Computer Graphics

Page: Subhash
Date: 01 Aug 17

```
void main()
{
int gdriver = DETECT, gmode;
int x1 = 200, y1 = 200;
int x2 = 400, y2 = 400;
# initgraph (&gdrive, &gmode, "C:\\Turbo C3\\BGI") #
  line (x1, x2, y1, y2);   \\line is an in-built function
  getch();
  closegraph();
}
```

→ **Types of Computer Graphics-**

1) Interactive (Active)
2) Non-interactive (Passive)

→ **Basic Terminology-**

1) **Pixel-** Defined as the smallest addressable screen element.
   - Each pixel has its own intensity named address by which we can calculate / control it's measure of screen resolution

2) **Resolution-** Defined as the ratio of width and height of output of the display device.

Image- Defined as the distance of one pixel from another pixel.
   - Also known as Pixel Spacing

Screen- Defined as the number of pixels in the horizontal and vertical direction of the screen.

1024x640 → means 1024 in horizontal direction
640 in vertical direction

3) Frame Buffer – It is a large continuous piece of memory into which the intensity value of all pixels are saved.
- It is like an array and graphical display devices can control and access this array.
- Thus, a frame buffer contains internal representation of an ~~array~~ image.

{Size of frame buffer = Resolution x (bits/pixels)}

4) Bit Map, Pixel Map and Opixmap –
There is one memory bit for each pixel in the Raster System.
If a pixel has only two-colour values (typically black & white), it can be encoded by 1 bit of info.
Thus an image of 2 colours (B & W) is called a Bit Map.

If more than 2 bits are used to represent the pixel, a large range of colours can be represented.

An image of more than 2 colours is called a Pixel Map.

With multiple bits for pixels, the frame buffer is called an Opixmap.

5) Aspect Ratio: Defined as the ratio of horizontal points to the vertical points required to produce equal length lines in both directions of the screen on different devices.

$$AR = \frac{\text{No. of horizontal lines}}{\text{No. of vertical lines}}$$

eg. (800 x 600) pixels in display

$$\frac{800}{600} = \frac{4}{3}$$

Ques) Suppose a screen of 8x10 inches has a resolution of 100 pixels/inch in each direction. If we want to store 6 bits of information per pixel in the FB, how much storage (in bytes) do we need for the FB? Also find the aspect ratio of the Raster System.

Resolution ~~Pixels~~ = (800 x 1000) pixels

Given that, 1 pixel stores 6 bits

$$\therefore FB = \frac{800 \times 1000 \times 6}{8} = 6 \times 10^5 \text{ bytes}$$

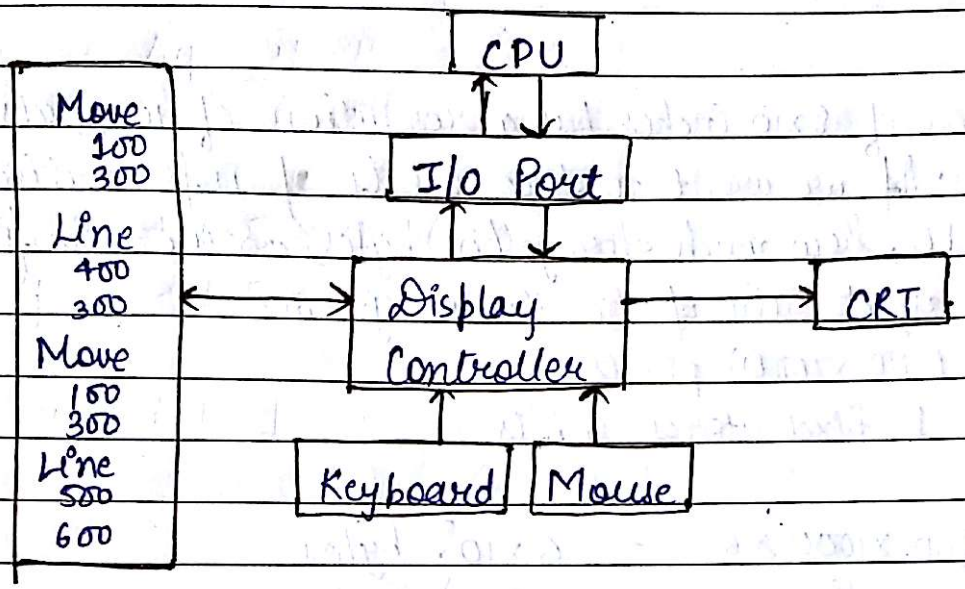
Aspect Ratio : $\frac{800}{1000} = 4:5$

Ques) Consider three different Raster Systems with resolution of -

a) 640 x 480 (b) 1280 x 1024 (c) 2560 x 2048

What size of FB (bytes) is needed for each of these systems to store 12 bits/pixel? How much storage is required for each of these systems if 24 bits/pixels are to be stored?

→ Random Scan Display-



→ DDA - Digital Differential Analyzer / Algorithm

1) The slope of a straight line, $m = \frac{\Delta y}{\Delta x}$ $\left\{ m = \frac{y_2 - y_1}{x_2 - x_1} \right.$

2) For given Δx along a line Δy will be

$$\Delta y = m \cdot \Delta x \Rightarrow \cancel{\Delta x} \; \Delta y = \Delta x \cdot \frac{y_2 - y_1}{x_2 - x_1}$$

3) If Δy is given then $\Delta x = \frac{\Delta y}{m} = \Delta y \cdot \left(\frac{x_2 - x_1}{y_2 - y_1} \right)$

4) Once the intervals are known,
the next value(s) of x & y are obtained as

$$x_{i+1} = x + \Delta x$$

i.e.

