SNS COLLEGE OF ENGINEERING

Kurumbapalayam(Po), Coimbatore – 641 916
Accredited by NAAC-UGC with 'A' Grade
Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

Department of Information Technology

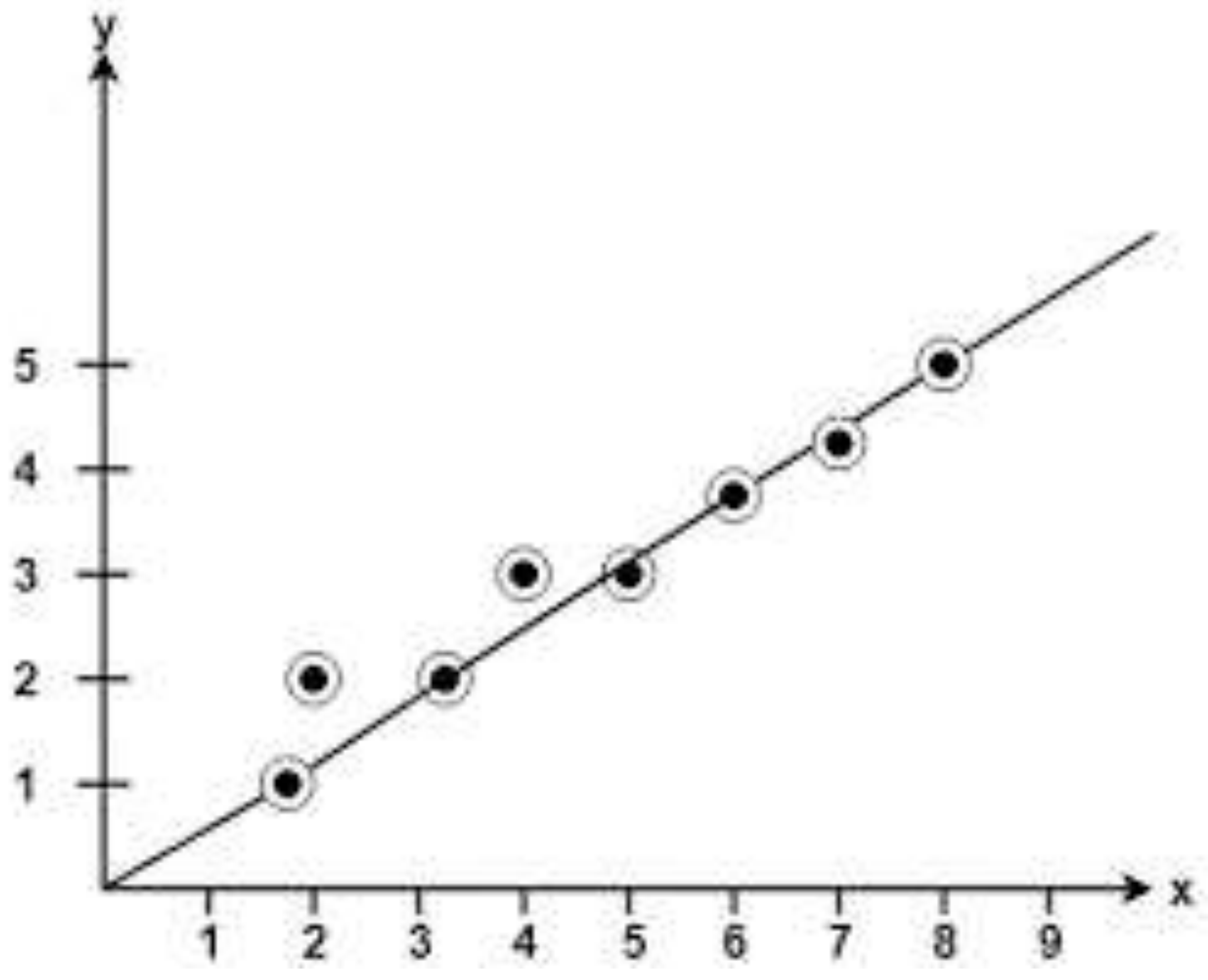
Course Name –Computer Graphics

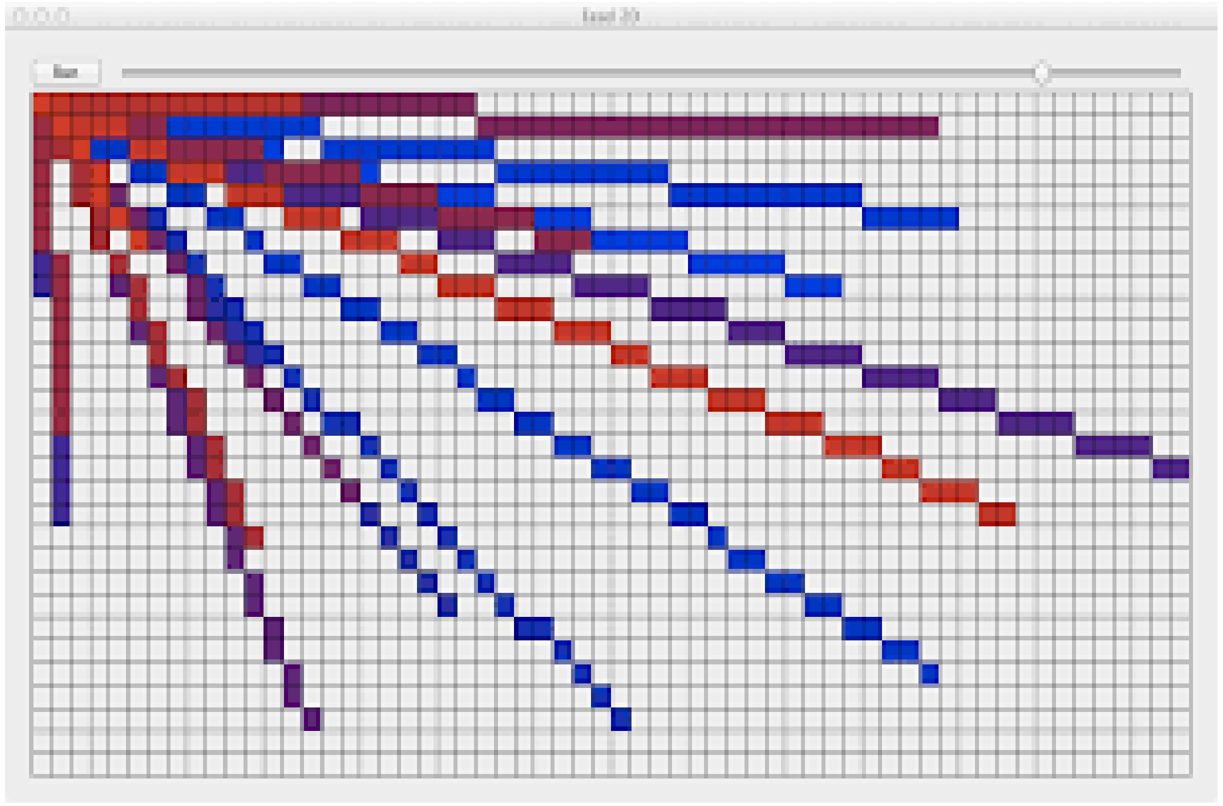
III Year / V Semester

Unit 1– INTRODUCTION TO COMPUTER GRAPHICS

Topic :Points and Lines , Line Drawing Algorithms

Relate the image to topic





Line

- A line in Computer graphics is a portion of straight line that extends indefinitely in opposite direction.
- It is defined by its two end points.
