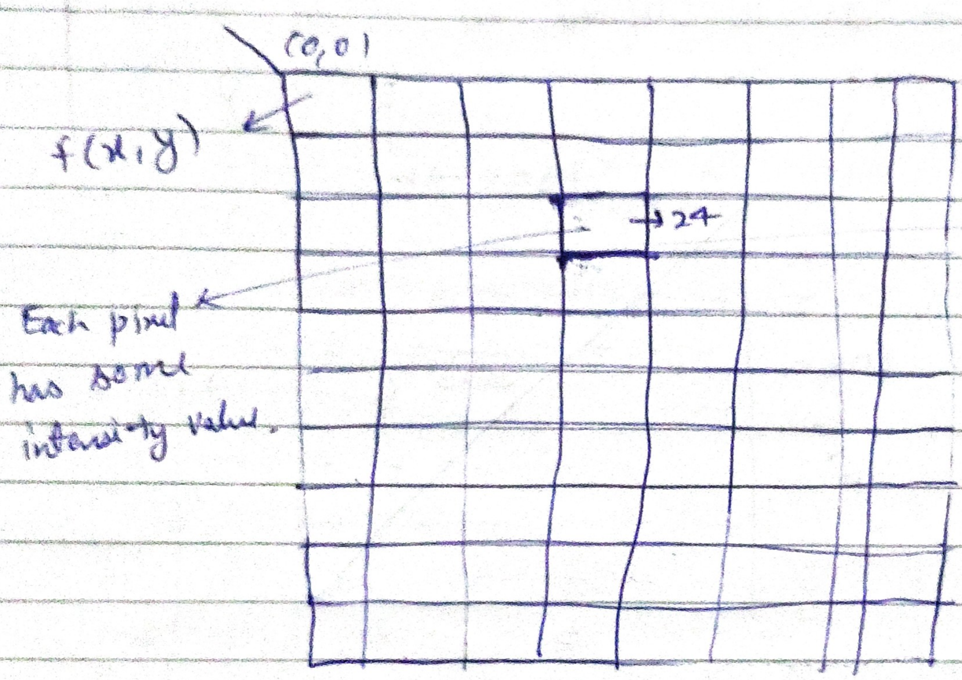
Computer Graphics

* It is the branch of engineering that deals with the generation and manipulation of an image.



Gray (0 - 255)

$2^3 + 2^4 = 24$

* Image is collection of pixels

Grey (0 - 255)
black white

Whenever an image is display on screen with the help of pixel values.

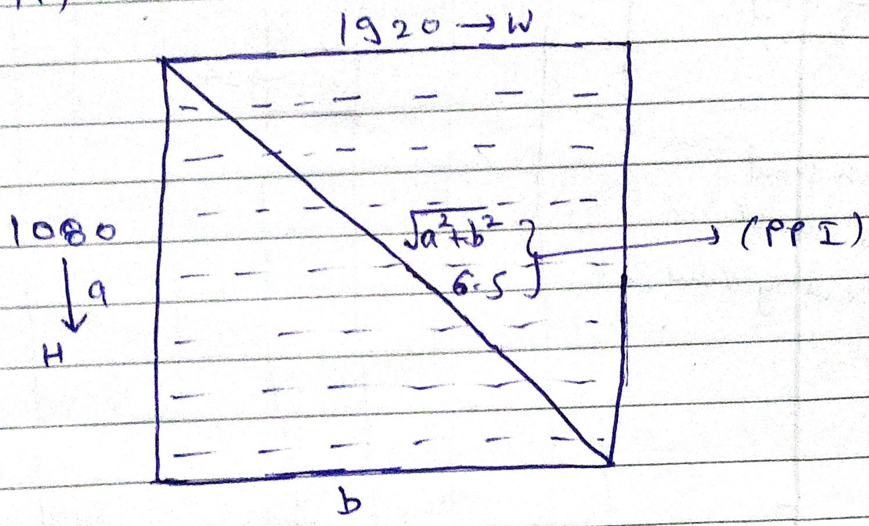
* Some basic technology in CGS are as follows :-

i) Pixel → It is the smallest unit of an image in monitor. It starts from the co-ordinate position (0,0) and It move from

Scanned with ACE Scanner

top to bottom and left to right. Each pixel value have some intensity value which display the image.

Resolution → It is defined as no. of pixels present as per unit area. It is generally denoted as (W × H)



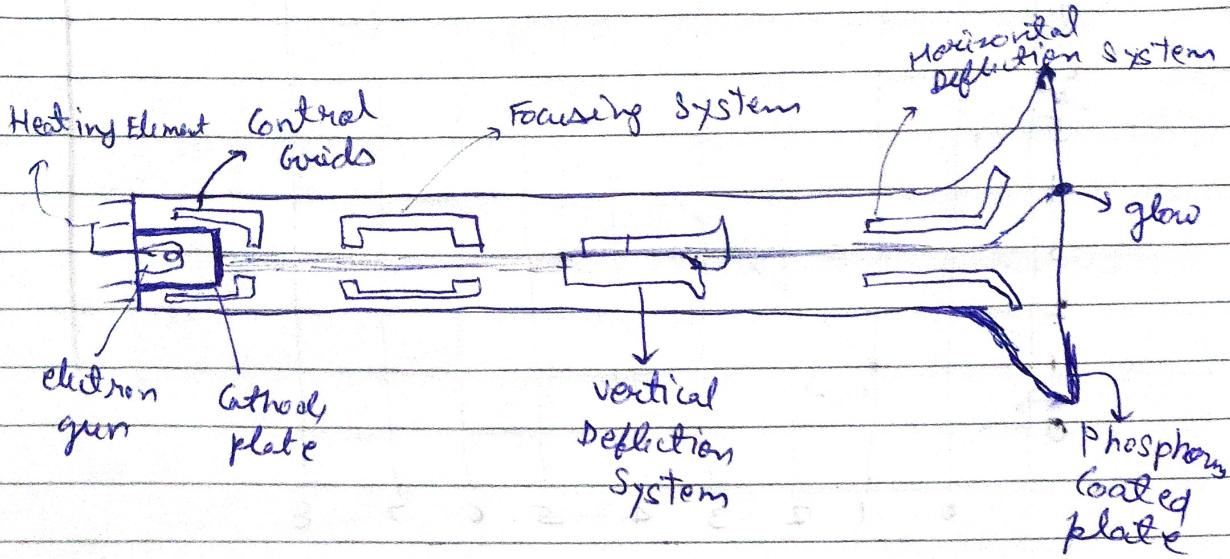
PPI → (Pixels per Inch)

Aspect Ratio → It is ratio of $\boxed{W/H}$

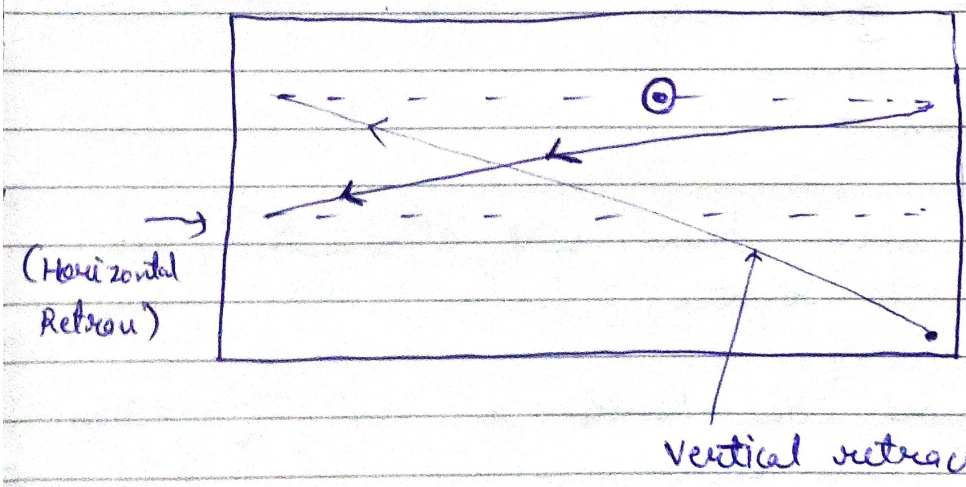
Persistence → The time it takes the emitted light from the screen to decay to one tenth of its original intensity.

Scanned with ACE Scanner

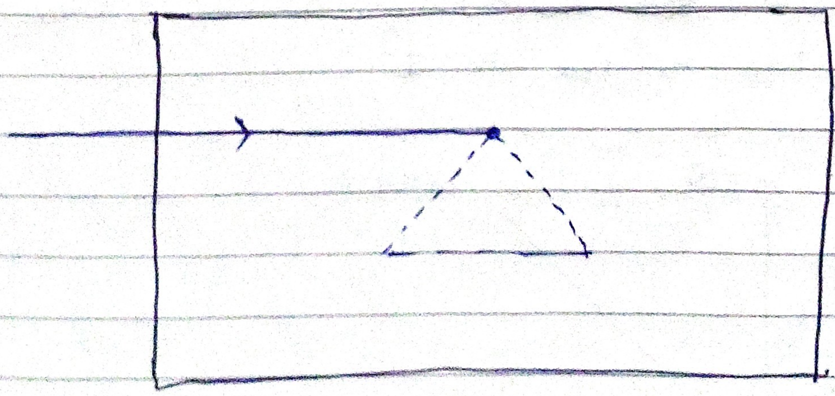
Cathode Ray Tube (CRT)



Raster Scan

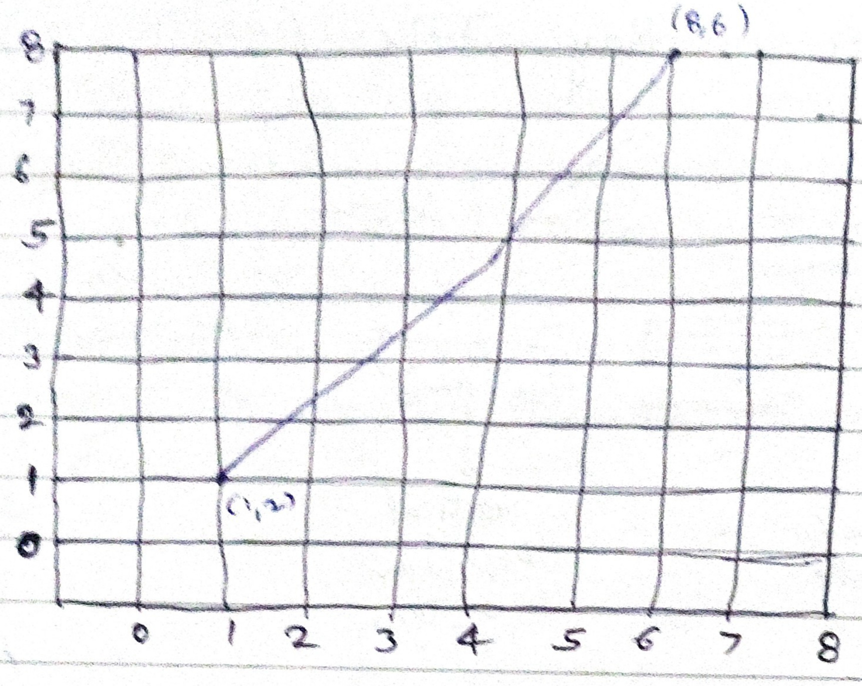


→ Random Scan → It is less time consuming than raster Scan.



Random display

Raster Scan and Sample -



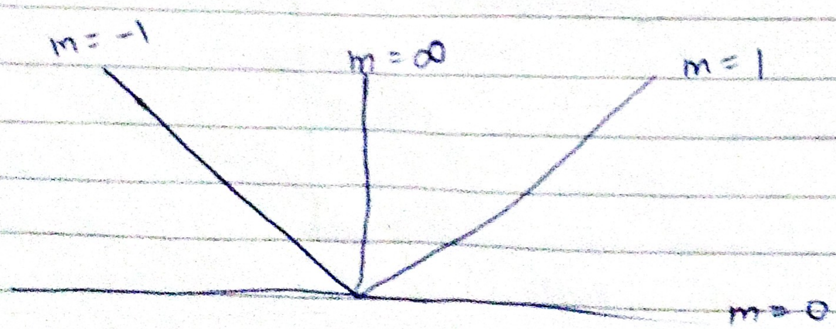
$y(3) = \frac{3}{5}(3)$

Raster Scan →

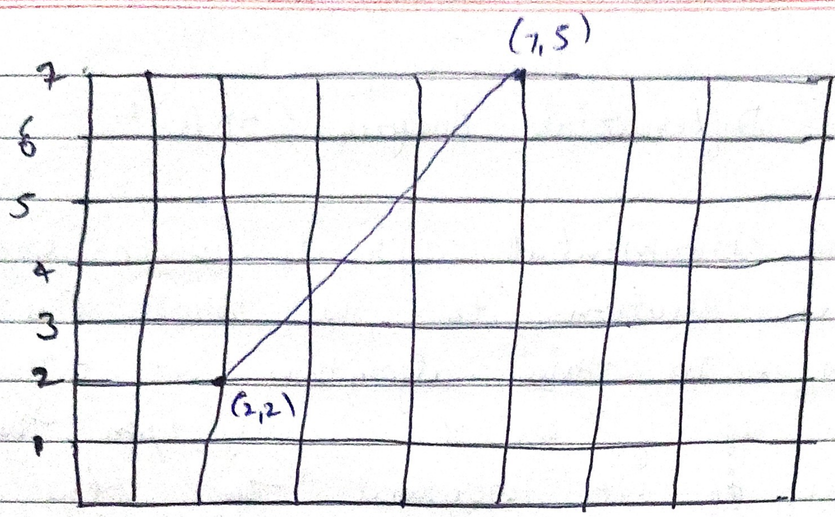
In raster scan beam will allow to move from left to right.

