Freeway-study

A freeway of learning

learn with ease

HANDWRITTEN NOTES

Course-B.C.A 4th Sem

Subject-Computer Graphics & Multimedia Application

Provided by - Sakib saifi

send us your query anytime
freewaystudy1234@gmail.com

Computer Graphics and multimedia :-

* **Computer Graphics :-** It is the use of computers to create and manipulate pictures on a display device. It comprises of software techniques to create, store, modify, represent pictures.

It is a method to display an image of any size on the computer screen using various algorithms and techniques.

* **Applications of computer Graphics :-**

1. **Education and training :-** computer generated model of the physical, financial and economics system is often used ~~physiological~~ as educational aids. or equipments help trainees to understand the operation of system.

   Eg:- flight simulator.

   AdV :- fuel saving, safety, etc.

2. **Use in Biology :-** Molecular biologist can display a picture of molecules and gain insight into their structure with the help of computer Graphics.

3. **Computer Generated Maps :-** Town planners and transportation engineers can use computer-generated maps which display data useful to them in their planning work.

4. **Presentation Graphics :-** Examples are - bar chart, line graph, pie chart and other displays showing relationship between multiple parameters.

5). **Computer Art:-** Computer Graphics are also used in the field of commercial arts. It is used to Generate television and advertising commercial.

6). **Others:-** Entertainment, visualization, Educational software, Printing tech.

* Interactive and passive Graphics:-

a) **Non-Interactive or Passive comp. Graphics:-**

In non-Interactive comp. Graphics, the picture is produced on the Monitor, and the user doesn't have any control over the image, i.e, the user cannot make any change in the rendered image.

Eg:- Titles show on T.V.

Non-Interactive Graphics involve only one-way commⁿ b/w the comp. and the user, User can see the produced image, but he can't make any change in the image.

b) **Interactive Computer Graphics:-** In Interactive computer Graphics user have some control over the picture, i.e, the user can make any change in the produced image.

Eg:- Ping-pong game.

Interactive computer graphics requires two-way communication b/w the computer and user. A user can see image and make any changes by sending his command with an input device.

Adv:-

1. High Quality.
2. More precise result or product.
3. Great productivity.
4. lower Analyse and design cost.
5. significantly enhance our ability to understand data and to perceive trends.

* Representative use / Application of comp. Graphics :-

1. Computer Art :- Using computer graphics we can create fine and commercial art which include animation packages, paint packages, cartoon drawing, paintings, logo design can also be done.

2. Entertainment :- computer graphics finds a major part of its utility industry (movie). In Game industry where focus and interactivity are the key players, comp. Graphics helps in providing such features in an efficient way.

3. Education :- Computer Generated Models are extremely useful for teaching huge number of concepts and fundamentals in an easy to understand and learn manner.

4. Training :-

Note :- Applications लिखे हो रखी है already.

* Types of Computer Graphics :-

1. Raster Graphics :- In raster, graphics pixels are used for an image to be drawn. It is also known as a bitmap image in which a sequences of images is into smaller pixels. Basically, a bitmap indicates a large no. of pixels together.

2. Vector Graphics :- In vector Graphics, mathematical formula are used to draw different types of shapes, lines, obj, and so on.

* Graphics hardware :- Graphics hardware is computer hardware that generates computer Graphics and allow them to be show on a display, usually using a graphics card (video card) in combination with a device driver to create the image on the screen.

Types of Graphics card :-

i) Integrated :- In-built with motherboard.

ii) dedicated :- external graphic card. More powerful than integrated.

* Graphics Software :- It is a type of comp. program that is used to create and edit image. There is a wide range of graphics software available on the market, ranging from simple programs that allow users to create detailed 3-D models and animation to basic images also.

Graphics software program is used to produced both Raster and vector images.

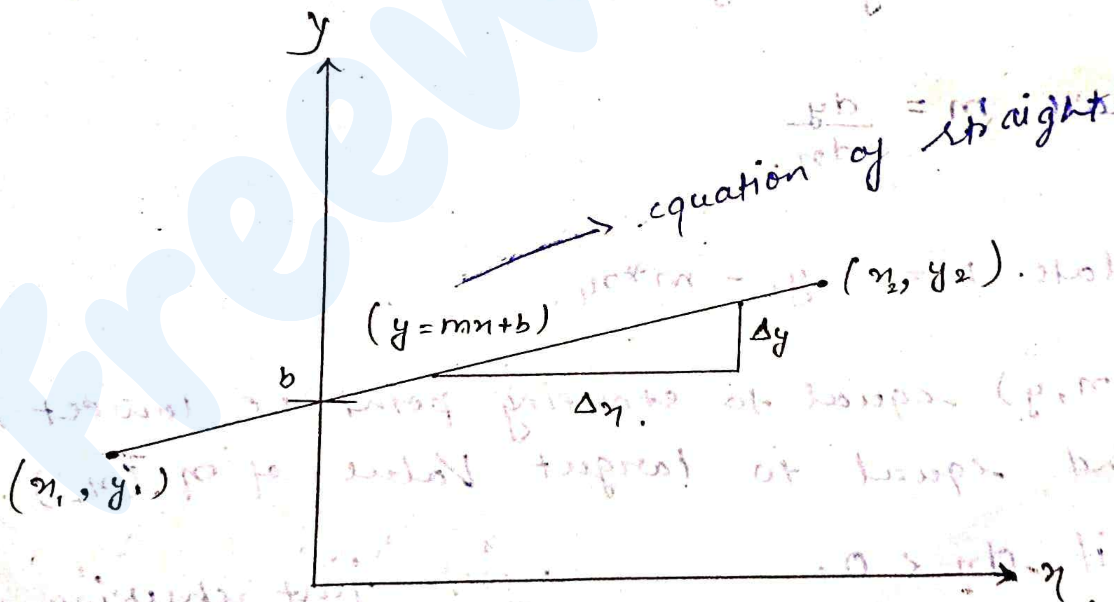
* Conceptual framework for Interactive Graphics :-

Conceptual framework has following elements.

- Graphics library -> between application and display hardware there is a graphics library / API.
- Application program -> An application program maps all the application obj. to image by invoking Graphics.
- Graphics System -> An interface that interacts b/w Graphics library and hardware.
- Modifications to image are the result of user interaction.

* Scan Converting a Straight line :-

A straight line may be defined by two endpoints and and eq^n. In fig. two end points are discribed by (x_1, y_1) and (x_2, y_2). The eq^n of line is used to determine the x, y co-ordinates of all the points that lie b/w these two end-points.



using the equation of straight line $y = mx + b$ where, $m = \frac{\Delta y}{\Delta x}$ & b = the y interupt, we can find value of y by incrementing x from $x = x_1$, to $x = x_2$. By Scan converting these calculated x, y values we represent the line as the sequence of pixel.

Note:- m is the slope (ढलान) of line. b where line intersect on y axis ↳ y-interupt.

* **Algo**:-

Step-1. Start Algo.

Step-2. Declare Variables. $x_1, x_2, y_1, y_2, dx, dy, m, b$.

Step-3. Enter values of x_1, x_2 and y_1, y_2.

$x_1, y_1 \rightarrow$ co-ordinates of starting point of straight line.

$x_2, y_2 \rightarrow$ co-ordinates of ending point of straight line.

Step-4. Calculate $dx = x_2 - x_1$

$dy = y_2 - y_1$.

Step-5. Calculate $m = \frac{dy}{dx}$.

Step-6. Calculate $b = y_1 - m * x_1$.

Step-7. set (x, y) equal to starting point. i.e lowest point and Xend equal to largest Value of x. (x_2).

if $dx < 0$.

than $x = x_2$.

$y = y_2$

$X_{end} = x_1$.

Note:- just checking which value of x is larger. and which is smaller if $dx > 0$

than $x = x_1$

$y = y_1$

$x_{end} = x_2$.

Step-8. check whether the complete line has been drawn if $x = x_{end}$, stop.

Step-9. Ploat a point at current (x, y) co-ordinates.

Step-10 Increment value of x, i.e., $x = x+1$.

Step-11 Compute next value of y from eq $y = mx+b$.

Step-12 Goto step-8.

Step-13. End.

* Program :-

```
#include< graphics.h>
#include < Stdlib.h >
#include < math.h >
# include < stdio.h >
# include< conio.h>
# include <iostream >
class breven {
      float x, y, x1, y1, x2, y2, dx, dy, m, b, Xend;
      Public :
            void  get();
            void  cal ();
};
```

```
void main () {
        breren b;
        b.get ();
        b.cal ();
       getch ();
 }.
 void   breren :: get() {
        Print
        cout<<"Enter Start and end points" << endl;
        cin >> x1 >> y1 >> x2 >> y2 ;
 }.
void  breren :: cal () {
        int gdriver = DETECT, gmode, errorcode;
      initgraph ( &gdriver, &gmode, " " );
      errorcode =  resultgraph(); ⟶ graphresult();
      if ( errorcode != grOK)
      {
           cout<<"Graphics Error :- " << grapherrormsg (errorcode);
           cout<<"Press any key to halt";
           getch();
           exit (1);
      }
      dx = x2 - x1;
      dy = y2 - y1;
      m = dy / dx;
      b = y1 - (m*x1);
      if (dx < 0) {
           x = x2;
           y = y2;
     }     xend = x1;
```

```
else {
    x = x1;
    y = y1;
    Xend = x2;
}.
while (x <= Xend) {.
        putpixel (x, y, RED);
        x++;
        y = (m*x) + b;
}.
}.
```

Methods :- Methods to scan converting a Straight line.

1. Direct method.
2. Simple DDA.
3. Incremental DDA
4. Integer DDA.
5. Bresenham's line Drawing.

line drawing
~~Direct~~ method :-

1. Direct Method :-

x	y
3	4
⋮	⋮
9	5

$(x_1, y_1) \rightarrow (3, 4)$

$(x_2, y_2) \rightarrow (9, 5)$.

$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{5-4}{9-3} = \frac{1}{6} < 1.$$

Cases.

$m \le 1$	$m > 1$
$x = x + 1$	$y = y + 1$
$y = mx + b$.	$x = (y - b)/m$.

x	y
3	4
4	$4.17 \to 4$
5	$4.33 \to 4$
6	$4.50 \to 5$
7	$4.66 \to 5$
8	$4.83 \to 5$
9	5

Note:- Find all values of y by putting x in eqn ($y = mx + b$).

And, $b = y_1 - (mx_1)$

$\Rightarrow 4 - \frac{1}{6} \times 3$

$\Rightarrow 4 - 1/2 = \frac{8-1}{2} = 7/2$.

$$\boxed{\begin{aligned} b &= 7/2 \\ m &= 1/6 \end{aligned}}$$

* Direct Method:-

Direct method.	Simple ~~~~ DDA — To reduce calculation.	Incremental DDA — To solve floating - Roll
i. $y = mx + b$	i. $y = mx + b$	i. $y = mx + b$
ii. $m = \frac{y_2 - y_1}{x_2 - x_1}$	ii. $m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{dy}{dx}$	ii. $m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{dy}{dx}$
iii. $m \le 1$: $x{+}{+}$, $y = mx + b$.	$m > 1$: $y{+}{+}$, $x = (y - b)/m$	iii. $m \le 1$: $x{+}{+}$, $y = y + m$.	$m > 1$: $y{+}{+}$, $x = x + \frac{1}{m}$	iii. Steps $= dx$ ($m \le 1$); steps $= dy$ ($m > 1$). $X = X + X_{inc}$. $Y = Y + Y_{inc}$. where, $X_{inc} = \frac{dx}{steps}$. $Y_{inc} = \frac{dy}{steps}$. ~~Steps~~

Note:- Comparison b/w different Algo's to drawing a line by comp. graphics.

* Bresnham's line drawing Algo :-

P (decision Parameter) => $2dy - dx$.

$$m = \frac{y_2 - y_1}{x_2 - x_1} \Leftrightarrow \frac{dy}{dx}.$$

- $m < 1$
  - $P < 0$
    - $x = x + 1$
    - $P = P + 2dy$
  - else.
    - $x = x + 1$
    - $y = y + 1$
    - $P = P + 2dy - 2dx$.
- else.
  - $P < 0$
    - $y = y + 1$
    - $P = P + 2dx$
  - else.
    - $x = x + 1$
    - $y = y + 1$
    - $P = P + 2dx - 2dy$.

Eg :- Scan converting a line from (1, 1) to (8, 5).

P	x	y
$P = 2dy - dx$. $\Rightarrow 8 - 7 = 1$.	(1)	(1).
$P = P + 2dy - 2dx$. $\Rightarrow 9 - 14 = -5$.	2	2.
	3	2.
$P = P + 2dy$. $\Rightarrow -5 + 8 = 3$.	3	2.
	4	3.
$P = 3$.	5	3.
$P = 5$	6	4.
$P = -1$	7	4.
$P = 7$	(8)	(5).

$dx = 7$

$dy = 4$

so, $m = 4/7$.

& $P = 1$.

Ans

* Scan Converting a circle :-

→ A circle is defined as set of points that all are at a same distance from common point.

→ It's an eight sided symmetrical figure divided into 4 Quadrants. By knowing only one point and reflecting it via every 45° axis, this symmetry aids constructing a circle on comp.

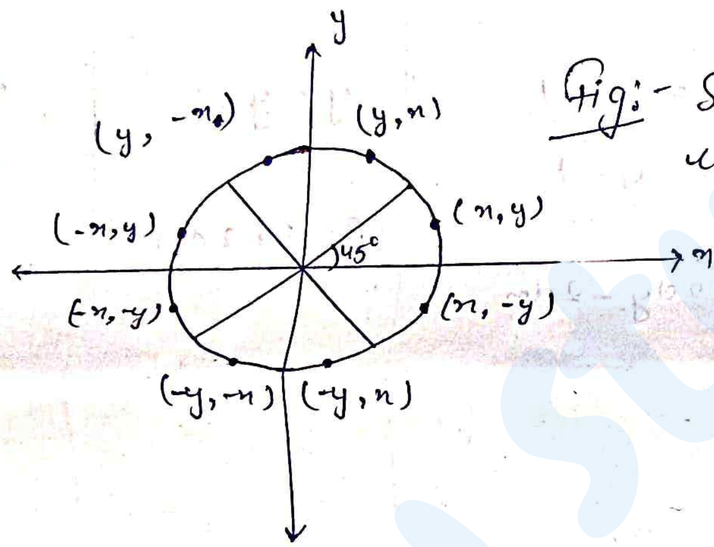


Fig:- Scan Conversion of circle.

* Methods :-

i) **Direct or polynomial method :-** In this method, a circle is defined with the help of a polynomial eqn. $\{(X-x)^2 + (Y-y)^2 = r^2\}$.

ii) **Polar co-ordinate method :-** In this method, the coordinates are converted into polar co-ordinate.

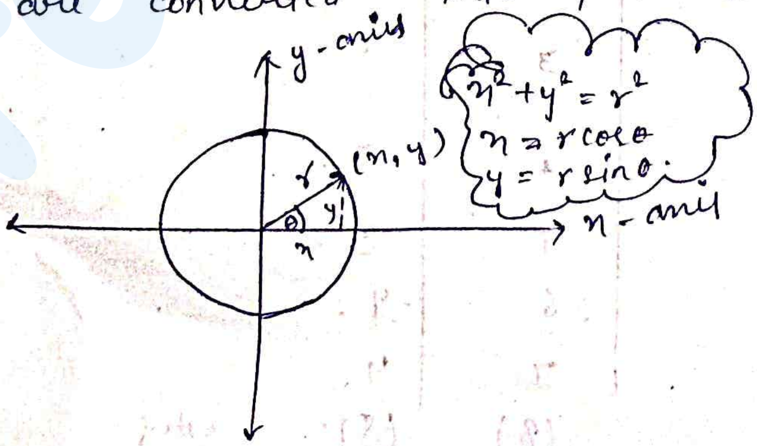


Fig:- polar form.

* Algo of Direct method :-

S-1 Input centre (x_c, y_c) and radius r.

S-2. Set initial value $X = x_c - r$, $Y = y_c$

S-3. Plot pixel $(\cancel{x}, y)$

S-4. while $(X \le x_c + r)$

increment $\cancel{x_c} = \cancel{x}+1$

Compute.

$y = y_c \pm \sqrt{r^2 - (x - x_c)^2}$ => {from the eqn of circle}

round off value of y

Plot pixel (X, Y)

End while.

S-5. Stop.

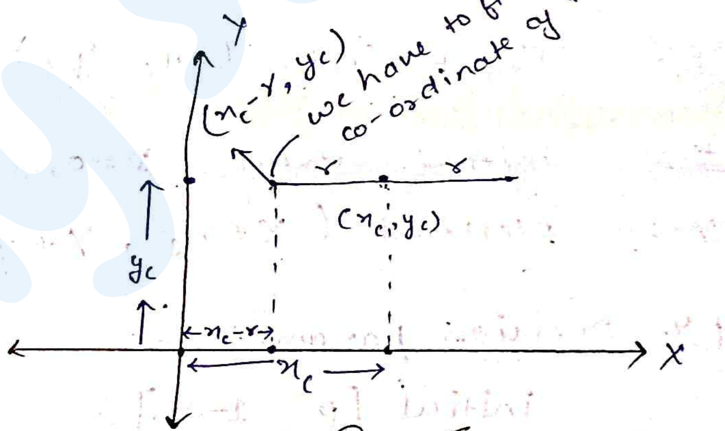


Fig:-

* Mid-point circle drawing Algo :-

use:- To avoid more calculation from polynomial / direct method.

