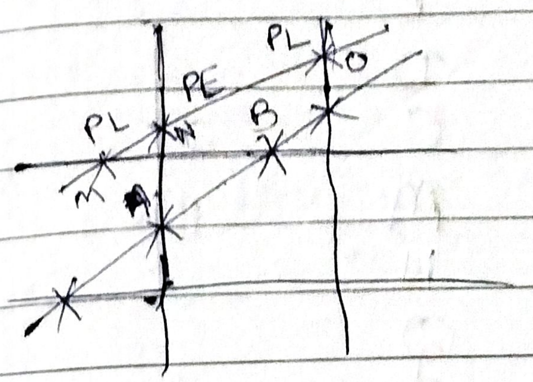
Liang - Barsky Algo



$PL \neq PE$

$t_L < t_E \Rightarrow$ Reject

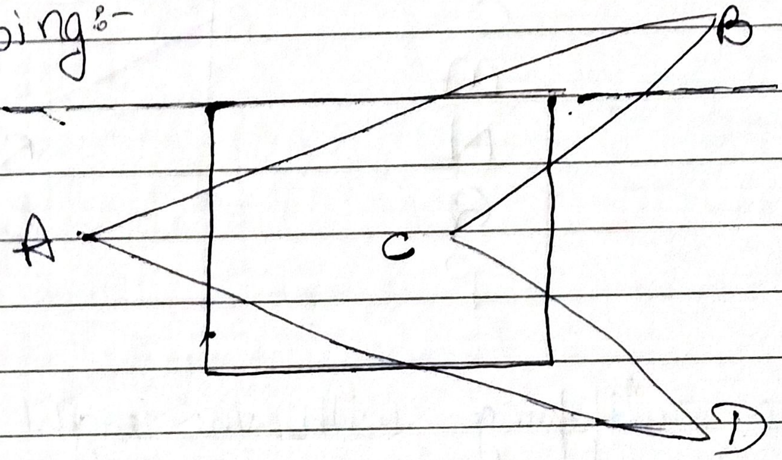
$t_m = PL$

$t_N = PE$

$t_o = PL$

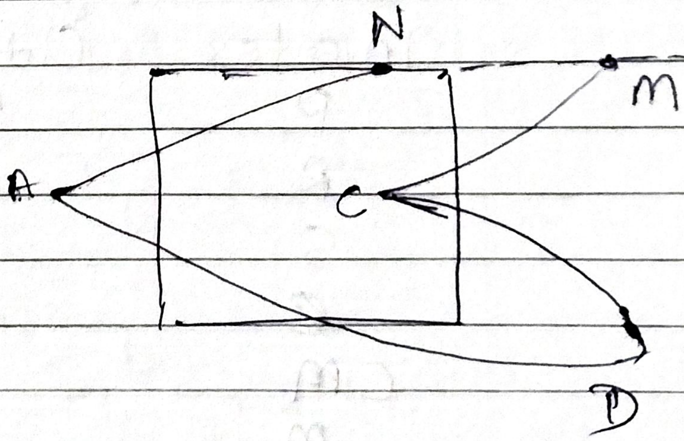
Sutherland Hodgeman Polygon Clipping Algo

The datastructures to be used in this algo ~~inver~~ invertex array and outvertex array. We always move in counter clockwise direction. There are 4 cases of polygon clipping:-



~~Invertex~~
D
D
C
B
A

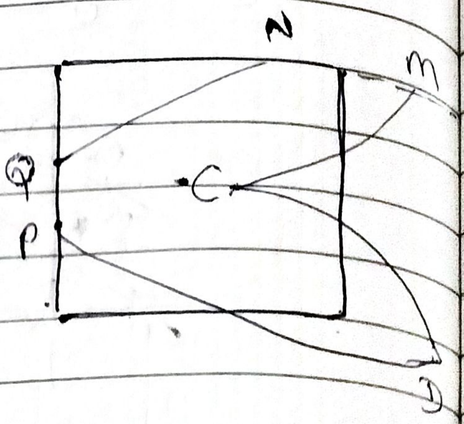
Invertex	Outvertex
A	A
D	D
C	C
B	M
~~A~~	N
A	A



Now we perform the clipping with left edge of the clip area.

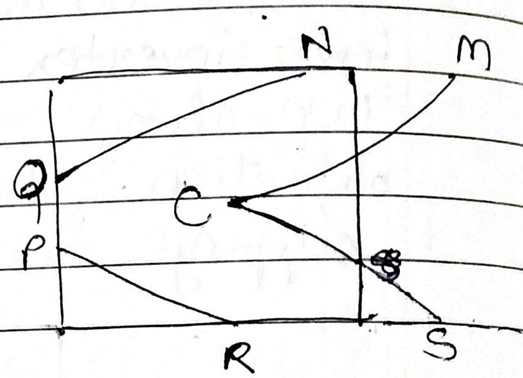
Date : / /
Page No.