- Its density should be independent of line length.

The slope intercept equation for a line:

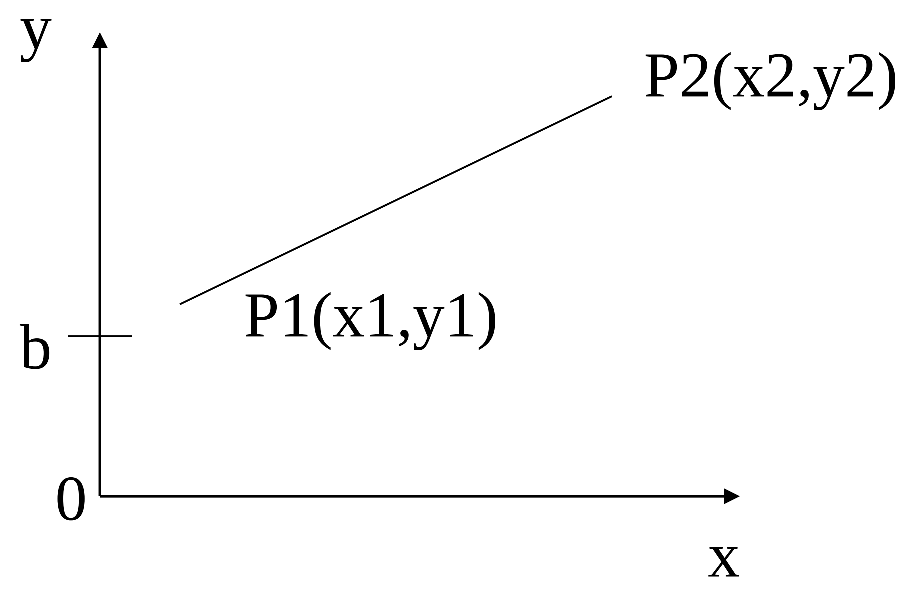
$$\mathbf{y = mx + b} \qquad (1)$$

where, **m** = Slope of the line

b = the y intercept of a line

Line Drawing Algorithm

The two endpoints of a line segment are specified at positions **(x1,y1)** and **(x2,y2).**



Line Drawing Algorithm

We can determine the value for slope m & b intercept as

m = y2-y1/x2-x1

i.e. m= Δy/ Δx (2)

Example:

The endpoints of line are(0,0) & (6,18). Compute each value of y as x steps from 0 to 6 and plot the result.