Benifits.

i) It is difficult to draw a circle of given center (x_c, y_c) with radius r, so, we'll calculate pixels position at origin $(0, 0)$.

ii) In this method we just have to calculate the first quadrant pixels. since, the circle is symmetric

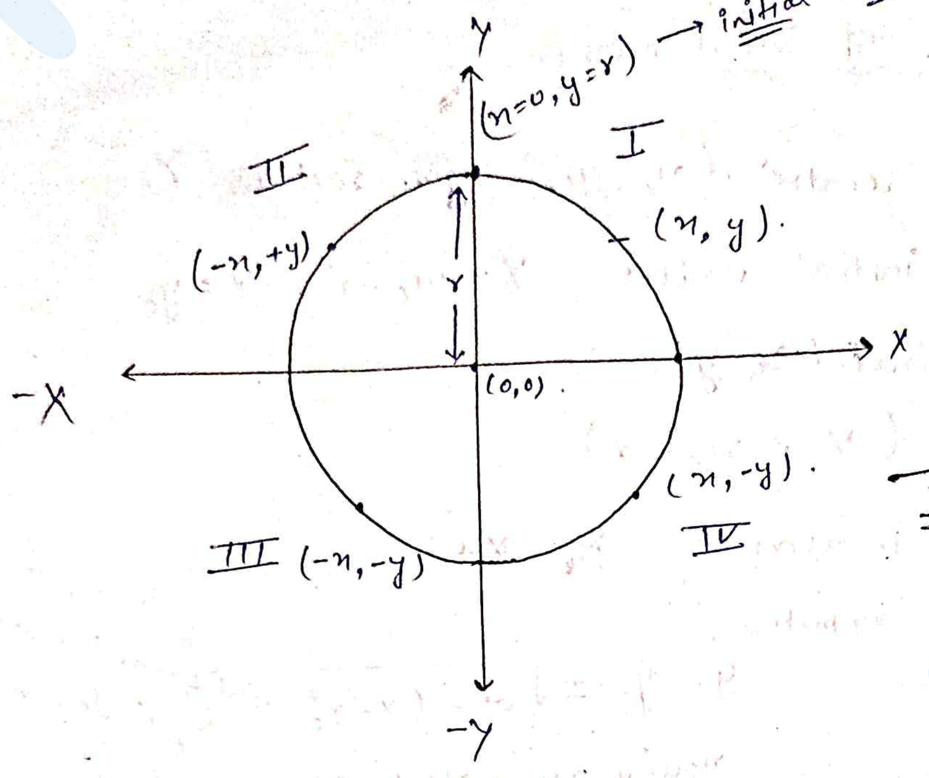


Fig:-

Algorithm :-

Step-1. Input centre (x_c, y_c) and radius r.

Step-2. initial value $X = 0$, $Y = r$.

Step-3. plotpixel $(X + x_c, Y + y_c)$ $\xrightarrow{\text{Note:-}}$ we work on origin but when plot on screen we add the x,y co-ordinate of circle.

Step-4. Decision parameter:-

initial $[P = 1 - r]$.

Step-5. while $X \le Y$.

if $p < 0$ then

$X = X + 1$

$P = P + 2X + 1$.

Plot pixel $(X + x_c, Y + y_c)$.

else $P \ge 0$.

$X = X + 1$

$Y = Y - 1$

$P = P + 2X + 1 - 2y$.

Plotpixel $(X + x_c, Y + y_c)$.

Endif.

End while.

Eg:- $(x_c, y_c) = (30, 40)$.

$r = 8$.

sol.

X	Y	P	$X + x_c$, $Y + y_c$		
0	8.	-7.	0+30,	8+40	(30, 48).
1	8	$P = P + 2x + 1$ ⇒ -4.	1+30,	8+40	(31, 48).
2	8	$P = P + 2x + 1$ ⇒ 1	2+30,	8+40	(32, 48).
3	7	$P = P + 2x + 1 - 2y$ ⇒ -6	3+30,	7+40	(33, 47).
4	7	$P = P + 2x + 1$ ⇒ 3.	4+30,	7+40	(34, 47).
5	6	$P = P + 2x + 1 - 2y$ ⇒ 2	5+30,	6+40	(35, 46).
6	5	$P = P + 2x + 1 - 2y$ ⇒ 5	6+30,	5+40	(36, 45).

All the pixels are:-

I	II	III	IV
(30, 48)	(-30, 48)	(-30, -48)	(30, -48)
(31, 48)	(-31, 48)	(-31, -48)	(31, -48)
(32, 48)	(-32, 48)	(-32, -48)	(32, -48)
(33, 47)	(-33, 47)	(-33, -47)	(33, -47)
(34, 47)	(-34, 47)	(-34, -47)	(34, -47)
(35, 46)	(-35, 46)	(-35, -46)	(35, -46)
(36, 45)	(-36, 45)	(-36, -45)	(36, -45).

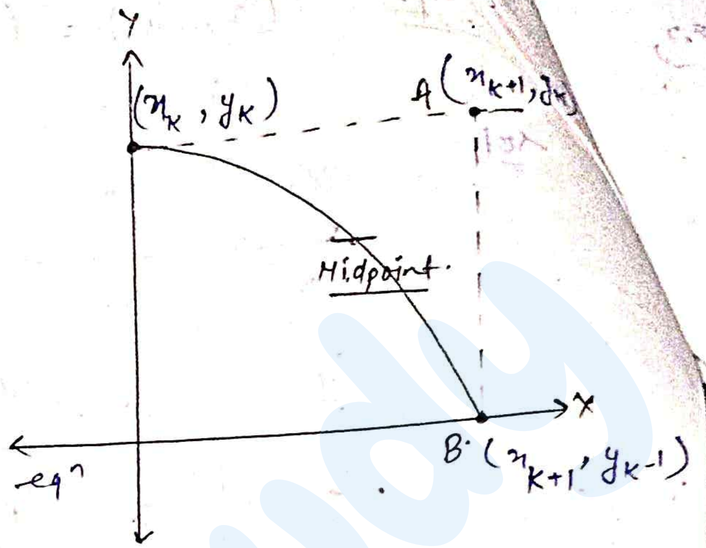
* Derivation :-

we know, eq of circle is

$$x^2 + y^2 = r^2.$$

or

$$x^2 + y^2 - r^2 = 0.$$



let, the co-ordinates of midpoint are (x_m, y_m). and equate in. eq^n of circle.

$(x_m)^2 + (y_m)^2 - r^2 = 0.$ → on solving we get -

Point. B. → 0 : points lies on circle.

point. A. → <0 : points lies inside the circle.

poin. B. → >0 : points lies outside the circle.

where,

$$x_m \Rightarrow \frac{x_{k+1} + x_{k+1}}{2} \quad \text{and} \quad y_m \Rightarrow \frac{y_k + y_{k-1}}{2}.$$

so, mid-points co-ordinates are :-

on solving!

$$\left(\frac{x_{k+1} + x_{k+1}}{2}, \frac{y_k + y_{k-1}}{2}\right) \Rightarrow \left(x_{k+1}, y_k - \frac{1}{2}\right).$$

After that,

we ~~select~~ compare the dis of point A and point B. from arc. or boundary and select the minimum one. if mid-point lies inside the arc, we select point A ~~xxx~~, outside the arc select the point B

$P_k \longrightarrow$ decision parameter.

$$\left[P_k = (x_k + 1)^2 + (y_k - \tfrac{1}{2})^2 - r^2 \right] \text{ 1st step}$$

$$\left[P_{k+1} = (x_{k+1} + 1)^2 + (y_{k+1} - \tfrac{1}{2})^2 - r^2 \right] \text{ 2nd step.}$$

now, subtract $P_{k+1} - P_k$.

and we know, x_{k+1} will not change so,

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

So, $P_{k+1} = \left((x_k+1)+1\right)^2 + \left(y_{k+1} - \frac{1}{2}\right)^2 - r^2$.

$$\Rightarrow (x_k+1)^2 + 1 + 2(x_k+1) + \left(y_{k+1} - \frac{1}{2}\right)^2 - r^2.$$

$$\Rightarrow (x_k+1)^2 + 2(x_k+1) + \left(y_{k+1} - \frac{1}{2}\right)^2 \left[-\left(y_k - \frac{1}{2}\right)^2 + \left(y_k - \frac{1}{2}\right)^2\right] - r^2 + 1.$$

$$\Rightarrow \underline{(x_k+1)^2} + 2(x_k+1) + \left(y_{k+1} - \frac{1}{2}\right)^2 - \left(y_k - \frac{1}{2}\right)^2 + \underline{\left(y_k - \frac{1}{2}\right)^2} - \underline{r^2} + 1.$$

$$\Rightarrow P_k + 2(x_k+1) + \left(y_{k+1} - \frac{1}{2}\right)^2 - \left(y_k - \frac{1}{2}\right)^2 + 1.$$

$$\Rightarrow P_k + 2(x_k+1) + 1 + \left(y_{k+1}^2 + \frac{1}{4} - \frac{\cancel{2}\, y_{k+1}}{\cancel{2}}\right) - \left(y_k^2 + \frac{1}{4} - \frac{\cancel{2}\, y_k}{\cancel{2}}\right).$$

$$\Rightarrow P_k + 2(x_k+1) + 1 + y_{k+1}^2 + \cancel{\frac{1}{4}} - y_{k+1} - y_k^2 - \cancel{\frac{1}{4}} + y_k.$$

$$\Rightarrow P_k + 2(x_k+1) + (y_{k+1}^2 - y_k^2) - (y_{k+1} - y_k) + 1.$$

$$\Rightarrow P_k + 2x_k + (y_{k+1}^2 - y_k^2) - (y_{k+1} - y_k) + 3.$$

if $P_k < 0$ then $y_{k+1} = y_k$.

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

$P_{k+1} = P_k + 2x_k + 3$. (put y_k in P_{k+1} eqn).

if $P_k \geq 0$ then

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

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

$P_{k+1} = P_k + 2x_k - 2y_k + 5$. (just put y_{k-1}).

Initial decision parameter:-

we know, initially $x_k = 0$

$y_k = r$.

we know,

$P_k = (x_k + 1)^2 + (y_k - \frac{1}{2})^2 - r^2$.

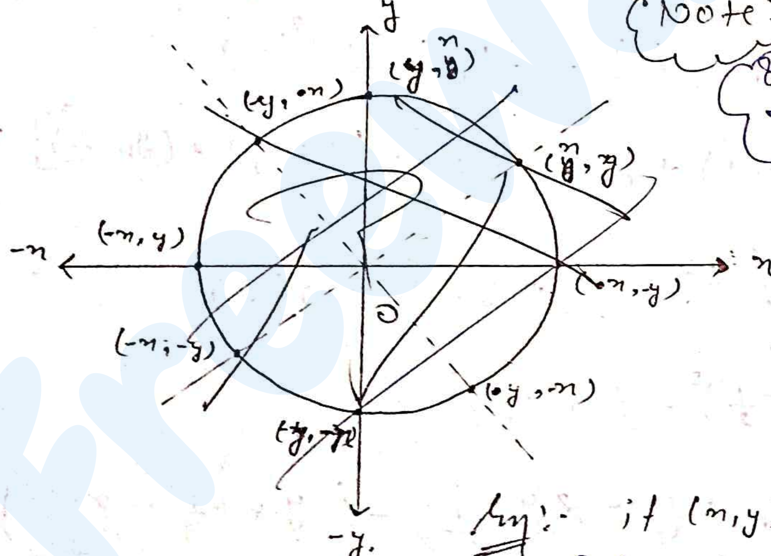
$= (0+1)^2 + (r - \frac{1}{2})^2 - r^2$.

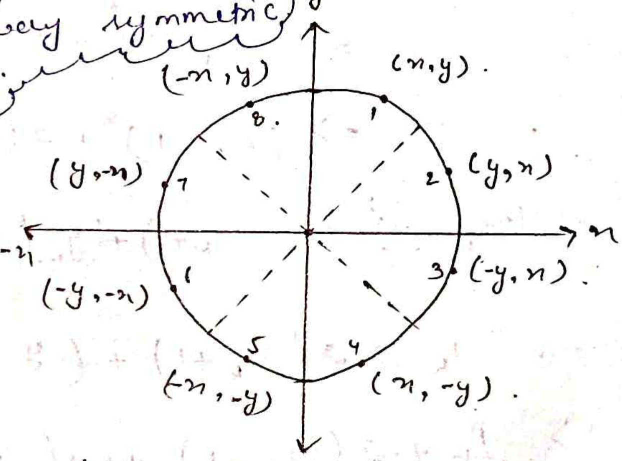
$= 1 + \cancel{r^2} + \frac{1}{4} - r - \cancel{r^2}$.

$\Rightarrow \frac{5}{4} - r. \Rightarrow 1.25 - r.$

$\Rightarrow 1 - r$. (after round off).

Note: Circle is 8-way symmetric fig.





Eg:- if (x,y) are (2,7) may:

P_1 (2,7) P_5 (-2,-7)

P_2 (7,2) P_6 (-7,-2)

P_3 (-7,2) P_7 (7,-2)

P_4 (2,-7) P_8 (-2,7)

we put $x_{k+1} = x_k + 1$ & $y_{k+1} = y_k$ if ($P<0$).

we know, $p_{k+1} = p_k + 2x_k + (y_{k+1}^2 - y_k^2) - (y_{k+1} - y_k) + 3$.

$\Rightarrow p_k + 2x_k + (\cancel{y_k^2} - \cancel{y_k^2}) - (\cancel{y_k} - \cancel{y_k}) + 3$.

$$\boxed{P_{k+1} \Rightarrow P_k + 2x_k + 3.}$$

else if ($P \geq 0$) so, $x_{k+1} = x_k + 1$ & $y_{k+1} = y_k - 1$.

$P_{k+1} = P_k + 2x_k + (y_{k+1}^2 - y_k^2) - (y_{k+1} - y_k) + 3$.

$\Rightarrow P_k + 2x_k + ((y_k - 1)^2 - y_k^2) - (\cancel{y_k} - 1 - \cancel{y_k}) + 3$.

$\Rightarrow P_k + 2x_k + (\cancel{y_k^2} + 1 - 2y_k - \cancel{y_k^2}) + 4$.

$$\boxed{P_{k+1} \Rightarrow P_k + 2x_k - 2y_k + 5.}$$

iv). Bresenham's circle drawing Algo :-

since, the circle is symmetric we divide 360° into 8 octants. and put the circle at Origin.

Algo:- [P (decision parameter) $\Rightarrow 3 - 2R$]:

$$\begin{bmatrix} X_c \Rightarrow 0 \\ Y_c \Rightarrow r \end{bmatrix}$$

Case I :- ($P < 0$)

$X_{k+1} = X_k + 1$

$Y_{k+1} = Y_k$

$P_{k+1} = P_k + 4x_k + 6$.

Case II :- ($P \geq 0$)

$X_{k+1} = X_k + 1$

$Y_{k+1} = Y_k - 1$.

$P_{k+1} = P_k + 4x_k - 4y_k + 10$.

$\rightarrow$ till ($x_{k+1} \leq Y_{k+1}$)

* **Ellipse** :- Ellipse is a four-way symmetry rather than eight-way.

a :- ~~diameter~~ radius (big). Major axis.

b :- ~~diameter~~ radius (small) minor axis.

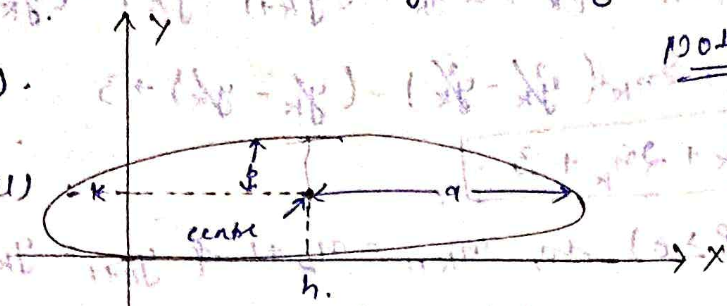


fig :- Ellipse

Note :- Eccentricity of ellipse

$[e = c/a]$ → focal length ; a → semi-major axis length.

(e is always > 1.)

$(c^2 = a^2 - b^2)$

so,

$$e = \frac{\sqrt{a^2 - b^2}}{a^2} \text{ or } \left(\sqrt{1 - \frac{b^2}{a^2}}\right)$$

Methods :-

i). **Polynomial Method** :- The ellipse has a major & minor axis a_1 & b_1. The centre of ellipse is (i, j).

The value of x will be ↑ till $i == a_1$ & value of y will be calculated using —

$$y = b_1 \times \sqrt{1 - \frac{x^2}{a_1^2}} + j. \quad / \quad y = b_1 \sqrt{1 - \frac{x - i}{a_1^2}} + j.$$

[eqⁿ of ellipse :- $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1.$]

or

$(c^2 = a^2 - b^2)$

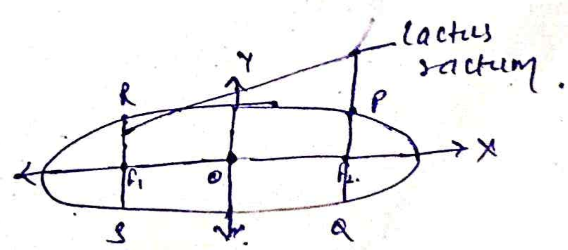
Area of ellipse ⇒ π × Semi-major axis (a) × Semi-minor axis (b).