$$\boxed{x_{i+1} = x + \frac{x_2 - x_1}{y_2 - y_1} \cdot \Delta y} \quad \text{—— (A)}$$

and $y_{i+1} = y_i + \Delta y$

$$\Rightarrow \boxed{y_{i+1} = y_i + \frac{y_2 - y_1}{x_2 - x_1} \cdot \Delta x} \quad \text{—— (B)}$$

5) For DDA either Δx or Δy whichever is larger is taken as Raster Unit.

ie. if $|\Delta x| > |\Delta y|$ then $\Delta x = 1$
else $\Delta y = 1$

6) Now if $\Delta x = 1$, then $y_{i+1} = y_i + \frac{y_2 - y_1}{x_2 - x_1}$

and $x_{i+1} = x_i + 1$

7) Similarly, if $\Delta y = 1$ then $x_{i+1} = x + \frac{x_2 - x_1}{y_2 - y_1}$

$y_{i+1} = y_i + 1$

1) Read line and points (x_1, y_1) & (x_2, y_2)

2) $\Delta x = |x_2 - x_1|$
$\Delta y = |y_2 - y_1|$

3) If $(\Delta x > \Delta y)$, then length $= \Delta x$
else length $= \Delta y$

4) Select Raster Unit

ie. $\Delta x = \frac{x_2 - x_1}{\cancel{y_2 - y_1}\ length}$

$\Delta y = \frac{y_2 - y_1}{length}$

5) $x = x_1 + 0.5 * sign(\Delta x)$

$y = y_1 + 0.5 * sign(\Delta y)$

6) Now plot the points $i = 1$

```
while (i ≤ length)
{
    plot ( integer(x) , integer(y))
        x = x + Δx
        y = y + Δy
        i = i + 1
}
```

7) STOP

Ques) Given $(0,0)$ and $(4,6)$
(x_1, y_1) (x_2, y_2)

$\Delta x = |x_2 - x_1| = 4$

$\Delta y = |y_2 - y_1| = 6$

Now, $\Delta x < \Delta y$

$\therefore length = \Delta y = 6$

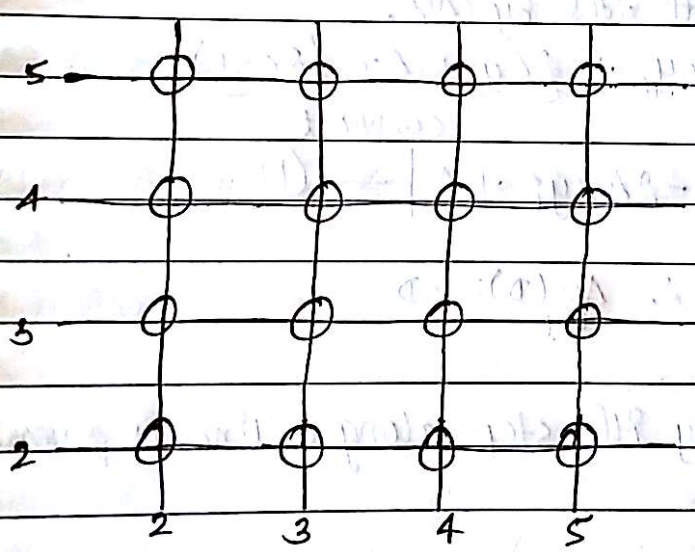
$\therefore R.U. = \Delta x = \frac{4}{6} = 0.66666 \simeq 0.667$

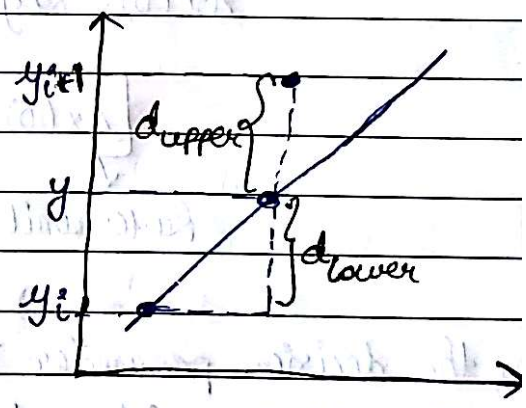
$\Delta y = \frac{6}{6} = 1$

Now, $x = 0 + 0.5 * \text{sign}(0.667)$
$= 0 + (0.5 * 1) = 0.5$
$y = 0 + 0.5 * \text{sign}(1)$
$= 0 + (0.5 * 1) = 0.5$

Plot $(\text{int}(x), \text{int}(y))$ i.e. plot $(0,0)$
$x = x + \Delta x$
$= 0.5 + 0.667 = 1.167$
$y = 0.5 + 1 = 1.5$

→ **Bresenham Line Algo**





d_{upper} = dist. of upper pixel from imaginary line
d_{lower} = " lower "

The y coordinate on the imaginary line at x_{i+1} is $y = m(x_i+1) + c$
∴

$d_{upper} = (y_i + 1) - y$
$= y_i + 1 - m(x_i + 1) - c$

$$d_{lower} = y - y_i$$
$$= m(x_i+1) + c - y_i$$

We can use these decision variables (d_{upper} & d_{lower}) to make a simple decision about a line i.e which pixel is closer to the imag. line.