Invertex	Outvertex
A	P
D	D
C	C
M	M
N	N
A	Q
	P



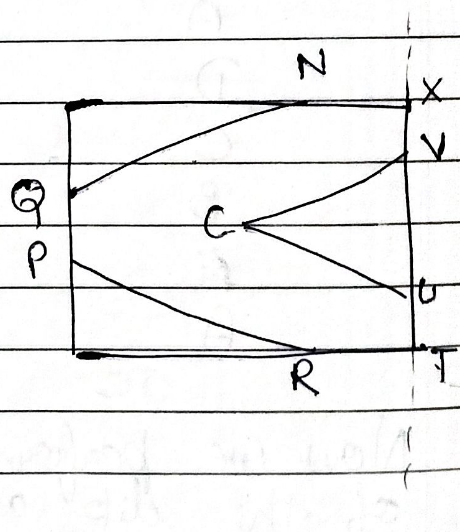
Now we are clipping with the bottom edge of the clip area

Invertex	Outvertex
P	P
D	R
	S
C	C
M	M
N	N
Q	Q
P	P



Now we are clipping with the right ~~edge~~ edge of the clip area

Invertex	Outvertex
P	P
R	R
S	~~S~~ T
~~C~~	~~C~~ U
C ~~M~~	C
M	V
	X
N	N

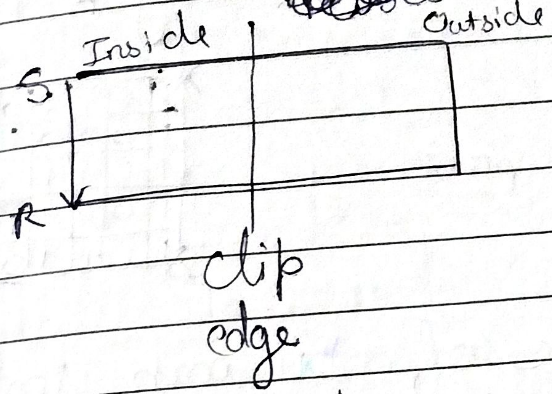


P Q   P Q

Sutherland Hodgeman Polygon Clipping Algo

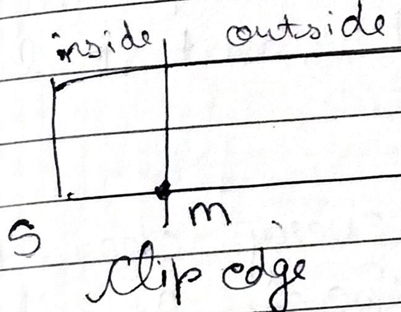
4 cases

I



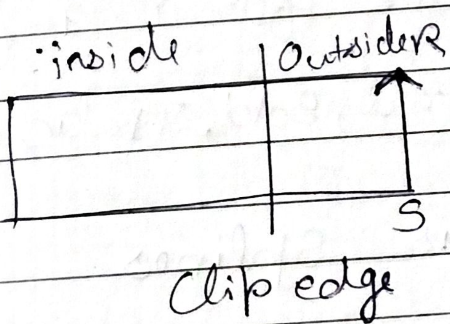
In	Outvertex
S	S
R	R

II



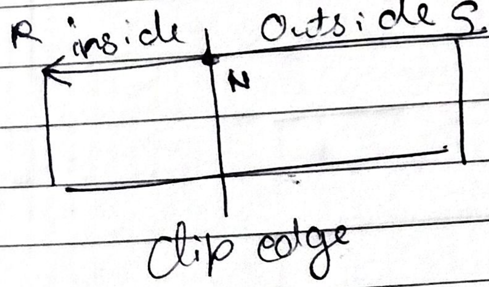
Invertex	Outvertex
S	S
R	M

III



Invertex	Out ~~side~~ vertex
S	NULL
R	

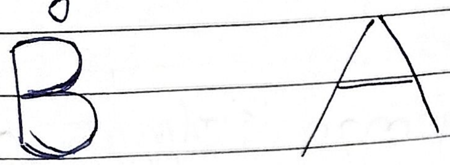
IV



Invertex	Outvertex
S	N
R	R

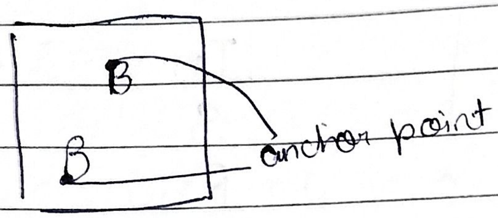
Date : / /
Page No.

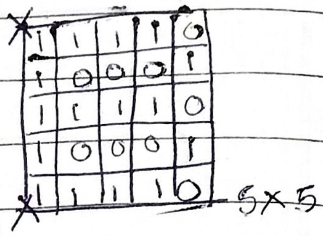
Generating Characters



2 methods

i). Using A Bitmap





Increasing the size of bit map improves the shape of the character, but it leads to higher memory usage.

Saving the bit map for every character takes a lot of memory space a separate bit map is to be save for all 4 faces.

Bold Italic, Normal, Bold, Italic

Generating Characters Splines

B(x, y) = ________

A(x, y) = ________

S(x, y) = ________

%(x, y) = ________

→ A lot of calculation is needed for generating the character on the screen. ~~Comp~~

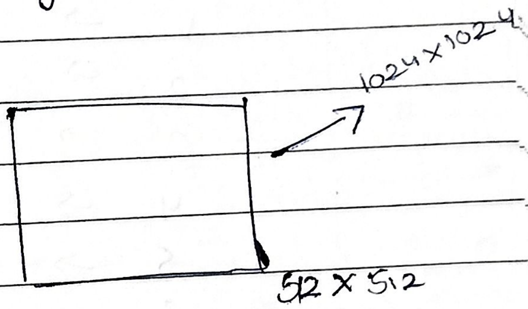
Date : / /
Page No.

- Complex to impliment.
- Diff: to draw the character with other 3 cases (Bold, Italic, Bold Italic)
- But it takes less memory space as compared to the bit map method.

Jaggies → Zig zag effect
→ Aliasing

Methods to deal with aliasing ~~are~~ called anti-aliasing technique.