↳ (πab)

Perimeter ⇒ ~~$2\pi \times \sqrt{\frac{a^2 + b^2}{2}}$~~

$$2\pi \times \sqrt{\frac{a^2 + b^2}{2}}.$$

Latus rectum ⇒ $L = \frac{2b^2}{a}.$



Eqn of ellipse:-

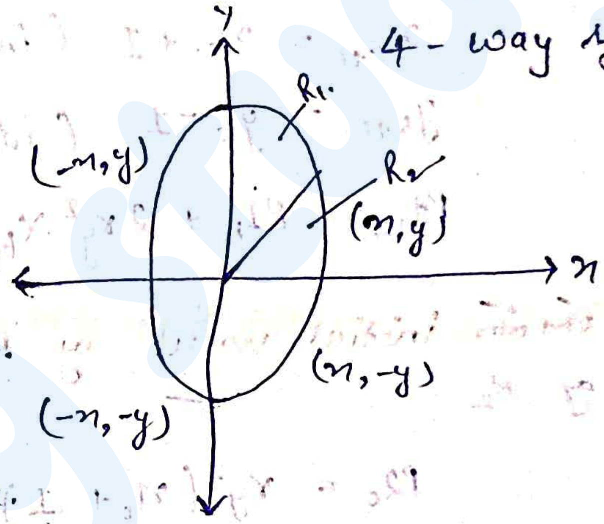
i) Centre at Origin & major axis along x-axis:-

$$\left\{ \frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \right\}$$

ii) Centre at Origin & major axis along y-axis:-

$$\left\{ \frac{y^2}{a^2} + \frac{x^2}{b^2} = 1 \right\}$$

* Mid-point ellipse:- 4-way symmetry



i) Function of ellipse:-

$$\left\{ F(x,y) = r_y^2 x^2 + r_x^2 y^2 - r_x^2 r_y^2 \right\}$$

if $F(x,y) < 0$, then (x,y) are inside the ellipse.

if $F(x,y) > 0$, then (x,y) are outside the ellipse.

if $(x,y) = 0$, then (x,y) are on ellipse.

ii) Decision parameter:-

in $R_1 \leftarrow P_1 \Rightarrow r_y^2 + \frac{1}{4 r_x^2} - r_x^2 r_y$, $P_2 \rightarrow$ initialize with last points of P_1.

* Algo:-

S-1. Take input radius along x and y axis and obtain centre of ellipse.

S-2. Initially we assume ellipse to be the centred at origin & first point as $[(x_0, y_0) = (0, r_y)]$

S-3. Now, Decision parameter for R_1.

$$\left[P1_0 = r_y^2 + \frac{1}{4}r_x^2 - r_x^2 r_y\right]$$

for each x_k position in R_1 :-
if $P1_k < 0$.

$$x_{k+1} = x_k + 1 \quad (\text{increment by } 1)$$

$$y_{k+1} = y_k \quad (\text{same})$$

$$P1_{k+1} = P1_k + 2r_y^2 x_{k+1} + r_y^2$$

while ($dx < dy$)

else

$$x_{k+1} = x_k + 1 \quad (\text{increment by } 1)$$

$$y_{k+1} = y_k - 1 \quad (\text{decrement by } 1)$$

$$P1_{k+1} = P1_k + 2r_y^2 x_{k+1} - 2r_x^2 y_{k+1} + r_y^2$$

S-4. obtain initial value of Region 2 using the last point (x_0, y_0) of R_1.

$$P2_0 = r_y^2\left(x_0 + \frac{1}{2}\right)^2 + r_x^2(y_0 - 1)^2 - r_x^2 r_y^2$$

for each y_k position in R_2 :- $(K=0)$

if $P2_k > 0$

while ($y >= 0$).

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

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

$$P2_{k+1} = P2_k - 2r_y^2 x_{k+1} + r_x^2$$

else

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

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

$$P2_{k+1} = P2_k + 2r_y^2 x_{k+1} - 2r_x^2 y_{k+1} + r_x^2$$

obtain this in all four quadrant until $dx < (2r_y^2 x >= 2r_x^2 y) \; dy$ for R_1.

Unit-Two.

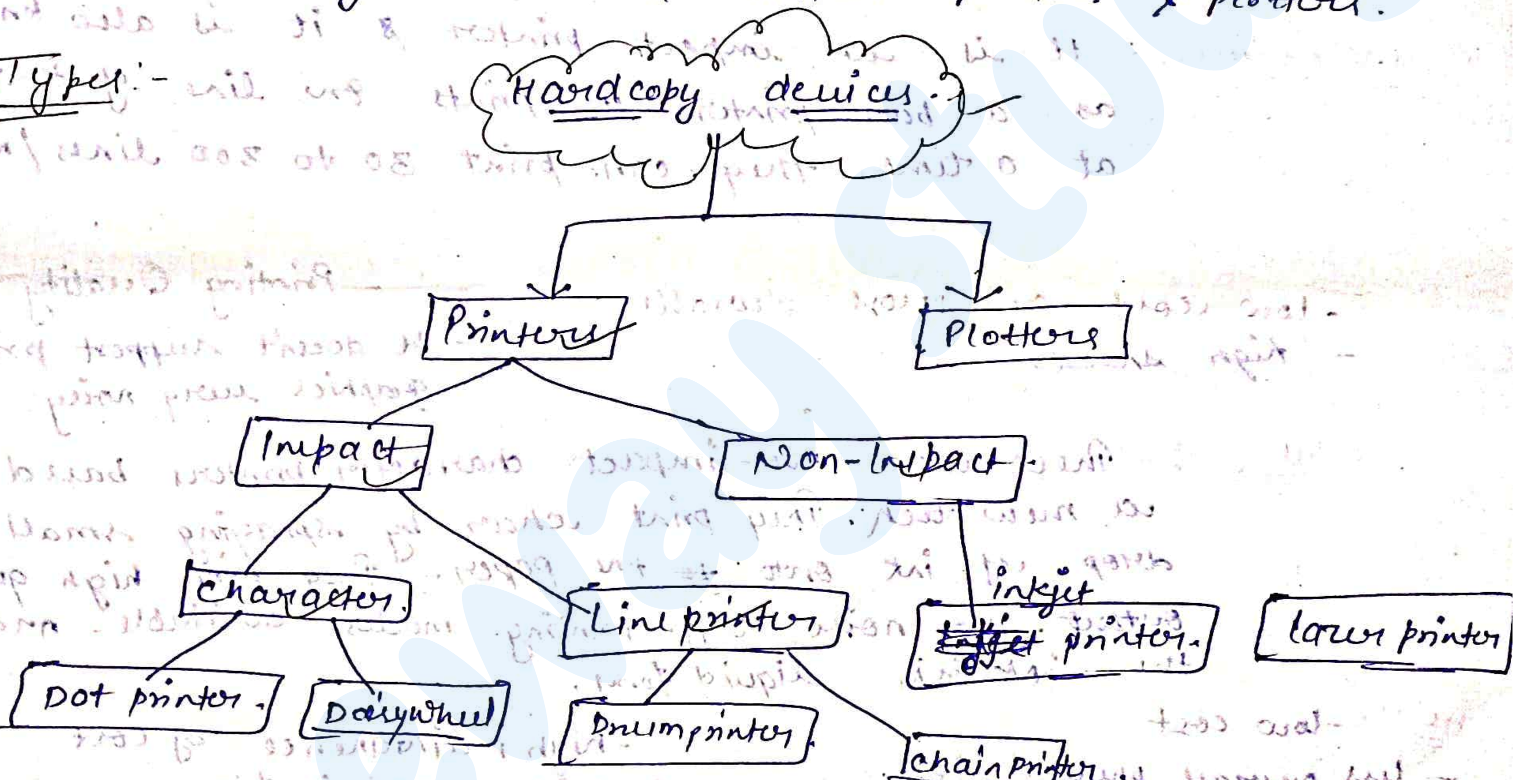
* Hardcopy devices:-

A hardcopy is a printed copy of info. from a comp. sometimes it is referred to as printer so it is called hard copy bcoz it exist as physical object.

Ex:- prinouts, microfilms, etc.

hardcopy output devices are the electromagnetic devices, which accepts data from comp. & translates them into forms understood by users. Ex: devices: printers & plotters.

Types:-



*: Printers:- It is the most imp. output device which is used to print data on paper. It is imp. in comp. Graphics bcoz comp. graphics have ultimately utilization in printed form.

- Impact :- It's an type of printer that works by direct contact with an ink ribbon with paper. These are typically loud but remain in use today because of their unique ability.
- Non-impact :- It's an type of printer that doesn't hit or impact a ribbon to the printer. They use xerographic, electrostatic technologies. They are generally much quiter.

- **Dot Printers:-** - most popular, economical, each char is in the form of patterns of dots, oldest tech, slow in speed.

- **Daisy wheel Printer:-** It is called daisy wheel Printers bcoz its mechanism looks like a daisy, at the end of each petal is a fully formed char. These are generally used for word processing.

Adv:- - More reliable
- better quality
- font can be changed easily.

Dis:- - noise
- More expensive.

- **line printer:-** It is an impact printer & it is also known as a bar printer. It prints one line of text at a time. they can print 30 to 300 lines/min.

Adv:-
- low cost and more durable.
- high speed.

Dis:- - Printing Quality is low
- It doesn't support printing graphics very noisy.

- **Ink-jet:-** These are non-impact character printers based on a new tech. They print char by spraying small drop of ink onto ~~to~~ the paper. They print high quality output. No noise, diff. printing modes available. and the ink in in liquid form.

Adv:- - low cost
- less overall physical size.
- high quality of printing.
- easy to use.

Dis:- - high maintanence of cost
- cartridges cost is high
- printing speed is less than laser printer.

- **Laser:-** These are non-impact printers which use laser lights to produced the dots needed to form a char to be printed on page. This type of printers make use of a focused beam of light to transfer text & images on paper.

Adv:- - not noisy, user friendly.
- fast, cost is low.
- Can print both text and img.

Dis:- - Costly.
- can't print multiple copies.
- These aren't economical.

Impact	Non-Impact
- produced char. & graphics on a piece of paper by striking is called impact printers.	- A type of printer that produces char. & graphics on a piece of paper without striking.
- It prints by hammering a set of metal pins.	- Printing is done by depositing ink in array form.
- electromagnetic devices are used.	- No electromagnetic devices are used.
- faster speed - 250 words/sec.	- slower speed - 1 page / 30 sec.
- Ex:- dot printers, line printers.	- Ex- Inkjet printers, Laser printers.

* Plotters :- A plotter is a special output device used to produced hard copies of large graphics and design on paper like - contributed maps & eng. drawings. plotter is either a peripheral device which you add to your comp. sys.

App:- - Architectural plans of buildings.
- CAD app. like design of aircraft.
- Many. eng. app.

Adv:- - can produce high quality output on large sheet.
- used to provide the high precession drawing.
- speed of producing output is high.

Types:- Drum plotter:- It consists of a drum. The paper on which design is made is kept on the drum. The drum can rotate in both dir^s. Plotters comprises one or more pen & a pen holder. The holders are mounted ⊥ to drum surface. Pen kept in holder & can move left to right & vice versa.

Flatbed Plotters:- It is used to draw complex designs & graphics. It can kept on table & consist of pen & holder. The pen can draw char. of various size. Multiple-pens. each pen have diff. color which helps to produce image of multi color.

* **Display Devices**:- These are the output devices used to represent the info. in the form of image (visual form). Display sys. are mostly called a video monitor. The purpose of display tech. is to simplify info. sharing

- CRT
- Flat panel Display.
- LCD
- Active Matrix TFT.
- MICRO Mirror devices. , etc

1. **CRT**:- CRT stands for cathode ray tube. It is a technology which is used in traditional comp., monitor & television.

It is a particular type of vaccume tube that displays images when an electron beam collides on the radiant surface.

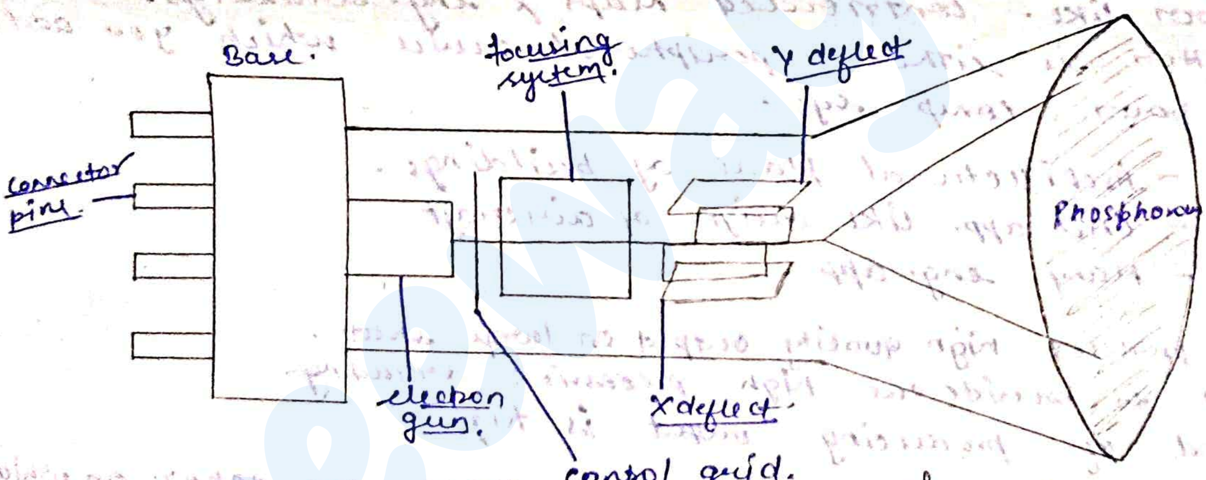


fig:- CRT

* **Working**:- खुद लिख लियो।
* **Components**:- i) **electron gun**:- It is made up of several elements, mainly a heating filament (heater) & a cathode. The electron gun is a source of electrons focused on a narrow beam facing the CRT.

ii) **Focusing and Accelerating Anodes**:- These are used to produce a narrow & sharply focused beam of electrons.

iii) Horizontal & Vertical deflecting plates :- These plates are used to guide the path of the electron beam. The plates produce an electromagnetic field that bends the electron beam through the area as it travels.

iv). Phosphorous Coated screen :- The phosphorous coated screen is used to produce bright spot when the high-velocity electron beam hits it.

* Raster Scan Display :- Raster Scan displays are most common type of monitor which employs CRT. It is based on television tech. In Raster Scan System electron beams sweep across the screen, from top to bottom covering one row at a time. A pattern of illuminated patterns of spots is created by turning beam ~~exp~~ intensity on and off as it moves across each row.

A memory area called Refresh buffer or frame buffer (Raster/bitmap) stores picture definitions. This memory area holds intensity values for all screen points. Stored Intensity values are restored from frame buffer and painted on screen one row at a time. Each screen point is referred to as pixels.

Refresh rate :- 60 - 80 frames / second.

At the end of each line scan, electron beams begins to display next scan line after returning to left side of screen.

The return to the left of screen after refresh of each scan line is known as horizontal retrace of electron beam. At the end of each frame electron beams returns to the top left corner & begins the next frame.

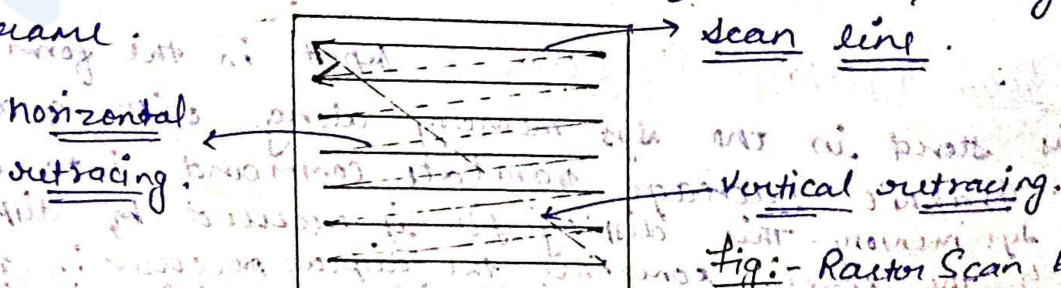


fig:- Raster Scan Display.

* **Raster Scan Display Processor:-** An imp. func of display processor is to digitize a picture definition given in an app. program into a set of pixels intensity value for storage in refresh buffer. This process is referred to as scan conversion. The purpose of display processor is to relieve the CPU from graphics jobs.

It can also perform various other tasks like:-
- creating diff. line styles.
- displaying color area.
- utilized to interface input devices.

* **Random - Scan Display:-** In Random Scan display electron beam is directed only to the areas of screen where a picture has to be drawn. It is also called vector displays. It can draw & refresh components lines of a pict. in any specified seq.

Eg:- Pen-Plotter.

The no. of lines regulates refresh rate on random-Scan display. An area of memory called refresh display file. store picture defn as a set of line drawing commands. Sys. returns back to first line command after all commands have been processed.

High-quality vector sys. can handle around 100,00 short lines at this refresh rate. Faster refreshing can burn the phosphorous. To avoid this every refresh cycle is delayed to prevent refresh rate greater than 60 frames per sec.

