University of Asia Pacific

Department of CSE

Course Title: Computer Graphics

Course Code: CSE 425

Assignment 01

Submitted By:

Apurba Datta
Reg Id: 20101040
Section: A

Submitted To:

Shammi Akhtar
Assistant Professor
Department of CSE(UAP)

Date of Submission: 14 / 05 / 2024

CamScanner

Composite Transformation

4.2:

a) From Prob 4.1

$$R_{30^\circ} = \begin{pmatrix} \cos 30^\circ & -\sin 30^\circ \\ \sin 30^\circ & \cos 30^\circ \end{pmatrix} = \begin{bmatrix} \sqrt{3}/2 & -1/2 \\ 1/2 & \sqrt{3}/2 \end{bmatrix}$$

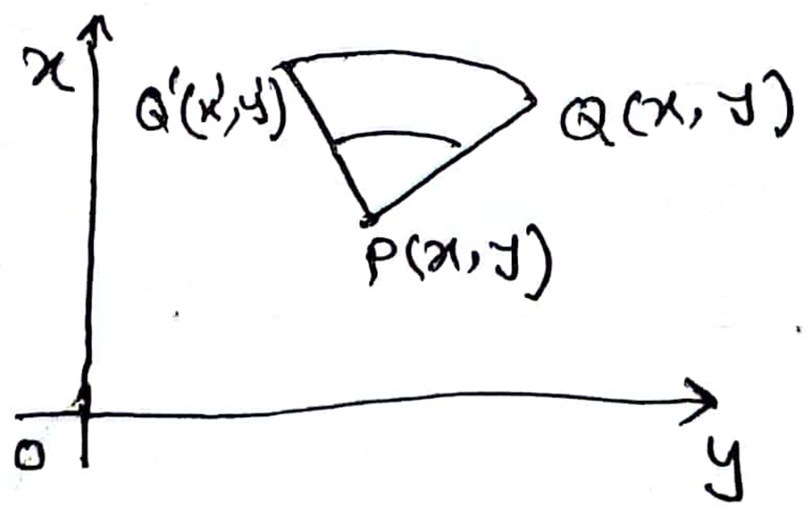
b) new coordinate of the point $P(2,-4)$

$$\begin{bmatrix} \sqrt{3}/2 & -1/2 \\ 1/2 & \sqrt{3}/2 \end{bmatrix} \begin{bmatrix} 2 \\ -4 \end{bmatrix} = \begin{bmatrix} \sqrt{3}+2 \\ 1-2\sqrt{3} \end{bmatrix}$$

4.3: describe transformation that rotate object

1. Translate center point P to the origin.
2. Perform θ degree rotation about the origin.
3. Translate P back to origin

$$R_{\theta P} = T_V \cdot R_\theta \, T_{-V}$$



4.4: General form of the matrix rotation

$$R_{\theta,P} = \begin{bmatrix} 1 & 0 & h \\ 0 & 1 & k \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -h \\ 0 & 1 & -k \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta & -\sin\theta & [-h\cos\theta + k\sin\theta + h] \\ \sin\theta & \cos\theta & [-h\sin\theta + (-k\cos\theta) + k] \\ 0 & 0 & 1 \end{bmatrix}$$

4.5: 45° rotation of triangle A(0,0), B(1,1), C(5,2)

$$ABC = \begin{bmatrix} 0 & 1 & 5 \\ 0 & 1 & 2 \\ 1 & 1 & 1 \end{bmatrix} \qquad R_{45^\circ} = \begin{bmatrix} \cos 45 & -\sin 45 & 0 \\ \sin 45 & \cos 45 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

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

$$A'B'C = R_{45^\circ}[ABC] = \begin{bmatrix} \sqrt{2}/2 & -\sqrt{2}/2 & 0 \\ \sqrt{2}/2 & \sqrt{2}/2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 5 \\ 0 & 1 & 2 \\ 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 3\sqrt{2}/2 \\ 0 & \sqrt{2} & 7\sqrt{2}/2 \\ 1 & 1 & 1 \end{bmatrix}$$