We'll take the decision variable as D, $\boxed{D = d_{lower} - d_{upper}}$

$$\text{or } D = m(x_i+1) + c - y_i - \{y_i + 1 - m(x_i+1) - \}$$
$$D = 2m(x_i+1) - 2y_i + 2c - 1$$

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

$$D = 2\frac{\Delta y}{\Delta x}(x_i+1) - 2y_i + 2c - 1$$

Multiplying both sides by Δx,

$$\Delta x(D) = 2\Delta y x_i - 2\Delta x y_i + \underbrace{2\Delta y + \Delta x(2c-1)}_{\text{constant}}$$

$$\boxed{\Delta x(D) = 2\Delta y x_i - 2\Delta x y_i + A} \quad \text{—— (I)}$$

Raster unit $\quad \therefore \underset{\to 1}{\Delta x}(D) = 1 \cdot D$

So, the decision parameter D_i at any ith step along a line is given by $D_i = 2\Delta y x_i - 2\Delta x y_i + A$

If $D_i = -ve$, we will choose the lower pixel
$D_i = +ve$, " upper pixel

Let at any step $i+1$, the decision parameter will be given as

$$\boxed{D_{i+1} = 2\Delta y x_{i+1} - 2\Delta x y_{i+1} + A} \quad \text{—— (II)}$$

Subtract (II) from (I),

$$D_{i+1} - D_i = 2\Delta y(x_{i+1} - x_i) - 2\Delta x(y_{i+1} - y_i)$$

But $x_{i+1} = x_i + 1$

$$D_{i+1} - D_i = 2\Delta y x_i + 2\Delta y - 2\Delta y x_i - 2\Delta x(y_{i+1} - y_i)$$

$$\boxed{D_{i+1} = D_i + 2\Delta y - 2\Delta x(y_{i+1} - y_i)} \quad \text{——— (III)}$$

Now, the first Decision Parameter D_0 at (x_0, y_0)

$$\boxed{D_0 = 2\Delta y - \Delta x} \quad \{\text{from eq}^n \text{ (I)}\}$$

ALGORITHM :-

1) Input line co-ordinates (x_1, y_1) & (x_2, y_2)

2) Calculate $\Delta x = x_2 - x_1$
$\Delta y = y_2 - y_1$
$2\Delta y$ and $2\Delta y - \Delta x$

3) Assign $i = 0$
$D_0 = 2\Delta y - \Delta x$

4) Plot pixels at (x_i, y_i)

5) For each integer x co-ordinate x_i
if $D_i < 0$, then
Plot pixels at (x_{i+1}, y_i)
$D_{i+1} = D_i + 2\Delta y$
else
Plot pixel at (x_{i+1}, y_{i+1})

$D_{i+1} = D_i + 2\Delta y - 2\Delta x$

Increment i

6) Repeat step 5 Δx times

7) EXIT

from eq^n (11),

if $D_i < 0$, choose d_{lower} i.e. (x_i+1, y_i)

then

$$D_{i+1} = D_i + 2\Delta y - 2\Delta x(y_i - y_i)$$

$$\boxed{D_{i+1} = D_i + 2\Delta y}$$

else if

$D_i \geq 0$, choose d_{upper} i.e. (x_i+1, y_i+1)

$$D_{i+1} = D_i + 2\Delta y - 2\Delta x\,(y_i+1-y_i)$$

$$\Rightarrow \boxed{D_{i+1} = D_i + 2\Delta y - 2\Delta x}$$

from (1), for first Decision Parameter D_0

$$D_0 = 2m(x_i+1) - 2y_i + 2c - 1$$

$$= 2mx_i + 2m - 2y_i + 2c - 1$$

$$= 2(\underbrace{mx_i + c - y_i}_{\to 0}) + 2m - 1$$

$$= 2m - 1$$

$$= \frac{2\Delta y}{\Delta x} - 1$$

$$\boxed{D_0 = 2\Delta y - \Delta x}$$

Hence we have

$$D_{i+1} = \begin{cases} D_i + 2\Delta y - 2\Delta x, & \text{if } D_i \geq 0 \\ D_i + 2\Delta y, & \text{if } D_i < 0 \end{cases}$$

and, ~~for~~ for using first step

$$D_0 = 2\Delta y - \Delta x$$

Ques) Draw the line using Bresenham Line Algo with end points (20,10) and (30,18)

i	D_i	x_{i+1}, y_{i+1}	i	D_i	x_{i+1}, y_{i+1}
0	6	(21,11)	5	6	(26,15)
1	2	(22,12)	6	2	(27,16)
2	-2	(23,12)	7	-2	(28,16)
3	14	(24,13)	8	14	(29,17)
4	10	(25,14)	9	10	(30,18)

$\Delta x = 30 - 20 = 10$

$\Delta y = 18 - 10 = 8$

Now. $D_0 = 2\Delta y - \Delta x = (2 \cdot 8) - 10 = 6$

Here $D_0 > 0$ => next point will be (21,11)

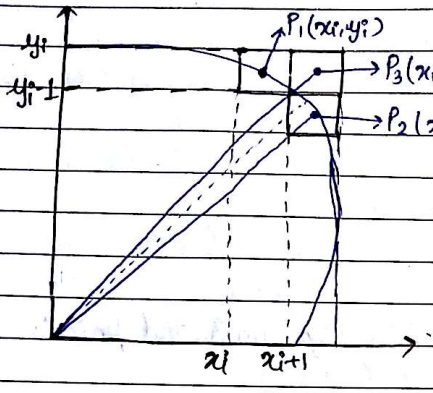
Now $D_{i+1} = D_i + 2\Delta y - 2\Delta x$

$= 6 + (2 \cdot 8) - (2 \cdot 10) = 6 + 16 - 20 = 2$

Again $D > 0$, next pixel will be (22,12)

Date

→ Bresenham Circle Algo-



Eq^n of circle: $x^2+y^2=r^2$

$\therefore dP_2 = (x_i+1)^2 + (y_i-1)^2 - r^2$ —— (I)

$dP_3 = (x_i+1)^2 + (y_i^2 - r^2)$ —— (II)

Now, $D = dP_3 - (-dP_2)$

or $D = dP_3 + dP_2$

$$D = (x_i+1)^2 + y_i^2 - r^2 + (x_i+1)^2 + (y_i-1)^2 - r^2$$
$$= 2(x_i+1)^2 + (y_i-1)^2 + y_i^2 - 2r^2 \quad \text{—— A}$$

for D_{i+1},

$$D_{i+1} = 2(x_{i+1}+1)^2 + (y_{i+1}-1)^2 + (y_{i+1})^2 - 2r^2 \quad \text{—— B}$$