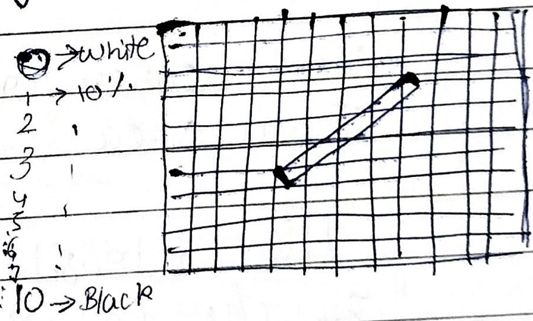
1) Increasing Resolution



Increasing resolution only reduce the aliasing effect as more pixel ~~needs~~ means better ~~at~~ precision.

Improvement in the zig-zag effect is at the cost of higher memory usage & increased scan ~~vortex~~ converting time.

2) Unweighted Area Sampling



There are 3 properties

I Intensity of pixcel intersected by a line edge decreases as the distance b/w the line edge & ~~primitive~~ pixcel increases.

II Primitive (is the line has to be drawn) can't influence the intesity, if the primitive doesn't intersect the pixcel.

III Equal areas contribute equally regardless of distance from the centre.

3) Weighted Area Sampling

Property I & II remains the same as unweighted area sampling.

III 0 → Black → white

1 → 10 %

2 → 20 %

3 → 30 %

4 → 40 %

5 → 50 %

6 → 60 %

7 → 70 %

8 → 80 %

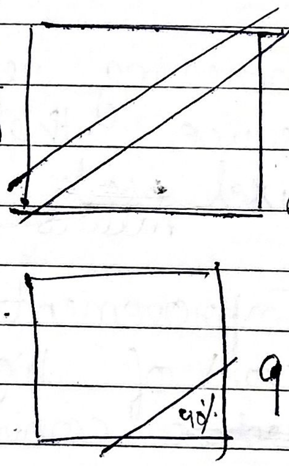
9 → 90 %

10 → Black

weighing function

4	2	4
3	5	3
4	2	4

2	1	2
1	3	1
2	1	2



40%

13 } centre darker colors

90%

9 } corner light color

Area nearer to the centre has more effect than area covered at the corners.

The intensity of the pixcel will be decided on the basis of % of pixcel covered & its weight.

Date : / /
Page No.

Z-buffer Algo

Z-buffer → $\overset{1024}{x} \times \overset{1024}{y}$

→ 0

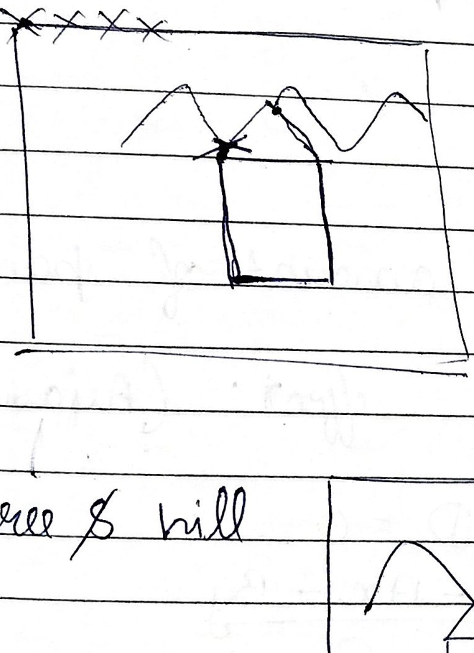
Frame buffer = bg colour initially

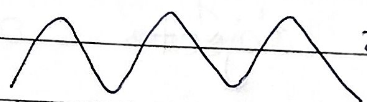
Frame buffer → $x \times y$ ↓ saves frame value for every xy of the screen.

Z-buffer → $\overset{1024}{x} \times \overset{1024}{y}$ → all zeroes initially

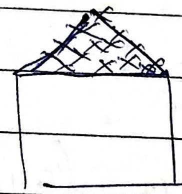
```
for (x=0 ; x < xmax ; x++)
    for (y=0 ; y < ymax ; y++)
    {
        Zbuffer [x] [y] = 0 ;
        Framebuffer [x] [y] = Background colour;
    }
```

hut & hill





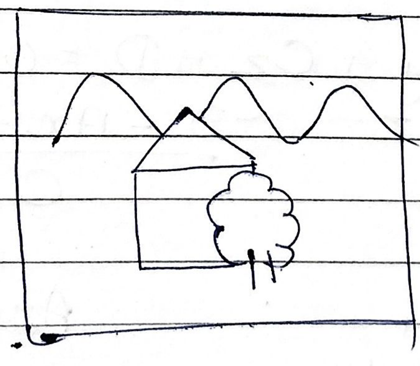
Z=2



Z = 5

Z=10

hut, tree & hill



Larger value will always in the front.

Date : / /
Page No.