Now, $A' = (0,0)$, $B'(0, \sqrt{2})$ and $C' = (3/2\sqrt{2}, 7/2\sqrt{2})$

4.6 :

a) Transformation (origin) a units in x direction

$P(x,y)$, point (ax, y)

Now, $S_{a \cdot 1} \cdot P$, $\begin{bmatrix} a & 0 \\ a & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} ax \\ y \end{bmatrix}$

b) b unist in Y direction ; $S_{1 \cdot b} \cdot P$. So,

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

c) a units in X and b units in Y direction, $S_{ab} \cdot P$

$$\begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} ax \\ by \end{bmatrix}$$

4.7 : Scaling matrix to fixed point $P(h,k)$

$$S_{ab} \cdot P = T_v \cdot S_{ab} \cdot T_v$$

$$= \begin{bmatrix} 1 & 0 & h \\ 0 & 1 & k \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -h \\ 0 & 1 & -k \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} a & 0 & -ah+h \\ 0 & b & -bk+k \\ 0 & 0 & 1 \end{bmatrix}$$

4.8:

$$S_{2\cdot 2\cdot C} = T_V \cdot S_{2\cdot 2} \cdot T_{-V}$$

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

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

$$[ABC] = \begin{bmatrix} 0 & 1 & 5 \\ 0 & 1 & 2 \\ 1 & 1 & 1 \end{bmatrix}$$

$$[A'B'C'] = S_{22} \cdot C \cdot [ABC] = \begin{bmatrix} 2 & 0 & -5 \\ 0 & 2 & -2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 5 \\ 0 & 1 & 2 \\ 1 & 1 & 1 \end{bmatrix}$$

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

CamScanner

Schaum's Outline

6.1:

a) $T = R_{\theta_y, J} \cdot R_{\theta_x, I}$

$$= \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x & 0 \\ 0 & \sin\theta_x & \cos\theta_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta_y & \sin\theta_y \sin\theta_x & \sin\theta_y \cos\theta_x & 0 \\ 0 & \cos\theta_x & -\sin\theta_x & 0 \\ -\sin\theta_y & \cos\theta_y \sin\theta_x & \cos\theta_y \cos\theta_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