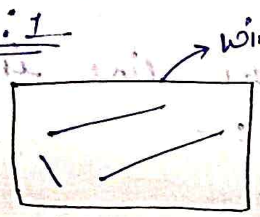
* **Random Scan Display Processor:-** Input in the form of app. prog is stored in the sys memory along with graphics package. Graphic package translate command into display file store in sys memory. This display file is accessed by display processor to refresh the screen. Sometime the display processor in random scan display is referred as display process. unit / Graphics Controller.

* <u>Clippling</u> :- when we have to display a large portion of picture, then not only scaling & translation is necessary, the visible part of pict. is also identified. This process is not easy, certain parts of the picture are inside while others are partially inside. The lines or elements which are partially visible will be omitted.

For deciding the visible & invisible portion, a particular process is used called clipping. clipping determines each elements into the visible & invisible portion. visible portion is selected & invisible portion is discarded.

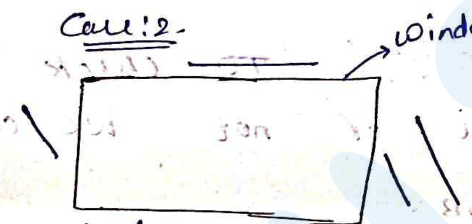
<u>Types of lines on screen</u> :-

<u>Case : 1</u>



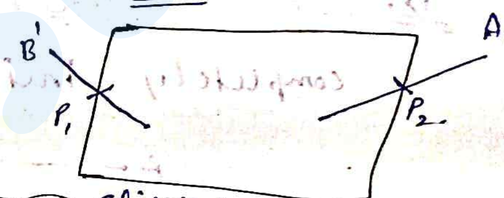
visible lines

<u>Case : 2</u>



Invisible lines.

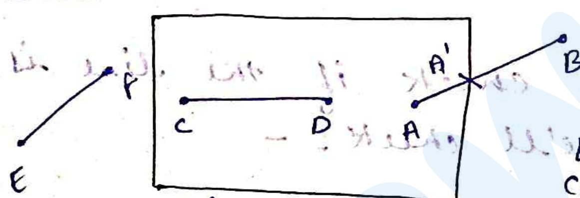
<u>Case : 3</u>



clipped lines

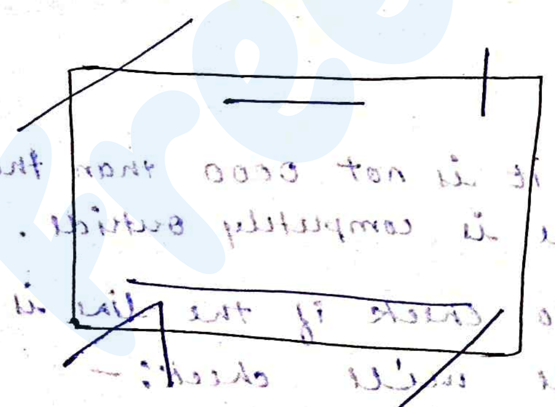
Note :- line P_2 to A_1 & P_1 to B' are discared or clipped.

<u>Case : 4</u>

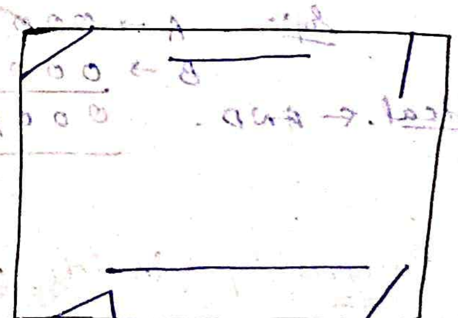


Mixed lines

EF → Invisible
CD → visible
A'B → clipped
AA' → visible



Origional pict / before clipping.

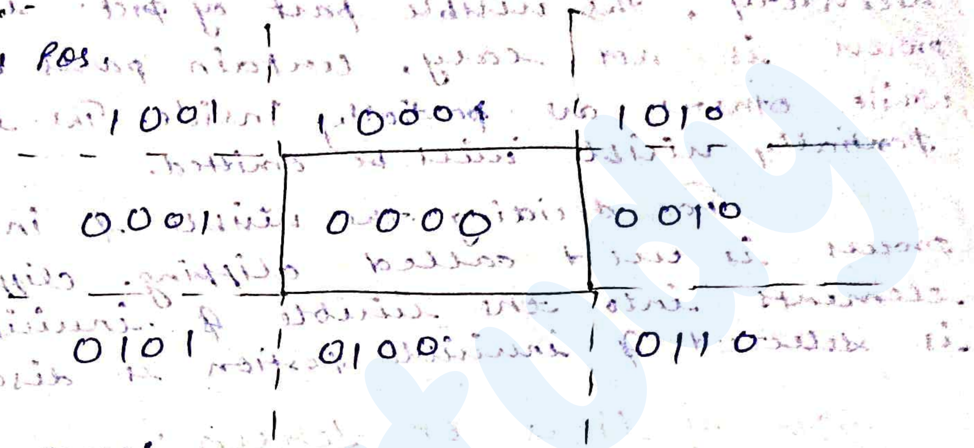


After clipping :-

* Sutherland - Cohen Algo :-

4-bit regional code :- which tells about the pos of window. (TBRL)

1-bit → Top.
2-bit → bottom.
3-bit → ~~left~~ Right
4-bit → ~~Right~~ left



Now, there are three cases :-

i). line completely Inside :- To check if the line is completely inside or not we check :-

A ——— B

if both the points A & B having the region code (0000) → inside the window.

ii). line completely outside :- To check if the line is completely outside or not we'll check :-

A ——— B

the regional code for A & B.

Eg:- A → 0001
B → 0001

logical AND. 0001 ——→ if it is not 0000 than this line is completely outside.

iii). Partially inside/outside :- To check if the line is Partially inside/outside will check :-

A ——— B

the regional code for A & B.

Ex:- $A \rightarrow 0001$

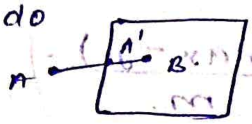
$B \rightarrow 1000$.

logical AND. $0000 \rightarrow$ partially inside/outside.

In first case we'll accept the whole line.

In second case we'll reject the whole line.

but in third case. we have to find that intersecting point from where we'll differentiate the acceptance & rejection of line. To do so we'll apply a Algo:-

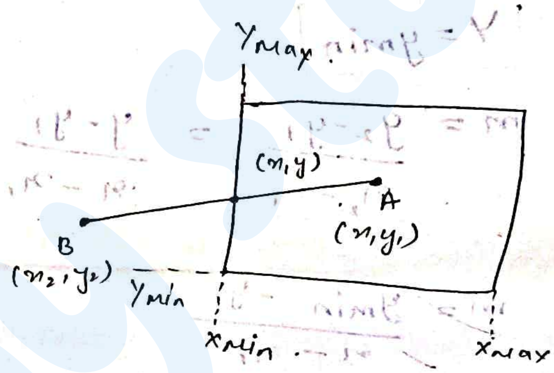


A'B — ✓

AA' — ✗.

Algo:-

→ Left:-



$$\boxed{X = x_{min}}$$

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

$$m \Rightarrow \frac{y - y_1}{x - x_1} \Rightarrow \frac{y - y_1}{x_{min} - x_1}$$

$$\Rightarrow y - y_1 = m(x_{min} - x_1)$$

$$\boxed{y = y_1 + m(x_{min} - x_1)} \qquad (x_{min},\ y_1 + m(x_{min} - x_1)).$$

→ Right:- $\boxed{X = x_{max}}$.

$$m = \frac{y_2 - y_1}{x_2 - x_1} \Rightarrow \frac{y - y_1}{x - x_1}$$

$$m = \frac{y - y_1}{x_{max} - x_1}$$

$$\boxed{y = y_1 + m(x_{max} - x_1)} \qquad (x_{max},\ y_1 + m(x_{max} - x_1)).$$

→ **Top** :-

$$\boxed{Y = Y_{max}}$$

$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{y - y_1}{x - x_1}$$

$$m = \frac{y_{max} - y_1}{x - x_1}$$

$$\boxed{x = x_1 + \frac{(y_{max} - y_1)}{m}} \quad \left(y_{max}, x_1 + \frac{(y_{max} - y_1)}{m}\right)$$

→ **bottom** :-

$$\boxed{Y = y_{min}}$$

$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{y - y_1}{x - x_1}$$

$$m = \frac{y_{min} - y_1}{x - x_1}$$

$$\boxed{x = x_1 + \frac{(y_{min} - y_1)}{m}} \quad \left(y_{min}, x_1 + \frac{(y_{min} - y_1)}{m}\right).$$

Note:- In case of corner we can visually find the points (just using x(min, max) & y(min, max)).

Q. BL: (1, 1)
UR: (7, 8)
line: (5, 2) (9, 6).

Sol:- $X_{min} = 1$, $Y_{min} = 1$.
$X_{max} = 7$, $Y_{max} = 8$.

Now, firstly check if line is totally inside or not.

$x_1 = 5$ $\quad$ $x_2 = 9$
$y_1 = 2$ $\quad$ $y_2 = 6$

so, i) $x_1 \geq x_{min}$.

$5 \geq 1$ satisfied.

ii) $y_1 \geq y_{min}$

$2 \geq 1$ satisfied.

iii) $x_2 \leq x_{max}$

$9 \leq 7$. unsatisfied

means the line is partially visible.

and the line is partially clipped from Right.

so, Intersection points are –

$x = x_{max}$

$x = 7.$

and $y = y_1 + m(x_{max} - x_1)$

and $m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{6-2}{9-5} = \frac{4}{4} = 1.$

$y = 2 + 1(7-5)$
$= 2 + 2.$
$= 4. \longrightarrow$ cordinates are $(7, 4)$.

so, visible line will be

$(5, 2)$ to $(7, 4)$

Ans.

* Cyrus - Beck line clipping Algo :-

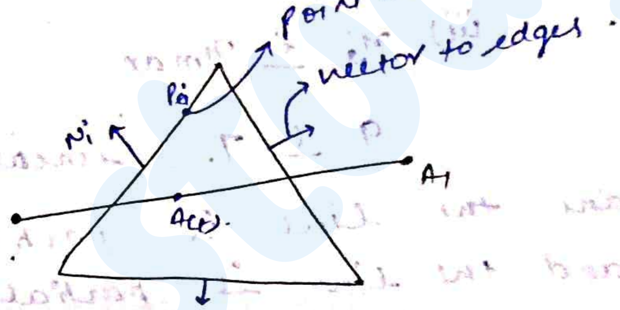
Dis-advantage in sutherland-cohen Algo :- There is only one disadvantage which is that the clipping window (polygon) can only be square or rectangle. It can't either be triangle or hexagon and so on.

Parametric eqⁿ :- $\left[A(t) = A_0 + t(A_1 - A_0) \right]$

Now,

$N_i(A(t) - P_{ei})$. ~~A(t)~~.

↳ less than 0 (inside CW).



if we perform :-

$N_i [A(t) - P_{ei}]$.

↳ < 0 , A(t) inside the clipping window.

↳ > 0 , A(t) is outside of clipping window.

↳ = 0 , A(t) is on CW means intersection point.

Now, we know,

$N_i [A(t) - P_{ei}] = 0$ and $A(t) = A_0 + t(A_1 - A_0)$

so, $N_i [A_0 + t_i(A_1 - A_0) - P_{ei}] = 0$

$N_i A_0 + N_i t_i A_1 - N_i t_i A_0 - N_i P_{ei} = 0$

$N_i t_i A_1 - N_i t_i A_0 = N_i P_{ei} - N_i A_0$

$t_i(N_i A_1 - N_i A_0) = N_i (P_{ei} - A_0)$.

only
window
be

$$t_i = \frac{N_i (P_{ei} - A_0)}{N_i A_1 - N_i A_0}$$

$$t_i = \frac{\cancel{\phi} (P_{ei} - A_0) N_i}{\cancel{\phi} (A_1 - A_0) N_i}$$

$$\boxed{t_i = \frac{(P_{ei} - A_0) \cdot N_i}{(A_1 - A_0) \cdot N_i}} \longrightarrow \begin{array}{l} d < 0 \text{ entering.} \\ d > 0 \text{ leaving.} \end{array}$$

Now, find the t_1, t_2, t_3 for the all edges e_1, e_2, e_3 respectively.

Case 1:-

$$e_1 \to t_1 \to \text{entering}$$
$$\left.\begin{array}{l} e_2 \to t_2 \to l \\ e_3 \to t_3 \to l. \end{array}\right\} \text{— take smallest}$$

Case 2:-

$$\left.\begin{array}{l} e_1 \to t_1 \to e \\ e_2 \to t_2 \to e \end{array}\right\} \text{— take greatest.}$$
$$e_3 \to t_3 \to \text{leaving.}$$

Now, we'll get two t's one of it's entering and the other one is leaving –

Now, by putting the value of t in

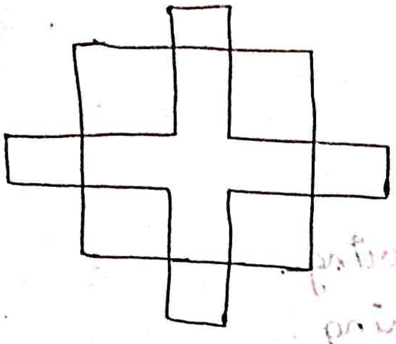
$$A(t) = A_0 + t(A_1 - A_0).$$

we get entering and leaving intersection coordinates.

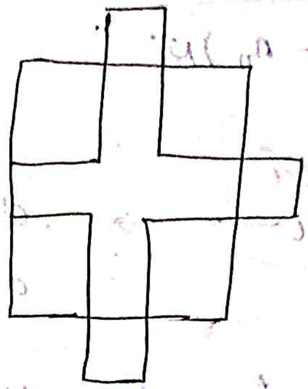
(8).

structure made by multiple lines

* Sutherland Hodgeman Polygon clipping Algo:-

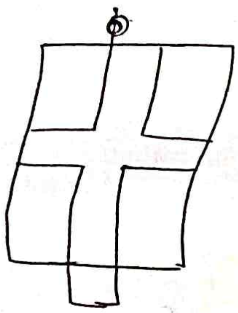


Origional pict.

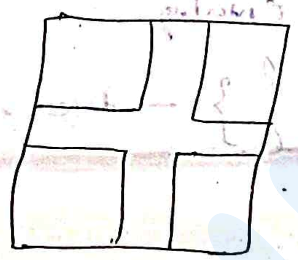


Left clipped.

Right clipped.



Top clipped.



bottom clipped.

some rules to follow:-

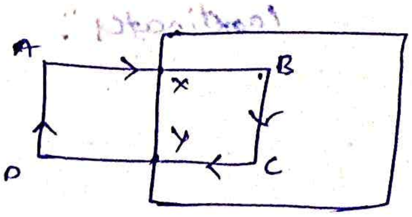
1). if points are

outside, O → I → (inside) take (intersection(x) to I). (save)

$I^{1st} \rightarrow I^{2nd}$ → take (I to I) save I^{2nd}.

I → O → take (I to Intersection (Y)) (save Y).

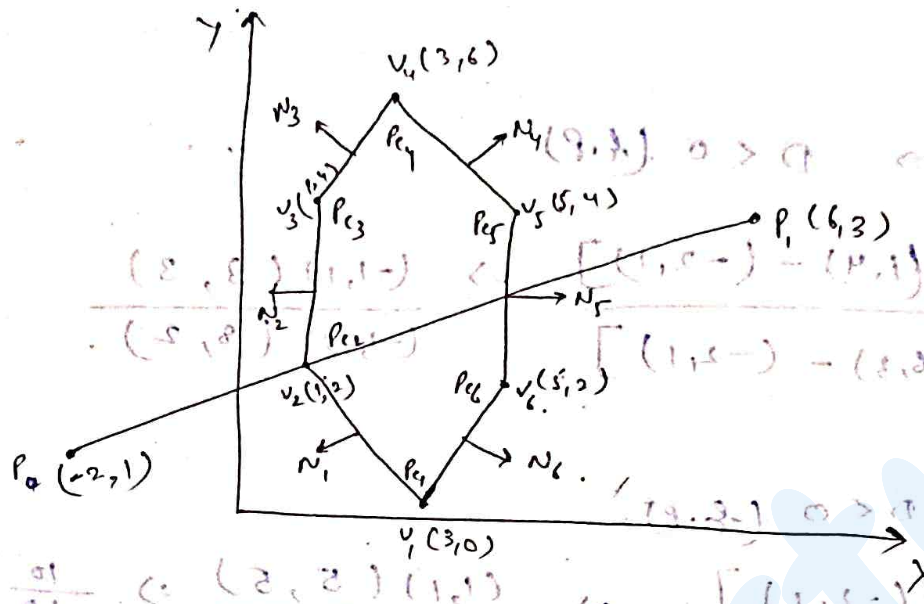
O → O → Ignore.



∴ now follow these steps for all 4 sides.

by follow these steps clipped the polygon and display the window.

Q.



sol. $N_i(x, y)$.

$N_1(-1,-1)$.