Memory used to handle on and off action is called frame buffer.

no. of horizontal ~~~~ retrace = no. of glace
no. of vertical retrace = 1



Scanned with ACE Scanner



$$m = \frac{5-2}{7-2} = 3/5$$

$$C = Y - mx$$

$$c = 2 - \frac{3}{5} \times 2$$

$$c = \frac{4}{5}$$

$$y = mx + C$$

$$Y(3) = \frac{3}{5} \times 3 + \frac{4}{5}$$

$$= \frac{13}{5} = 2.6 \approx 3$$

$$Y(4) = \frac{3}{5} \times 4 + \frac{4}{5} = \frac{16}{5} = 3.2 = 3$$

$$Y(5) = \frac{3}{5} \times 5 + \frac{4}{5} = \frac{19}{5} = 3.8 = 4$$

$$Y(6) = \frac{22}{5} = 4$$

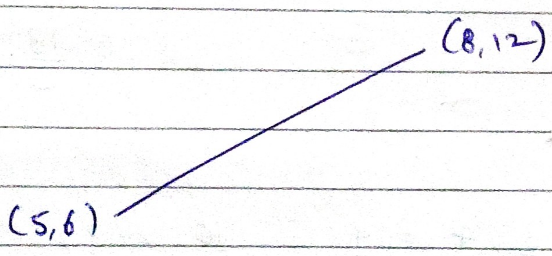
(3, 3)
(4, 3)
(5, 4)
(6, 4)
(7, 5)

Scanned with ACE Scanner

Digital Differential Analyzer (DDA) :-

It is a incremental approach which provides a faster Solution to the normal line drawing algorithms. In this algorithm we either increment the value of y or x depending upon the value of slope if we increment the value of x then we need to approximate the value of y.

Q → Using DDA draw a line starting from 5,6 and ending at 8,12.



Step 1 - Take initial point as $(x_1, y_1) \approx (5,6)$ and $(x_2, y_2) \approx (8,12)$

Step 2 - Calculate Δx and Δy

$$\Delta x = 8 - 5 = 3$$
$$\Delta y = 12 - 6 = 6$$

Step 3 - Set Δx and Δy as the step of length by using

$$\text{Step length} = \max(\Delta x, \Delta y)$$

Scanned with ACE Scanner

Step 4 - Find the slope $m = \frac{\Delta y}{\Delta x} = \frac{6}{3} = 2$

if $m>1$ x will be approximated and increment X

if $m<1$ or $(-1 + 01)$ y ~~we~~ will be approximate the value of y

Step 5 - Create a table that will store the different value of pixel position and round of the value.

	x_1	y_1	x_k	y_k	Round OF
	5	6			
$m=2>1$			5.5	7	(6,7)
			6	8	(6,8)
			6.5	9	(7,9)
			7	10	(7,10)
			7.5	11	(8,1)
			8	12	(8,12)

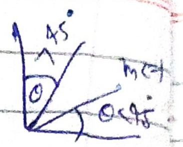
$y = mx + c$
$7 = 2x - 4$
$x = \frac{11}{2}$

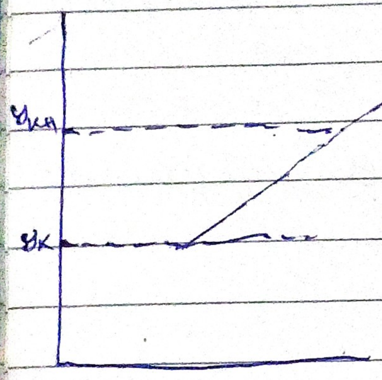
$y = mx + c$
$8 = 2x - 4$
$x = 6$

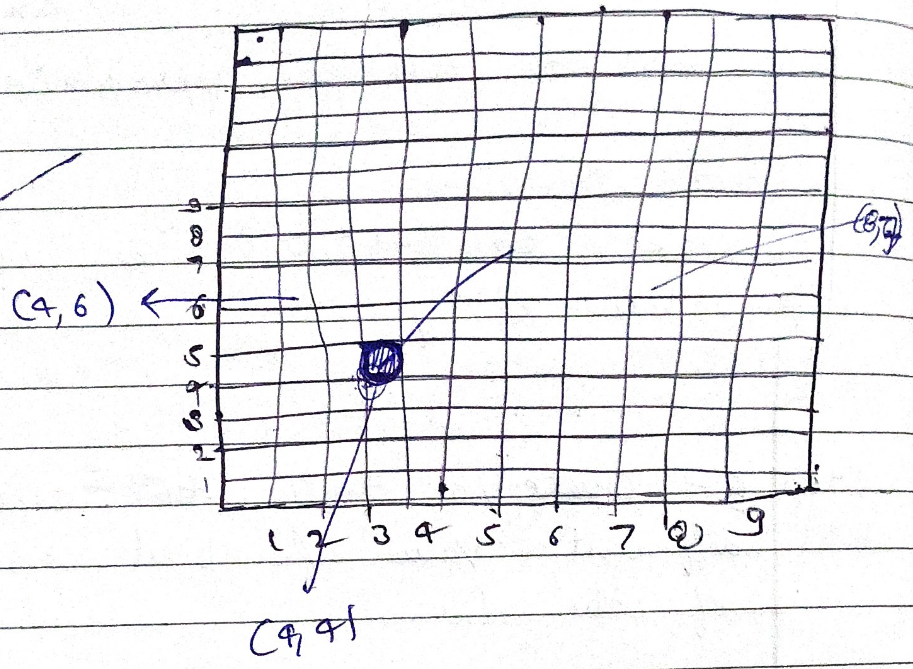
$y = mx + c$
$9 + 4 = 2x$
$x = \frac{13}{2} = 6.5$

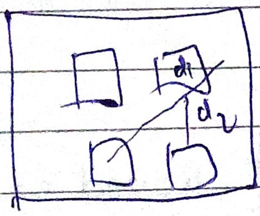
$y = mx + c$
$6 = 2 \times 5 + c$
$c = -4$

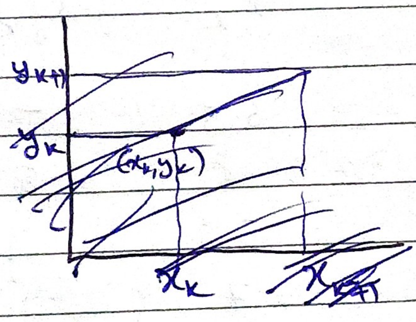
Breshanharg Line Drawing Algorithm

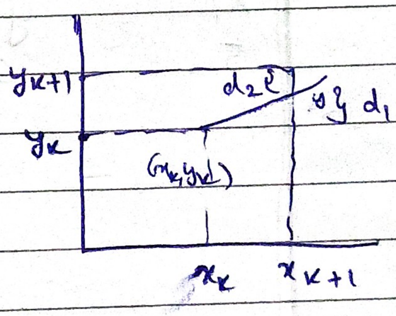












$$y = mx + b \quad -(1)$$
$$= m(x_k + 1) + b$$

$$d_1 = y - y_k \quad -(2)$$
$$d_2 = y_k + 1 - y \quad -(3)$$

$$d_1 = m(x_{k+1}) + b - y_k$$
$$d_2 = y_k + 1 - m(x_{k+1}) - b$$

$$d_1 - d_2 = m(x_{k+1}) + \cancel{b} - y_k - y_k + 1 + m(x_k + 1) \cancel{+b}$$

$$\cancel{\boxed{d_1 - d_2 = 2m(x_{k+1}) - y_k - y_{k+1}}}$$

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

~~by multiply~~

$$d_1 - d_2 = 2\frac{\Delta y}{\Delta x}(x_k+1) - 2y_k + 2b - 1$$

$$\Delta x(d_1-d_2) = 2\Delta y(x_k+1) - 2y_k\Delta x + \overbrace{\boxed{2b\Delta x - \Delta x}}^{c}$$

$$\Delta x(d_1-d_2) = 2\Delta y\, x_k - 2\Delta x\, y_k + c \quad \text{— (4)}$$

Let us assume that $\Delta x(d_1-d_2)$ is taken as decision parameter p_k so we need to find out the next decision parameter p_{k+1} as p_{k+1}

$$p_{k+1} = 2\Delta y\, x_{k+1} - 2\Delta x\, y_{k+1} + c \quad \text{— (5)}$$

Here the parameter c is constant which is equal to $c = 2\Delta y + \Delta x(2b-1)$ is independent therefore the pixel at y_k is closure to the line path then the pixel at y_k+1 since $d1 < d2$ next pixel position is y_k+1

In that case we plot the lower pixel.

We obtain the value of successive decision parameter by using incremental integer calculation

$$p_{k+1} - p_k = 2\Delta y(x_{k+1} - x_k) - 2\Delta x(y_{k+1} - y_k) \quad \text{— (6)}$$

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

~~Case — (1)~~

~~If $p_k < 0$, Next point will be~~

~~$\therefore p_{k+1} = p_k + 2\Delta y$~~

Scanned with ACE Scanner

Case 1:-

if $P_k < 0$ Next point will be (x_{k+1}, y_k).

$$P_{k+1} = P_k + 2\Delta y - 2\Delta x(y_k - y_k)$$

$$\boxed{P_{k+1} = P_k + 2\Delta y}$$

Case 2:-

if $P_k > 0$, Next point (x_{k+1}, y_{k+1})

$$P_{k+1} = P_{k+2} - 2\Delta x$$

$$P_0 = 2\Delta y - \Delta x$$

~~step~~ The first decision Parameter P_0 will be obtain with the help of starting pixel position (x_0, y_0) and slope $m = \frac{\Delta y}{\Delta x}$ and

it is calculated as $p_0 = 2\Delta y - \Delta x$

$$p_k = 2\Delta y \cdot x_k - 2\Delta x \cdot y_k + 2\Delta y + \Delta x \cdot (2b-1)$$

$$p_k = 2\Delta y \cdot x_k - 2\Delta x \cdot y_k + 2\Delta y + \Delta x \cdot (2b-1).$$

$$P_0 = 2\Delta y \cdot x_0 - 2\Delta x \cdot y_0 + 2\Delta y + \Delta x(2b-1)$$

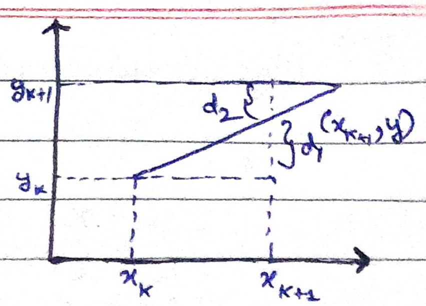
$$\boxed{p_0 = 2\Delta y - \Delta x}$$

Scanned with ACE Scanner

$y = mx + b = m(x_k + 1) + b$ — ①

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

$d_2 = y_{k+1} - y = y_k + 1 - m(x_k + 1) - b$ — ③



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

$= \frac{\Delta y}{\Delta x}(x_k + 1) + b - y_k - y_k - 1 + \frac{\Delta y}{\Delta x}(x_k + 1) + b$

Assume $\underbrace{\Delta x(d_1 - d_2)}_{P_k} = 2\Delta y x_k - 2\Delta x y_k + 2\Delta y + \Delta x(2b - 1)$

$P_k = 2\Delta y x_k - 2\Delta x y_k + 2\Delta y + \Delta x(2b-1) \rightarrow c$

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

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

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

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

Case 1 $\boxed{\text{if } P_k < 0}$ $y_{knew} = y_k$

Next Point (x_{k+1}, y_k)

$\boxed{P_{k+1} = P_k + 2\Delta y}$

Case 2 if $P_k > 0$, Next Point (x_{k+1}, y_{k+1})

$\boxed{P_{k+1} = P_k + 2\Delta y - 2\Delta x}$

From eq ⑤

$P_k = 2\Delta y x_k - 2\Delta x y_k + 2\Delta y + \Delta x(2b - 1)$ — ⑤

$P_0 = 2\Delta y x_0 - 2\Delta x y_0 + 2\Delta y + \Delta x(2b - 1)$

$2\Delta y x_0 - 2\Delta x y_0 + 2\Delta y + \Delta x(2$

$y = mx + b$

$y_0 \quad y = \frac{\Delta y}{\Delta x} x + b$

$b = \frac{y\Delta x - x\Delta y}{x}$

$y = mx + b \rightarrow \frac{\Delta y}{\Delta x} x + b \qquad y\Delta x - x\Delta y - b\Delta x = 0$

$b\Delta x + x\Delta y - y\Delta x = 0$

$\boxed{P_0 = 2\Delta y - \Delta x} \rightarrow$ Initial Decision Parameter

Scanned with ACE Scanner

Algo Bresenham line (x_1, y_1, x_2, y_2)

$\Delta x = 7$, $\Delta y = 4$ $\quad P_1(1,1)$, $P_2(8,5)$

```
{
    x = x1 ;
    y = y1 ;
    Δx = x2 - x1 ;
    Δy = y2 - y1 ;
     P0 = 2Δy - Δx ;
    while (x <= x2)
    {
     putpixel (x, y) :
       x++ ;
       if (P < 0)
      {
        Pk+1 = Pk + 2Δy
      }
      else {
          Pk+1 = Pk + 2Δy - 2Δx
          y++ ;
      }
    }
}
```

x	y	p
1	1	1
2	2	-5
3	2	3
4	3	-3
5	~~4~~ 3	5
6	4	-1
7	4	7
8	5	1

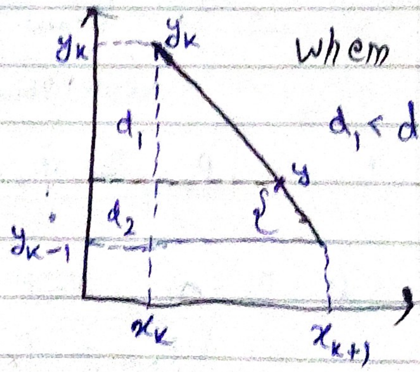
→ $\Delta x = 8 - 1 = 7$

$\Delta y = 5 - 1 = 4$

$P_0 = 2+4-7 = 1$

$P_{k+1} = P_k + 8$ [for $P_k < 0$]

$P_{k+1} = P_k + (-6)$ [for $P_k > 1$]



when $m < 0$

$d_1 < d_2$

$y = mx + b \longrightarrow y = m(x_k + 1) + b$ — (1)

$d_1 = y_k - y$ — (2)

$d_2 = y - y_{k-1}$ — (3)

$(d_1 - d_2) < 0 \longrightarrow (x_{k+1}, y_k)$

$(d_1 - d_2) > 0 \longrightarrow (x_{k+1}, y_{k-1})$

Scanned with ACE Scanner

$2\sigma_1 y^2 x > 2\sigma_1 x^2 y$

$$d_1 - d_2 = (y_k - y) - (y - y_{k-1})$$
$$= y_k - y - (y - (y_k - 1))$$
$$= y_k - y - y - y_k + 1$$
$$= 2y_k - 2y - 1$$
$$= 2y_k - 2(mx_k + m + b) - 1 \quad \text{[From eq-1]}$$
$$= 2y_k - 2mx_k - 2m - 2b - 1$$