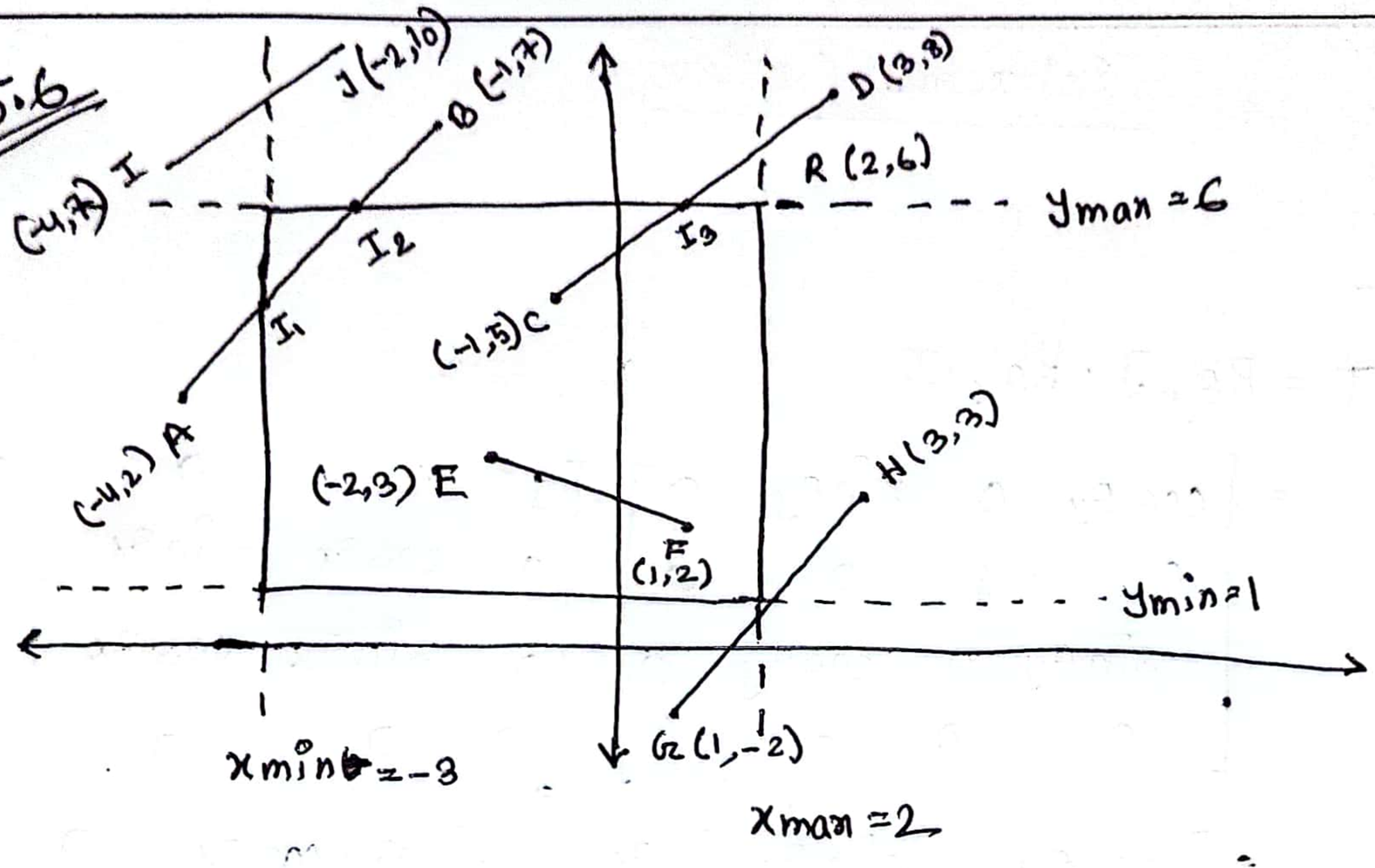
b) We multiply $R_{\theta_x} \cdot I \cdot R_{\theta_y} \cdot J$ to obtain matrix

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

CamScanner

Viewing and clipping

5.6



Bit 1 = sign(y - ymax) = sign(y-6)

Bit 2 = sign(ymin - y) = sign(1-y)

Bit 3 = sign(x - xmax) = sign(x-2)

Bit 4 = sign(xmin - x) = sign(-3-x)

$$\text{sign}(a) = \begin{cases} 1 & \text{positive } (a) \\ 0 & \text{other} \end{cases}$$

A(-4,2) → 0001

B(-1,7) → 1000

C(-1,5) → 0000

D(3,8) → 1010

E(-2,3) → 0000

F(1,2) → 0000

G(1,-2) → 0100

H(3,3) → 0010

I(-4,7) → 1001

J(-2,10) → 1000

5.8:

E or F = 0000 ⟶ visible

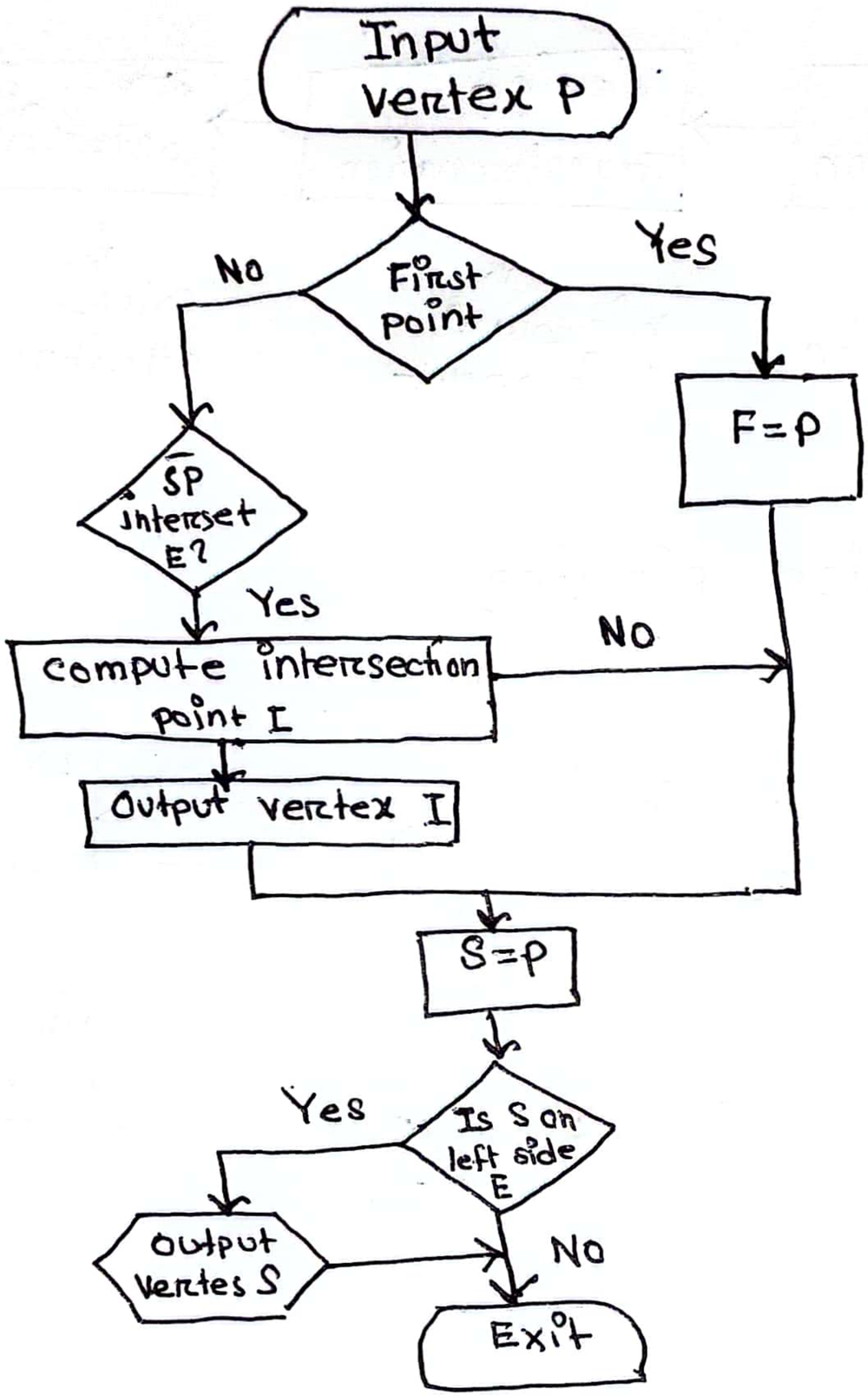
I and J = 1000 ⟶ not visible (not 0000)