For each object (polygon) in the scene (for each x, y of the current object)

```
{ pz = z value of object for x,y
    if ( pz > zbuffer [x] [y])
    {  zbuffer [x] [y] = pz;
       Framebuffer [x] [y] = colour of xy;
    }
}
```

Advantages of Zbuffer algorithm :-

1 It is used to draw any object of any shape.

2 Easy to implement.

3 Only 1 zvalue comparison is done for each objects x, y.

Disadvantages :-

1 It requires a large amount of space.

2 Might face zig-zag effect. (Aliasing).

$$Ax + By + Cz + D = 0$$

$$z = \frac{-D - Ax - By}{C}$$

Solve for z

$x_1 = x_1$ Assuming; $y_1 = y_1$

$x_2 = x_1 + 1$ $y_2 = y_1$

Date : / /
Page No.

$$z_1 = \frac{-D - Ax_1 - By_1}{C}$$

$$z_2 = \frac{-D - Ax_1 - By_1 + 1}{C}$$

$$z_2 = \frac{-D - Ax_2 - By_2}{C}$$

$$z_2 = \frac{-D - A(x_1 + 1) - By_1}{C}$$

$$z_2 = z_1 - \frac{A}{C}$$

Solve for y

$x_1 = x_2 = x_1$

$$z_1 = \frac{-D - Ax_1 - By_1}{C}$$

$y_1 = y_1$

$y_2 = y_1 + 1$

$$z_2 = \frac{-D - Ax_2 - By_2}{C}$$

$$z_2 = \frac{-D - Ax_1 - B(y_1 + 1)}{C}$$

$$z_2 = z_1 - \frac{B}{C}$$

5	3	3	5	1
1	0	2	1	
4	3	2		
1	2			
8				

Final z-buffer

5	3	3	5	1
2	3	4	5	6
4	3	3	4	5
8	9	6	5	4
8	2	1	0	0

5
2
3
0
6

0	0	0	0	0
0	3	4	5	6
1	2	3	4	5
8	9	6	5	4
3	2	1	0	0

A-buffer algo

A-buffer algo is z-buffer algo for trasparent / translucent objects.

Depth Sort Algo → ~~Par~~ Painter's algo

1 Sort on z-value.

2 Resolve any ambiguity or split.

3 Scan (→draw) convert each polygon starting from the smallest z value.
(nearest to the screen or farthest from the view point)

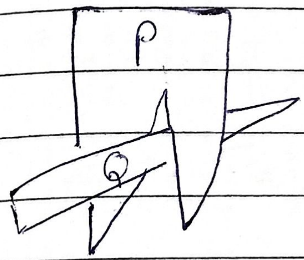
Note:- If we ignore the second step of spliting the object the algo is called painter's algo.

Date: / /
Page No.

To check is the split is required or not :-

- 5 ~~steps~~ tests these tests are ↑ order of complexity if any test comes true we know there is an ambiguity & split is needed. If the test is false we check the next test. If all the 5 tests fails that means there is no ambiguity & you can draw a polygon.

If we have to draw the polygon P we will check the polygon P against all the other polygon (Q) & all the 5 tests musts fails. for all other Q's.



T-1 X - ~~extends~~ extents don't overlap ⟶ T / F

$x_{min} - x_{max}$

T-2 Y- extents do not overlap → T/F

T-3 Is P entirely on the opposite side of Q's plain from the view point.

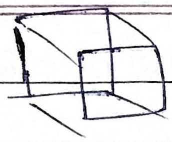
↳ Is P completely behind Q

T-4 Is Q entirely on the the same side of the P's plain on the view point

↳ P is completely in front of Q.

Date : / /
Page No.

T-5 3D - 2D projection



Perform ~~2D to 3D~~ 3D to 2D projection & check if the projects don't overlap.

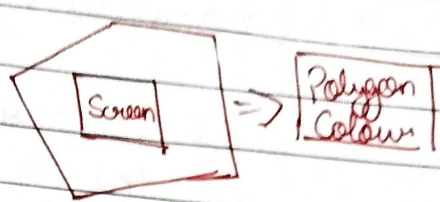
If all the stests fails then split has to be perform.

Date : / /
Page No.

Warnock's Algorithm

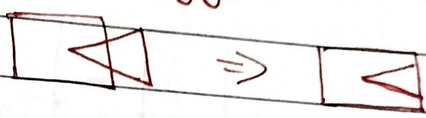
This is area sub-division algo. It uses divide & conquer strategy.

1) Surrounding Polygon



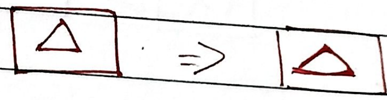
On the screen you will able to see:-
~~If it~~ the colour of the polygon.

2) Intersecting Polygon



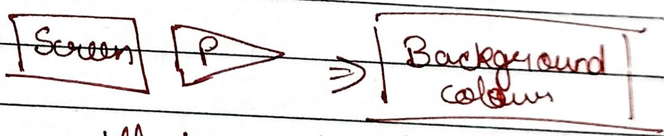
Only the part which is inside the screen is intersecting polygon

3) Contain Polygon



Complete polygon is on the screen is contain polygon.

4) Disjoint Polygon



Only background colour will be visible

Case I All the polygons are disjoint from the areas:
Background colour visible on the screen.

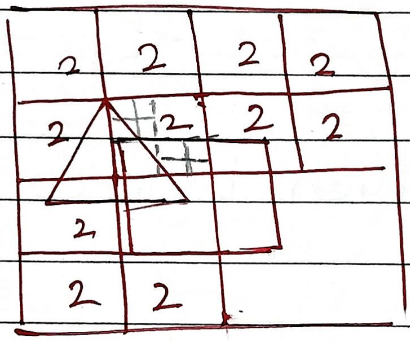
Case II: Only one intersecting or only one contain polygon:
Whatever part of polygon in the screen will be ~~dr~~

Case III There is one surrounding polygon, neither intersecting nor contained polygon:
Colour of the polygon on the screen.

Case IV More than one polygon is intersecting, contained or surrounding but the surrounding polygon is in front ~~the~~ of all others the surronding polygon has the maximum z value:

The colour of the surrounding polygon is shown on the screen.