Slope $= -ve = -\frac{\Delta y}{\Delta x}$

$$d_1 - d_2 = 2y_k + 2\frac{\Delta y}{\Delta x}x_k + 2\frac{\Delta y}{\Delta x} - 2b - 1 \quad \text{— 4}$$

$$\underbrace{\Delta x(d_1 - d_2)}_{P_k} = 2y_k \Delta x + 2\Delta y x_k + \underbrace{2\Delta y - \overset{2b\Delta x}{\cancel{2b\Delta x}} - \Delta x}_{c}$$

$$P_k = 2\Delta x y_k + 2\Delta y x_k + c$$

$$P_{k+1} = 2\Delta x y_{k+1} + 2\Delta y x_{k+1} + c$$

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

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

Case 1 → If $P_k < 0$, then $P_{k+1} = P_k + 2\Delta y$

Case 2 → If $P_k \geq 0$, then $P_{k+1} = P_k + 2\Delta y - 2\Delta x$

and the initial decision parameter $\boxed{P_0 = 2\Delta y - \Delta x}$

Scanned with ACE Scanner

Algorithm DDA (x_1, y_1, x_2, y_2)

```
{
    x = x1
    y = y1
    dx = x2 - x1
    dy = y2 - y1
    if (abs(dx) > abs(dy))
        step = abs(dx)
    else
        step = abs(dy)
    xinc = dx / Step

    yinc = dy / Step

    for (i=1; i<=Step ; i++)
    {
        putpixel (x1, y1)
        x1 = x1 + xinc
        y1 = y1 + yinc
    }
}
```

(5,4) - (12,7)   (5,7)   (10,15)

$m<1$, P_2

$\Delta x = 12-5 = 7$

$\Delta y = 7-4 = 3$

Step $= 7$, $m = \frac{3}{7}$

x	y	y round
5	4	4
6	4.4	4
7	4.8 $4+\frac{3}{7}$	5
8	$4.8+\frac{3}{7}$ 5.2	5
9	$5.2+\frac{3}{7}$ 5.6	6

$m>1$

$\Delta x = 10-5 = 5$

$\Delta y = 15-7 = 8$

$m = \frac{8}{5}$ Step $= 8$

x round	x	y
5	5	7
6	5.6	8
6	6.2	9
7	6.8	10
7	7.4	11

$m=1$ (12,9) (17,14)

$\Delta x = 17-12 = 5$

$\Delta y = 14-9 = 5$

$m = 1$ Step $= 5$

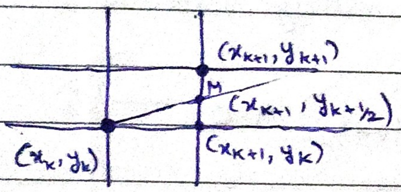
Scanned with ACE Scanner

x	y	y_{round}	x_{rand}	x	y
10	$\frac{5.6+3/7}{6}$	6	8	8	12
11	~~5.06~~ $6+3/7$	6	9	8.6	13
			9	9.2	14
12	6.7	7	10	9.8	15

→ Mid point line drawing algorithm : Assumption

(x_0, y_0)

$0<m<1$, $x_2<x_1$ and $y_2>y_1$



For a given 2 points $P1(x_0, y_0)$ and $P2(x_1, y_1)$ We have to find all the intermediate points ~~to~~ to draw the line assume ~~that~~ the slope lies b/w 0 and 1 since every pixel has integer coordinate so from the current pixel position $P(x_k, y_k)$ we will find the next closest pixel from either (x_{k+1}, y_{k+1}), (x_{k+1}, y_k). The ~~middle~~ coordinate of the middle points ~~is~~ will be $M\left(\frac{x_{k+1}+x_k+1}{2}, \frac{y_{k+1}+y_k}{2}\right)$

If the mid point M is above the line the next point will be (x_{k+1}, y_k). If the mid point below the line the next point will be (x_{k+1}, y_{k+1}).

Intersect ~~form~~ eqn form of line is given by $y = mx + b$

Implicit ~~eqn~~ of line $\Rightarrow ax + by + c = 0$ — ①

$y = \frac{dy}{dx} x + b$

$dy(x) - dx(y) + B(dx) = 0$ — ②

comparing ① and ②, we get a =

$a = dy$

$b = -dx$

$c = B(dx)$

1) ~~If~~ A given point $P(x, y)$ passes through a line then $f(x, y) = 0$

$F(x, y) = ax + by + c$

2) A given point $P(x,y)$ passes through above the line then $f(x,y) < 0$.

3) A given point $P(x,y)$ passes through below the line then $f(x,y) > 0$.

Assume that decision parameter d_k is $F(x,y)$

The above 3 conditions will be check on the mid point

Case 1: If the point M is above the line then next point will be (x_{k+1}, y_k) i.e ~~dk~~ $d_k < 0$

$$dk = F(x,y) = F(x_{k+1}, y_{k+\frac{1}{2}})$$

$$= a(x_{k+1}) + b(y_{k+1/2}) + c \quad \text{---} (3)$$

$$dk+1 = F(x_{k+1+1}, y_{k+1+1/2})$$

$$= F(x_{k+2}, y_{k+1/2})$$

$$= a(x_k+2) + b(y_k+\frac{1}{2}) + c \quad \text{---} (4)$$

$$= (4) - (3) \Rightarrow d_{k+1} - d_k = a(x_k+2) + b(y_k+\frac{1}{2}) + c - a(x_{k+1}) - b(y_k+\frac{1}{2}) - c$$

$$a \Rightarrow d_{k+1} = d_k + dy$$

$$d_{k+1} = d_k + a$$

Case 2 → If the point M is below the line, then next point will be (x_{k+1}, y_{k+1}) i.e. $d_k > 0$.

Here decision parameter d_k will be

$$d_k = F(x,y) = F(x_{k+1}, y_{k+1})$$

$$= a(x_{k+1}) + b(y_{k+\frac{1}{2}}) + c \quad \text{---} (5)$$

$$d_{k+1} = F(x_{k+1+1}, y_{k+1+\frac{1}{2}}) + c \quad \text{---} (6)$$

$$= a(x_k + 2) + b(y_{k+1+\frac{1}{2}}) + c \quad -(6)$$

$$(6) - (5)$$

$$d_{k+1} = d_k + a + b$$

$$d_{k+1} = d_k + a + b \Rightarrow \boxed{d_{k+1} = d_k + dy - dx}$$

Initial Decision parameter $d_0 = F(x_0 + 1, y_0 + \frac{1}{2})$

$$d_0 = ax_0 + by_0 + a + \frac{b}{2} + c$$

$$d_0 = \underline{ax_0 + by_0 + c} + a + \frac{b}{2}$$

$$\boxed{d_0 = a + \frac{b}{2}}$$

$$\boxed{d_0 = dy - \frac{dx}{2}}$$

Q → Using the mid point Line drawing algorithm calculate all the intermediate point starting point P1 (20, 10) to the ending point P2 (30, 18).

Ans) $(x_0, y_0) = (20, 10)$ | Step 1 : $dx = 30 - 20$

$(x_n, y_n) = (30, 18)$ | $dy = 18 - 10$

Step 2 : Initial decision parameter $d_0 = dy - \frac{dx}{2}$

$$= 8 - \frac{10}{2} = 3$$

Dinitial	Dnew	x_{k+1}	y_{k+1}
-	-	20	10
3.	1	21	11
1	-1	22	12
-1	7	23	12
7	5	24	13
5	3	25	14

$d_k > 0$ x_{k+1}, y_{k+1}

$d_{k+1} = d_k + dy - dx$

$= 3 + 8 - 10$

$= +1$

$1 + 8 - 10$

$= -1$

Scanned with ACE Scanner

$d_k < 0 \quad (x_{k+1}, y_k) \rightarrow d_{k+1} = d_k + dy$

Draw a line after

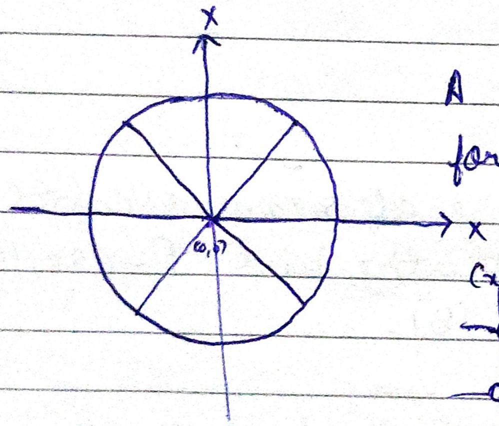
3	1	26	15
1	-1	27	16
7	7	28	16
7	5	29	17
5	3	30	18

$\frac{7}{7} = 1$

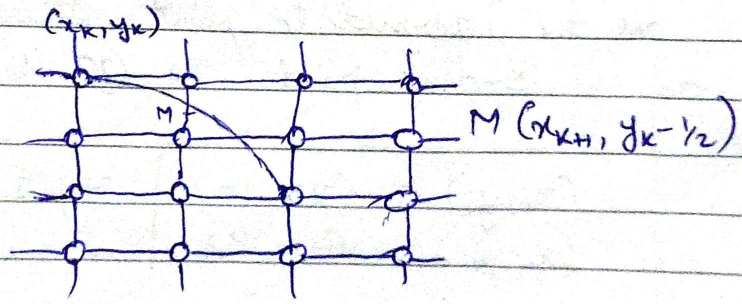
$7 - \frac{7}{2} \Rightarrow \frac{7}{2}$

Q → For 2 given points (5, 9) and (12, 16) calculate all the intermediate points using mid point line algorithm.

Mid Point Circle Drawing Algorithm



A circle posses 8-way symmetry property for a given radius r and center (0, 0).



According to the symmetry property of a circle whenever we consider 1 point in 1 octant then we can get the other point in the next octant with the help of previous octant. In this way we will calculate the pixel position in all the octant.

General eqn of circle

$f(x, y) = x^2 + y^2 - r^2$

if $f(x,y) = 0$ pt lies on the circle
" $f(x,y) < 0$ " " inside the circle
" $f(x,y) > 0$ " " outside the circle

Mid pt $= (x_k+1, y_k - 1/2)$.
Assume Decision parameter $P_k = f(x_k+1, y_k+1)$ we have to check whether the mid point lies on the circle, inside, outside the circle.

$$P_k = f(x_k+1, y_k - 1/2) = (x_k+1)^2 + (y_k - 1/2)^2 - r^2 \quad -(1)$$

mid point

if $P_k < 0$ $f(x_k+1, y_k - 1/2) < 0$ lies inside the circle

$x_{k+1} = x_k + 1$ ~~$x_{k+1} = x_k+1$~~ (x_k+1, y_k)

$y_{k+1} = y_k$ ~~$y_{k+1} = y_k$~~

mid point

if $P_k > 0$ $f(x_k+1, y_k - 1/2) > 0$ lies outside the circle

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

$y_{k+1} = y_k)$

We need to calculate the successive decision parameter from eq $-(1)$

$$P_{k+1} = (x_{k+1}+1)^2 + (y_{k+1} - 1/2)^2 - r^2 \quad -(2)$$

(2) $-$ (1)

$$P_{k+1} - P_k = (x_{k+1}+1)^2 + (y_{k+1} - 1/2)^2 - r^2 - (x_k+1)^2 - (y_k - 1/2)^2 + r^2$$

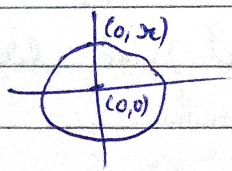
$$= (x_k+1)^2 + 1 + 2(x_k+1) + y_k + 1^2 + \frac{1}{4} - 2 \cdot \frac{1}{2}(y_k+1)$$

$$- (x_k+1)^2 - \cancel{2(x_k+1)} - y_k + 1^2 + 2(y_k+1)\frac{1}{2} - \frac{1}{4}$$

$1 + 1 + \frac{1}{4} - 1 - \frac{1}{4}$

$$P_{k+1} - P_k = (x_k+2)^2 + 1 + 2(x_k+1) + y_k + 1^2 + \frac{1}{4} - 2\cdot\frac{1}{2}(y_k+1) + (x_k+1)^2 - (y_k - 1/2)$$

$$P_{k+1} = P_k + 1 + 2x_{k+1} + \left((y_{k+1})^2 - y_{k+1}\right) - (y_k^2 - y_k)$$



The initial decision parameter P_0 will be

$$P_0 = f_c(0, r) \rightarrow P_0 = f(x_k+1, y_k - 1/2) = (x_k+1)^2 + (y_k - 1/2)^2 - r^2$$
$$= (0+1)^2 + (r - 1/2)^2 - r^2$$

$$P_0 = 1 - r + \frac{1}{4} = \frac{5}{4} - r$$

$$P_0 = \frac{5}{4} - r$$

if the radius given as integer value then $\frac{5}{4} \approx 1$

Example $P_0 = \frac{5}{4} - r$

$$P_k = (x_{k+1}+1)^2 + (y_k - 1/2)^2 - r^2 = (x_{k+1}, y_k)$$

$$P_{k+1} = P_k + 1 + 2(x_k+1) + (y_{k+1}^2 - y_k^2) - (y_{k+1} - y_k)$$

Case 1 → When $P_k < 0$, $P_{k+1} = P_k + 1 + 2(x_k+1)$

$x_{k+1} = x_k + 1$
$y_{k+1} = y_k$

Case 2 → When $P_k > 0$ $P_{k+1} = P_k + 1 + 2(x_k+1) - ((y_{k-1})^2 - y_k^2) - (y_{k+1} - y_k)$

$x_{k+1} = x_k + 1$
$y_{k+1} = y_k - 1$

Scanned with ACE Scanner

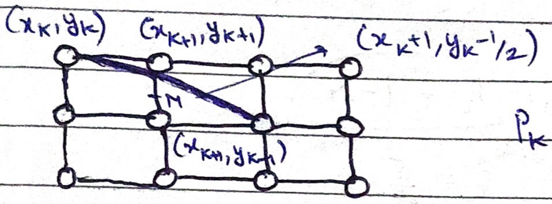
Imp:-

$$P_{k+1} = P_k + 1 + 2(x_k+1) + (y_k^2 - 2y_k + 1 - y_k^2) + 1$$

$$P_{k+1} = P_k + 1 + 2(x_k+1) + 1 - 2y_k + 1$$

$$= P_k + 1 + 2(x_k+1) - 2y_k + 1$$

$$P_{k+1} = P_k + 1 + 2(x_k+1) - 2y_{k+1}$$

→ 

$$P_k = f_c(x_k+1, y_k - 1/2)$$
$$= (x_k+1)^2 + (y_k - 1/2)^2 - r^2 \quad \text{--- (1)}$$