Subtract B−A,

put $x_{i+1} = x_i + 1$

$$D_{i+1} - D_i = 2(x_i+2)^2 + (y_{i+1}-1)^2 + (y_{i+1})^2 - \cancel{2r^2} - [2[x_i+1]^2 + (y_i-1)^2 + y_i^2 - \cancel{2r^2}]$$

$$= \cancel{2x_i^2} + 8 + 8x_i + y_{i+1}^2 + \cancel{1} - 2y_{i+1} + y_{i+1}^2 - \cancel{2x_i^2} - 2 - 4x_i - \cancel{y_i^2} - \cancel{1} + 2y_i - y_i^2$$

or $D_{i+1} - D_i = 4x_i + 6 + 2y_{i+1}^2 - 2y_i^2 - 2y_{i+1} + 2y_i$

$$\Rightarrow \boxed{D_{i+1} = D_i + [4x_i + 6 + (y_{i+1}-1)^2 + y_{i+1}^2 + 2y_i - 2y_i^2 - 1]} \quad \text{—— (C)}$$

If $D_i < 0$, then $dP_2 > dP_3$ $\Rightarrow$ select Pixel P_3
i.e. $y_{i+1} = y_i$

$\therefore$ eq^n (C) becomes, $D_{i+1} = D_i + 4x + 6 + (y_i - 1)^2 + (y_i)^2 + 2y_i - 2y_i^2 - 1$
$= D_i + 4x + 6 + \cancel{y_i^2} + \cancel{1} - \cancel{2y_i} + \cancel{y_i^2} + \cancel{2y_i} - \cancel{2y_i^2} - \cancel{1}$

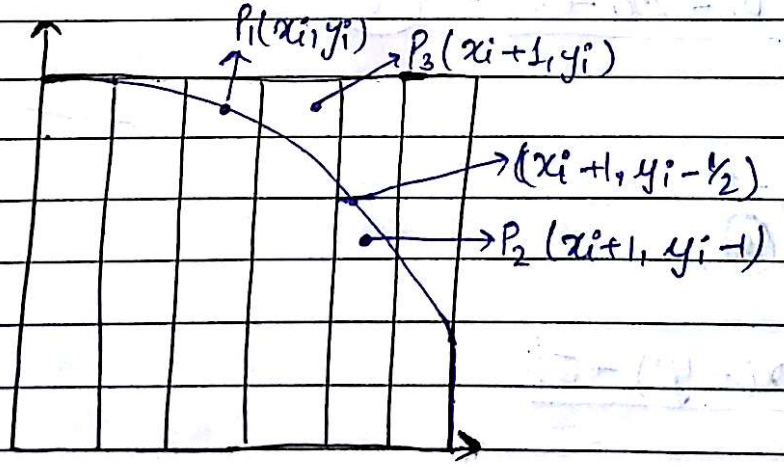
$$\Rightarrow \boxed{D_i + 1 = D_i + 4x + 6} \quad \text{— (D)}$$

if $D_i > 0$, then $dP_3 > dP_2$ $\Rightarrow$ select Pixel P_2
i.e. $y_{i+1} = y_i - 1$

$$\Rightarrow \boxed{D_i = D_i + 4(x_i - y_i) + 10} \quad \text{— (E)}$$

for first co-ordinate, $(x_i, y_i) = (0, r)$
Putting these values in eq^n (A), we get $\boxed{D = 3 - 2r}$

→ Mid-point Circle Algo-



Consider mid-point exactly b/w the points P_3 and P_2 with co ordinates $(x_i+1, y_i - 1/2)$

Eq^n of a circle $= x^2+y^2-r^2$

$\Rightarrow$ Decision parameter, $\boxed{D_i = (x_i+1)^2 + (y_i - 1/2)^2 - r^2}$ — (I)

Hence, $D_{i+1} = (x_{i+1}+1)^2 + (y_{i+1} - 1/2)^2 - r^2$

Since $x_{i+1} = x_i + 1$

$\Rightarrow D_{i+1} = (x_i+2)^2 + (y_{i+1} - 1/2)^2 - r^2$ — (II)

(II) − (I),

$D_{i+1} - D_i = (x_i+2)^2 + (y_{i+1} - 1/2)^2 - \cancel{r^2} - (x_i+1)^2 - (y_i - 1/2)^2 + \cancel{r^2}$

or $D_{i+1} = D_i + 2x_i + (y_{i+1} - 1/2)^2 - (y_i - 1/2)^2 + 2$ — (III)

If $D_i < 0$, select P_3 i.e. $y_{i+1} = y_i$

Put in eq^n (III),

$D_{i+1} = D_i + 2(x_i+1) + 1$

or $\boxed{D_{i+1} = D_i + 2x_i + 3}$ — (IV)

If $D_i > 0$, select P_2

i.e. $y_{i+1} = y_i - 1$

Put in eq^n (III),

$\boxed{D_{i+1} = D_i + 2(x_i - y_i) + 5}$

Now, put $\left.\begin{matrix} x_i = 0 \\ y_i = r \end{matrix}\right\}$ in (I)

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

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

or $D = \frac{5}{4} - r$

or $\boxed{D = 1 - r}$

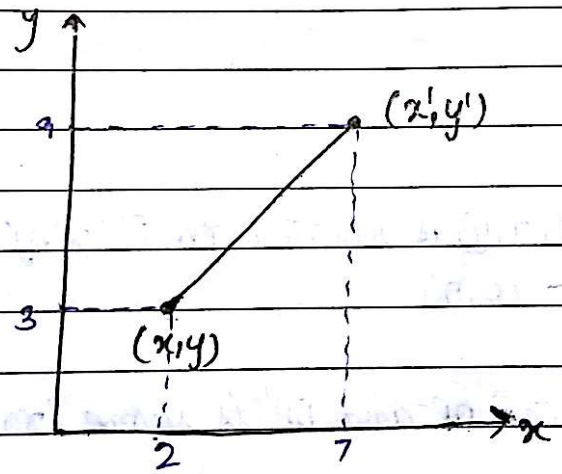
→ Transformation-

Translation
Scaling
Rotation

Reflection

1) Translation:

$t_x, t_y \rightarrow$ translation factors



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

In matrix form,

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

2) Scaling:

$x' = x * S_x$
$y' = y * S_y$

In matrix form,

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

Ques) Consider a triangle (0,0), (1,1), (5,2).

a) Magnify the Δ to twice its size

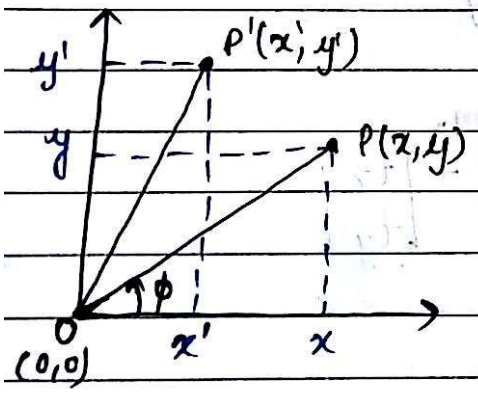
b) Reduce the Δ to half of its size

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} A \\ B \\ C \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 5 \\ 0 & 1 & 2 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 2 & 10 \\ 0 & 2 & 4 \\ 1 & 1 & 1 \end{bmatrix}$$

i.e. $\begin{bmatrix} \\ \\ \end{bmatrix} = \begin{bmatrix} \\ \\ \end{bmatrix}$

3) Rotation-



Here the point P(x,y) is rotated to P'(x',y') about the origin (0,0)

Now the distance OP and OP' is same (radi

Here,

$$\left.\begin{aligned} x &= r\cos\phi \\ y &= r\sin\phi \end{aligned}\right\} \text{—— (1)}$$

and

$$\left.\begin{aligned} x' &= r\cos(\theta+\phi) = r\cos\theta \cdot \cos\phi + r\sin\phi \cdot \sin\theta \\ y' &= r\sin(\theta+\phi) = r\sin\theta\cos\phi + r\sin\phi\cos\theta \end{aligned}\right\} \text{(2)}$$

Substituting eqⁿ (1) in (2), we get

$$\left\{\begin{aligned} x' &= x\cos\theta - y\sin\theta \\ y' &= x\sin\theta + y\cos\theta \end{aligned}\right\} \text{—— (III)}$$

Date

In matrix form,

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

Suppose we want to ~~ro~~ rotate P' back to P.
i.e. We want to perform Inverse Rotation.
Then the rotation angle will be $-\theta$.

$$T^{-1} = \begin{bmatrix} \cos(-\theta) & -\sin(-\theta) \\ \sin(-\theta) & \cos(-\theta) \end{bmatrix}$$

Or $$T^{-1} = \begin{bmatrix} \cos\theta & +\sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

Now, $[T][T^{-1}] = I$

Q) Write 2x2 trans matrix for the

i) Counterclockwise by π ii) cw by $\pi/2$

Ques) Rotate a Δ with co-ordinates A(0,0), B(6,0), C(3,3) by 90° about origin in anti-clockwise dir^n.

→ Scaling w.r.t any arbitrary point -

1) Translate ~~on~~ ~~of~~ a reference point (x_r, y_r) to origin

$$T_{trans} = \begin{bmatrix} 0 & 0 & -x_r \\ 0 & 0 & -y_r \\ 0 & 0 & 1 \end{bmatrix}$$

2) Scale the object by S_x and S_y is

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

3) Translate the (x_r, y_r) to its original position

i.e. $$T'_{trans} = \begin{bmatrix} 0 & 0 & x_r \\ 0 & 0 & y_r \\ 0 & 0 & 1 \end{bmatrix}$$

$$\boxed{T_{composite} = T_{trans} \times T_{scale} \times T^{-1}_{trans}}$$

$$\begin{bmatrix} x_r \\ y_r \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & x_r(S_x - 1) \\ 0 & S_y & y_r(S_y - 1) \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

Ques) Find the transformation matrix A, B, C, D with co-ordinates A (0,0), B(0,4), C(4,4), D(4,0) where the center is (2,2). Reduce the size of the square to its half. Find the new co-ordinate

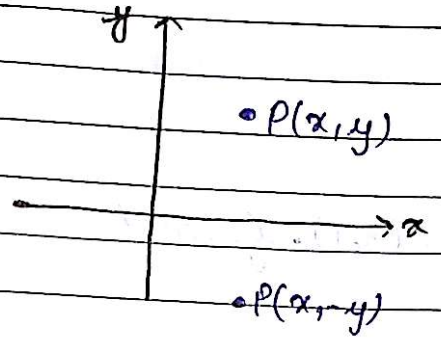
→ Rotation through an arbitrary point-

Same 3 steps

$$[T_{comp}] = T_{trans} \cdot R_\theta \cdot T^{-1}_{trans}$$

$$= \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ -x_r(\cos\theta - 1) + y_r \sin\theta & -y_r(\cos\theta - 1) - x_r \sin\theta & 1 \end{bmatrix}$$

→ Reflection



1) through x-axis -

i.e. $x' = x$, $y' = -y$

Matrix form: $R_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$

Then,

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

$$\boxed{X^* = R_x [X]}$$

2) through y-axis -

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

3) through the straight line $y = x$ -

$x' = y$; $y' = x$

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

Ques) Consider a box ABCD where A(2,6), B(-3,6), C(-3,3), D(2,3). Generate the reflection image through $y = x$.