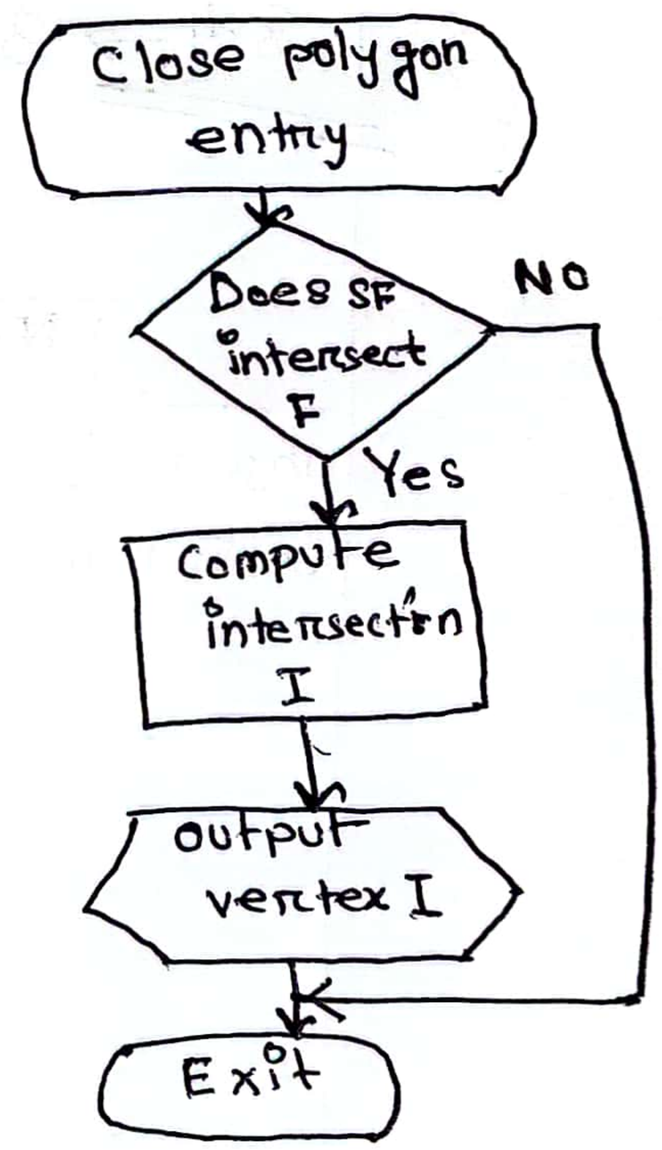
A and B = 0000 → cliping candidate

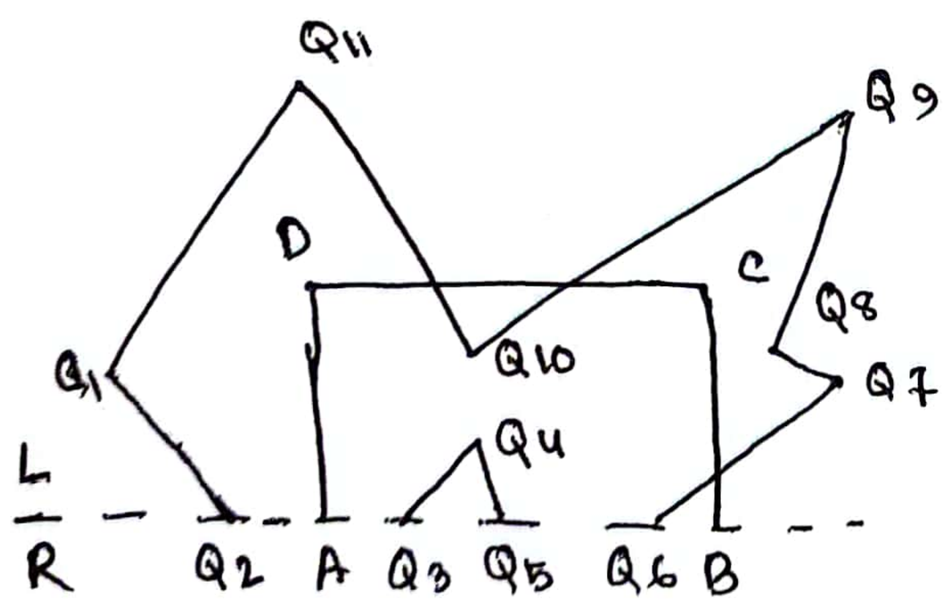
C and D = 0000 → "