Screen



Date : / /
Page No.

Ch-11

2D Transformations

$[x \quad y]$ → Row Matrix → $[x \quad y \quad z]$

$\begin{bmatrix} x \\ y \\ z \end{bmatrix}$ → ~~Colummn~~ Column Matrix

$$[x \quad y] \begin{bmatrix} a & b \\ c & d \end{bmatrix} = [(ax+cy) \; (bx+dy)]$$

Case1 $a = d = 1$ $b = c = 0$

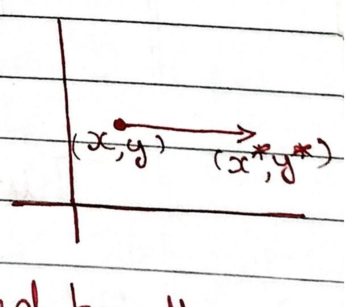
$$T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \Rightarrow \text{Identity}$$

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

position vector remains unchanged.

Case2 $d = 1$ $b = c = 0$

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



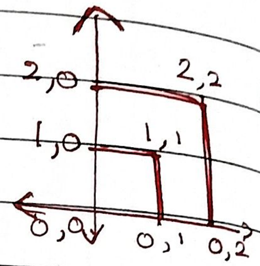
Note:- The point P's x-coordinate is scaled by the factor of a.

Case 3 $\quad b = c = 0$

$$[x \quad y]\begin{bmatrix} a & 0 \\ 0 & d \end{bmatrix} = [ax \quad dy] = [x^* \quad y^*]$$

x is scaled by factor of a.
y is scaled by factor of d.

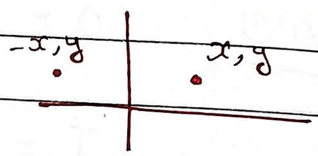
if $a = d \Rightarrow$ equal scaling
$a \neq d \Rightarrow$ unequal scaling



$a > 1 \Rightarrow$ enlargement is done.

$0 < a = d < 1 \Rightarrow$ compression is done.

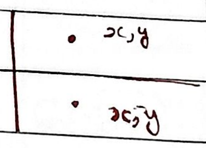
Case 4 $\quad b = c = 0 \qquad a = -1 \qquad d = 1$



$$[x \quad y]\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} = [-x \quad y]$$

This matrix is for a reflection through y-axis
y-axis ($x = 0$).

Case 5 $\quad b = c = 0 \qquad a = 1 \qquad d = -1$



$$[x, \ y]\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = [x \quad -y]$$

This is a reflection through the x-axis ($y = 0$)

Case 6 Reflection through the origin.
$b = c = 0 \qquad d = a < 0$

e: Both reflection & scaling involve only the diagonal terms of the transformation matrix.

Case 7

$a = d = 1 \qquad c = 0$

$$[x \quad y] \begin{bmatrix} 1 & b \\ 0 & 1 \end{bmatrix} = \cancel{[(x+b) \quad y]}$$

$$= [x \quad \underbrace{(bx+y)}_{\text{Shearing}}]$$

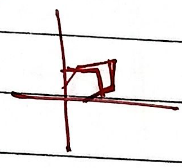
~~Shearing~~ Shearing

Shearing is performed in y proportional to x coordinate.

Case 8

$$[x \quad y] \begin{bmatrix} 1 & 0 \\ c & 1 \end{bmatrix} = [(x+cy) \quad y]$$

Shearing is performed in x proportional to y coordinate.



$P = [0 \quad 0]$

$$[0 \quad 0] \begin{bmatrix} a & b \\ c & d \end{bmatrix} = [0 \quad 0]$$

Note: The origin is invariant with respect to 2×2 transformation matrix.

↳ We can move the origin.

This is the reason why we use the homogenious coordinates.

Que

$A = [0 \quad 1]$

$B = [2 \quad 3]$

$T = \begin{bmatrix} 1 & 2 \\ 3 & 1 \end{bmatrix}$

Transform the line ab using the transformation matrix T.

$[A]\,[T] = [A^*]$

$[B]\,[T] = [B^*]$

$$\begin{array}{c} A \\ B \end{array} \begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 1 \\ 11 & 7 \end{bmatrix} \begin{array}{c} A^* \\ B^* \end{array}$$

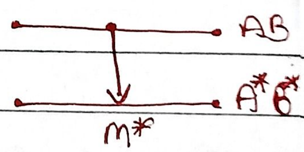
To Prove

1. There is a one to one correspondance b/w the points on ~~of~~ the line ab & the $a^* b^*$.

$A = [x_1 \quad y_1]$

$B = [x_2 \quad y_1]$

General T matrix = $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$



$$\begin{array}{c} A^* \\ B^* \end{array} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} (x_1 a + y_1 c) & (x_1 b + y_1 d) \\ (x_2 a + y_2 c) & (x_2 b + y_2 d) \end{bmatrix}_{2\times 2} \begin{array}{c} \rightarrow A^* \\ \rightarrow B^* \end{array}$$

mid point of $A^*B^* = \frac{A^* + B^*}{2}$ = ~~$ax_1 + y_1 c + a$~~

$$= \left[\frac{ax_1 + cy_1 + ax_2 + cy_2}{2} \quad \frac{bx_1 + dy_1 + bx_2 + dy_2}{2} \right] \quad \text{— (1)}$$

mid-point of AB = $m = \frac{A+B}{2}$

$$= \left[\frac{x_1 + x_2}{2} \quad \frac{y_1 + y_2}{2} \right] = m$$

$$m[T] = \left[\frac{x_1 + x_2}{2} \quad \frac{y_1 + y_2}{2} \right] \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$\Rightarrow \left[\frac{ax_1 + ax_2 + cy_1 + cy_2}{2} \quad \frac{bx_1 + bx_2 + dy_1 + dy_2}{2} \right]$$

$$\Rightarrow \left[\frac{ax_1 + ay_1 + ax_2 + cy_2}{2} \quad \frac{bx_1 + dy_1 + bx_2 + dy_2}{2} \right] \quad \text{— (2)}$$

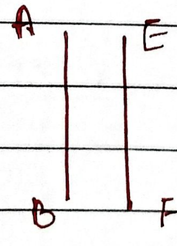
Though (1) = (2) Hence, proved.