for A': $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 2 \\ 6 \end{bmatrix}$

$x' = 6$

$y' = 2$

for B: $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} -3 \\ 6 \end{bmatrix}$

$x' = 6$

$y' = -3$

4) through $y=-x$ –

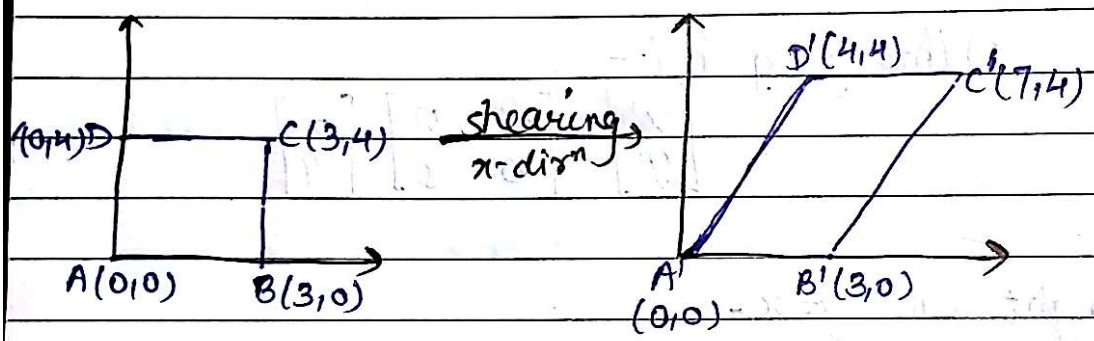
$x' = -y$; $y' = -x$

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

Ques) Consider a ΔABC where $A(2,1)$, $B(2,3)$ & $C(3,1)$. Generate the reflected image against $y=-x$.

$A'(-1,-2)$ $B(-3,-2)$ $C(-1,-3)$

→ <u>Shearing</u>–



1) In x-shearing, y-coordinate remains same / unchanged

Let $b=1$

↳ shearing factor

$$[x' \ y'] = [x \ y] \begin{bmatrix} 1 & 0 \\ b & 1 \end{bmatrix}$$

$$= [x+yb \ \ y]$$

$x' = x + yb$

$y' = y$

2) In y-shearing, x-co-ordinate remains unchanged

$a \rightarrow$ shearing factor

$$[x' \ y'] = [x \ y] \begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix}$$

$$= [x \ \ ax+y]$$

$x' = x$

$y' = y + ax$

eg. $A(1,1)$, $B(2,4)$, $C(4,3)$, $D(1,0)$; shearing factor $= 2$
$\parallel$
a

Generate shearing image along y-axis.

for A: $x' = x = 2$

$y' = y + ax$

$= 1 + 2(1) = 3$

$A'(1,3)$

for B: $x' = 2$

$y' = 4 + 2(2)$

$= 8$

$B'(2,8)$

3) Shearing along both x and y axes -

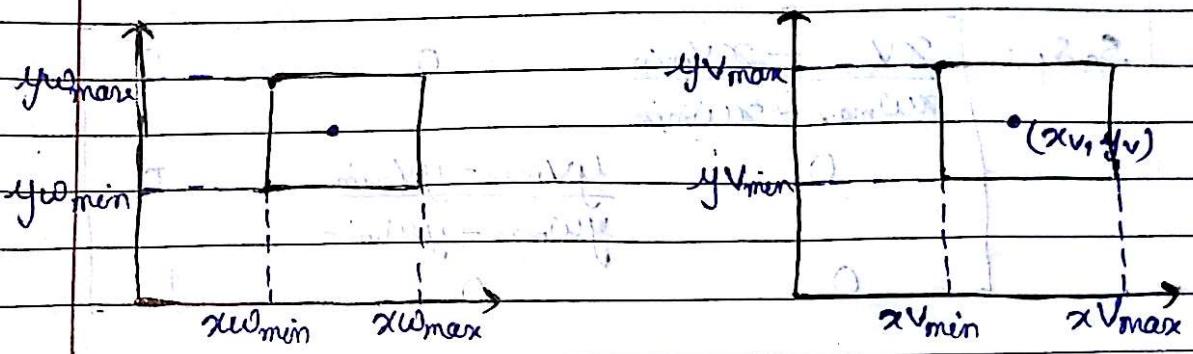
Both co-ordinates will change

$$[x' \ y'] = [x \ y] \begin{bmatrix} 1 & a \\ b & 1 \end{bmatrix} \cancel{[x \ y]}$$

$$= [x + yb \quad y + ax]$$

$$\Rightarrow \boxed{\begin{aligned} x' &= x + yb \\ y' &= y + ax \end{aligned}}$$

eg.



(Window) (Viewport)

To maintain relative placement in viewport as in window we require

$$\frac{x_w - x_{w_{min}}}{x_{w_{max}} - x_{w_{min}}} = \frac{x_V - x_{V_{min}}}{x_{V_{max}} - x_{V_{min}}}$$

and

$$\frac{y_w - y_{w_{min}}}{y_{w_{max}} - y_{w_{min}}} = \frac{y_V - y_{V_{max}}}{y_{V_{max}} - y_{V_{min}}}$$

$$\Rightarrow x_V = \frac{x_{V_{max}} - x_{V_{min}}}{x_{w_{max}} - x_{w_{min}}} (x_w - x_{w_{min}}) + x_{V_{min}}$$

$$y_V = \frac{y_{V_{max}} - y_{V_{min}}}{y_{w_{max}} - y_{w_{min}}} (y_w - y_{w_{min}}) + y_{V_{min}}$$

$$V = N \cdot W$$

$$N = T_w^{-1} \cdot S_x S_y T_v$$

$$T_w^{-1} = \begin{bmatrix} 1 & 0 & -x_{w_{min}} \\ 0 & 1 & -y_{w_{min}} \\ 0 & 0 & 1 \end{bmatrix}$$

