_1$ and $i = 0$

⑤ Plot (x, y)

⑤ Plot (x, y)

$x = x + dx$

$y = y + dy$

⑥ Repeat ⑤ until i <= length

Example :- (0, 0) & (8, 8)

① $x_1 = 0$ $y_1 = 0$ $x_2 = 8$ $y = 8$

② abs $(x_2 - x_1) = 8$

abs $(y_2 - y_1) = 8$

length = 8

③ $dx = 1$

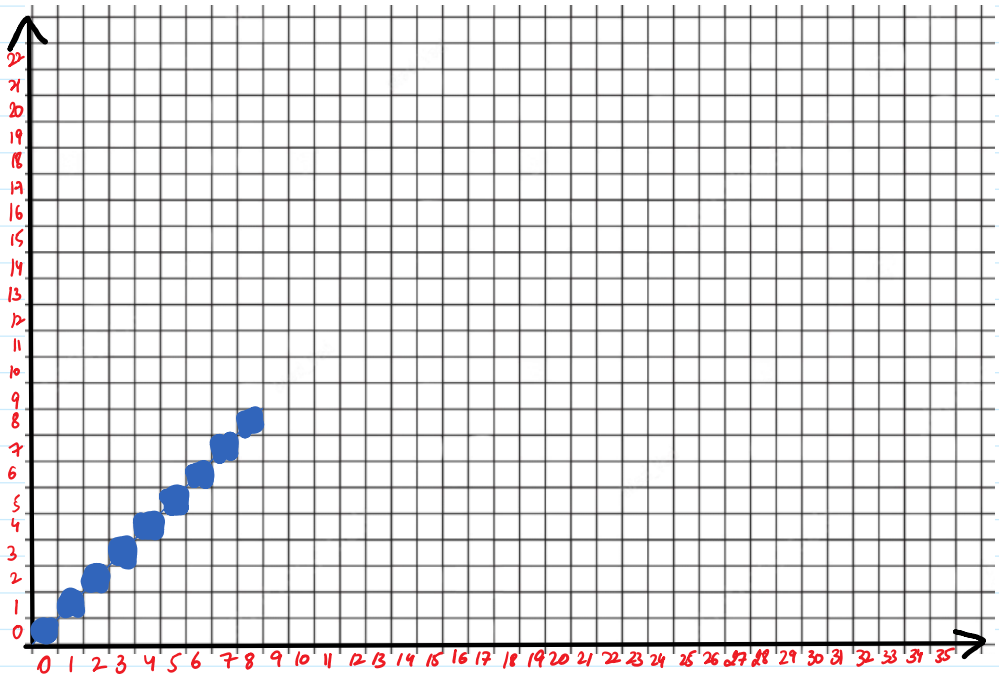
$dy = 1$

④ Set (x, y) $x = 0$, $y = 0$

⑤ Execute until i <= length

i	x	y	Points
0	0	0	(0,0)
1	1	1	(1,1)
2	2	2	(2,2)
3	3	3	(3,3)
4	4	4	(4,4)
5	5	5	(5,5)
6	6	6	(6,6)
7	7	7	(7,7)
8	8	8	(8,8)

8 | 8 | 8 | (8,8)

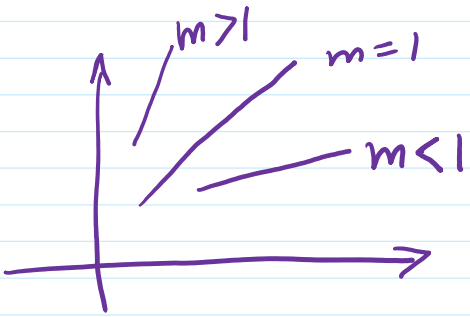


"Floating Point Calculations"