Successive decision parameter will P_{k+1}, $x_k \to x_{k+1}$, $y_k \to y_{k+1}$

$$P_{k+1} = (x_{k+1}+1)^2 + (y_{k+1} - 1/2)^2 - r^2 \quad \text{--- (2)}$$

(2) − (1)

$$P_{k+1} - P_k = (x_k+1)^2 + 2(x_k+1)\cdot 1 + 1 - (x_k+1)^2 + (y_{k+1} - 1/2)^2 - (y_k - 1/2)^2 - r^2 + r^2$$
$$= 1 + 2x_k + 1 + y_{k+1}^2 - y_{k+1} + \frac{1}{4} - y_k^2 + y_k - \frac{1}{4}$$

$$= 1 + 2(x_k+1) + y_{k+1}^2 - y_k^2 + y_k$$

when $P_k < 0$, $y_{k+1} \to y_k$, $x_{k+1} \to x_k + 1$ → $\boxed{P_{k+1} = P_k + 1 + 2(x_k+1)}$

$$P_{k+1} - P_k = 1 + 2(x_k+1) + y_k^2 - y_k - y_k^2 + y_k$$

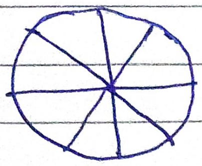
When $P_k > 0$, $y_{k+1} \to y_k - 1$, $x_{k+1} \to x_k + 1$] 0

$$P_{k+1} = P_k + 1 + 2(x_{k+1}) + (y_k - 1)^2 - (y_k - 1) - y_k^2 + y_k$$

$$\boxed{P_{k+1} = P_k + 2(x_{k+1}) + 1 - 2y_{k+1}}$$

Initial Decision parameter P_0 will be obtained by taking

$x = 0,\ y = r,\quad f_c(x,y) = (x_k+1)^2 + (y_k - 1/2)^2 - r^2$



$f_c(0,r) = (0+1)^2 + (r - 1/2)^2 - r^2$

$f_c(0,r) = 1 + \cancel{r^2} + \frac{1}{4} - r - \cancel{r^2}$

$$\boxed{f_c(0,r) = \frac{5}{4} - r} = P_0$$

if ~~r~~ r is integer $P_0 = 1 - r$

else $P_0 = \frac{5}{4} - r$

Ex → Mid point circle drawing algorithm draw a circle for a given radius $r = 10$ and center $(0,0)$

Ans Rough →

$P_0 = 1 - 10 = -9$

$X_0 = 0$

$y_0 = 10$

K	P_{k+1}	X_{k+1}	y_{k+1}
0	-9	1	10
1	-6	2	10
2	-1	3	10
3	6	4	9
4	-3	5	9
5	8	6	8
6	5	7	7
[illegible]	[illegible]	[illegible]	[illegible]

$P_1 = -9 + 2 + 1$

$P_1 = -6$

$P_2 = -6 + 4 + 1$

$P_2 = -2 + 1$

$P_3 = -1 + 6 + 1$

$= 6$

$P_4 = 6 + 8 + 1 - 18$

$= -3$

$P_5 = -3 + 10 + 1$

$= 8$

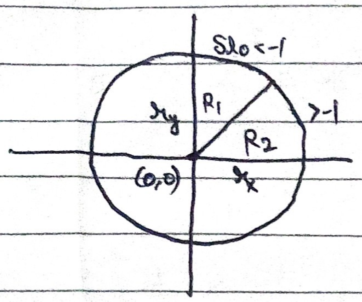
$P_6 = 8 + 12 + 1 - 18$

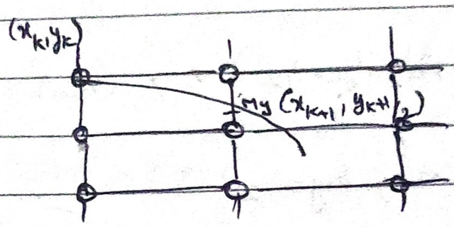
$= 5$

$P_7 = 5 + 14 + 1 - 16$

$= 5 + 1 - 2 = 5 - 1$

~~Algorithm~~ Ellipse Drawing using Mid point algorithm





x_{k+1}, y_{k+1} → x_{k+1}, y_k ; x_{k+1}, y_{k-1}

$$\frac{x^2}{r_x^2} + \frac{y^2}{r_y^2} - 1 = 0$$

$P_k = f(x,y) =$

$\rightarrow x^2 r_y^2 + y^2 r_x^2 - r_x^2 r_y^2 = 0$ — ①

General Equation of an Ellipse → $\frac{x^2}{r_x^2} + \frac{y^2}{r_y^2} - 1 = 0$ — ①

In Region R_1 → Mid point $= \left(\frac{x_{k+1} + x_k + 1}{2}, \frac{y_k + y_k - 1}{2}\right)$

$= \left(x_k + 1, y_k - \frac{1}{2}\right)$

$P_k = (x_{k+1})^2 r_y^2 + (y_k - \frac{1}{2})^2 r_x^2 - r_x^2 r_y^2$ — ②

$P_{k+1} = (x_{k+1} + 1)^2 r_y^2 + (y_{k+1} - \frac{1}{2})^2 r_x^2 - r_x^2 r_y^2$ — ③

$$P_{k+1} - P_k = (x_{k+1} + 1)^2 r_y^2 - (x_k + 1)^2 r_y^2 + (y_{k+1} - \tfrac{1}{2})^2 r_x^2 - (y_k - \tfrac{1}{2})^2 r_x^2 - r_x^2 r_y^2 + r_x^2 r_y^2$$

$$= r_y^2 \{(x_k+1)^2 + 2\cdot(x_k+1)\cdot 1 + 1 - (x_k+1)^2\} + r_x^2 \{y_{k+1}^2 - y_k^2 - y_{k+1} + y_k\}$$

$$= r_y^2 [2\cdot(x_k+1) + 1] + r_x^2 [y_{k+1}^2 - y_k^2 - y_{k+1} + y_k]$$

$P_k < 0$, $y_{k+1} \rightarrow y_k$

$P_{k+1} = P_k + r_y^2 [(2x_{k+1}) + 1]$

$$\boxed{P_{k+1} = P_k + r_y^2 [(2x_{k+1}) + 1]}$$

$P_k \geq 0$ $y_{k+1} = y_k - 1$

$P'_{k+1} = P'_k + 2x_{k+1} r_y^2 + r_y^2 + 2r_x^2(1 - y_k)$

Initial Decision parameter in R_1, when $x_k = 0$, $y_k = r_y$

$$P_0' = f_e(x_k+1, y_k - 1/2) \rightarrow (x_{k+1})^2 r_y^2 + (y_k - 1/2)^2 r_x^2 - r_x^2 r_y^2$$

$$= f_e(0+1, r_y - 1/2)$$

$$P_0' = r_y^2 + (r_y - 1/2)^2 r_x^2 - r_x^2 r_y^2$$

$$P_0' = r_y^2 + (r_y^2 - r_y + \frac{1}{4}) r_x^2 - r_x^2 r_y^2$$

$$\boxed{P_0' = r_y^2 + \frac{r_x^2}{4} - r_y r_x^2}$$

E

$$\boxed{P_0' = r_y^2 + \frac{r_x^2}{4} - r_y r_x^2}$$

d

$$x^2 r_y^2 + y^2 r_x^2 - r_x^2 r_y^2 = 0$$

Taking derivative both side

$$\frac{dy}{dx}(x^2 r_y^2 + y^2 r_x^2 - r_x^2 r_y^2) = 0$$

$$\Rightarrow 2x r_y^2 + \frac{d}{dy}(y^2 r_x^2)\frac{dy}{dx} - \frac{dy}{dx}(r_x^2 r_y^2)$$

$$\Rightarrow 2x r_y^2 = -2y r_x^2 \frac{dy}{dx}$$

$$\Rightarrow \frac{dy}{dx} = \frac{-2x r_y^2}{2y r_x^2}$$

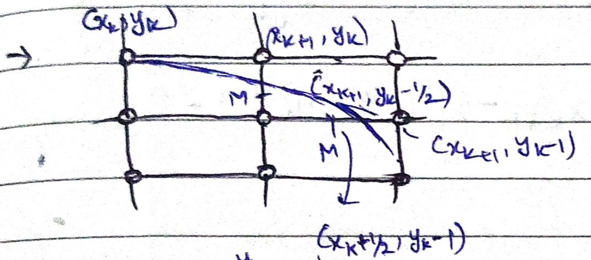
At boundary point R_1 and R_2 $\rightarrow + \frac{2x r_y^2}{2y r_x^2} = +1$

The slope of curve Since $\frac{dy}{dx} = -1$ we reach boundary point become -1 So

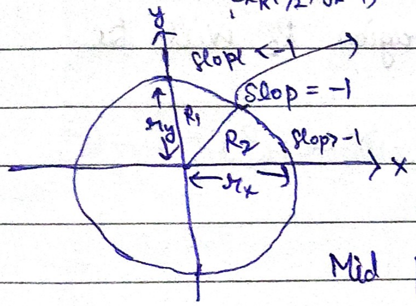
For Region R_2 the next pixel position will be (x_{k+1}, y_{k+1})

$$(x_{k+1}, y_{k+1}) \begin{cases} \to x_k, y_{k-1} \\ \to x_{k+1}, y_{k-1} \end{cases}$$

Mid point $(x_k + 1/2, y_k - 1)$



For Region R2
y_k will be decremented by unit interval and x_k will be estimated.



$$\text{Next P1} \begin{cases} \to x_k, y_k - 1 \\ \to x_k + 1, y_k - 1 \end{cases}$$

$$\text{Mid point} = \left(\frac{x_k + x_k + 1}{2}, \frac{y_k - 1 + y_k - 1}{2} \right)$$

$$= \left(x_k + 1/2, y_k - 1 \right)$$

General Equation of an ellipse $f_{xy} = \frac{x^2}{r_x^2} + \frac{y^2}{r_y^2} - 1 = 0$

$$P_k^2 = f_e\left(x + \frac{1}{2}, y_k - 1\right) = \left(x_k + \frac{1}{2}\right)^2 r_y^2 + (y_k - 1)^2 r_x^2 - r_x^2 r_y^2 \quad \text{--- } 4$$

It is symbol not square

Similarly Next decision parameter P_{k+1}^2

$$P_{k+1}^2 = \left(x_{k+1} + \frac{1}{2}\right)^2 r_y^2 + (y_{k+1} - 1)^2 r_x^2 - r_x^2 r_y^2 \quad \text{--- } 5$$

$$5 - 4$$

$$P_{k+1}^2 - P_k^2 = r_y^2 \left[\left(x_{k+1} + \frac{1}{2}\right)^2 - \left(x_k + \frac{1}{2}\right)^2 \right] + r_x^2 \left[(y_{k+1} - 1)^2 - (y_k - 1)^2 \right]$$

$$P^2_{k+1} - P^2_k = r_y^2\left[x^2_{k+1} + \frac{1}{4} + x_{k+1} - x_k^2 - \frac{1}{4} - x_k\right] + r_x^2\left[(y_k - 1 - 1)^2 - (y_k - 1)^2\right]$$

$$P^2_{k+1} - P^2_k = r_y^2\left[x^2_{k+1} + \frac{1}{4} + x_{k+1} - x_k^2 - \frac{1}{4} - x_k\right] + r_x^2\left[\cancel{(y_k-1)^2} + 1 - 2(y_k - 1)\cdot 1 - \cancel{(y_k-1)^2}\right]$$

$$P^2_{k+1} - P^2_k = r_y^2\left[x^2_{k+1} - x_k^2 + x_{k+1} - x_k\right] + r_x^2\left[1 - 2(y_k - 1)\right]$$

$$P^2_{k+1} - P^2_k = r_y^2\left[x^2_{k+1} - x_k^2 + x_{k+1} - x_k\right] + r_x^2\left[1 - 2y_{k+1}\right]$$

$$\boxed{P^2_{k+1} = P^2_k + r_y^2\left[x^2_{k+1} - x_k^2 + x_{k+1} - x_k\right] + r_x^2\left[1 - 2y_{k+1}\right]}$$

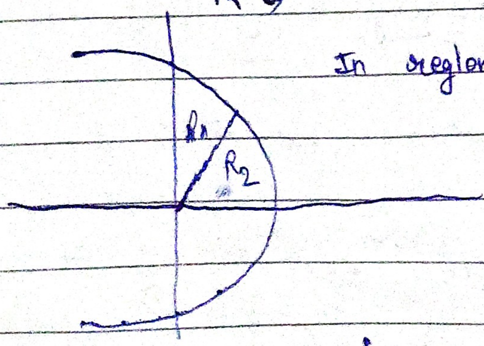
Initial Decision Parameter for region R_2 will be

$$\boxed{P^2_0 = r_y^2\left[x_k + \frac{1}{2}\right]^2 + r_x^2 (y_k - 1)^2 - r_x^2 r_y^2} \quad (6)$$

The intial pixel position of region R2 will be the last pixel position of region R1.

→ For a given center (0,0) and major axis (r_x) = 8 and minor axis (r_y) = 6, draw an ellipse using mid point algorithm.