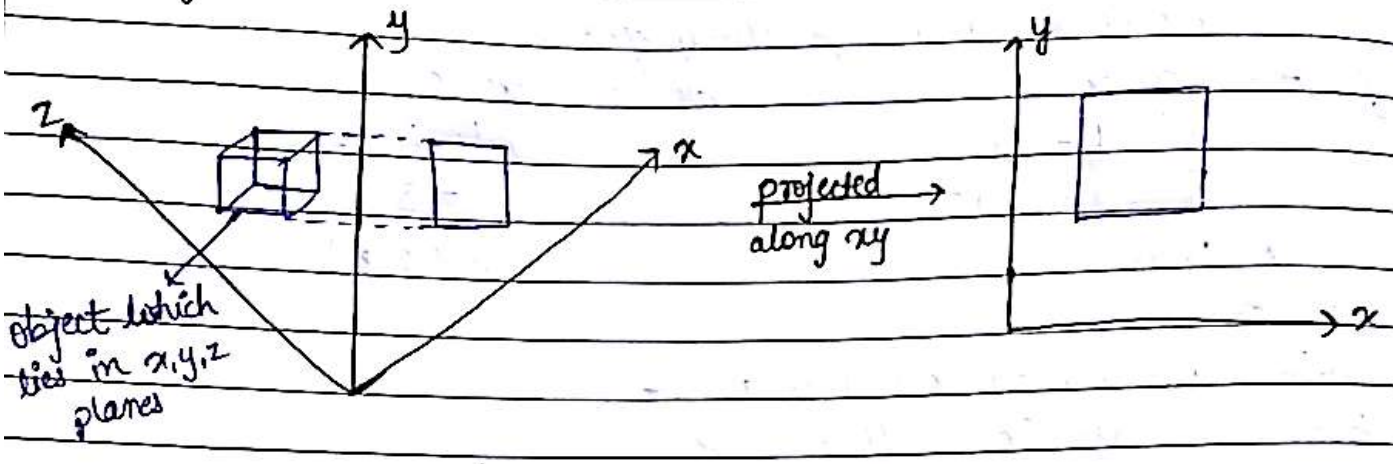
$$S_xS_y = \begin{bmatrix} \frac{xV_{max} - xV_{min}}{xW_{max} - xW_{min}} & 0 & 0 \\ 0 & \frac{yV_{max} - yV_{min}}{yW_{max} - yW_{min}} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$T_V = \begin{bmatrix} 1 & 0 & xV_{min} \\ 0 & 1 & yV_{min} \\ 0 & 0 & 1 \end{bmatrix}$$

Find the normalised transformation matrix for Windows to viewport which uses the rectangle whose lower left corner is at (2,2) and upper right corner is at (6,10) a a window. The viewport has lower left corner at (0,0) and upper right corner at (1,1).

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

→ Projection-



Consider an eq^n for a line which is passing through a line at point $(x_1, y_1$ and the direction of the projection having parametric eq^n as

$$x = x_1 + x_p \cdot u \quad \text{—— (I)}$$

$$y = y_1 + y_p \cdot u \quad \text{—— (II)}$$

$$z = z_1 + z_p \cdot u \quad \text{—— (III)}$$

In parallel projection, we assume that z-axis = 0.

⇒ from eq^n (III), $0 = z_1 + z_p \cdot u$

or $u = \frac{-z_1}{z_p}$

Put this value in eq^n (I) & (II),

$$x = x_1 + x_p\left(\frac{-z_1}{z_p}\right)$$

$$y = y_1 - z_1\left(\frac{y_e}{z_p}\right)$$

or $[x' \; y' \; z' \; 1] = [x \; y \; z \; 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -x_p/z_p & -y_p/z_p & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Types of Projection -

- Projection
  - Parallel
    - Orthographic
    - Axnometric
      - Isometric
      - Di-metric
      - Tri-metric
    - Oblique
      - Cabinet
      - Cavalier
  - Perspective
    - 1pt.
    - 2pt.
    - 3pt.

Orthographic – Is a part of parallel projⁿ in which projectors are perpendicular to the projection plane

at $x=0$

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

at $y=0$

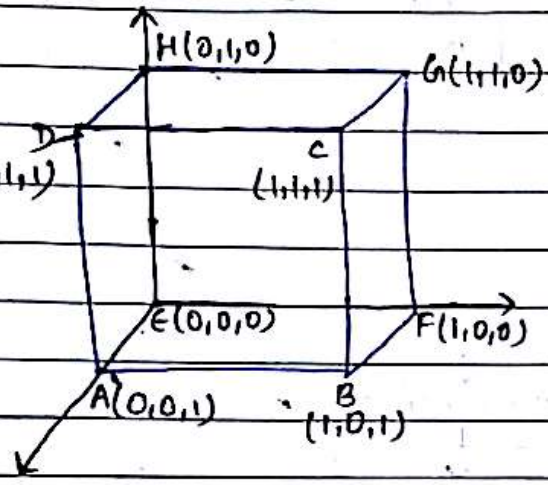
$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

at $z=0$

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

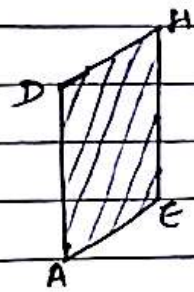
V. Imp.

Ques) Find out the orthographic projection of the unit cube along x, y and z=0 plane.



$$\begin{array}{c} A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \end{array} \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \rightarrow \text{cubic matrix}$$

at $x=0$, [cubic matrix] $\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{matrix} A \\ \\ D \\ \\ E \\ \\ H \\ \\ \end{matrix}$



Date

Axnometric-

1) Isometric- The direction of projection makes equal angles with all the three principle axes. It is a way of presenting a drawing in 3-D.