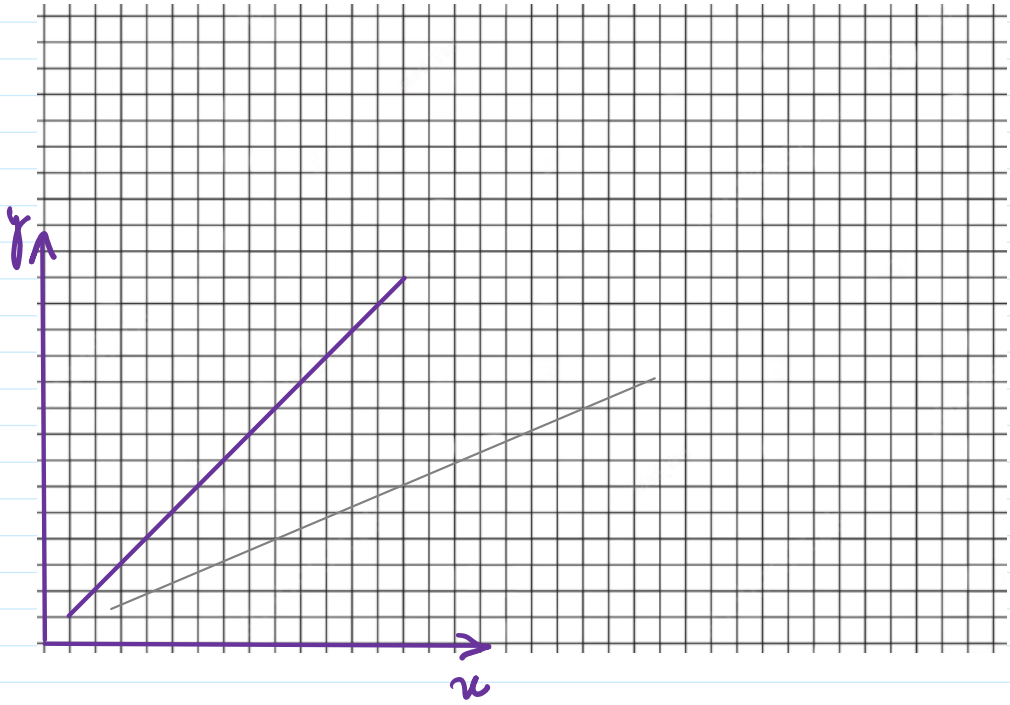
③ Bresenham's Line Algorithm

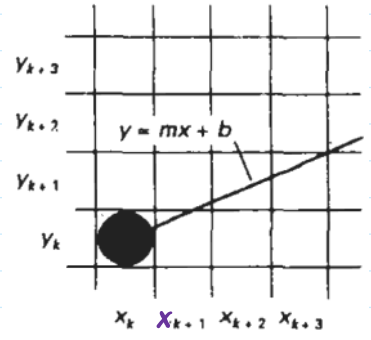
Jack Elton Bresenham, 1962
(IBM)

Using only integer calculations.



For slope $m < 1$,
sampling at unit x intervals.





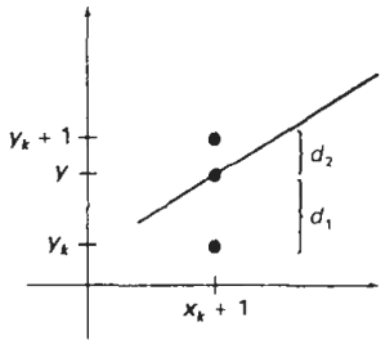
$m<1$

(x_k, y_k) next ?

(x_k+1, y_k) (x_k+1, y_k+1)

$y = m(x_k+1) + b$

At x_k+1, separating pixel from line,



from figure,

$$d_1 = y - y_k$$
$$= m(x_k+1) + b - y_k$$

and

$$d_2 = (y_k+1) - y$$
$$= y_k + 1 - m(x_k+1) - b$$

$$d_1 - d_2 = 2m(x_k+1) - 2y_k + 2b - 1$$

$m = \Delta y / \Delta x$

$$p_k = (d_1 - d_2)\Delta x = 2\Delta y \cdot x_k - 2\Delta x \cdot y_k + c$$

$$p_{k+1} = 2\Delta y \cdot x_{k+1} - 2\Delta x \cdot y_{k+1} + c$$

$$p_{k+1} - p_k = 2\Delta y(x_{k+1} - x_k) - 2\Delta x(y_{k+1} - y_k)$$

but $x_{k+1} = x_k + 1$

$$p_{k+1} = p_k + 2\Delta y - 2\Delta x(y_{k+1} - y_k)$$

Algorithm :-

① $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$

② Calculate

$dx = x_2 - x_1$

$dy = y_2 - y_1$

③ Calculate decision parameter P

$P = 2dy - dx$

④ Set initial point

$x = x_1$, $y = y_1$

and $i = 0$

⑤ Plot (x, y)