Ans



In region R1, $P_k^1 < 0$

Initial Decision Parameter

$$P_0^1 = r_y^2 + \frac{r_x^2}{4} - r_x^2 r_y$$

$$P^1_{k+1} = P^1_k + r_y^2(1 + 2(x_k + 1))$$

Next point → $x_{k+1} \rightarrow x_{k+1}$

$$y_{k+1} \begin{cases} \rightarrow y_k \\ \rightarrow y_{k-1} \end{cases}$$

$36 \times \frac{64}{4} - 64 \times 6$

$36 \times 16 - 384$

$40 \times 36 (1 \times 8)$

For Region R1

K	P_K	x_{k+1}, y_{k+1}	$x r_y^2 = y r_x^2$
0	-332	(1, 6)	72, 768
1	-224	(2, 6)	144, 768
2	-44	(3, 6)	216, 768
3	208	(4, 5)	288, 640
4	-108	(5, 5)	360, 640
5	288	(6, 4)	432, 512
6	244	(7, 3)	504, 304
7			

For region R2 $\quad p^2_{k+1} = p^2_k + r_x^2 - 2r_x^2 y_{k+1}$ for $p^2_k > 0$

$p^2_{k+1} = p^2_k + r_y^2(2x_{k+1}) + r_x^2 - 2r_x^2 y_{k+1}$, for $p_k \leq 0$

$$P_0 = r_y^2 (x_0 + \tfrac{1}{2})^2 + r_x^2 (y_0 - 1)^2 - r_x^2 r_y^2$$

For region R2

$\cancel{5\ldots + 64 - 2 \times 64 \times 3}$

$= -\ldots$

K	P_K^2	x_{k+1}, y_{k+1}
0	-23	(8, 2)
1	361	(8, 1)
2	297	(8, 0)

$\cancel{288}+$

64×4

$244 + 36 - 2 \times 8$

64

$p2 = -23 + dx + 2\ldots \quad dy + 2\ldots + r_x^2$

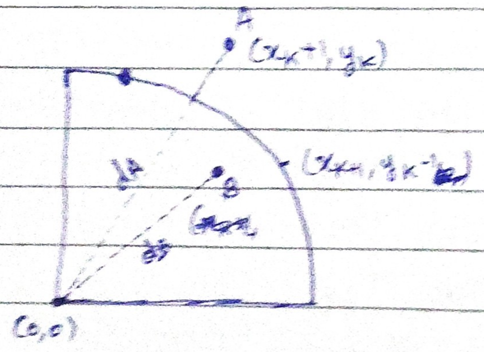
$= -23 + 2(36)(8) + 2(36) - 2(64)(2) + 64$

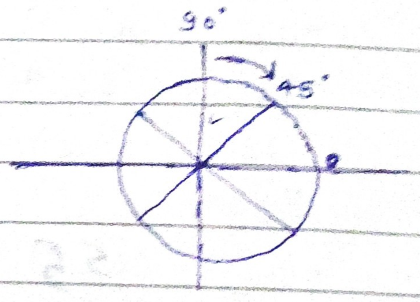
$= -23 + 576 + 72 - 256 + 64$

$= -23 + 2(36)(8) - 2(64)(2) + 64$

$=$

Bresenham Circle drawing algorithm





$$90° - 45° \to x_{k+1} \to x_k+1$$
$$y_{k+1} \to y_k \text{ or } y_{k-1}$$

Since a circle follows 8-way symmetry property if we can find out the pixel position of 1 octate then we can easily find out the pixel position of other octant. x will be increment by unit interval where as y will have 2 choices y_k or y_{k-1} from general eq^n of circle

$$f_c(x,y) = x^2+y^2-r^2 \to r \text{ is radius}$$

$(d_A + d_B) = 0$

$(d_A + d_B) < 0$

$(d_A + d_B) > 0$

For A,

$$f_{c_A}(x_{k+1}, y_k) = (x_{k+1})^2 + y_k^2 - r^2$$

For B,

$$f_{c_B}(x_{k+1}, y_{k-1}) = (x_{k+1})^2 + (y_k-1)^2 - r^2$$

Assume that the decision parameter is

$$d_k = f_c(A) + f_c(B)$$

if $d_k < 0$, it means $f_c(B) > f_c(A)$

$\therefore$ Next Pixel $\to (x_{k+1}, y_k)$

otherwise if $d_k > 0$ i.e. $f_c(A) > f_c(B)$

$\therefore$ Next Pixel $\to (x_{k+1}, y_{k-1})$

Scanned with ACE Scanner

$$d_k = ((x_k+1)^2 + y_k^2 - r^2) + ((x_k+1)^2 + (y_k-1)^2 - r^2) \quad -①$$

$$d_{k+1} = (x_{k+1}+1)^2 + y_{k+1}^2 - r^2 + (x_{k+1}+1)^2 + (y_{k+1}-1)^2 - r^2 \quad ②$$

Perform ② − ①

$$d_{k+1} - d_k = (x_{k+1})^2 + 2 \cdot x_{k+1} \cdot 1 + 1 + y_{k+1}^2 - r^2 + x_{k+1}$$

$$d_k = (x_k+1)^2 + y_k^2 - r^2 + (x_k+1)^2 + (y_k-1)^2 - r^2$$
$$= 2(x_k+1)^2 + y_k^2 + (y_k-1)^2 - 2r^2 \longrightarrow ③$$

$$d_{k+1} = 2(x_{k+1}+1)^2 + (y_k+1)^2 + (y_{k+1}-1)^2 - 2r^2$$

$$= 2(x_k+2)^2 + y_{k+1}^2 + (y_{k+1}-1)^2 - 2r^2 \longrightarrow ④$$

$$d_{k+1} - d_k = 2(x_k+2)^2 - 2(x_k+1)^2 + y_{k+1}^2 - y_k^2 + (y_{k+1}-1)^2 - (y_k-1)^2$$

$$= 2[x_k^2 + 4x_k + 4 - x_k^2 - 2x_k - 1] + y_{k+1}^2 - y_k^2 + y_{k+1}^2 - 2y_{k+1} + 1 - y_k^2 + 2y_k - 1$$

$$d_{k+1} - d_k = 4x_k + 6 + 2y_{k+1}^2 - 2y_k^2 - 2y_{k+1} + 2y_k$$

$$\boxed{d_{k+1} = d_k + 4x_k + 6 + 2y_{k+1}^2 - 2y_k^2 - 2y_{k+1} + 2y_k}$$

Case 1: when $d_k < 0 \rightarrow$ Next

$$\boxed{d_{k+1} = d_k + 4x_k + 6} \qquad \boxed{\begin{array}{l} x_{k+1} \rightarrow x_k + 1 \\ y_{k+1} \rightarrow y_k \end{array}}$$

Case 2: when $d_k > 0$ Next pt

$$\boxed{d_{k+1} = d_k + 4(x_k - y_k) + 10} \qquad \boxed{\begin{array}{l} x_{k+1} \rightarrow x_k + 1 \\ y_{k+1} \rightarrow y_k - 1 \end{array}}$$

Scanned with ACE Scanner

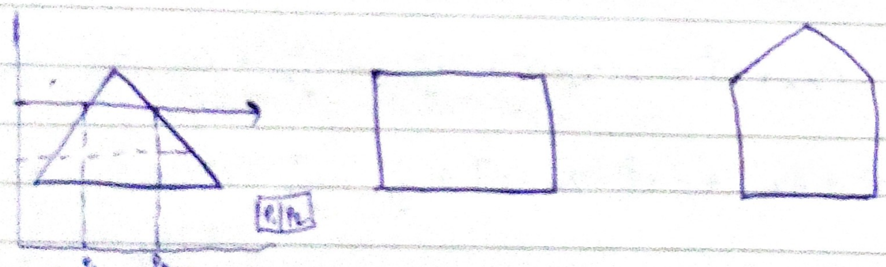
Intial decision will be calculated by taking the starting pixel position to $(0, r) \Rightarrow d_k = 2(x_k+1)^2 + y_k^2 (y_k-1)^2 - 2r^2$ $\boxed{d_0 = 3-2r}$

$d_0 = 2(0+1)^2 + r^2 + (r-1)^2 - 2r^2 \Rightarrow 3-2r$

Q→ Draw a circle with the radius $r=10$ and center $(0,0)$

Iteration	d_k	x_k	y_k	Decision Parameter
	-17	0	10	Initial DP $d_0 = 3-2\times10 = -17$
	-11	1	10	$d_1 = \cancel{d_0}\ -17+4(0)+6 = -11$
	-1	2	10	$d_2 = -11+4(1)+6 = -1$
	13	3	10	$d_3 = -1+4\times2+6 = 13$
	-5	4	9	$d_4 = 13+4(3-10)+10 = -5$
	17	5	9	$d_5 = d_4+4\times4+6 = 17$
	11	6	8	$d_6 = d_5+4(5-9)+10 = 11$
	13	7	7	$d_7 = d_6+4(6-8)+10 = 13$

Filled Area Primitives



a) Scan line

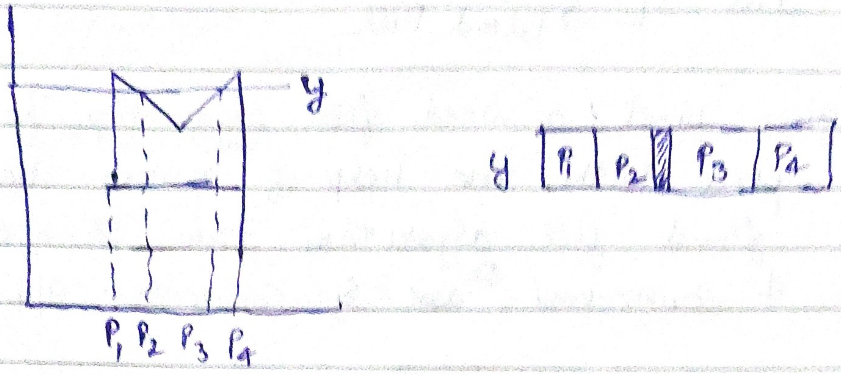
b) Seed Fill

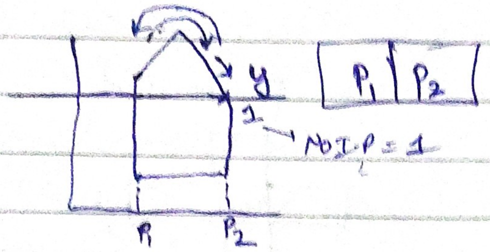
For a given polygon we need to fill a polygon with a specified colour,

A polygon is a collection of edges or line segment with the same starting and ending point. Now to fill a polygon we required to components 1) Interior Pixel

2) Colour to be filled to Pixel

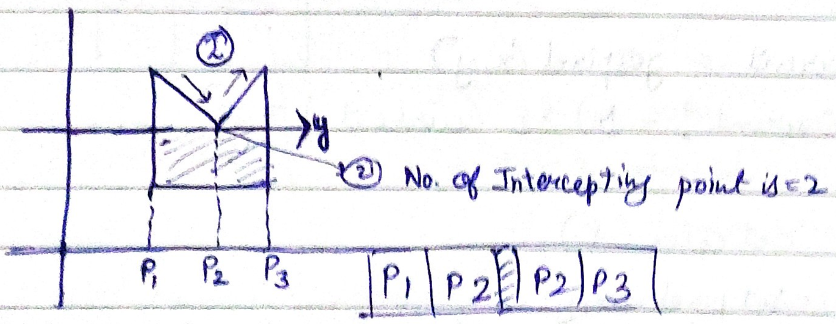
1) Find Interior Pixel 2) Fill colour to Pixel

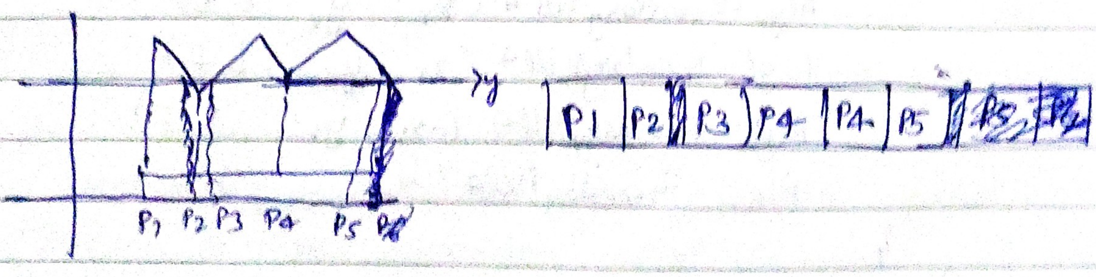




Case 1 → If the y value is monotonically increasing or decreasing it is considered as 1

Case 2 → otherwise it is considered as 2.

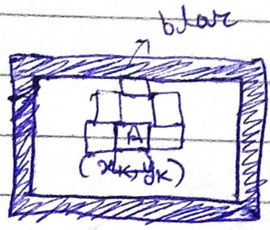




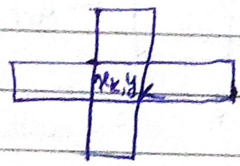
b) Seed Fill → Boundary Fill / → Flood Fill

For a given pixel a seed fill algorithm fills a enterior pixel with the help of either boundary fill or flood fill algorithm this 2 algorithms follows 4 connected or 8 connected approach.

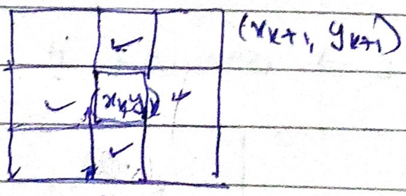
1) Boundary Fill



4 Connected



8 connected



Algorithm → Algo-Boundary_Fill 4 (int x, int y, int b, int f)

```
{
    int current = getpixel (x, y)
    if (current != b) && (current != f)
    {
        set Color (f)
        Set pixel (x, y)
        Algo - Boundary Fill4 (xk, yk+1, b, f)
        Algo - Boundary Fill 4 (xk-1, yk, b, f)
        Algo - Boundary Fill4 (xk, yk-1, b, f)
        Algo - Boundary Fill4 (xk, yk-1, b, f)
        Algo - Boundary Fill4 (xk+1, yk, b, f)
    }
}
```

Scanned with ACE Scanner

1) CRT → Random
→ Raster (Working Principle) ✓

2) Circle → Mid-point ✓, ~~BBB~~ ✓
→ Breshenman

3) Line ✓.

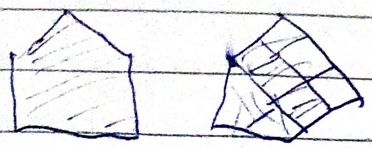
4) Derivation → Ellipse (only) X

Output Primitives

The geometric structures use to describe an image is called output primitives.
Every primitives have some attributes which are used to draw the different structure. For eg. line is an output primitive

(i) Line
→ line_color (lc)
→ line_type (solid, dotted, dashed)
eg.
lt = 1, Solid line
lt = 2, dashed
lt = 3, dotted
lt = 4, dashed_dotted
→ line_width (Std, thicker thinner)
eg. lw = 1, Std width
lw > 1. thicker width
lw < 1 thinner width

(ii) Area Attribute - Determine whether it is filled with solid color or some pattern



Scanned with ACE Scanner

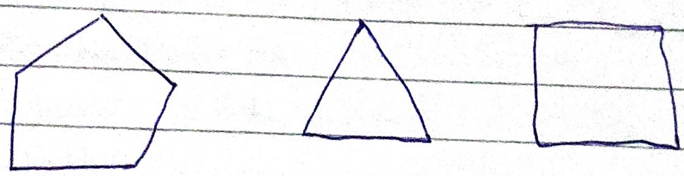
(iii) Text → Text color
→ Type
→ Spacing
→ Width

(iv) Text Path → Horizontal
→ Vertical

1) Filled Area Primitives

It deals with the procedures to filled a polygon with specified color.