$N_2(-1,0)$ just direction of x & y axis:-

$N_3(-1,1)$

$N_4(1,1)$

$N_5(1,0)$

$N_6(1,-1)$

$A_1 = (6,3)$

$A_0 = (-2,1)$

$$\{A(t) = A_0 + t(A_1 - A_0)\}$$

$$\left\{ t_i = \frac{N_i(P_{ei} - A_0)}{N_i(A_1 - A_0)} \right\}$$

Now, for t_1 :-

$$t_1 = \frac{(-1,-1)[(3,0)-(-2,1)]}{(-1,-1)[(6,3)-(-2,1)]} \Rightarrow \frac{(-1,-1)(5,-1)}{(-1,-1)(8,2)}$$

$$\Rightarrow \frac{-4}{-10} \quad D<0 \ (E.P).$$

$$t_2 = \frac{(-1,0)\,[(1,2)-(-2,1)]}{(-1,0)\,[(6,3)-(-2,1)]} \Rightarrow \frac{(-1,0)(3,1)}{(-1,0)(8,2)}$$

$$\Rightarrow \frac{-3}{-8} \longrightarrow D<0 \;(\text{L.P})$$

$$t_3 = \frac{(-1,1)\,[(1,4)-(-2,1)]}{(-1,1)\,[(6,3)-(-2,1)]} \Rightarrow \frac{(-1,1)(3,3)}{(-1,1)(8,2)}$$

$$\Rightarrow \frac{0}{-6} \longrightarrow D<0 \;(\text{E.P})$$

$$t_4 = \frac{(1,1)\,[(3,6)-(-2,1)]}{(1,1)\,[(6,3)-(-2,1)]} \Rightarrow \frac{(1,1)(5,5)}{(1,1)(8,2)} \Rightarrow \frac{10}{10} \rightarrow d>0 \;(\text{L.P})$$

$$t_5 = \frac{(1,0)\,[(5,4)-(-2,1)]}{(1,0)\,[(6,3)-(-2,1)]} \Rightarrow \frac{(1,0)(7,3)}{(1,0)(8,2)} \Rightarrow \frac{7}{8} \rightarrow d>0 \;(\text{L.P})$$

$$t_6 = \frac{(1,-1)\,[(5,2)-(-2,1)]}{(1,-1)\,[(6,3)-(-2,1)]} \Rightarrow \frac{(1,-1)(7,1)}{(1,-1)(8,2)} = \frac{6}{6} \rightarrow d>0 \;(\text{L.P})$$

Now, greatest d for ~~...~~ entering & smallest d for leaving.

so, entering point $= \frac{4}{10}$ (t_1)

leaving point $= \frac{7}{8}$.

Now,

$$P(t) = (-2,1) + \frac{4}{10}\,[(6,3)-(-2,1)]$$

$$\Rightarrow (-2,1) + \frac{4}{10}(8,2) \Rightarrow (-2,1) + \left(\frac{32}{10}, \frac{8}{10}\right)$$

$$\Rightarrow \left(\frac{6}{5}, \frac{9}{5}\right) = P(t).$$

Now, $P(t)$ where $t = (7/8)$.

$$P(t) = (-2,1) + \frac{7}{8}\left[(6,3) - (-2,1)\right].$$

$$= (-2,1) + \frac{7}{8}\left[(8,2)\right]. \Rightarrow (-2,1)+(7, 7/4)$$

$$\Rightarrow \left(5, \ 11/4\right).$$

So, entering co-ordinates are – $\left(\frac{6}{5}, 9/5\right)$

and leaving co-ordinates are – $(5, 11/4)$.

* Mid-point Subdivision Algorithm :-

Sutherland - cohen line clipping Algo req. floating point cal. to find intersection point and we can avoid this cal. by mid-point subdivision Algo.

It repeatedly subdivides the line at its midpoint.

for line :-

- ↳ If line is completely visible → draw.
- ↳ If line is completely invisible → Ignore
- ↳ If line is partially visible than it is sub-divided into two equal parts
  - the visible test is applied on each half.
  - the subdivision process is repeated until we get completely visible or Invisible line segments.

condi:-

If region code for both end points: 0000

↳ line visible.

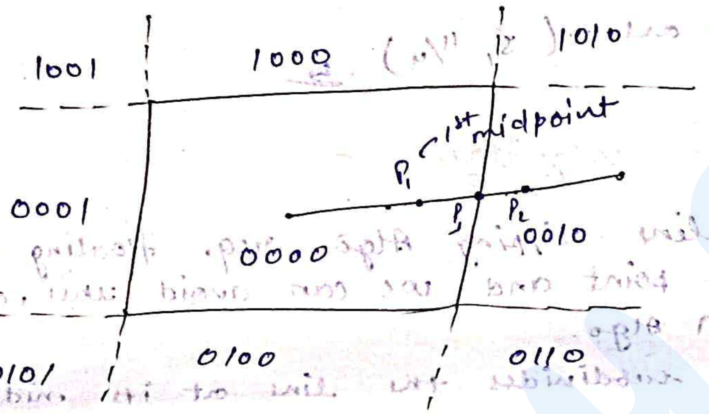
If region code for both end points after performing logical AND op. is also non-zero

↳ line invisible.

visibility Test

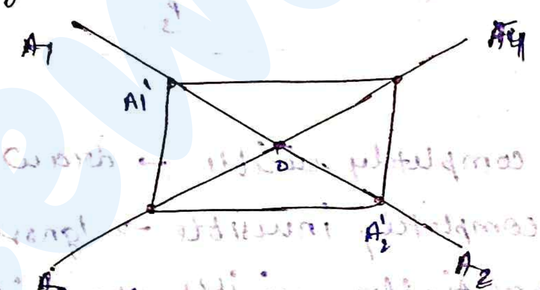
Else

↳ partially visible



After finding first mid point check for both & perform the midpoint subdivision on partial part again & again. After finding P_3 we get fully visible & Invisible line.

Q. Apply mid-point Subdivision Algo & clip the line segment to the window edge.



Sol!-

$A1$ to A_2 → $(A1, O)$ to $(O$ to $A_2)$

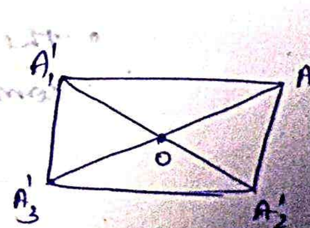
↳ $(A1'$ to $O)$ ↳ $(O$ to $A_2')$

vis. vis.

A_3 to A_4 → (A_3, O) to (A_4, O)

↳ (A_3', O) ↳ (A_4', O) → vis.

vis.



↳ After clipping.

~~Xxx~~-
Unit-3. Geometrical Transformation :-

Def^n :- computer graphics can provide the facility of viewing objects from different angles. ~~The architect can study~~. The purpose of using computers for drawing is to provide facility to user to view the obj. from diff angles, enlarging or reducing the scale or shape of object called as Transformation.

Point of view to describe obj. transform :-

i) Geometric Transformation :- The object is transformed related to system co-ordinate.

ii) Co-ordinate transformation :- The obj. is held stationary while the co-ordinate system is transformed relative to obj.

* Types :- i) Translation :- It is a straight line movement of an object from one position to another is called translation. Here, the obj. is positioned from one co-ordinate location to another.

Translation of point :- To translate a point from co-ordinate pos. (x, y) to another (x_1, y_1). we add the translation dis T_x, T_y to origional co-ordinates :

$$\begin{Bmatrix} x_1 = x + T_x . \\ y_1 = y + T_y . \end{Bmatrix}$$

here, T_x, T_y are shift vector.

line → end points translate

polygon → each vertex translate.

circle/ellipse → centre translate.

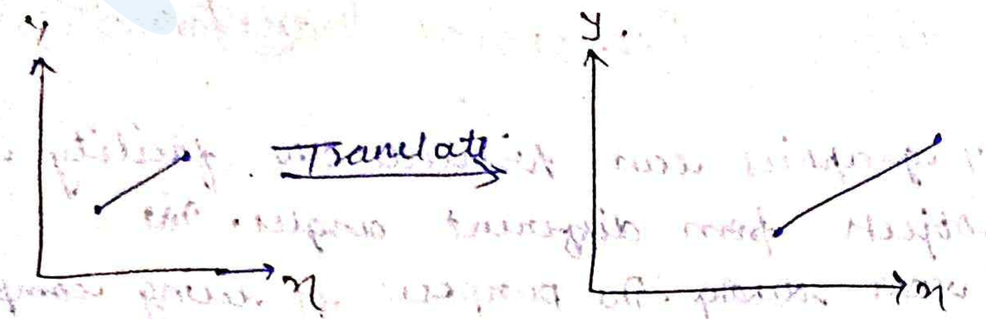


fig:- Translation of line

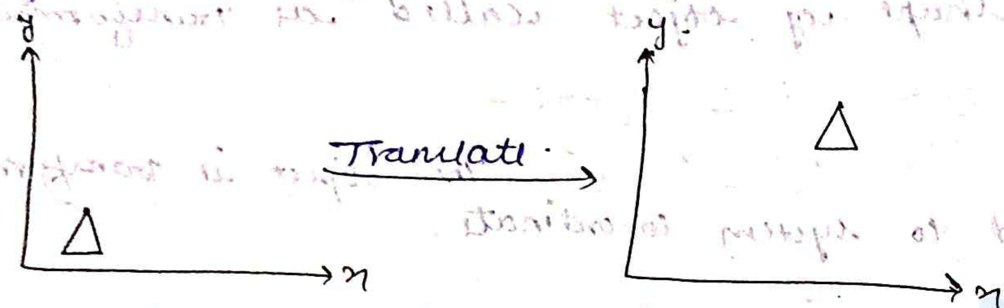


fig:- Translation of polygon.

Matrix of Translation :-

$$\begin{vmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{vmatrix} \quad or \quad \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{vmatrix}$$

Q. Point A(3, 4). translate it with translation factor $t_x = 3, t_y = 5$.

Sol. $A = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$, $T = \begin{bmatrix} 3 \\ 5 \end{bmatrix}$.

we know,

$A' = A + T$.

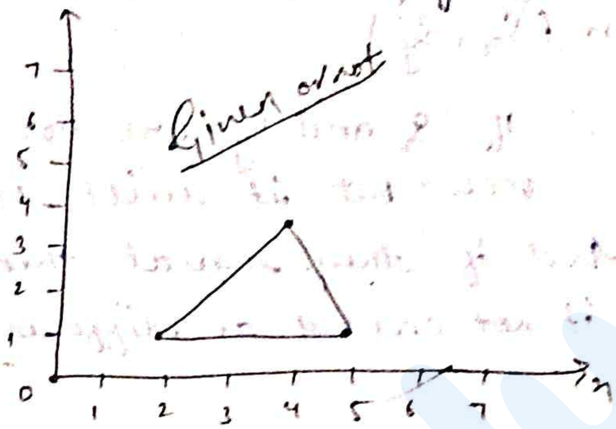
$A' = \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \end{bmatrix} + \begin{bmatrix} 3 \\ 5 \end{bmatrix} \Rightarrow \begin{bmatrix} 6 \\ 9 \end{bmatrix}$.

so, $x' = 6$.
$y' = 9$. Ans.

Q. Co-ordinates of polynomial $A(2,1)$; $B(5,1)$ & $C(4,4)$ with translation factor $t_x = 4$, $t_y = 2$.

sol. we know,

$$[A' = A + T]$$



for $A(2,1)$:-

$$A' = A + T$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix} + \begin{bmatrix} 4 \\ 2 \end{bmatrix} = \begin{bmatrix} 6 \\ 3 \end{bmatrix}.$$

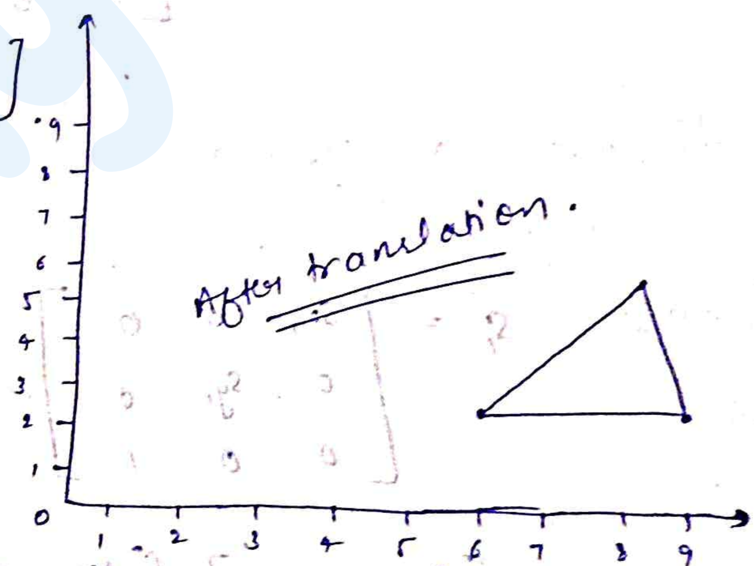
$x' = 6$
$y' = 3$

for $B(5,1)$:-

$$A' = A + T$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 5 \\ 1 \end{bmatrix} + \begin{bmatrix} 4 \\ 2 \end{bmatrix} = \begin{bmatrix} 9 \\ 3 \end{bmatrix}$$

$x' = 9$
$y' = 3$



for $C(4,4)$:-

$$A' = A + T$$

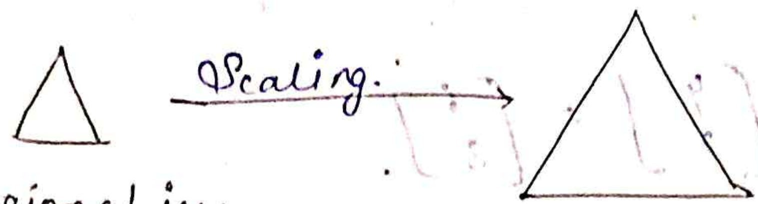
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 4 \\ 4 \end{bmatrix} \begin{bmatrix} 4 \\ 2 \end{bmatrix} = \begin{bmatrix} 8 \\ 6 \end{bmatrix}.$$

$x' = 8$
$y' = 6.$

New co-ordinates will be. $A(6,3)$, $B(9,3)$ & $C(8,6)$ Ans.

ii). Scaling :- It is used to alter or change the size of obj. The change is done using scaling factors (S_x, S_y).

Note :- If S_x and S_y are not equal than scaling will occur but it will elongate or distort the picture. And if they equal then it'll be uniform scaling. if not called as differential scaling.



$[A' = A \cdot S]$

fig:- origional img

fig:- enlarged img.

Matrix of scaling :-

$$\begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

here, S_x, S_y are scaling vectors.

Q. Prove that 2-D Scaling transformation are commutative i.e, $S_1 \cdot S_2 = S_2 \cdot S_1$.

Sol.
$$S_1 = \begin{bmatrix} S_{x_1} & 0 & 0 \\ 0 & S_{y_1} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad S_2 = \begin{bmatrix} S_{x_2} & 0 & 0 \\ 0 & S_{y_2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$S_1 \cdot S_2 = \begin{bmatrix} S_{x_1} & 0 & 0 \\ 0 & S_{y_1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S_{x_2} & 0 & 0 \\ 0 & S_{y_2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} S_{x_1} \cdot S_{x_2} & 0 & 0 \\ 0 & S_{y_1} \cdot S_{y_2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \longrightarrow (1)$$

So,

$$S_2 \cdot S_1 = \begin{bmatrix} S_{x_2} & 0 & 0 \\ 0 & S_{y_2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} S_{x_1} & 0 & 0 \\ 0 & S_{y_1} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} S_{x_2} \cdot S_{x_1} & 0 & 0 \\ 0 & S_{y_2} \cdot S_{y_1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{——— (ii)}$$

from eq^n (i) & (ii).

$$\left[S_1 S_2 = S_2 S_1 \right]$$ Hence proved.

for 2-D :- $A' = A \cdot S$

$$= \begin{bmatrix} x & y \end{bmatrix}_{1\times 2} \cdot \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix}_{2\times 2}$$

$$\Rightarrow \begin{bmatrix} x \cdot S_x & y \cdot S_y \end{bmatrix}$$

Q. Three points A(2,1), B(5,1) and C(4,4). Scale them with scaling factor $S_x = 3$ and $S_y = 2$ unit.

Sol:- We know,

$$[A' = A \cdot S]$$

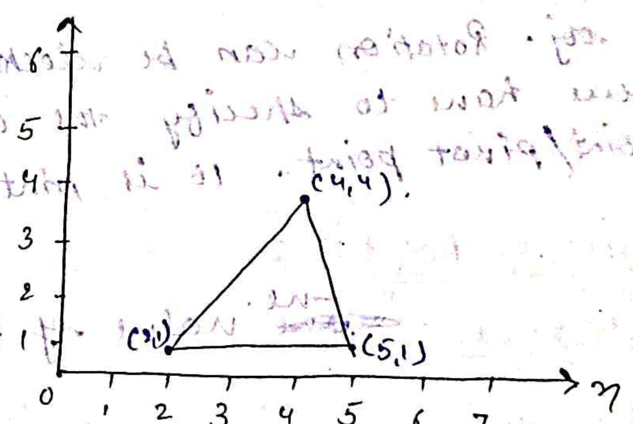


fig:- Origional figure :-

so, for A(2,1)

$$A' = \begin{bmatrix} 2 & 1 \end{bmatrix} \times \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix}$$

$$A[x' \; y'] = [6 \quad 2]$$

so, $x' = 6$ and $y' = 2$.