Normally, designs are usually drawn at 30° in isometric projection. Since all the directions of projection make equal angles

$$\therefore [d_x] = [d_y] = [d_z]$$

Transforming the unit vector along x-axis using isometric projn

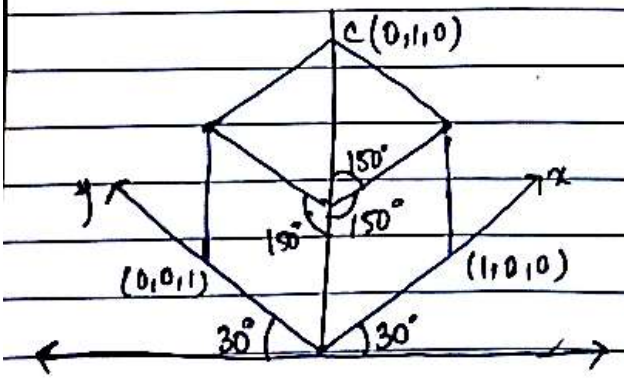
$$[U_x] = [1\ 0\ 0\ 1] \begin{bmatrix} \cos\phi & \sin\phi\sin\theta & 0 & 0 \\ 0 & \cos\theta & 0 & 0 \\ \sin\phi & -\cos\phi\sin\theta & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= [\cos\phi \quad \sin\phi\sin\theta \quad 0 \quad 1] \text{ — (A)}$$

along y-axis, $[0 \quad \cos\theta \quad 0 \quad 1]$ — (B)

along z-axis, $[\sin\phi \quad -\cos\phi\sin\theta \quad 0 \quad 1]$ — (C)

eqⁿs (A), (B) & (C) are called 'Foreshortening Factors'.



Foreshortening factor is the ratio of projected length to its true length.

$$\Rightarrow F_x^2 = \cos^2\phi + \sin^2\phi\sin^2\theta$$

$$F_y^2 = \cos^2\theta$$

$$F_z^2 = \sin^2\phi + \cos^2\phi\sin^2\theta$$

Date

In isometric projⁿ, $F_x^2 = F_y^2 = F_z^2$
ie. all the foreshortening factors are equal.

equating F_x^2 & F_y^2

$$\cos^2\phi + \sin^2\phi \sin^2\theta = \cos^2\theta$$

$$\Rightarrow \sin^2\phi = \frac{\sin^2\theta}{(1-\sin^2\theta)} \quad \text{—— (I)}$$

Similarly, $F_y^2 = F_z^2$

$$\sin^2\phi = \frac{1-2\sin^2\theta}{1-\sin^2\theta} \quad \text{—— (II)}$$

~~Adding~~ Equating (II) & (I)

$$3\sin^2\theta = 1$$

$$\sin\theta = \pm\sqrt{1/3}$$

$$\{\theta = \pm 35.26\}$$

2) Dimetric – Direction of projection makes equal angle with exactly two of the principle axes

Consider a rotation about y-axis by angle ϕ followed by rotation about x-axis by an angle θ and $z=0$.

$$[T] = [R_y][R_x][R_z]$$

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

Date

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

3) Tri-metric – Directions of proj^n make unequal angles with the three principle axes.
- Formed by arbitrary rotation in arbitrary order about any or all the co-ordinate axes followed by parallel projection onto $z=0$ plane.
- For each projected principle axis, four shortening factors are different.

For any tri-metric proj^n, the four S.F. are obtained by applying the concantenation of the unit vectors along the principle axis.

i.e. $[U][T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} [T] = \begin{bmatrix} x_x' & y_x' & 0 & 1 \\ x_y' & y_y' & 0 & 1 \\ x_z' & y_z' & 0 & 1 \end{bmatrix}$

$$F.F = F_x^2 = x_x^2 + y_x^2$$
$$F_y^2 = x_y^2 + y_y^2$$
$$F_z^2 = x_z^2 + y_z^2$$

→ Oblique Projection –

When the angle b/w the projector and the plane of projection $\neq 90°$, then the projection is called oblique proj^n.

This form of proj^n is seen in the shadows of objects formed by sunlight.

1) Cavalier - Projⁿ makes 45° with the plane of projⁿ.
- The line of projⁿ is ⊥ to the viewplane and as the same length as the line itself.

2) Cabinet - Makes 63.4° with the plane of projⁿ
- The line of projⁿ is ⊥ to the viewplane and is projected at half of their actual length.

Projⁿ co-ordinates are,

$$\boxed{\begin{aligned} x_p &= x + Z_p L\cos\theta \\ y_p &= y + Z_p L\sin\theta \end{aligned}} \quad \begin{matrix} \text{— (I)} \\ \text{— (II)} \end{matrix}$$

In matrix form,

$$\begin{bmatrix} x_p \\ y_p \\ Z_p \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & L\cos\theta & 0 \\ 0 & 1 & L\sin\theta & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ Z \\ 1 \end{bmatrix}$$

→ Perspective Projⁿ -

→ 1 pt
→ 2 pt
→ 3 pt

} Vanishing Point

In 1 vanishing pt., when plane of projⁿ intercepts only one principle axis.
In 2 vanishing pt. → POP intercepts two PAs
In 3 " → POP " all the three PAs

In 1 vanishing point, when POP intercepts at x-axis

$$[x' \; y' \; z' \; 1] = [x \; y \; z \; 1] \begin{bmatrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where p → point where 1 vanishing pt. will intercept x-axis

$$[x \quad y \quad z \quad px+1]$$

$$\left[\frac{x}{px+1} \quad \frac{y}{px+1} \quad \frac{z}{px+1} \quad 1\right] =$$

→ Spline Representation -

Spline Curves
- Piecewise cubic spline
- Berger curve
- B-spline

Splines are smooth curves passing through a set of given points.
The curve refers to composite curve which is formed by polynomial