G and H = 0000 → cliping candidate

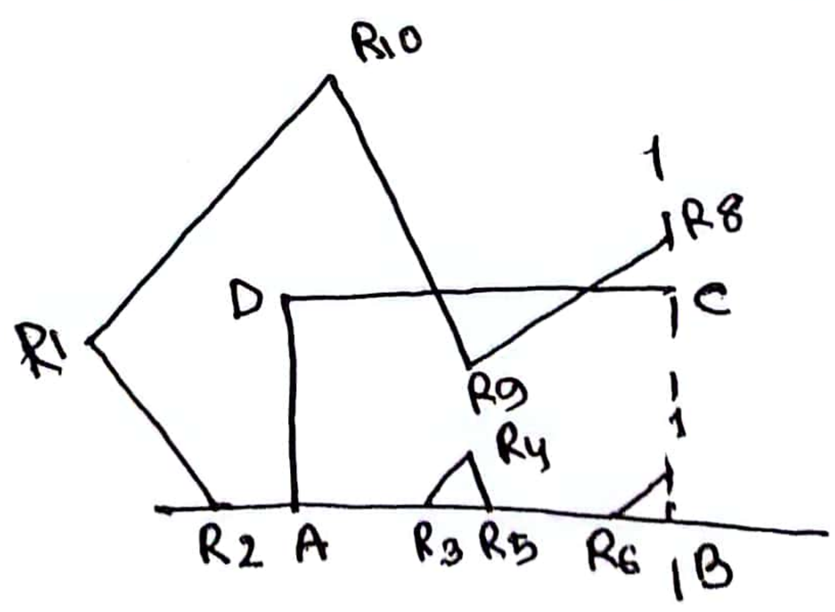
5.16 : Clip the polygon $P_1, \ldots, P_8$ using Sutherland-Hodgman algo