for B(5,1) :-

$A' = A \cdot S.$

$A = \begin{bmatrix} 5 & 1 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix}.$

$\begin{bmatrix} x' & y' \end{bmatrix} = \begin{bmatrix} 15 & 2 \end{bmatrix}.$

so, $x' = 15$
$y' = 2.$

for C (4,4) :-

$A' = A \cdot S.$

$A' = \begin{bmatrix} 4, & 4 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix}$

$\begin{bmatrix} x' & y' \end{bmatrix} = \begin{bmatrix} 12 & 8 \end{bmatrix}.$

So, $x' = 12$
$y' = 8.$

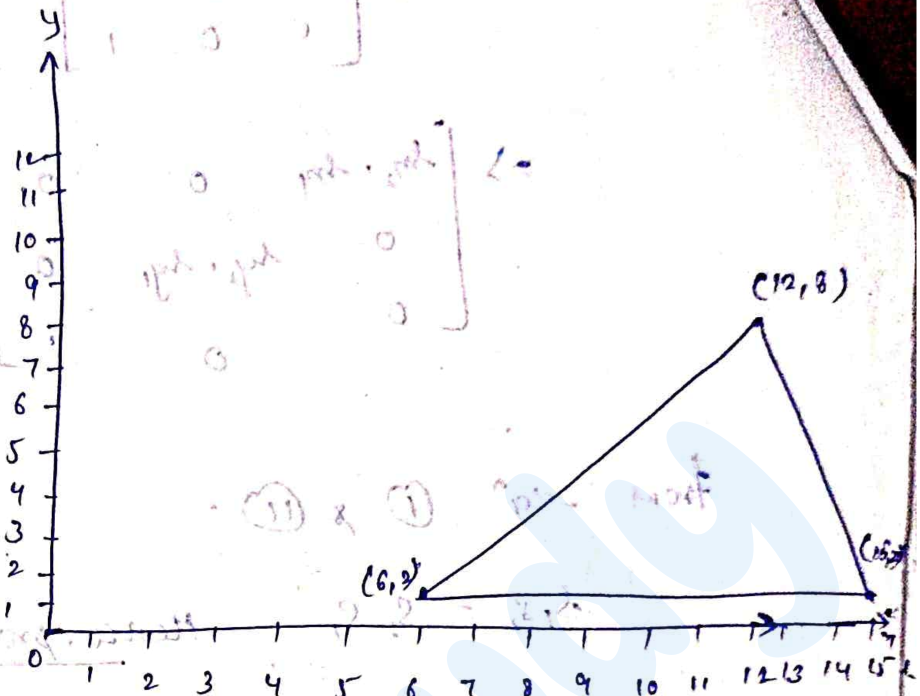


fig:- The figure after Scaling:-

So, the new co-ordinates after scaling are A (6,2), B(15,2) & C (12,8). Ans.

iii). Rotation:- It is a process of changing the angle of obj. Rotation can be clockwise or anticlockwise. for rotation we have to specify the angle of rotation and rotation point/pivot point. It is point about which the object is rotated.

Types of Rotation:-

i) clockwise :- -ve value of pivot point rotates the obj. clockwise.

ii) Anti-clockwise :- the value of pivot point rotates the obj. anti-clockwise.

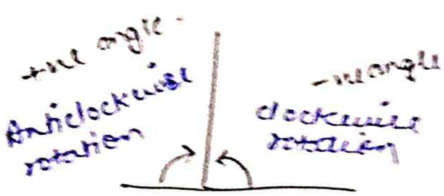
Straight line:- rotate the endpoints with same angle and redraw the line b/w new endpoints.

Polygon:- Rotate every vertex with same angle.

Curved line:- ~~draw~~ rotate all the points and draw again.

Circle:- positioned centre by specified angle.

Ellipse:- rotate major and minor axis by desired angle.



$$\left.\begin{aligned} x' &= x\cos\phi - y\sin\phi \\ y' &= y\cos\phi + x\sin\phi \end{aligned}\right\} \text{anticlockwise}$$

Matrix for Rotation:-

$$\left.\begin{aligned} x' &= x\cos\phi + y\sin\phi \\ y' &= y\cos\phi - x\sin\phi \end{aligned}\right\} \text{clockwise}$$

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \rightarrow \text{clockwise}, \quad R = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \rightarrow \text{anti-clockwise}$$

for homogenous co-ordinate rotation:-

$$R = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \rightarrow \text{clockwise}$$

$$R = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \rightarrow \text{Anti-clockwise}$$

***Rotation about an arbitrary/ pivot point:-**

i) translate the pivot point to origin

ii) Rotate the point about origin.

iii) Again translate it to original place.

Q. Rotate a line CD whose end points are (3,4) & (12,15) about origin through 45° anti-clockwise dirc^n. *

sol:- The point (3,4).

$$R = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}.$$

$\theta = 45°$.

Let, $R = \begin{bmatrix} \cos 45° & \sin 45° \\ -\sin 45° & \cos 45° \end{bmatrix}$

$$R = \begin{bmatrix} \cancel{0.707} \; 1/\sqrt{2} & \cancel{0.707} \; 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$$

The point (3,4) after rotation will be:-

$$[x, y] = [3, 4] \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} \frac{3}{\sqrt{2}} - \frac{4}{\sqrt{2}} & \frac{3}{\sqrt{2}} + \frac{4}{\sqrt{2}} \end{bmatrix} \Rightarrow \begin{bmatrix} -1/\sqrt{2} & 7/\sqrt{2} \end{bmatrix}.$$

so, $[x, y] = [-0.707 \quad 4.949]$. Ans.

The rotation of point B(12,15).

$$[x, y] = [12 \quad 15] \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} \frac{12}{\sqrt{2}} - \frac{15}{\sqrt{2}} & \frac{12}{\sqrt{2}} + \frac{15}{\sqrt{2}} \end{bmatrix} \Rightarrow \begin{bmatrix} \frac{-3}{\sqrt{2}} & \frac{27}{\sqrt{2}} \end{bmatrix}$$

so, $[x, y] = [-2.121 \quad 19.094]$.

so, the new line after rotation at 45° become.

~~A(...)~~ $A(-0.707, 4.949)$ and $B(-2.121, 19.094)$ **Ans.**

Q. Rotate a line CD whose endpoints are (2,5) & (6,12) about origin through a 30° clockwise dirc^n.

Sol:- we know, to rotate a point clockwise we can use.

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}.$$

here, $\theta = 30°$

$$R = \begin{bmatrix} \cos 30° & -\sin 30° \\ \sin 30 & \cos 30 \end{bmatrix} \Rightarrow \begin{bmatrix} \sqrt{3}/2 & -1/2 \\ 1/2 & \sqrt{3}/2 \end{bmatrix}.$$

After rotating the point (2,5) :-

$$[x, y] \Rightarrow [2, 5] \begin{bmatrix} \sqrt{3}/2 & -1/2 \\ 1/2 & \sqrt{3}/2 \end{bmatrix}.$$

$$\Rightarrow \left[\sqrt{3} + \frac{5}{2} \quad \frac{5\sqrt{3}}{2} - 1\right] \Rightarrow [4.232 \quad 3.330]$$

so, $[x, y] = [4.232 \quad 3.330]$.

After rotation of point (6, 12).

$$[x, y] \Rightarrow [6. 12] \begin{bmatrix} \sqrt{3}/2 & -1/2 \\ 1/2 & \sqrt{3}/2 \end{bmatrix}.$$

$\Rightarrow \begin{bmatrix} 3\sqrt{3}+6 & 6\sqrt{3}-3 \end{bmatrix} \Rightarrow \begin{bmatrix} 11.196 & 7.392 \end{bmatrix}$.

so, $[x, y] = [11.196 \quad 7.392]$

so, the new line after rotating 30° clockwise will be

$A(4.232, 3.33)$ & $B(11.196, 7.392)$ <u>Ans</u>.

<u>iv) Reflection</u> :- It is a transformation which produces a mirror image of an obj. The mirror img. can be either about x-axis or y-axis. The obj. is rotated by 180°.

<u>Types</u>:-

i) About <u>x-axis</u> :- In this transformation value of x will remain same & value of y will become -ve.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \rightarrow \underline{\text{matrix}} \quad \text{or} \quad \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix}.$$

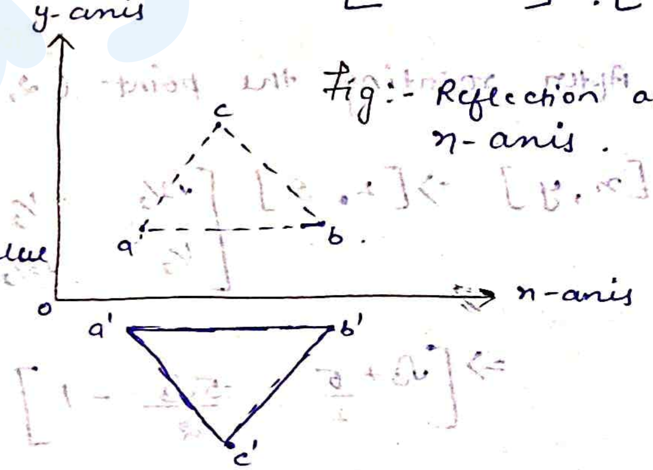


Fig:- Reflection about x-axis.

ii) About <u>y-axis</u> :- Here, the value of x will removed & value of y will remain same

$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \underline{\text{matrix}}.$$

or

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix}.$$

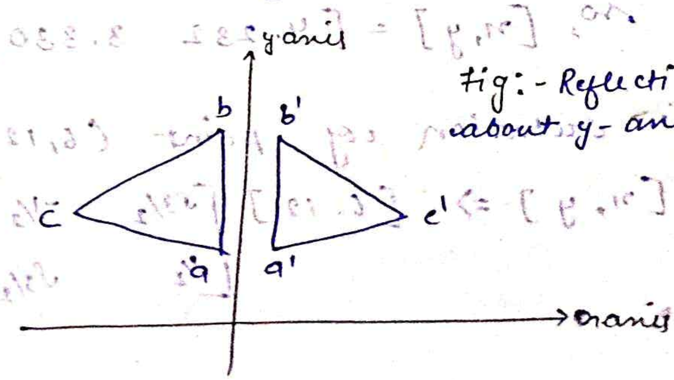
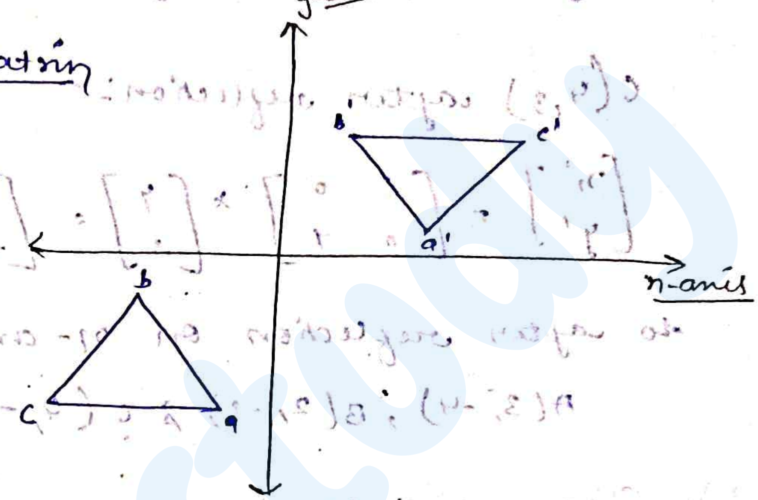


Fig:- Reflection about y-axis.

iii). Reflection about an axis passing ~~throug~~ ⊥ to xy plane passing through origin :- both the value of x & y will reversed (-ve).

$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ — matrix}$$

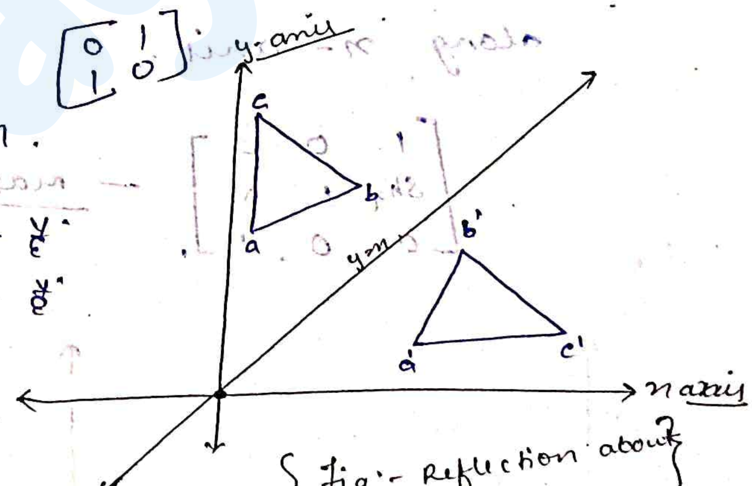


iv). Reflection about line $y=x$:-

First of all obj. is rotated at 45°. The dirn of rotation is clockwise. After its reflection is done concerning x-axis. The last step is the rotation of $y=x$ back to its origional pos., that is a clockwise at 45°.

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \rightarrow \text{matrix.} \qquad \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

— x के co-ordinates y बन जाते हैं
— y के co-ordinates x बन जाते हैं



{fig:- Reflection about $y=x$.}

Q: A(3,4), B(2,3) & C(4,3).
Reflection of x and y axis :-

Sol:- for x-axis :-
A(3,4) after reflection.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \times \begin{bmatrix} 3 \\ 4 \end{bmatrix} \Rightarrow \begin{bmatrix} 3 \\ -4 \end{bmatrix}.$$

for B(2,3) after reflection:-

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} * \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 2 \\ -3 \end{bmatrix}.$$

C(4,3) after reflection:-

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} * \begin{bmatrix} 4 \\ 3 \end{bmatrix} = \begin{bmatrix} 4 \\ -3 \end{bmatrix}.$$

so after reflection on x-axis new co-ordinates are -

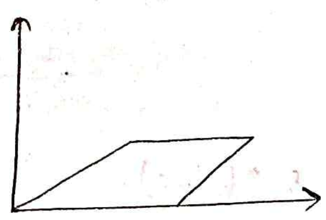
A(3, -4) ; B(2, -3) & C(4, -3) Ans.

vi) Shearing:- It is a type of transformation which changes the shape of obj. The sliding of layers of obj. occurs. The shear can be in one or two dirc^n.

i) In x dirc^n:- Here shearing is done by sliding along x-axis.

$$\begin{bmatrix} 1 & 0 & 0 \\ Sh_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ — matrix} \quad \left\{ \begin{aligned} X' &= X + Sh_x \cdot Y \\ Y' &= Y \end{aligned} \right\}$$

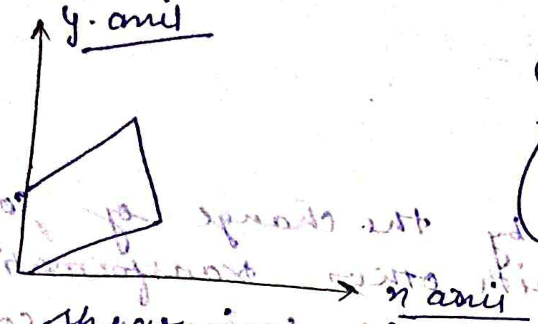
Origional.



shear in x-dirc^n.

ii) In y dirc^n:- Here shearing is done by sliding y-axis.

$$\begin{bmatrix} 1 & Sh_y & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ — matrix.}$$



shear in y-dirn

$$\left\{ \begin{aligned} x' &= x. \\ y' &= y + sh_y \cdot x \end{aligned} \right\}$$

(ii) Shear in X-Y dirn :- More layers will be slided in both x- and y dirn. The shape of obj will be distorted.

$$\begin{bmatrix} 1 & sh_y & 0 \\ sh_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \left\{ \begin{aligned} x' &= x + sh_x \cdot y \\ y' &= y + sh_y \cdot x . \end{aligned} \right\}$$

Q. $A(2,2)$, $B(1,1)$, $C(3,1)$

$sh_x = 1$

$sh_y = 1.$

Sol:- In x-axis :-

$A(2,2).$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ sh_x & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \times \begin{bmatrix} 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \end{bmatrix}.$$

$B(1,1)$

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

$C(3,1)$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 3 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 \\ 1 & 1. \end{bmatrix} = \begin{bmatrix} 4 \\ 1 \end{bmatrix} \text{Ans}$$

$\Longrightarrow A(4,2), B(2,1)$ & $C(4,1)$ Ans

* ## Homogenous Co-ordinates :-

~~Moving~~

Translation of point by the change of co-ordinate can't be combined with other transformation by using simple matrix app. so, we use homogenous co-ordinates to combine these transformation (if we have to change the pivot other than origin than we have to use these transfomation).

Eg :- i) $\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$ — translation.

{Perform only matrix multiplication in whole homogenous co-ordinates}

ii) $\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$ — Rotation (clockwise).

* ## Composition transformation :-

A number of transfomation or seq. of tranfomation can be combined into single one called as composition. The resulting matrix is known as composite matrix. The process of combining is called as Concatination.