Polygon → Polygon is a collection of line segment with the same starting and ending point.



To fill a polygon with a specified color we need to perform 2 task

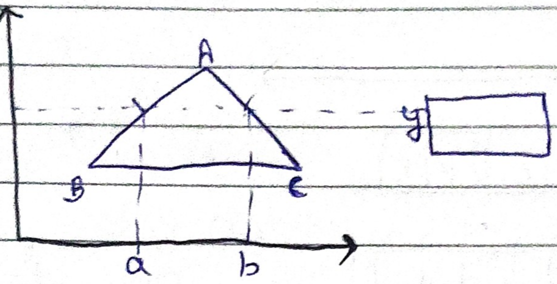
(i) Find the interior pixel.
(ii) Fill the interior pixel with some specified color.

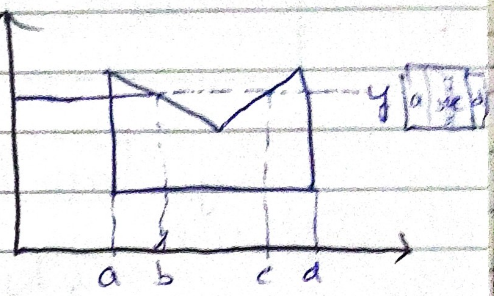
a) Scan line Filling algorithm
b) Seed Fill Algorithm

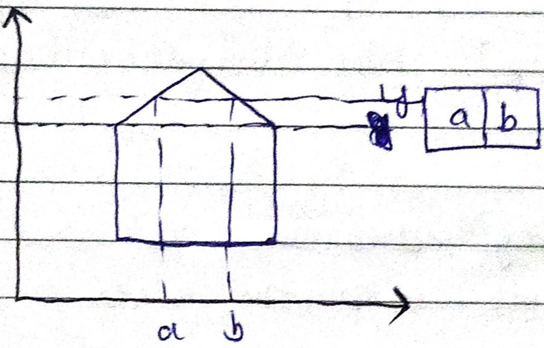
Scanned with ACE Scanner

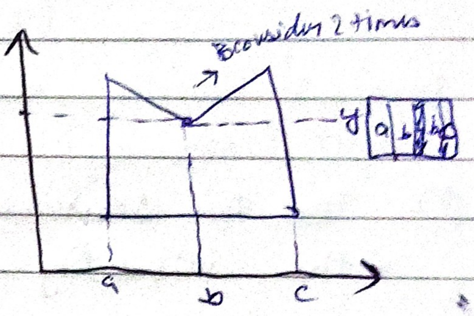
Scan Line Filling Algorithm

In scan line filling algorithm we need to find out the intersecting point with the edges of the polygon with the help of scan line (y).



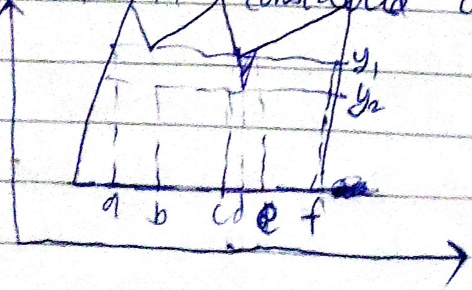


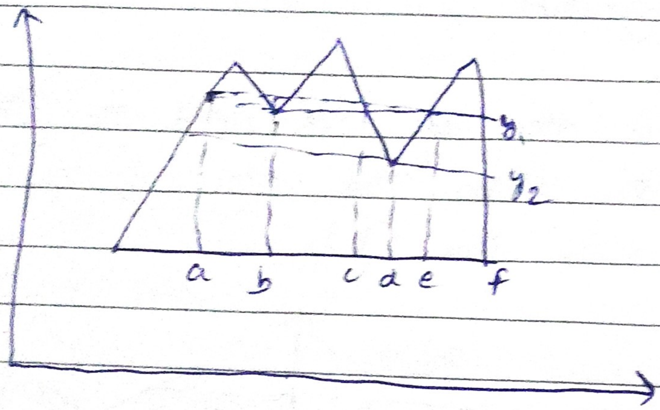




In scan line algorithm an intersecting point will be consider as either 1 or 2 depending upon the value of y so we need to perform a clock-wise or anti clock wise traversal along the edges.

If the value of y is ~~considered as~~ monotically increasing or decreasing then the intersecting point will considered as 1 otherwise it is considered as 2.





y_1

a	b	▨	b	c	▨	e	f

y_2

a	d	▨	d	f

Steps

① ~~The~~ The scan line (y) traverse to and locate the intersecting points with the edges.

② Short the intersecting point from left to right

③ Draw the Interior intersecting point pairwise.

Seed Fill algorithm

This algorithm is perform by 2 approaches

1) Boundary Fill
2) Flood Fill

Boundary Fill → In Boundary Fill algorithm the boundary of a polygon is filled with a given color and we have to find the neighbouring pixel of given interior pixel and filled it with an given color.

To find the neighbouring pixel we have to perform either 4 connected or 8 connected algorithm we required 3 parameters

a) Current Pixel (x,y)
b) Boundary color (b)
c) Fill color (f)

This is for 4
for 8

Imp
Algorithm - boundaryfill4(int x, int y, b, f)

```
{
    int current = getpixel (x,y)
    if ((current != b) && (current != f))
    {
        set color (f)
        set pixel (x,y) / putpixel (x,y)
        boundary fill 4 (x, y+1, b, f)
        boundary fill 4 (x-1, y, b, f)
                        (x, y-1, b, f)
                        (x+1, y, b, f)
    }
}
```

you do it for 8 times

Drawbacks

any of

If the interior pixel is filled with the boundary colour then the recursive call to boundary filled will terminate hence the polygon could not be filled.

Scanned with ACE Scanner

Steps

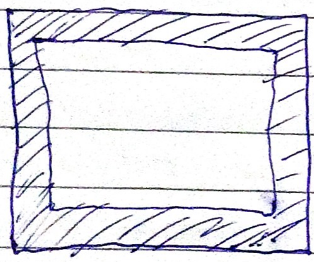
1) ~~Start~~ For a given interior pixel, check whether its neighbouring pixels are filled with the boundary color or not.

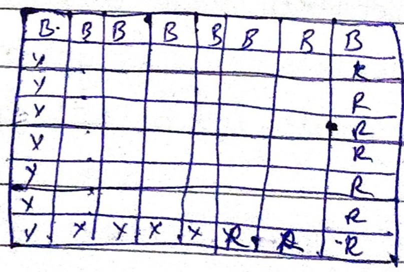
2) If the neighbouring pixel is not filled with the boundary color fill it with the fill-color and assign it ~~a~~ with current pixel.

3) This process continue until all the pixels till boundary has been tested.

Flood Fill Algorithm

It deals with the filling of a polygon with a specified color where the boundary is not specify with a single color.





~~Imp~~

Algorithm flood fill 4 (int x, int y, int old color, int new color)

{

~~setpixel~~ if getpixel((x,y) == old color)

{

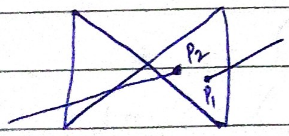
putpixel (x, y, new color)

flood fill (x+1, y, oldcolor, newcolor)
" (x, y+1, " , ")
" (x, y-1, " , ") — 4 connected
" (x-1, y-1, " , ")
(x-1, y+1, " , ")
(x+1, y-1, " , ")
(x+1, y+1, " , ") — 8-connected

When filling a polygon either through a scan line or seed fill algorithm we need to check whether a point lies inside or outside the polygon for this purpose we perform 2 method

1) Inside-out test
2) Non-zero Winding number rule.

1) Inside-out Test → In this method we draw a line segment from a point of a polygon towards the outside the boundary of the polygon and count the



number of intersection that the line segment have met we the boundary of that polygon.

Case 1 - If the number of count is even then point p is exterior

Case 2 - If the number of count is odd then point p is interior

Note → The line segment from that point should not passes any vertices or the intersection of 2 edges of that polygon.

★★ 2marks Imp.

2) Non-Zero Winding Rule - In this method we count the number of times the edges of a polygon wind around a point p either in clockwise or in anti clockwise direction.

Step 1 - Initialize the winding number WN = 0

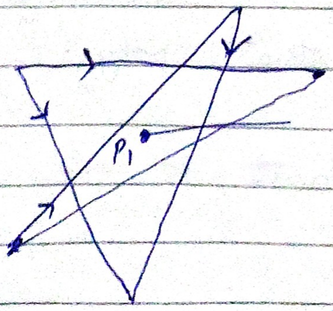
Step 2 - Draw a line segment from point P towards outside the polygon in such a way that it should not passes through any vertex of that polygon.

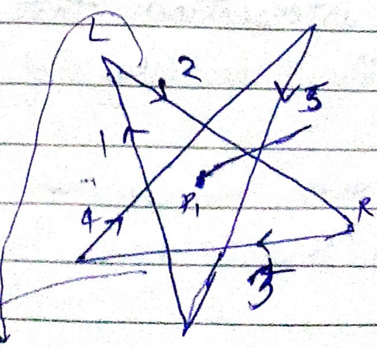
Step 3 - Add 1 to the winding number WN if the edge crosses this line in 1 direction and subtract 1 to WN if the edge crosses this line in opposite dirⁿ.

Step 4 - If the sum of winding number is non-zero then p is consider as interior otherwise / else point p is exterior point.

(right → left) or (top → bottom) ⇒ -1

(left → right) or (bottom → top) ⇒ $+1$

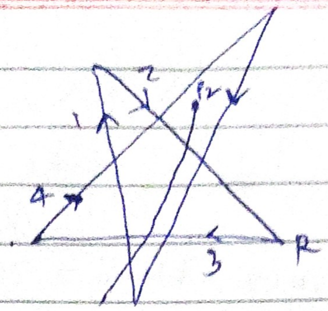




$W_N = 0+0+1+$
$0+0+1$
$= 2$
P is interior

$W_N = 0+0-1+0+0+1 = 0$

Initial value

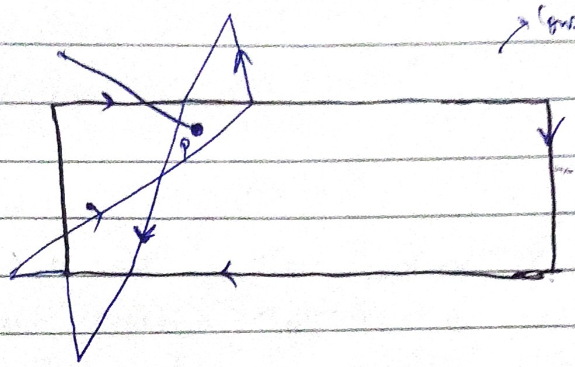


→ ([illegible])

$W_N = 0 - 1 + 1 + 1 = +1$

→ (Top to Bottom)

$W_N = +1 + 1 - 1 = +1$

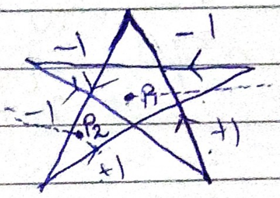


→ Considering Both

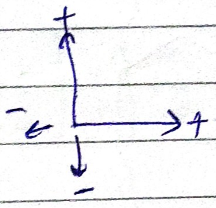
$W_N = 0 - 1 + 1 = 0$

P is exterior

Non-zero Winding Number

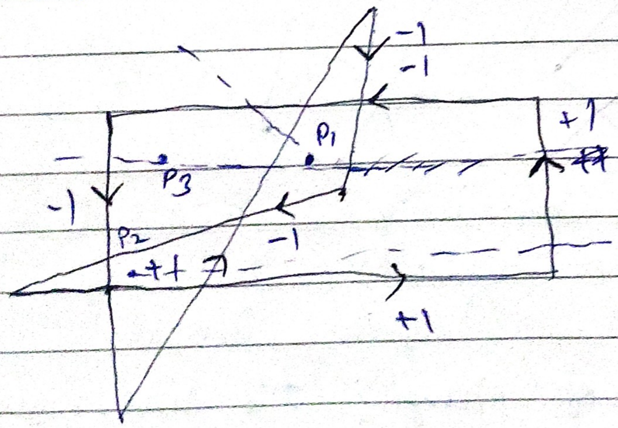


$WN(P_1) = +1 + 1 = 2$



L-R / B→U (+1)

R-L / T-B (-1)



exterior

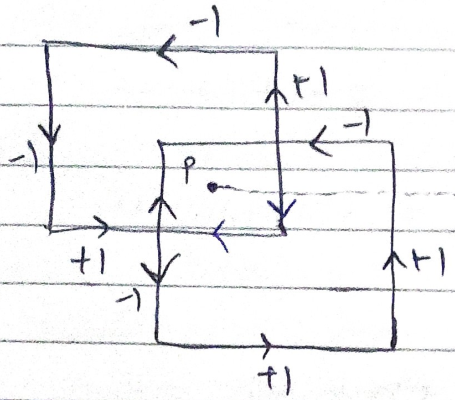
$WN(P_1) = +1 - 1 = 0$

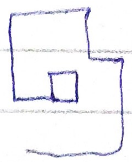
$WN(P_2) = +1 + 1 = 2 \neq 0$

interior

$WN(P_3) = -1 + 1 - 1 \neq 0$

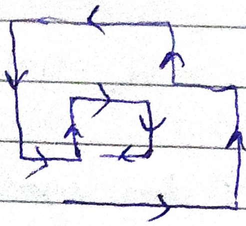
exterior

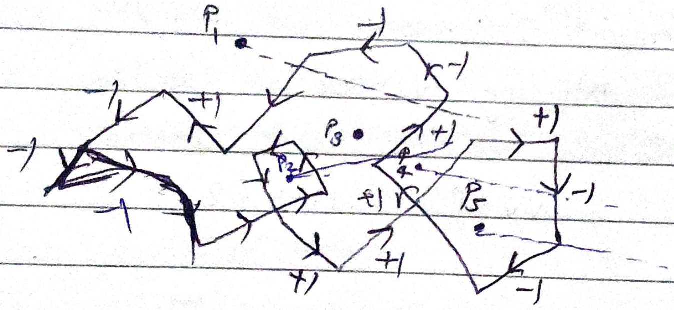




X $WN(P) = +1+1 = 2$ interior

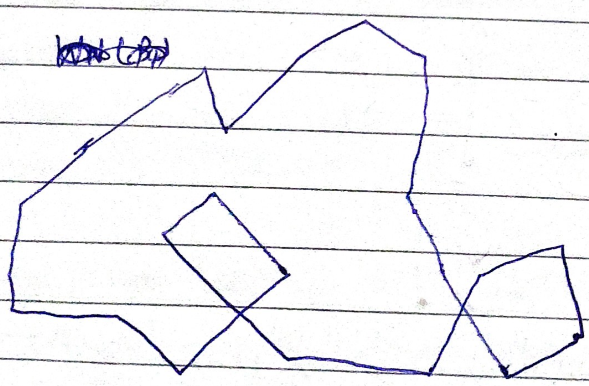
$WN(P) = +1-1 = 0$ P exterior





Draw polygon fill only interior points not exterior

~~WN(P)~~



Transformation

It is the process of changing the size, shape, ~~object and~~ orientation w.r.t. to a height ~~cordine~~ coordinate system
of object.