if $P < 0$ then

$x = x + 1$

$P = P + 2dx$

else

$x = x + 1$

$y = y + 1$

$P = P + 2dy - 2dx$

⑥ Repeat ⑤ until $i <= dx$

Example :- Draw (20,10) and (30,18)

① (20,10) & (30,18)

$x_1 = 20$ $y_1 = 10$ $x_2 = 30$ $y_2 = 18$

② $dx = x_2 - x_1 = 10$

$dy = y_2 - y_1 = 8$

③ Decision parameter

$p = 2dy - dx$

$= 6$

④ Set (x, y)

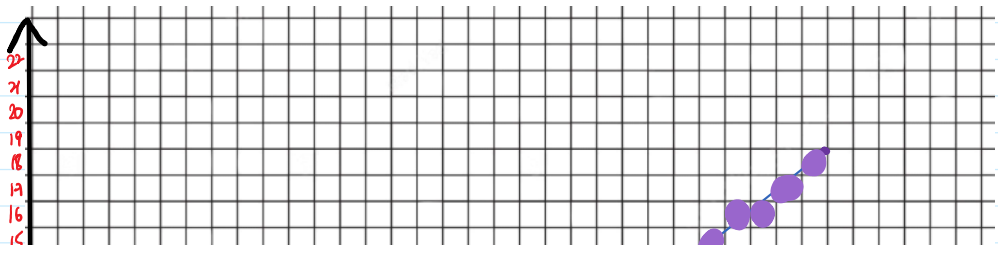
$x = x_1 = 20$

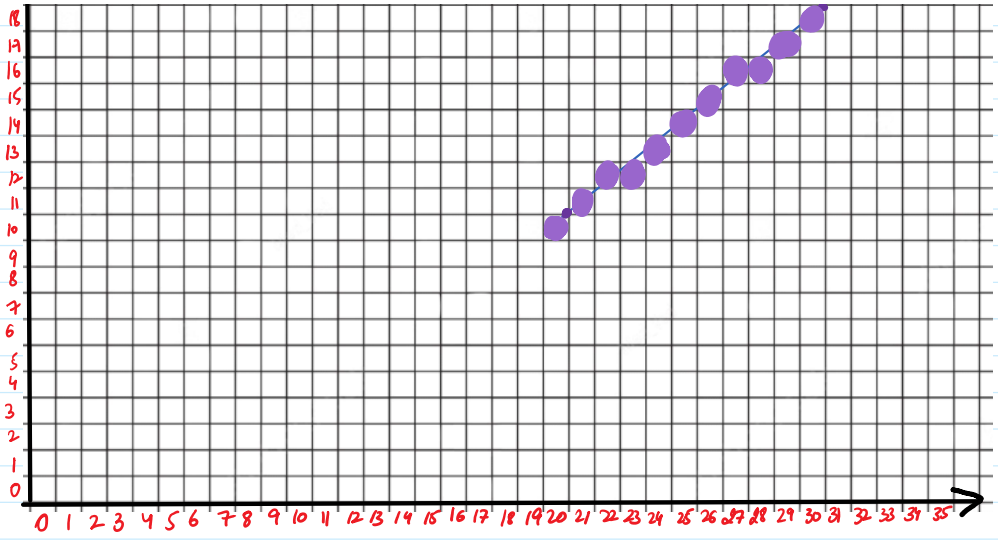
$y = y_1 = 10$

and $i = 0$

⑤ Execute until $i \leq dx$

i	p	Points
0		(20, 10)
1	6	(21, 11)
2	2	(22, 12)
3	−2	(23, 12)
4	14	(24, 13)
5	10	(25, 14)
6	6	(26, 15)
7	2	(27, 16)
8	−2	(28, 16)
9	14	(29, 17)
10	10	(30, 18)




18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35