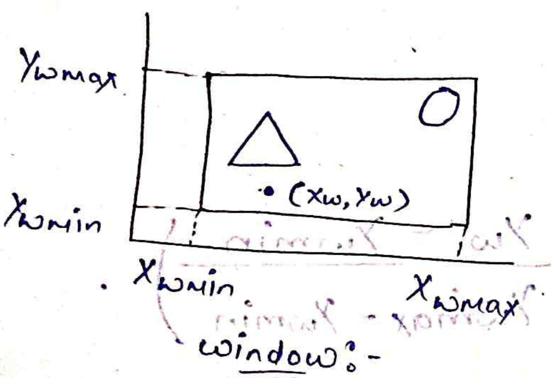
NOTE :- The ordering seq. of these no. of transf. must not be changed. go from left to right first multiply matrix than the resultant of that must multiply with new transfomation.

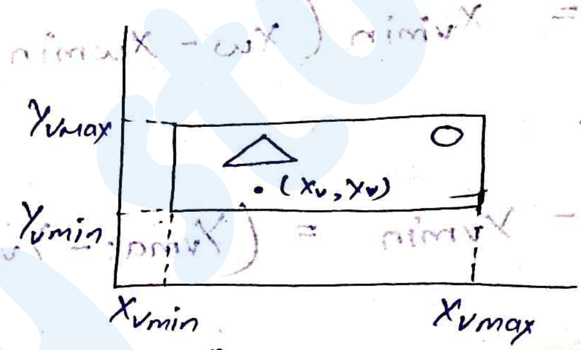
ADV :-
- It makes transformation compact.
- The no. of operations will be reduced.
- Easy to Understand.

* Window to viewport tranformation:-

Window:- The method of selecting a portion of drawing is known as window. The area chosen for this display is called window.

Viewport:- An area on display device to which a window is displayed.





Note:- Relative position will not change only size will change acc. to window & viewport.

Now, let us have (X_w, Y_w) given, we have to find (X_v, Y_v).

$$\frac{X_w - X_{wmin}}{X_{wmax} - X_{wmin}} = \frac{X_v - X_{vmin}}{X_{vmax} - X_{vmin}}$$ [∵ relative pos. same.]

$$\frac{Y_w - Y_{wmin}}{Y_{wmax} - Y_{wmin}} = \frac{Y_v - Y_{vmin}}{Y_{vmax} - Y_{vmin}}$$ [∵ relative pos. same.]

So,

$$X_v - X_{vmin} = (X_{vmax} - X_{vmin}) \times \left(\frac{X_w - X_{wmin}}{X_{wmax} - X_{wmin}}\right)$$

$$X_v - X_{vmin} = (X_w - X_{wmin}) \times \underbrace{\left(\frac{X_{vmax} - X_{vmin}}{X_{wmax} - X_{wmin}}\right)}_{\text{Scaling factor } (S_x)}$$

$$\left[X_v = X_{vmin} (X_w - X_{wmin}) \times S_x\right] \longrightarrow (1)$$

Again.

$$Y_v - Y_{vmin} = (Y_{vmax} - Y_{vmin}) \times \left(\frac{Y_w - Y_{wmin}}{Y_{wmax} - Y_{wmin}}\right)$$

$$Y_v - Y_{vmin} = (Y_w - Y_{wmin}) \times \underbrace{\left(\frac{Y_{vmax} - Y_{vmin}}{Y_{wmax} - Y_{wmin}}\right)}_{\text{Scaling factor } (S_y)}$$

$$\left[Y_v = Y_{vmin} (Y_w - Y_{wmin}) \times S_y\right] \longrightarrow (11)$$

So,
$$\left.\begin{bmatrix} X_v = X_{vmin} (X_w - X_{wmin}) S_x \\ Y_v = Y_{vmin} (Y_w - Y_{wmin}) S_y \end{bmatrix}\right\}$$

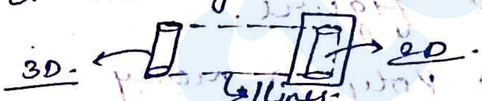
* Three - D Graphics :- In 3-D transformations are extension of 2-D transformation. In 2-D, two co-ordinates are used x, y and In 3-D three co-ordinates are used x, y & z. We used 3-D to produce realistic 3D scenes -- it requires the third dimension. i.e, depth.

* Producing Realism in 3-D :- The 3-D obj are made using comp. graphics. The tech. used for two-D displays of 3-D objects is called projections.
↳ It is also defined as mapping or transf. of obj in proj. plane.

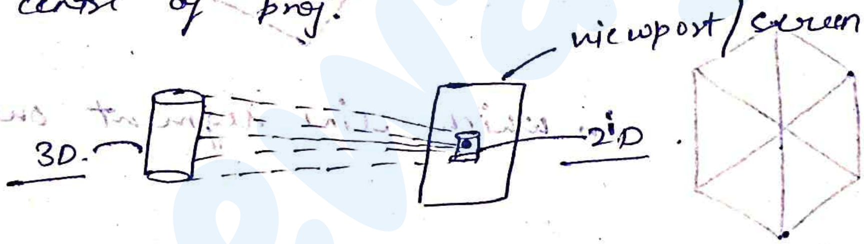
Types of projections :-

i) Parallel Projection :- In this projection point on the screen is identified within a point in 3-D object by a line ⊥ to the object screen. The architect drawing, i.e, plane, front view, side view are nothing but lines of // projection.

Note :- Parallel lines from 3D to viewport to make a 2D obj (the length or size of obj. will not change).



ii) Perspective projection :- This proj. has property that it provides idea about depth. Further the subject from viewer, smaller it will appear. All the lines in perspective projection will converge to a centre point called as the centre of proj.



* Transformation in 3D :-

i) Translation :-

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

ii) Scaling :-

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \cdot s_x \\ y \cdot s_y \\ z \cdot s_z \\ 1 \end{bmatrix}$$

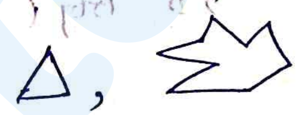
Unit - 4.

* Representing curves and surfaces :-

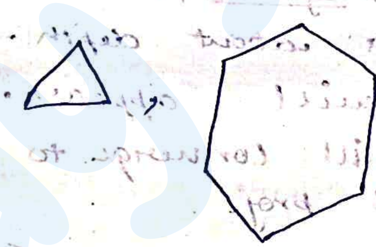
In comp. graphics, curves and surfaces are typically represented using mathematical models such as, NURB (Non-uniform rational B-splines), Bezier curve and Bezier surfaces. These models ~~can be manipulated~~ allow for precise control over the shape of curve or surface & make it easy to manipulate & animate them.

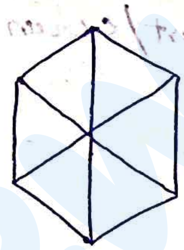
* Polygon :- Parametric :-

A polygon is any 2-D shaped object formed with straight line segments and it must be a closed shaped figure.

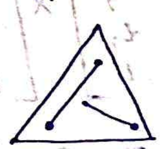
Eg :- 

greek work ← { Poly → many, gon → angle }

Types :- i> simple :-  : doesn't overlap.

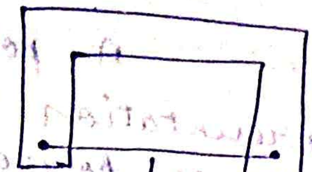
ii>. Complex :-  in which line segment overlaps.

iii>. Convex :- If line connecting two interior points of polygon lies completely inside the polygon than it is said to be complex polygons.

Eg :- 

iv) Concave :- If line connecting two interior points of polygon doesn't lie completely inside of polygon than it is said to be concave polygon.

Eg:-



* **Polygon Table:-** The polygon table contains all surfaces (vertex, edges, surfaces) info. about the polygon with set of co-ordinates & associated parameter.

Eg:-

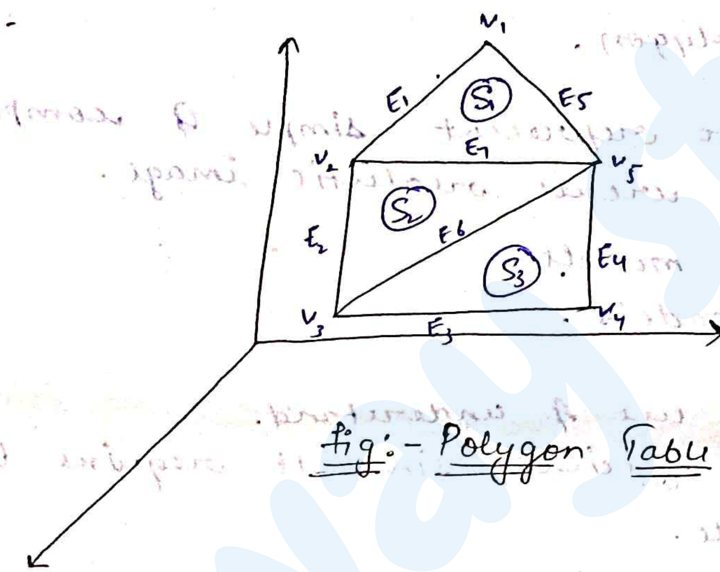


fig:- Polygon Table:-

Now, polygon table of this fig:-

Vertex table

$V_1 = (x_1, y_1, z_1)$

$V_2 = (x_2, y_2, z_2)$

$V_3 = (x_3, y_3, z_3)$

$V_4 = (x_4, y_4, z_4)$

$V_5 = (x_5, y_5, z_5)$.

Edge Table.

$E_1 = V_1, V_2$

$E_2 = V_2, V_3$

$E_3 = V_3, V_4$

$E_4 = V_4, V_5$

$E_5 = V_5, V_1$

$E_6 = V_3, V_5$

$E_7 = V_2, V_5$.

Surface table.

$S_1 = E_1, E_5, E_7$

$S_2 = E_7, E_2, E_6$

$S_3 = E_3, E_4, E_6$.

* Polygon meshes Parametric :- A polygon mesh is a type of surface representation in comp. graphics which is made up of set of polygons.

These polygons can be either triangular or quadrilateral & defined by set of vertices.

Parametric representation of mesh polygon refers to the use of mathematical equation to define the pos. of each vertex of polygon.

Use :- - used to represent simple & complex shapes.
- Manipulate to create realistic image.
- to create 3D models.
- 3D printing models.

Adv :- - Easy to use & understand.
- It is highly efficient since, It requires less polygons.
- It is versatile.

* Curves :- A curve is an infinitely large set of points. Each points has two neighbours except end points.

Types :-

i> Implicit curve :- Implicit curve representation define the set of points on a curve by employing a procedure that can test if a point is on the curve.

$[f(x,y) = 0]$ It can represent multiple value of x here.

Eg :- $[x^2 + y^2 - r^2 = 0]$ — eqⁿ of circle.

ii). **Implicit curve:-** A mathematical function $y = f(x)$ can be plotted as a curve. Such a funcⁿ is the implicit representation of the curve. It is not generally used since, it can't represent vertical lines & is also single-valued. For each value of x only a single value of y can be computed.

iii). **Parametric curve:-** Curves figuring parametric form are called parametric curve. Implicit & explicit representation can be used only when the funcⁿ is known. But in General, parametric curves is are used.

↳ Linear (1)
↳ Polynomial (2) (2,3,4...)
↳ cubic curve (3). Piecewise.

Imp.

Represent:-

$x = f(y)$
$y = f(x)$] — They aren't dependent instead they depend on

$f(x, y) = f(v)$ ⟶ Parameter.

Now, the points (x, y) are obtained when the value of (v) varied over a certain Interval.

* **Bezier Curve:-** Bezier curve is discovered by french engineer Pierre bezier. These curves can be generated under the control of other points. Approximate tangents by using control points are used to generate curve.

$$\left[\text{Representation:-} \quad \sum_{K=0}^{n} P_i B_i^n(t)\right]$$

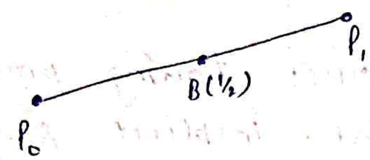
where, P_i → set of points.
B_i^n → Bernstein polynomial:

$$\left[B_i^n(t) = \binom{n}{i} (1-t)^{n-i} t^i\right]$$

where, n → Polynomial degree.
t → variable.
i → Index.

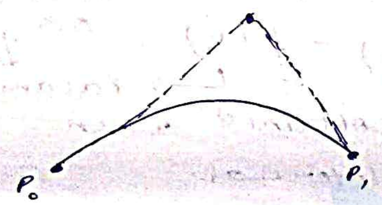
Types :-

i) Simplest Bezier curve :- It is the simplest curve that formed from two points P_0 & P_1 and formed a straight line.



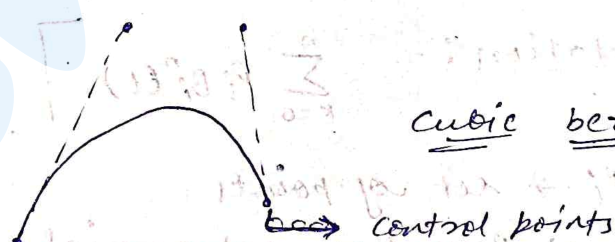
simple bezier curve.

ii) Quadratic Bezier curve :- This curve can be formed or determine by three control points P_0, P_1 & P_2.



Quadratic bezier curve.

iii) Cubic Bezier curve :- This curve can be formed or determine by four control points. P_0, P_1, P_2 and P_3.



Cubic bezier curve.

* Properties :- i) They generally follow the shape of control polygon, which consist of segments joining the control points

ii) They always pass through 1st & last control points.

iii) The degree of poly. defining the curve is 1 less than the no. of defining the polygon points.

iv) No straight line intersect a Bezier curve more times than it intersect its control polygon.

v) moving a control point alter the shape of whole bezier curve.

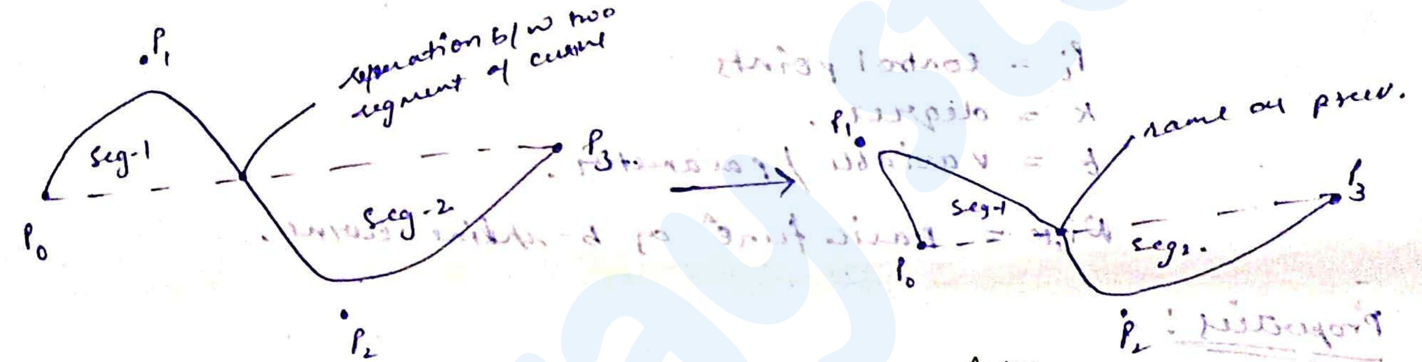
vi) A given bezier curve can be subdivided into two bezier segments which can join together at a point.

* **Dis-advantages in B-Curve:-** when we change any of the control point respective location the whole curve shape gets changed. But

↳ control points import global control over curve.

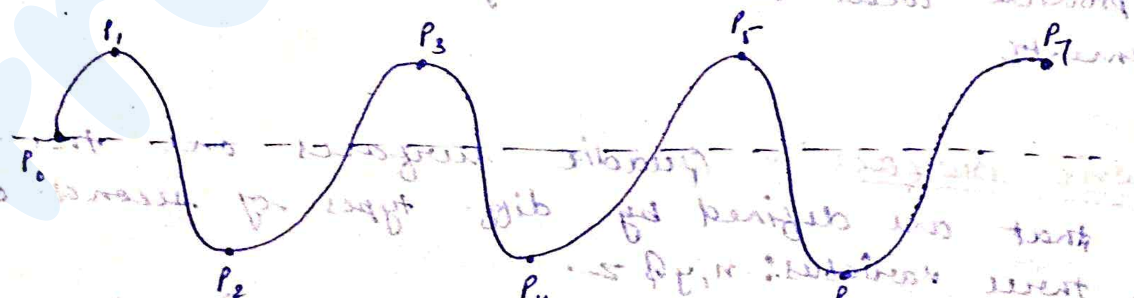
In B-spline curve, only a specific segment of the curve shape gets changes or affected by the changing of the corresponding loc of the control points.

↳ Import local control over curve



Before. After.

* **B-spline Curve:-** B-spline curve are independent of the no. of control points are made up of joining the several segments smoothly, where each segment shape is decided by some specific control points that come in that region of segment.



degree = ~~control point~~ -1 $\Rightarrow$ 7. [no of control points = n+1 = 7+1 = 8.]

Total no. of segment = $n-k+2$.

Note:- The point b/w two segments of a curve that join each other such points are known as knots in b-spline curve.

↳ uniform (periodic)
↳ open-uniform
↳ non-uniform.

Eqn:- $$Q(t) = \sum_{i=0}^{n} P_i \cdot N_{i,k}(t).$$

P_i = control points
k = degree.
t = variable / parameter.
$N_{i,k}$ = Basic funcⁿ of b-spline curve.