Scanned with ACE Scanner

Basic Transformation

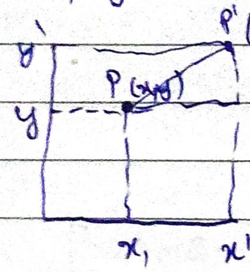
1) Translation
2) Rotation
3) Scaling

Two derived transformation
4) Reflection
5) Shearing.

~~Transformation~~

Translation - It is the process of repositionary an object from one position to another position along a straight line. Performed using a translation / Shift vector.

(t_x, t_y)



$x' = x + t_x$

$y' = y + t_y$

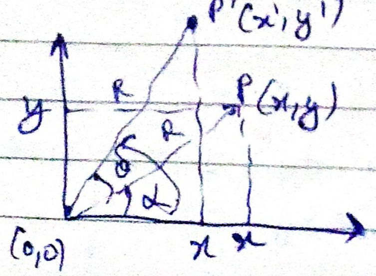
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \end{bmatrix}$$

P(3,5) T(2,-5)

P' = (5,10)

Rotation - It is a process of repositionary of an object from one angle to another angle by keeping distance same from origin.

P' = P.R



$\cos\alpha = \frac{x}{r} \Rightarrow x = r\cos\alpha$

$\sin\alpha = \frac{y}{r} \Rightarrow y = r\sin\alpha$

Scanned with ACE Scanner

$\cos(\theta+\alpha) = \frac{x'}{r} \Rightarrow x' = r\cos\theta\cos\alpha - r\sin\theta\sin\alpha$

$\sin(\theta+\alpha) = \frac{y'}{r} \Rightarrow y' = r\cos\theta\sin\alpha + \frac{r\cos\alpha\sin\theta}{r}$

$$\boxed{\begin{aligned} y' &= y\cos\theta + x\sin\theta \\ x &= x\cos\theta - y\sin\theta \end{aligned}}$$

$\theta = +$ Anti

$\theta = -$ Clock

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

Q→ A point P(4,3) has been rotated in counter clockwise direction by an angle of 45° then find the rotation matrix R and the resultant points.

Ans

$$\cancel{y' = y\cos 45^\circ + x\sin 45^\circ}$$
$$\cancel{x' = x\cos 45^\circ - y\sin 45^\circ}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 4 \\ 3 \end{bmatrix} \begin{bmatrix} \cos 45^\circ & -\sin 45^\circ \\ \sin 45^\circ & \cos 45^\circ \end{bmatrix}$$

It is a type of transformation in which the size of ~~the~~ on object is either enlarge or shrink by a scaling factor (S_x, S_y) therefore if point P prime is new point then it is represented

$P' = P \cdot S_{x,y}$

$0 < S_x, S_y < 1 \rightarrow$ Size decrease (↓)

$S_x, S_y > 1 \longrightarrow$ Size increase (↑)

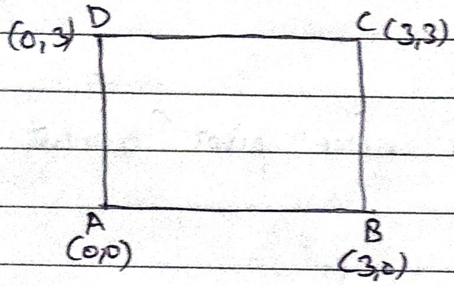
$S_x = S_y \rightarrow$ uniform scaling

Scanned with ACE Scanner

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix}$$

$$\boxed{\begin{aligned} x' &= x \cdot S_x \\ y' &= y \cdot S_y \end{aligned}}$$

Q-) For ABCD translate a polygon with A(0,0) B(3,0) C(3,3) D(0,3) by 2 units in both the direction and scale it by 1.5 units in ~~Translate~~ x direction and 0.5 units in y dirⁿ. Then determine the result position of ABCD.



T_x =	A' = (2,2)	~~Sx.. Sy~~	(3,1)
	B = (5,2)		(7.5,1)
	(5,5)		(7.5,2.5)
	(2,5)		(5, 2.5)

Q. Magnify a Δ with vertices (0,0) (1,1) and (5,2) to twice of its size while keeping (5,2) fixed.

Q. Scale a Polygon with coordinate A(2,5), B(7,10) and C(10,2) by 2 units in x-dirⁿ and 3-units in y-dirⁿ.

A(2,5)

$$A' = \begin{bmatrix} x' \\ y' \end{bmatrix} = \cancel{\begin{bmatrix} x \\ y \end{bmatrix}} \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

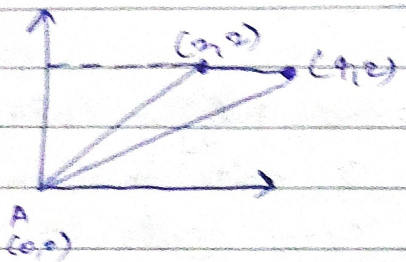
B'

C'

Scanned with ACE Scanner

Q→ Rotate a Δ A(0,0), B(2,2) and C(4,2) about the origin by an angle of 45°

Ans $R = R_1 X_1 \neq X_1 R_1$



Q→ Rotate the same Δ about some pivot point element (-2,-2)

Ans

$$\overset{R_o}{\begin{bmatrix} & \\ & \end{bmatrix}} = \overset{T_x}{\begin{bmatrix} & \\ & \end{bmatrix}} \overset{P}{\begin{bmatrix} & \\ & \end{bmatrix}}$$

A' $(-2, 2\sqrt{2}-2)$

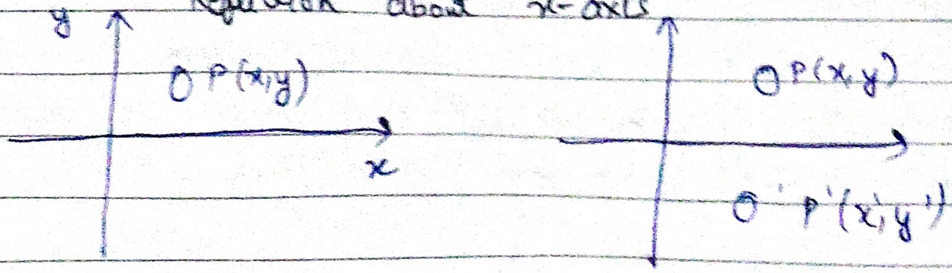
B' $(-2, 4\sqrt{2}-2)$

C' $(\sqrt{2}-2, 5\sqrt{2}-2)$

~~Composite Tran~~ Reflection

Reflection is a transformation that produces a mirror image of an object w.r.t an axis it is generated by rotating an object by an angle of 180° w.r.t the reflection axis.

(a) Reflection about x-axis

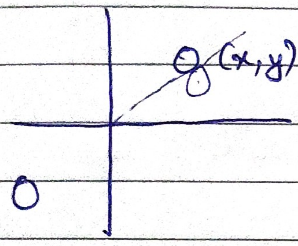


$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

b) Reflection about y-axis

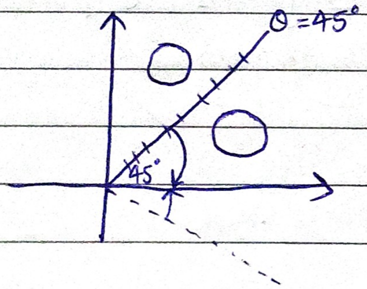
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

c) Reflection about origin



$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

d) Reflection about $y=x$ line



$P' = R(-\theta)\, R_{Ref_x}\, R(\theta)$

Reflection about the line $y=mx+c$

Steps.

1) Translate the line $y=mx+c$ so that it passes through the origin.

2) Perform rotation with an angle of θ.

3) Apply Reflection.

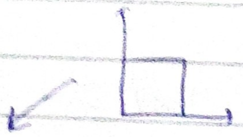
4) Perform Rotation with an angle of $-(\theta)$

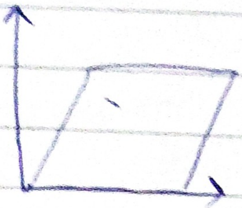
5) Perform reverse Translation

Shearing

It is a transformation in which the shape of an object is distorted. It is of 3 types

a) X-shearing



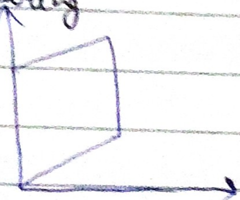


$x' = x + Sh_x y$

$y' = y$

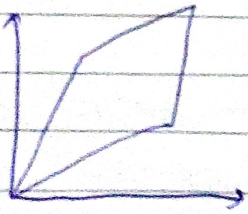
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & Sh_x \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

b) Y-Shearing



$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ Sh_y & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

c) about XY



$x' = x + Sh_x y$

$y' = y + Sh_y \cdot x$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & Sh_x \\ Sh_y & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Q) Shear a polygon A (0,0), B (1,0), C (1,1) and D (0,1) by a shearing vector $Sh_x = 2$ and determine the new Coordinate.

Ans

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Scanned with ACE Scanner

$x' = x + 2 \cdot y$
$y' = y$

$A \rightarrow \quad x' = 0 + 0 = 0$
$y' = 0$

$B \rightarrow \quad x' = 1 + 0 = 1$
$y' = 0$

$C \rightarrow \quad x' = 1 + 2 = 3$
$y' = 1$

$D \rightarrow \quad x' = 0 + 2 = 2$
$y' = 1$

Q → A polygon with A (0,0), B (1,0), C (1,1) and D (0,1) ~~[illegible]~~. Perform the following transformation and calculate the result of shearing.

1) X-Shearing with A = 2 followed by Y-Shearing with B = 3

2) Simultaneous X and Y shearing with A = 2 and B = 3

Composite Transformation

A transformation system where 2 or more than 2 successive are apply is called composite transformation. For eg. a compositive transformation which apply 2 translation

$T_1 (t_{x_1}, t_{y_1})$
$T_2 (t_{x_2}, t_{y_2})$

Scanned with ACE Scanner

Q) Imp. $\boxed{P' = T_2 \cdot T_1 P}$

[Prove that the reflection about the line $Y = -X$ is equivalent to the reflection relative to y-axis followed by anti clockwise rotation of 90 degree.]

Homogeneous Transformation System

It is a special type of representation system in which an object in 2D System is represented in triplet form that is in 3 elements vector $P_h(x_h, y_h, h)$ where h is a homogenous factors such that $x = \frac{x_h}{h}$ and $y = \frac{y_h}{h}$ and it is a non-zero value. For eg.

$P(3, 4) \rightarrow (6, 8, 2)$

Why we need Homogenous coordinate System

In this coordinate system all the transformation are represented in a matrix multiplication form. The reason is in 2D system, translation involve matrix multiplicative term as well as additive terms whereas rotation and scaling involves only matrix multiplicative terms therefore we require a general representation system where all the basic transformations can be represented in a single form which is matrix multiplication. Thus, a 2D system is converted into homogenous terms where (x, y) represented in a form $(x, y, 1)$.

Translation - $x' = x + t_x$

$y' = y + t_y$

~~In H.S.~~ $\cancel{\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

(i) In H.S.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

(ii) In H.S. Rotation

$$x' = x\cos\theta - y\sin\theta$$
$$y' = x\sin\theta + y\cos\theta$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

(iii) Scaling

$$x' = x \cdot S_x \qquad \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$
$$y' = y \cdot S_y$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Scanned with ACE Scanner

(iv) Reflection in Homogenous Systems

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

X axis Y-axis Y=X plane

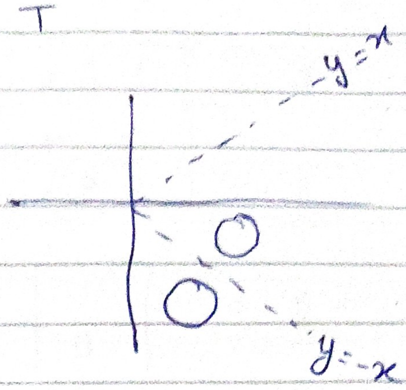
(V) Shearing → x, y, xy

$$\begin{array}{c} X \\ \begin{bmatrix} 1 & Sh_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \end{array}$$

$$\begin{array}{c} XY \\ \begin{bmatrix} 1 & Sh_x & 0 \\ Sh_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \end{array} \qquad \begin{array}{c} Y \\ \begin{bmatrix} 1 & 0 & 0 \\ Sh_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \end{array}$$

Imp.