Next statement to Prove

(2) || lines remain || even after transformation.
This also implies a parallelogram remains a parallelogram even after the transformation

$[A] = [x_1 \quad y_1]$

$[B] = [x_2 \quad y_2]$



Slope of AB = $\frac{y_2 - y_1}{x_2 - x_1}$ = m = slope

$$\begin{bmatrix} A^* \\ B^* \end{bmatrix} = \begin{bmatrix} A \\ B \end{bmatrix} [T] = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} ax_1 + cy_1 & bx_1 + dy_1 \\ ax_2 + cy_2 & bx_2 + dy_2 \end{bmatrix} \begin{matrix} \to A^* \\ \to B^* \end{matrix}$$

m^* = slope of A^*B^* = $\frac{(bx_2 + dy_2 - bx_1 - dy_1)}{(ax_2 + cy_2 - ax_1 - cy_1)}$

$$= \frac{b(x_2 - x_1) + d(y_2 - y_1)}{a(x_2 - x_1) + c(y_2 - y_1)}$$

$\Downarrow$

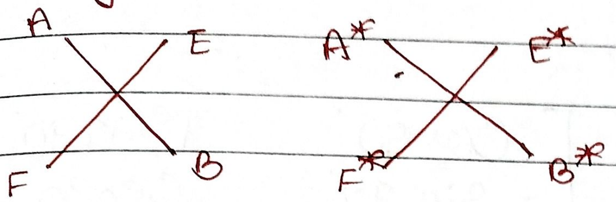
divide the num & den by $x_2 - x_1$

$$m^* = \frac{b + d\left(\frac{y_2 - y_1}{x_2 - x_1}\right)}{a + c\left(\frac{y_2 - y_1}{x_2 - x_1}\right)} = \frac{b + dm}{a + cm}$$

m^* is dependent on a, b, c, d, m (original slope of AB & EF). All these variables are same for ~~ab~~ AB & EF therefore, A^*B^* will be $\parallel$ to E^*F^*. Hence, proved.

Next

(3) Intersecting lines remains intersecting even after ~~intersecting~~ intersection.

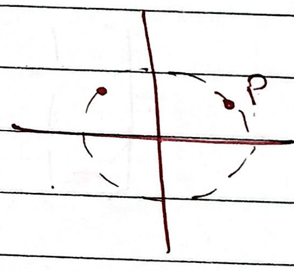


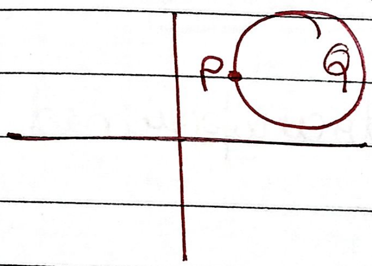
⇒ Need to be done ~~[illegible]~~ from book.

Rotation

(1) This matrix we rotate the point counter clock wise about the origin.

$$T = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$ → anti-clockwise.





Determinant is $= 1$; $\det[T] = 1$

All the transformation with ~~to~~ the determinant identically equal to 1 give the pure rotation.

(2) $[T]^T = [T]^{-1}$

The transpose of T will be equal to -1.

(3) $\cos(-\theta) = \cos\theta$
$\sin(-\theta) = -\sin\theta$

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

Clockwise

Date : / /
Page No.

Que Given a ΔABC. Rotate the Δ 90° about the origin in the ~~coun~~ anticlockwise direct[ion]

A (3, -1) B(4, 1) C (2, 1)

$$\begin{bmatrix} 3 & -1 \\ 4 & 1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} \cos 90^\circ & +\sin 90^\circ \\ -\sin 90^\circ & \cos 90^\circ \end{bmatrix}$$

$$\underset{3\times 2}{\begin{bmatrix} 3 & -1 \\ 4 & 1 \\ 2 & 1 \end{bmatrix}} \underset{2\times 3}{\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}}$$

$$\begin{bmatrix} 1 & 3 \\ -1 & 4 \\ -1 & 2 \end{bmatrix} \begin{matrix} A^* \\ B^* \\ C^* \end{matrix}$$

A^* B^* C^* is a transformed ~~ra~~ Δ.

Date: / /
Page No.

Rotation Matrix

$$\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \quad \theta = 180^\circ$$

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

$$\theta = 270 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

$$\theta = 360 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Reflection

x-axis $(y=0)$ $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$

y-axis $(x=0)$ $\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$

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

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

If the determinent of transformation matrix is identically equal to '-1' then the transformation is said to be a pure reflection.

Que 1. Given a $\triangle ABC$; $A(4,1)$, $B(5,2)$, $C(4,3)$ reflect the $\triangle$ about x-axis & about the line $x = -y$.

Date : / /
Page No.