Solution : Equation of line is y= mx +b

m = y2-y1/x2-x1= 18-0/6-0 = 3

DDA Algorithm

- The Digital differential analyzer (DDA) algorithm is an incremental scan-conversion method.
- Such an approach is characterized by performing calculations at each step using results from the preceding step.

Algorithm

(x1,y1) (x2,y2) are the end points and dx, dy are the float variables.
Where dx= abs(x2-x1) and dy= abs(y2-y1)

```
(i)    If dx >=dy then
               length = dx
         else
               length = dy
         endif
ii) dx = (x2-x1)/length
    dy = (y2-y1)/length
(iii)x = x1 + 0.5
     y = y1 + 0.5
(iv)  i = 0
(v)Plot ((x), (y))
(vi) x = x + dx
      y = y + dy
(vii) i = i + 1
```

Algorithm

(viii) If i < length then go to step (v)
(ix) Stop

Example

x1	y1	x2	y2	L	dx	dy	i	x	y	Result	Plot
3	2	4	7	5	.2	1	0	3.5	2.5	3.5, 2.5	3,2
							1	3.7	3.5	3.7,3.5	3,3
							2	3.9	4.5	3.9,4.5	3,4
							3	4.1	5.5	4.1,5.5	4,5
							4	4.3	6.5	4.3,6.5	4,6
							5	4.5	7.5	4.5,7.5	4,7

Limitations of DDA

- The rounding operation & floating point arithmetic are time consuming procedures.
- Round-off error can cause the calculated pixel position to drift away from the true line path for long line segment.

Bresenham Line Algorithm

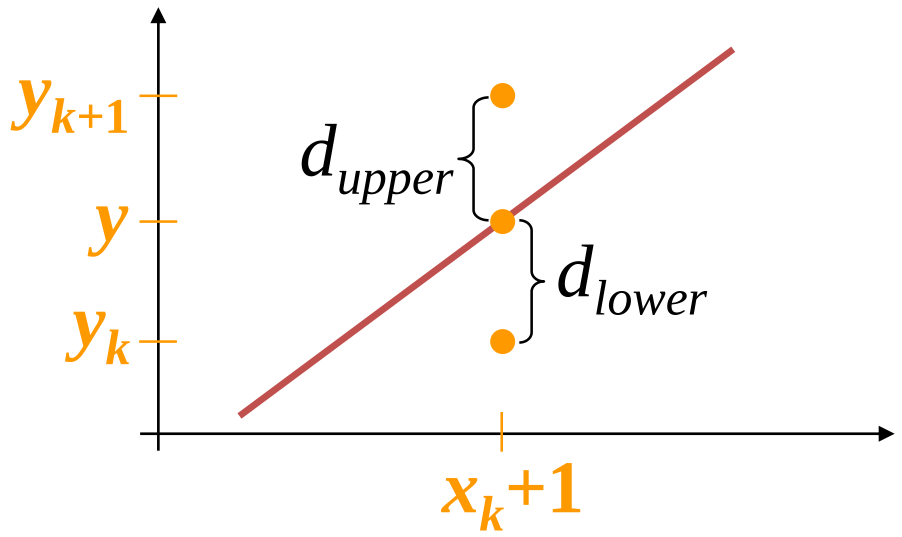
- The Bresenham algorithm is another incremental scan conversion algorithm
- The big advantage of this algorithm is that it uses only integer calculations

Deriving The Bresenham Line Algorithm

At sample position x_k+1 the vertical separations from the mathematical line are labelled d_{upper} and d_{lower}

The y coordinate on the mathematical line at x_k+1 is:

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



BRESENHAM’S LINE DRAWING ALGORITHM

1. Input the two line end-points, storing the left end-point in (x_1, y_1)
2. Calculate the constants Δ*x i.e. dx*, Δ*y i.e. dy*, 2Δ*y* and 2 Δx, get the first value for the decision parameter as

$$e = 2\Delta y - \Delta x$$

3. Initialize starting
4. Initialize i=1 as a counter, $e = e + 2\Delta y$

   Otherwise, the next point to plot is (x_k+*1*, y_k+*1*) and:

   $$p_{k+1} = p_k + 2\Delta y - 2\Delta x$$

5. Repeat step 4 (Δ*x* – 1) times

Adjustment

For m>1, we will find whether we will increment x while incrementing y each time.

After solving, the equation for decision parameter p_k will be very similar, just the x and y in the equation will get interchanged.

Bresenham Example

Let's plot the line from (20, 10) to (30, 18)

First off calculate all of the constants:

- Δx: 10
- Δy: 8
- $2\Delta y$: 16
- $2\Delta y - 2\Delta x$: -4

Calculate the initial decision parameter p_0:

- $p0 = 2\Delta y - \Delta x = 6$

THANK YOU