Ans) Reflection about $Y = -X$ is T



$$T_{Ref.} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$T_{Ray} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Rotation about $\theta = 90°$ R_θ

$\theta = 90°$

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

multiply both

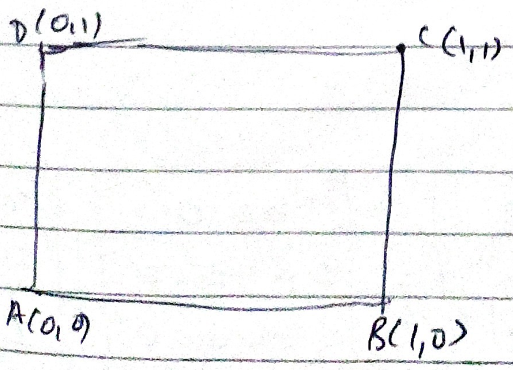
$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

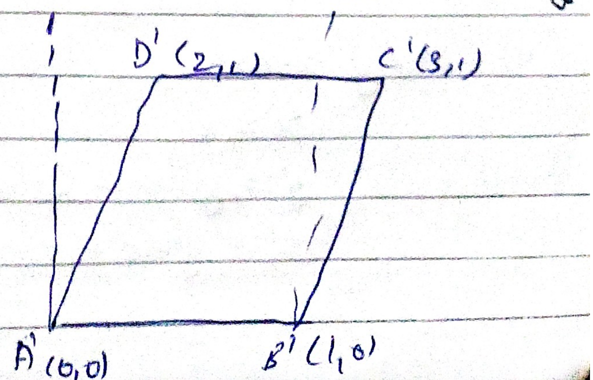
3-D Transformations

Polygon A(0,0) B(1,0) ~~Take place in 3D plane~~ C(1,1) D(0,1) Perform transformation and calculate result shearing

a) x-shearing with $a=2$ followed by y-shearing with $b=3$

b) Simultaneous shearing x and y with $a=2$ and $b=3$ respectively





$$A' = \begin{bmatrix} 1 & 2 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$B' = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \quad C' = \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix} \quad D' = \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}$$

3-D transformation

When the transformation textplay in 3-D plane or x, y, z-axis such that each axis is normal to each other then it is called 3-D transformation. In 3-D transformation each basic Transformation like translation, rotation and shearing like represented as a 4x4 matrix

1) 3-D Translation

$x' = x + t_x$
$y' = y + t_y$
$z' = z + t_z$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

2) 3D - Rotation

a) Along x-axis

$x' = x$
$y' = y\cos\theta - z\sin\theta$
$z' = y\sin\theta + z\cos\theta$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Scanned with ACE Scanner

b) Along y-axis

$$y' = y$$
$$x' = x\cos\theta + z\sin\theta$$
$$z' = z\cos\theta - x\sin\theta$$
$$x' = z\sin\theta + x\cos\theta$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

c) along z-axis

$$z' = z$$
$$x' = x\cos\theta - y\sin\theta$$
$$y' = x\sin\theta - y\cos\theta$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

3) 3D - Scaling

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

3D - Scaling in x-direction by factor a   y-dirⁿ b   z dirⁿ c

$$\begin{bmatrix} a & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Scanned with ACE Scanner

d) Scaling in x, y, z-direction by factor a, b, c respectively

$$\begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(e) Uniform Scaling in x-direction

$$\begin{bmatrix} a & 0 & 0 & 0 \\ 0 & a & 0 & 0 \\ 0 & 0 & a & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

4) 3D-Reflection a) about XY plane (z=0)

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

yz plane (x=0)

b) $$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

c) $$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$ xz plane (y=0)

5) 3D-Shearing

a) In x, y, z direction

$$x' = x + y \cdot a + z \cdot b$$
$$y' = \cancel{xxx}\ y + x \cdot c + z \cdot d$$
$$z' = z + x \cdot e + y \cdot f$$

$$\begin{bmatrix} 1 & a & b & 0 \\ c & 1 & d & 0 \\ e & f & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Scanned with ACE Scanner

b) 3D - Shearing in X-dir^n

$$Sx = \begin{bmatrix} 1 & Sy & Sz & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Sz = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ Sx & Sy & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Sy = \begin{bmatrix} 1 & 0 & 0 & 0 \\ Sx & 1 & Sz & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Q → For a given matrix, apply rotation of 45° about y-axis followed by rotation of 45° about x-axis and determine the resultant matrix.

$$M = \begin{bmatrix} 2 & 0 & 1 & 0 \\ 1 & 3 & 0 & 0 \\ 4 & 0 & 1 & 0 \\ 0 & 3 & 6 & 1 \end{bmatrix}$$

Ans $R_{y\theta} = 45°$, $R_{x\theta} = 45°$

$$R_x \cdot R_y \cdot [M]$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{2} & 0 \\ -\frac{1}{2} & \frac{1}{\sqrt{2}} & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

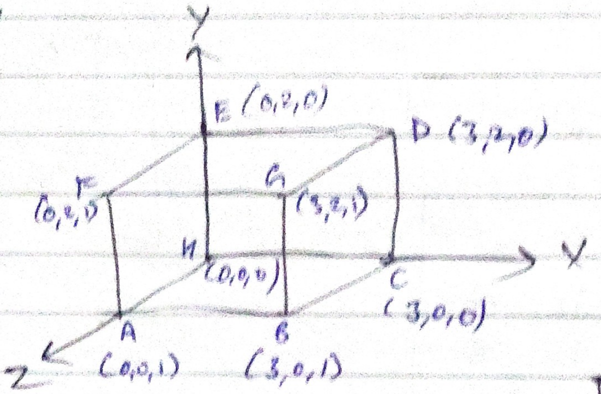
$$RM = \begin{bmatrix} \sqrt{2} & 0 & 0 & 0 \\ 0 & \sqrt{3}/2 & 0 & 0 \\ 0 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Scanned with ACE Scanner

Q → A rectangular parrallellopiped has its length 3 units 2 units on x, y and z-axis respectively perform a rotation of 90° clockwise about x-axis.

$P' = R_{x,\theta=90°} \cdot P$

rectangle in 2-D



$$P = \begin{array}{c} \begin{array}{cccccccc} A & B & C & & & & & H \end{array} \\ \begin{bmatrix} 0 & 3 & 3 & 3 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 2 & 2 & 2 & 2 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

A (0,0,1) E (0,2,0)

B (3,0,1) F (0,2,1)

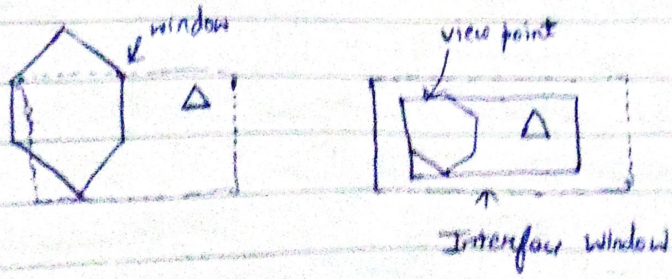
C (3,0,0) G (3,2,1)

D (3,2,0) H (0,0,0)

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 0 & 0 & 0 \\ 3 & 1 & 0 \\ 3 & 0 & 0 \\ 3 & 0 & -2 \end{bmatrix}$$

$$\begin{matrix} 0 & 0 & -2 \\ 0 & 1 & -2 \\ 3 & 2 & -2 \\ 0 & 0 & 0 \end{matrix}$$

2D - Viewing



Object Coordinate

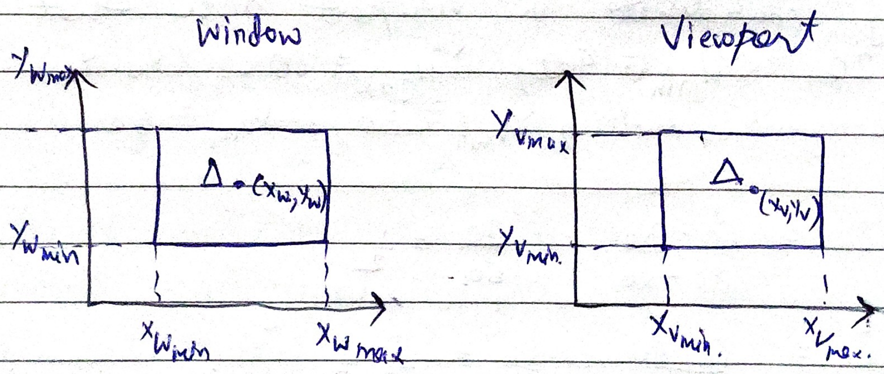
The coordinate in which object are created or define.

Screen coordinate - The space in which the image is displayed.
World coordinate - The space in which application model is define or several objects are assembled into scene.

Window → The rectangular area "What we want to see" is called window.

Viewport - The rectangular area "Screen" for viewing the content of the window. (where we want to display).

Windowing ~~Transport~~ Transformation → The process of transforming the window coordinate to viewport coordinate is called windowing transformation.



For a given coordinate the relative position of an object will not be changed only the size of the object will be change according to the windows and viewport coordinate. Thus for an given object (X_W, Y_W) in the window coordinate we will find out its position (X_V, Y_V) in the viewport using the formula

$$\boxed{\frac{X_W - X_{Wmin}}{X_{Wmax} - X_{Wmin}} = \frac{X_V - X_{Vmin}}{X_{Vmax} - X_{Vmin}}}$$

$$\boxed{\frac{Y_W - Y_{W_{min}}}{Y_{W_{max}} - Y_{W_{min}}} = \frac{Y_V - Y_{V_{min}}}{Y_{V_{max}} - Y_{V_{min}}}}$$

Scaling factor

$$\boxed{X_V = X_{V_{min}} + (X_W - X_{W_{min}}) \cdot S_x}$$

$$\boxed{S_x = \frac{X_{V_{max}} - X_{V_{min}}}{X_{W_{max}} - X_{W_{min}}}}$$

$$\boxed{Y_V = Y_{V_{min}} + (Y_W - Y_{W_{min}}) \cdot S_y}$$

$$\boxed{S_y = \frac{Y_{V_{max}} - Y_{V_{min}}}{Y_{W_{max}} - Y_{W_{min}}}}$$

Q For a given coordinate of an object in window find the corresponding coordinate in viewport given that $X_{W_{max}} = 80$, $X_{W_{min}} = 20$, $Y_{W_{min}} = 40$, $Y_{W_{max}} = 80$ $(X_W, Y_W) = (30, 60)$, $Y_{V_{max}} = 60$, $Y_{V_{min}} = 40$, $X_{V_{min}} = 30$, $X_{V_{max}} = 60$.

Ans

$$S_x = \frac{60-30}{80-20} = \frac{30}{60} = \frac{1}{2}$$

$$S_y = \frac{60-40}{\cancel{6}0-\cancel{4}0} = \frac{20}{\cancel{4}0} = \frac{\cancel{2}}{\cancel{3}}\ \frac{1}{2}$$

$$X_V = 30 + (30-20) \cdot \frac{1}{2} = 30 + 5 = 35 \qquad (35, 50)$$

$$Y_V = 40 + (60-40) \cdot \frac{1}{2} = 40 + 10 = 50$$

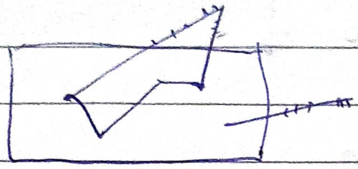
2D viewing Pipeline

~~It describe~~

A series of transformation which are performed by geometric data to end up as a image data by displaying image data being displayed on a device.

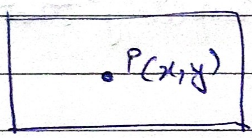
2D - Clipping

Cliping is a process of finding the interior portion of an object or an image is called clipping.



1) Point Clipping

A point P(x,y) will be displayed if it satisfies the equation :-

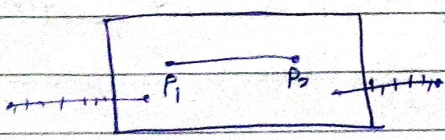


$$X_{min} \le x \le X_{max}$$
$$Y_{min} \le y \le Y_{max}$$

2) Line Clipping

The process of removing the line or some portion of a line outside the clipping windows is called line clipping.

Imp
Imp Cohen - ~~Southerland~~ Sutherland Algorithm



It determines the visibility of a line that whether a line or portion of a line is visible or not. In this algorithm entire region of a window is subdivided into 9 sub regions which has unique code. We ~~assign~~ assign

a unique code at ~~~~ each region using TBRL
Top Bottom Right Left

1001	1000	1010
0001	P_1 — P_2 0000	0010
0101	0100	0110

5marks
The Pseudo code for this algorithm.

Step-1 → Assign the region code for 2 end points of given line.

Step-2 → If both the end points have region code → 0000 then line is ~~~~ accepted completely that is ~~~~ C_0 OR C_1 = 0000

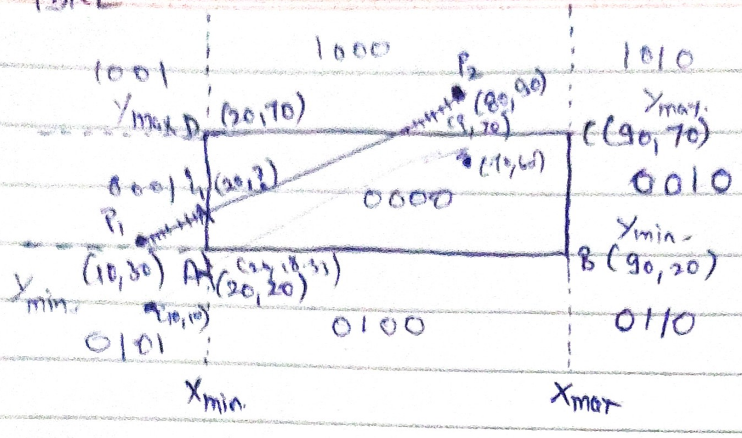
0001
0000
0001

Step-3 → Else perform the logical AND operation on the region code of the 2 end points and check
(a) If the result of C_0 AND C_1 ≠ 0000 then line is outside.
Else the line ~~~~ is partially inside.
(i) Choose an end point of the line that is outside the rectangle
(ii) Find the intersection point $y = mx + c$ ~~~~ $x = \frac{1}{m}x + c$
(iii) Replace the end point with the intersection point.
(iv) Repeat Step-2 until the line is accepted or rejected.

Note- If any endpoints lies on the boundary of the windows then its region code is 0000.

Q→ For a given rectangular window ABCD with A (20, 20) B (90, 20) C (90, 70) D (20, 70) Find the region code for the end points of the line P_1 (10, 30) and P_2 (80, 90) and clip the line using Cohen Sutherland Algo.

TBRL



Steps

1) Assign region code
2) Perform logical OR P_1 OR P_2

0001
1000
1001 $\neq 0$,

3) Perform logical AND P_1 (AND) P_2

0001
1000
0000

4) Find slop $m = \frac{Y_2 - Y_1}{X_2 - X_1} = \frac{90-30}{80-10} = \frac{60}{70} = 0.857$

5) $y = m(X_{min} - x_1) + y_1 = 0.857(20-10) + 30$
$= 38.57 \approx 39$

$x = \frac{1}{m}(Y_{max} - Y_1) + x_1 = \frac{7}{6}(70-30) + 10 = 56.67$

P_3 (10,10) and P_4 (70,60) → Clip these points

$y = m(x_{min} - x_1) + y_1 = \frac{5}{6}(20-10) + 10 = 18.33$

Disadvantages :-

Clipping window must be square and rectangular shape in order to clip so new algo come