Properties:-

i). degree of b-spline curve doesn't depend on the no of control points which make it more reliable than using Bezier curve.

ii). The sum of basic functions for a given parameter is one.

iii) It provide local control through control points over control segments.

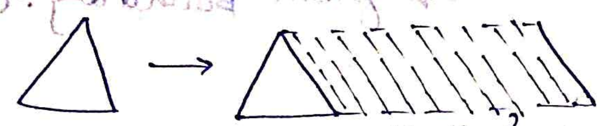
* **Quadric Surface:-** Quadric surfaces are the surfaces that are defined by diff. types of second-order eqⁿ with three variables: x, y & z.

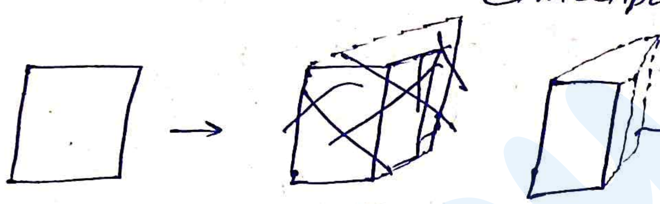
General form:- $Ax^2 + By^2 + Cz^2 + Dxy + Exz + Fyz + Gx + Hy + Jz + K = 0$

* Solid modeling:- It is a method to create and display the 3-D objects on computer screen by comp. grap. is known as solid modeling.

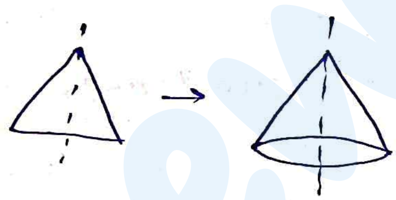
* Sweep representation:-

a) Translation:- obj. को रखकर एक dir^n में खिंचना

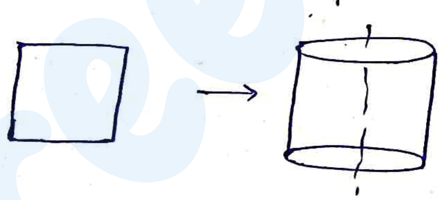




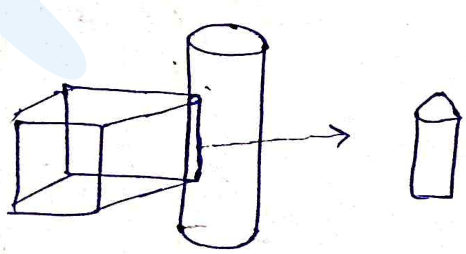
b) Rotational:-



axis से obj. को घुमा देना।



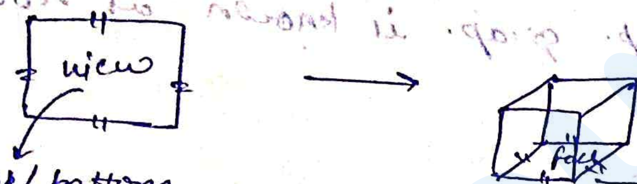
* Constructive solid Geometry:-



→ to use a part of mixed objects is undergoes as constructive solid Geometry.

* Boundary Representation :-

CAD
↳ comp. Aided design.

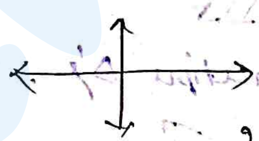


TOP/bottom

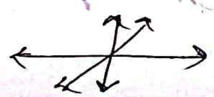
→ That view is a face in 3D obj. This is known as boundary representation since it is made up from boundary. (B-Reps).

used mostly in CAD modeling.

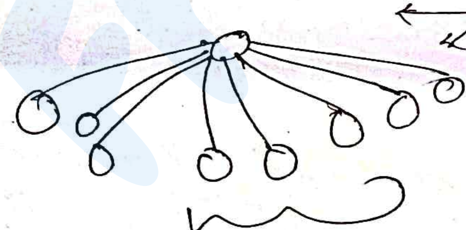
* Octree :-



$2^2 = 4$ quadrant.



$2^3 = 8$ quad.



↳ tells all about all quadrants.

* Multimedia:-

many ↙ ↘ material through which something can be transmitted or send.

It combined all the media elements like. text & graphics to make it more effective & attractive.

* Components:-

1) Text:- soft. MSword, notepad, wordpad, etc. text file format DOC, Txt, etc.

2) Audio:- This component ↑ understandibility & improve clarity of concept.

soft. → window media player, Real player.

3) Graphics:- Every multimedia presentation is based on graphics which makes it more effective & presentable

Soft → window pict, Internet explorer, Adobe photoshop

4) Video:- moving pict with sound it is the best way to communicate with each other.

5) Animation:- Animation is used to make changes to the images so, the seq. of img. appear to be moving pict.

* Applications:-

1) Education:-

2) Entertainment:- movies, Games

3) Business:- video conferencing. marketing & advertisement.

Edutainment
Mix of edu. entertain.

* Animation:- It refers to a simulated motion picture depicting movement of obj.

features:- i) picture, ii) Motions iii) Simulated

* Use of multimedia animation :-

1) To attract attention.
2) Demonstration
3) Simulation
4) To inform about state of process. (% of process completion)
5) Motivation
6) Org.
7) Representation.

* Types of Animation tech :-

1) Traditional / classical or 2D animation. ← by hand pen/pencil
2) 2D animation — 2D space में 100 frame bnake movement dikha do
3) 3D animation — unreal char. into real one.
4) Topography animation
5) Clay "
6) Sand "
7) flip book "
8) Stop-motion " — physical static obj. make them move or talk.
9) Mechanical Animation — Mach. को बनाकर उसका animation bnao which shows its functionality.
10) Audio-animatronics - repeats a pre-prog. over & over.
11) Autonomatronics — camera, sensor and, & ability to make their own choices about what you say & do
12) ~~[illegible]~~.
12) Puppetry animation — create animation in life of puppets
Ex - The Humpty Dumpty circus (1908)

4 Stages in multimedia project.

1) Planning & costing. —

2) designing & producing. —

3) Testing & debug. —

4) Delivering.

* Hardware req. in multimedia.

1) Sys device.
   - motherboard — backbone
   - microprocessor. — engine
   - GPU (— boost the performance .

2) Memory & storage dev.
   - RAM
   - ROM
   - Hard disk
   - DVD
   - Compact disk (CD)

3) Input devices
   - keyboard.
   - pointing device.
   - composite " (combine two devices)
   - Scanner
   - Audio Input dev.

4) output devices.
   - Printer
   - monitor
   - Amplifier

* Soft. req. for multimedia.
   - Painting & drawing tools
   - 3-D modeling
   - Image editing tool
   - Sound editing tool
   - Animation video

2016.

Refresh-Rate :- It refers the no. of times / sec the img is redrawn.

the content of frame buffer is displayed on CRT at high enough rate to avoid flicker called refresh rate.

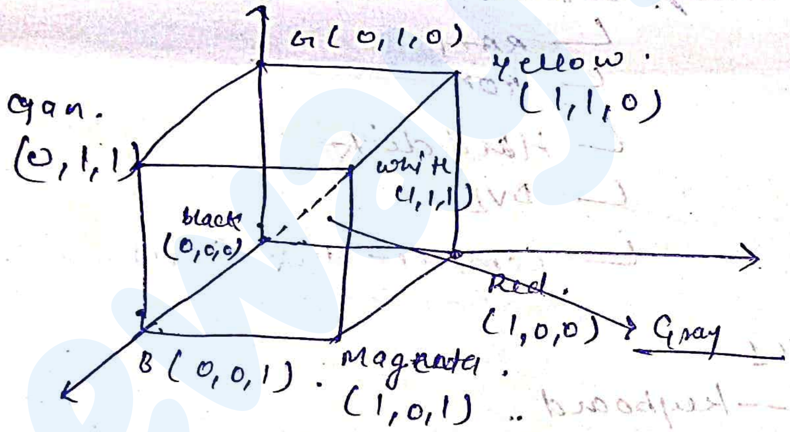
60 frames/sec = 60 Hz.

normally 60-80 f/s.

120 f/s तक आता है।

Q. Most commonly used color models in CG.

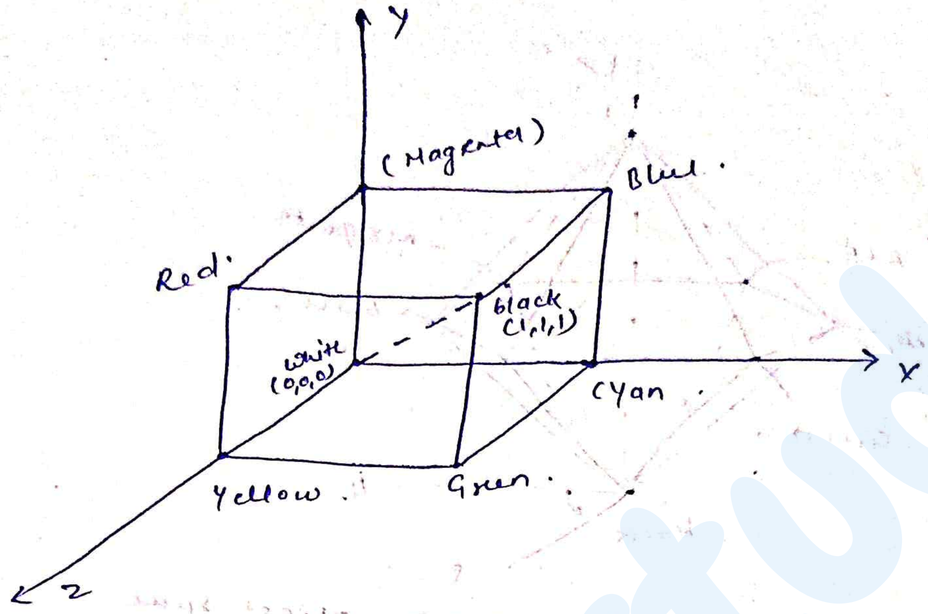
1) RBG ($\underbrace{\text{Red, blue, green}}_{\text{resultant color}}$)



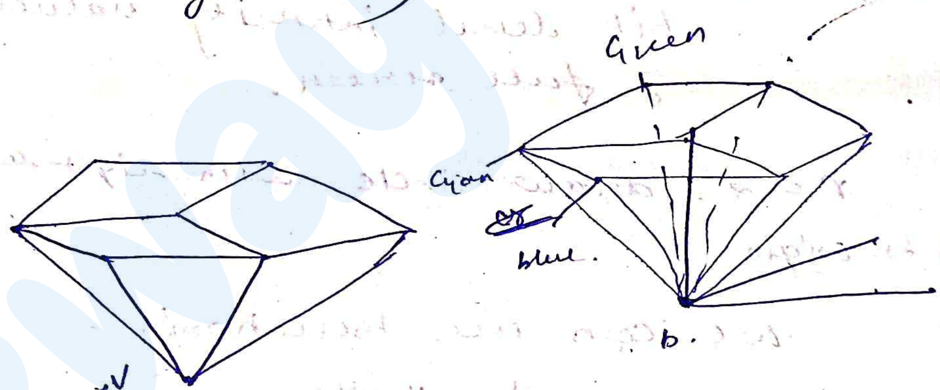
$$\boxed{C_\lambda = RR + GG + BB}$$

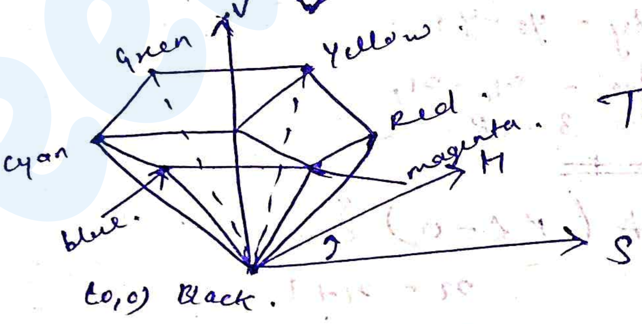
2) CMY. (cyan Magenta Yellow)

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$



3). HSV (Hue — distinguish b/w colors red, green yellow etc)
(Saturation - How far color is from Gray)
(Value - level of brightness)

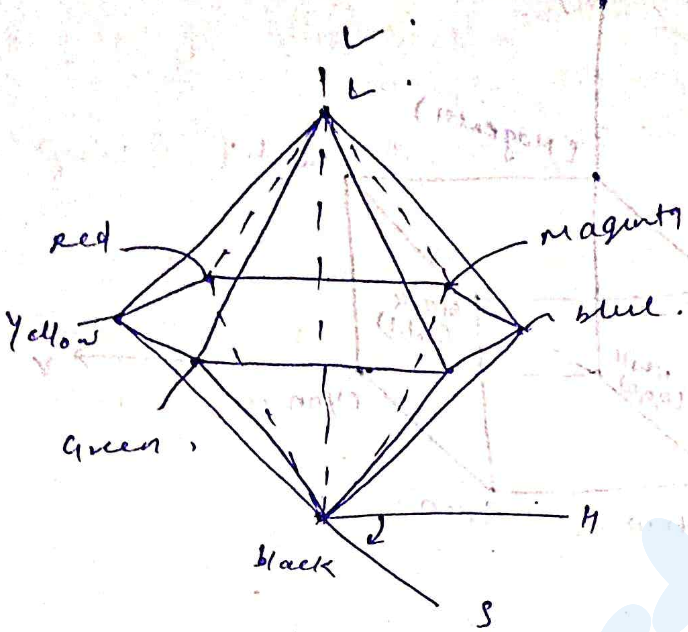




The HSV hexcone.

4) HLS :- hue -
lightness - ~~At~~ At $L = 0$ (black
$L = 1$ white.
In b/w gray.

Saturation -



3D visible surface detection → object space.
Img. space meth. ⟶ Alg (Z-buffer),

bit map/plane:- It is a block of memory which stores bit level intensity values for each pixel of full screen.

Q. $r = 2$. draw circle with eight unique points.
center → origin

Sol:- we can use bresenham's.

$dy = y_2 - y_1$

$dx = x_2 - x_1$

$P = 3 - 2R$

~~P = P +~~

if (P <= 0) {

$x = x + 1$

$P = P + 4x + 6$

}

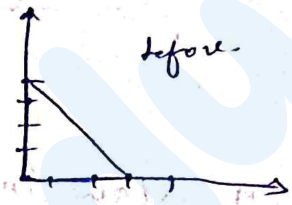
$x = x + 1$

$y = y - 1$

$P = P + 4x - 4y + 10$

Q. $A(0,0)$ $B(4,0)$, $C(0,4)$
$S_x = 2$, $S_y = 3$. (shearing)

In X- dirⁿ.

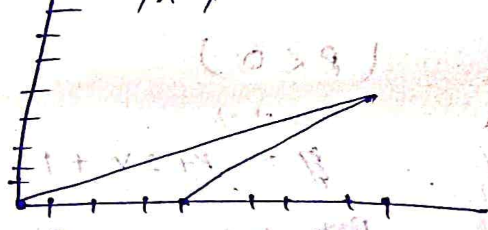


$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 4 & 0 & 1 \\ 0 & 4 & 1 \end{bmatrix}.$$

$$\Rightarrow \begin{bmatrix} 0 & 0 & 1 \\ 4 & 0 & 1 \\ 0 & 4 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 1 \\ 4 & 0 & 1 \\ 8 & 4 & 1 \end{bmatrix}$$

New.
$A(0,0)$ $B(4,0)$ $C(8,4)$

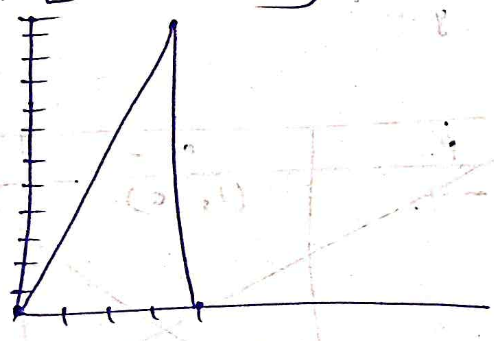
In x-dirⁿ



In Y- dirⁿ.

$$\begin{bmatrix} 0 & 0 & 1 \\ 4 & 0 & 1 \\ 0 & 4 & 1 \end{bmatrix} \begin{bmatrix} 1 & 3 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 4 & 12 & 1 \\ 0 & 4 & 1 \end{bmatrix}$$

$A(0,0)$ $B(4,12)$ $C(0,4)$



Video controller - Interactive raster graphics sys typically used several processing unit in add. to CPU called video controller. It is used to control the op. of display devices.

* Basic rule of animation:-

1) Squash & stretch:-

Eg- rubber ball on floor

2) Slow in & slow out:- camera view

3) Maintaining 3D effect.

Q. $r = 10$. origin . (mid point circle).

$P = 1 - r = 1 - 10 = -9$.

$(P < 0)$

$x++$

$P = P + 2x + 1$

else

$P = P + 2x - 2y + 1$

$x++$

$y--$

$x = 0$

$y = 10$

X	Y	P
0	10	-9
1	10.	-6.
2	10	-1
3	10	6.
4	9.	-3 .
5		
⋮	⋮	⋮

$P = P + 2X + 1 = -9 + 3 = -6$.

$P = -6 + 5 = -1$

$P = -1 + 7 = 6$.

$P = P + 2X - 2Y + 1 = 6 + 8 - 18 + 1 = -3$.

- - -