A (4, 1)
B (5, 2)
C (4, 3)

$$X = \begin{bmatrix} 4 & 1 \\ 5 & 2 \\ 4 & 3 \end{bmatrix}$$

$3x + 2y = 0$
$8x + 4y = 0$

$$\begin{bmatrix} 3 & 8 \\ 2 & 4 \end{bmatrix}$$

$$T_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$T_2 = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$$

$$[X'] = [X][T_1][T_2]$$

$$[X'] = \begin{bmatrix} 4 & 1 \\ 5 & 2 \\ 4 & 3 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$$

$$[X'] = \begin{bmatrix} 4 & -1 \\ 5 & -2 \\ 4 & -3 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & -4 \\ 2 & -5 \\ 3 & -4 \end{bmatrix}$$

1

Q. Perform rotation of 270° anticlockwise about the origin.

A (4, 1)
B (5, 2)
C (4, 3)

$$X = \begin{bmatrix} 4 & 1 \\ 5 & 2 \\ 4 & 3 \end{bmatrix}$$

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

$$[X'] = [X][T]$$

$$= \begin{bmatrix} 4 & 1 \\ 5 & 2 \\ 4 & 3 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} -1 & 4 \\ -2 & 5 \\ -3 & 4 \end{bmatrix} \quad — \quad 2$$

Date: / /
Page No.

Reflection about x-axis followed by $x = -y$ is same as rotation about 270°.

To prove : Matrix multiplication is non-commutative.

Or
The order of the transformations perform can't be reversed.

Let's assume matrix multiplication is commutative.

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \qquad B = \begin{bmatrix} 1 & 0 \\ 2 & 2 \end{bmatrix}$$

$$[X] = [A]\,[B] \quad \text{—— (1)}$$

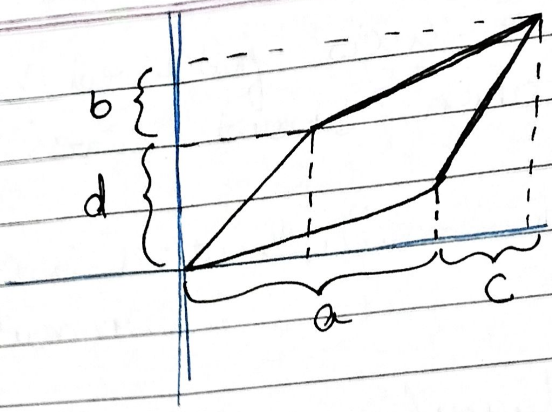
$$[Y] = [B]\,[A] \quad \text{—— (2)}$$

$$[X] = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 2 & 2 \end{bmatrix} = \begin{bmatrix} 5 & 4 \\ 11 & 8 \end{bmatrix}$$

$$[Y] = \begin{bmatrix} 1 & 0 \\ 2 & 2 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 8 & 12 \end{bmatrix}$$

$[x]$ = Unit square at the origin $= \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}$ $\qquad T = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$

$$[X]\,[T] = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 \\ c & d \\ a & b \\ a+c & b+d \end{bmatrix}$$



The origin remains invariant with respect to 2x2 transformation matrix

\# a \$ d are scaling factors
b \$ c perform ~~shearing~~ shearing.

Area Property

Area of the transformed figure (A_t) = Area of the original figure X det [T]

$\triangle ABC$ A (1,0) $T = \begin{bmatrix} 3 & 2 \\ -1 & 2 \end{bmatrix}$
B (0,1)
C (-1,0)

What will be the area of transformed Δ.

Area of the ΔABC = $\frac{1}{2} \times (x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2))$

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

$= \frac{1}{2}(1+1) = \frac{1}{2} \times 2 = 1$

Date : / /
Page No.

Area of the $\Delta ABC = 1$

$\det[T] = 8$

Area of the Transformed $\Delta = 1 \times 8 = 8$ unit sq.

$$[X'] = [X][T]$$

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

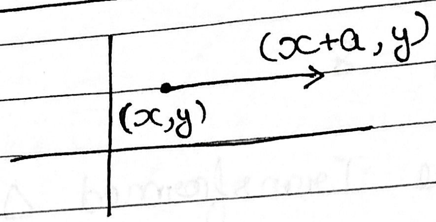
$$= \begin{bmatrix} 3 & 2 \\ -1 & +2 \\ -3 & -2 \end{bmatrix}$$

$$\text{Area of } X' = \frac{1}{2}\left(3(2+2)+(-1)(-2-2)+(-3)(2-2)\right)$$

$$= \frac{1}{2}(12+4) = \frac{16}{2} = 8 \text{ unit sq.}$$

Homogenous Coordinates

(x, y)



$$T = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$\begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 1 & 0 \\ a & 1 \end{bmatrix} = \begin{bmatrix} x+a & y \end{bmatrix} \ (\times)$$

1) We can't perform translation with 2x2 transformation matrix.

2) The origin ~~in~~ is invariant, this 2x2 transformation matrix.

1) $\begin{bmatrix} x & y & h \end{bmatrix}$

$h \downarrow$ homogenous plane $(h=1)$.

$$T = \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ m & n & 1 \end{bmatrix}$$

$m, n \rightarrow$ will perform translation.

$$\begin{bmatrix} x & y \end{bmatrix}_{1\times 2} \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}_{3\times 3} = \begin{bmatrix} x+a & y \end{bmatrix}$$

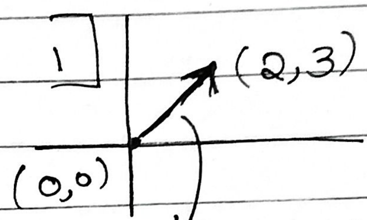
not possible.

Date : / /
Page No.

$$\underset{1\times 3}{\begin{bmatrix} x & y & 1 \end{bmatrix}} \underset{3\times 2}{\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ a & 0 \end{bmatrix}} = \begin{bmatrix} x+a & y \end{bmatrix}$$

$$\underset{1\times 3}{\begin{bmatrix} x & y & 1 \end{bmatrix}} \underset{3\times 3}{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ a & 0 & 1 \end{bmatrix}} = \begin{bmatrix} x+a & y & 1 \end{bmatrix}$$

2) $\begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 3 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 1 \end{bmatrix}$



Shifting from origin to (2,3).

formation

Que Which point from the given homogenous coordinate can be represented in 2-d.

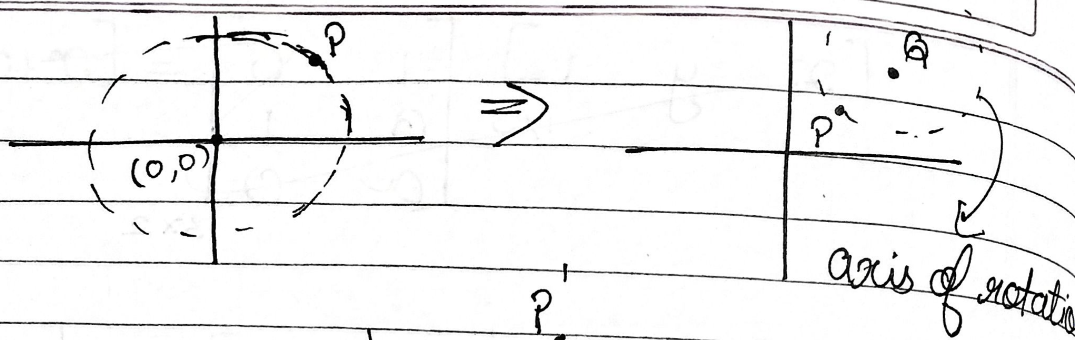
$\begin{bmatrix} 4 & 6 & 2 \end{bmatrix} \rightarrow \begin{bmatrix} 2 & 3 & 1 \end{bmatrix}$

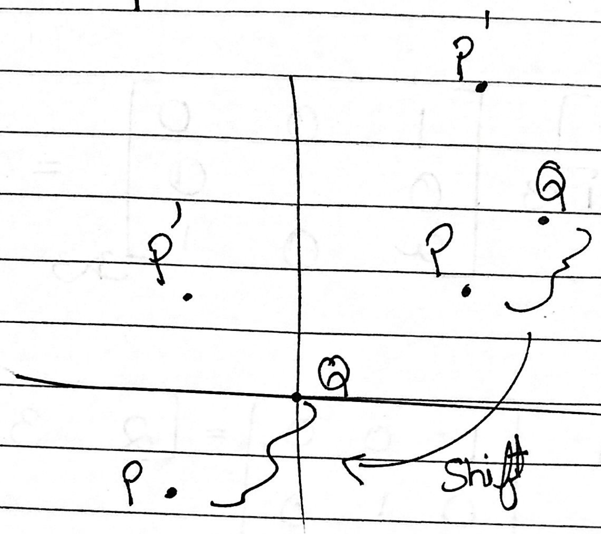
$\begin{bmatrix} 10 & 15 & 5 \end{bmatrix} \rightarrow \begin{bmatrix} 2 & 3 & 1 \end{bmatrix}$

$\begin{bmatrix} 20 & 30 & 10 \end{bmatrix} \rightarrow \begin{bmatrix} 2 & 3 & 1 \end{bmatrix}$

They represent the same point (2,3) in 2-D.

Rotation about the arbitrary point 'Q'





Step I Translate the point 'Q' (m,n) to the origin.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -m & -n & 1 \end{bmatrix}$$

Step II Perform the rotation asked in the Que.

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

Step III Perform inverse translation.

$$\underset{180^\circ}{\underline{T_1}} \quad \underset{-3,-2}{\underline{T_2}} \quad \underset{90^\circ}{\underline{T_3}} \quad \text{Rot} \quad \underset{-90^\circ}{\underline{T_3^{-1}}} \quad \underset{3,2}{\underline{T_2^{-1}}} \quad \underset{-180^\circ}{\underline{T_1^{-1}}}$$

Date : / /
Page No.

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

Que Rotate the point (10, 13) about the point (4,3) 90° counter clockwise.

$P(10, 13)$ $Q(4, 3)$

$[T_1]$ = Translate the point Q to the origin

$X = [10 \quad 13 \quad 1]$

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

$$[T_2] = \begin{bmatrix} \cos 90° & \sin 90° & 0 \\ -\sin 90° & \cos 90° & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$[T_3] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 4 & 3 & 1 \end{bmatrix}$$

$$[X'] = [X][T_1][T_2][T_3]$$

$$[X'] = [10 \quad 13 \quad 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -4 & -3 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} [T_3]$$

$$= [10 \quad 13 \quad 1] \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 3 & -4 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 4 & 3 & 1 \end{bmatrix}$$

Ch-3

Ch-15 (Only z-buffer algorithm)

$$= \begin{bmatrix} 10 & 13 & 1 \end{bmatrix}_{1\times3} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 7 & -1 & 1 \end{bmatrix}_{3\times3}$$

$$= \begin{bmatrix} -6 & 9 & 1 \end{bmatrix}_{1\times3}$$