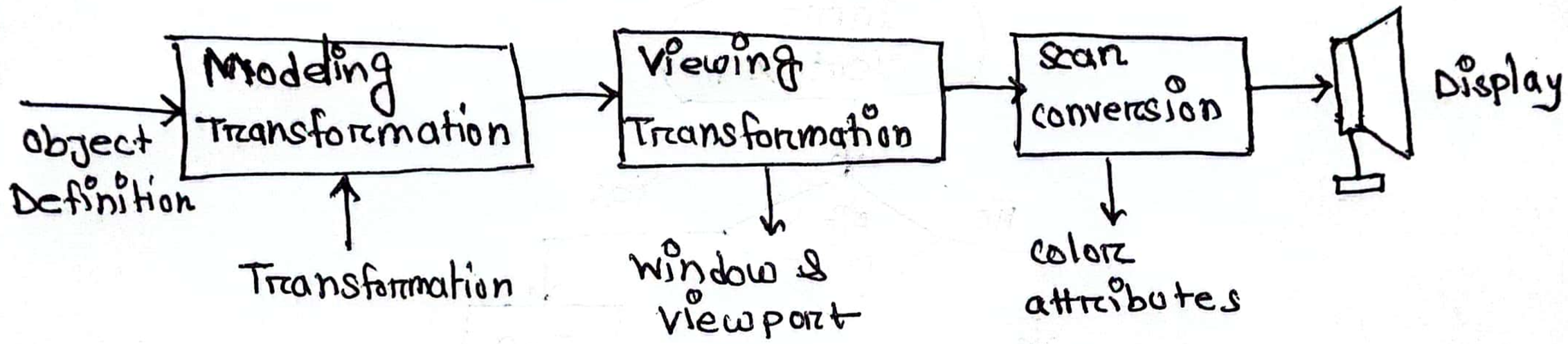


Clip againts AB

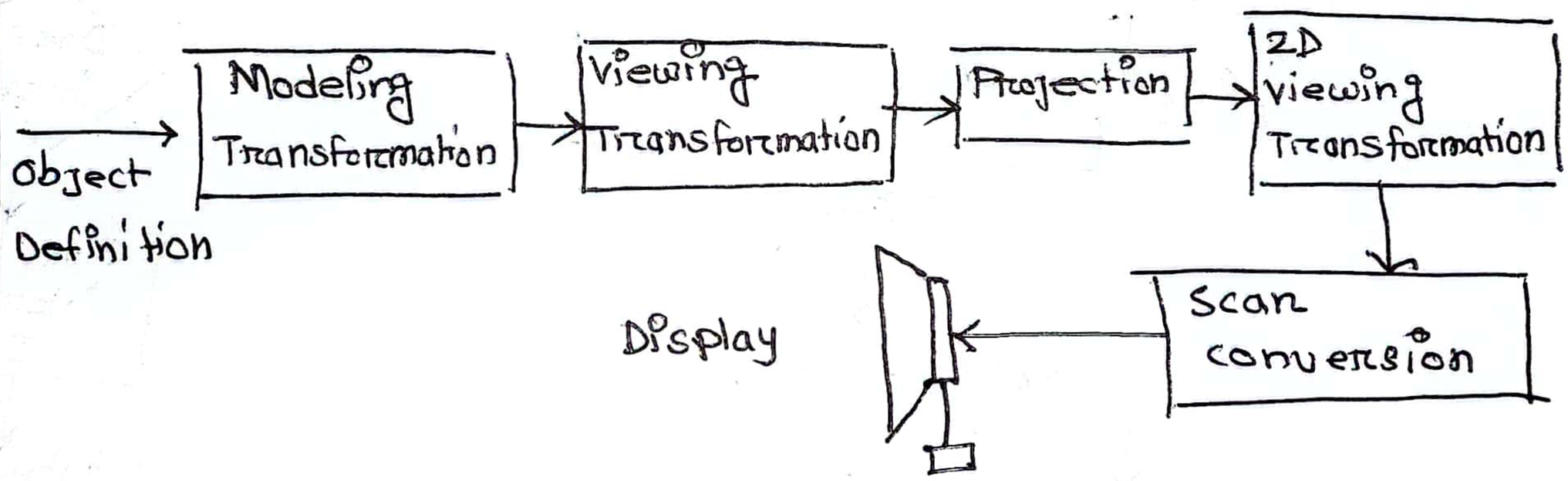


Clip against BC

5.5 → 2D graphic pipeline



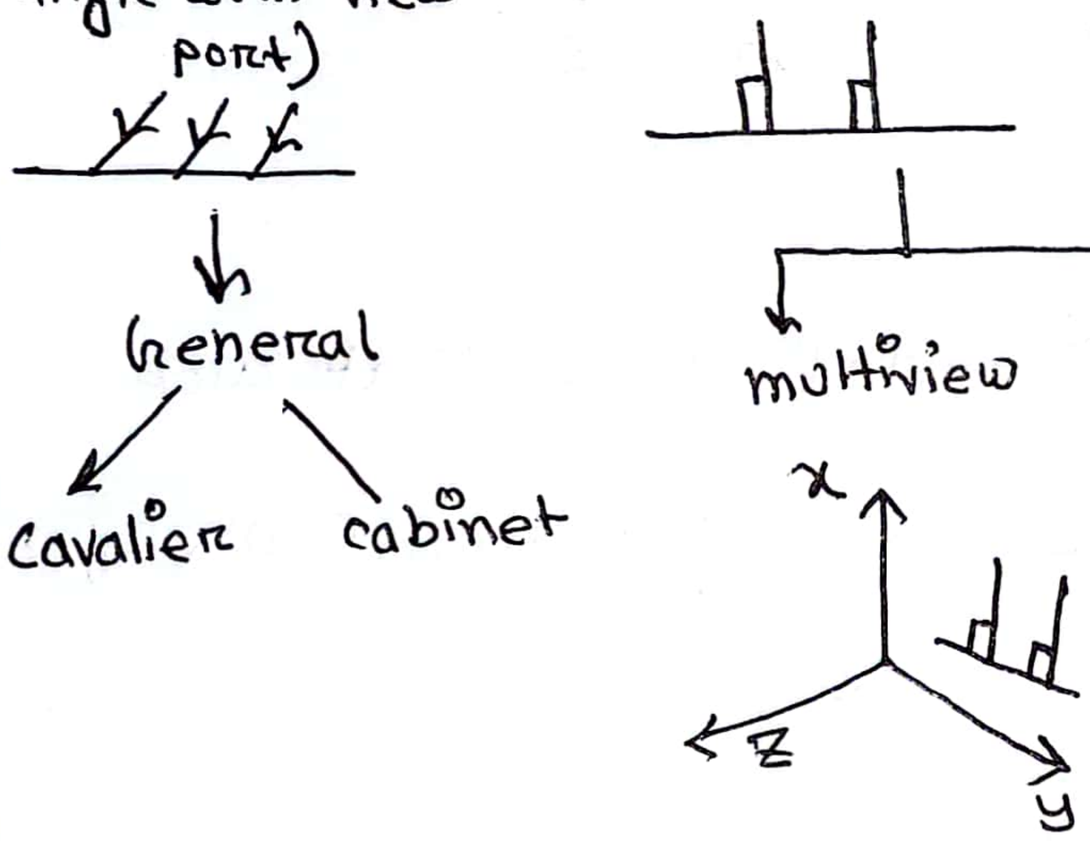
8.4 → 3D graphics pipeline



Projection

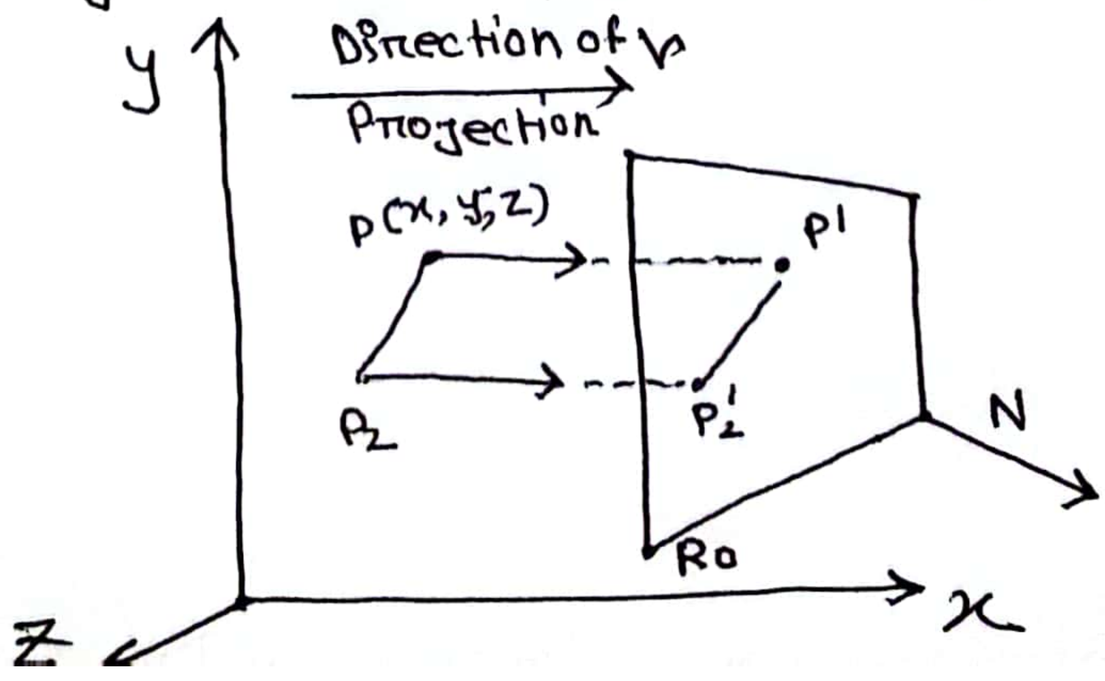
⊛ Texonomy of projection:

- **Projection**
  - **Parallel** (Projection line parallel to each other)
    - **oblique** (bend with same angle with view port)
      - **General**
        - **Cavalier**
        - **cabinet**
    - **Orthographic** (90° with view port)
      - **multiview**
      - **Axemometric** (not make 90° angle with principle axis)
  - **Perspective** (projection approching to the fix point)
    - **one point** (one principle vanishing point)
    - **two point**
    - **three point**



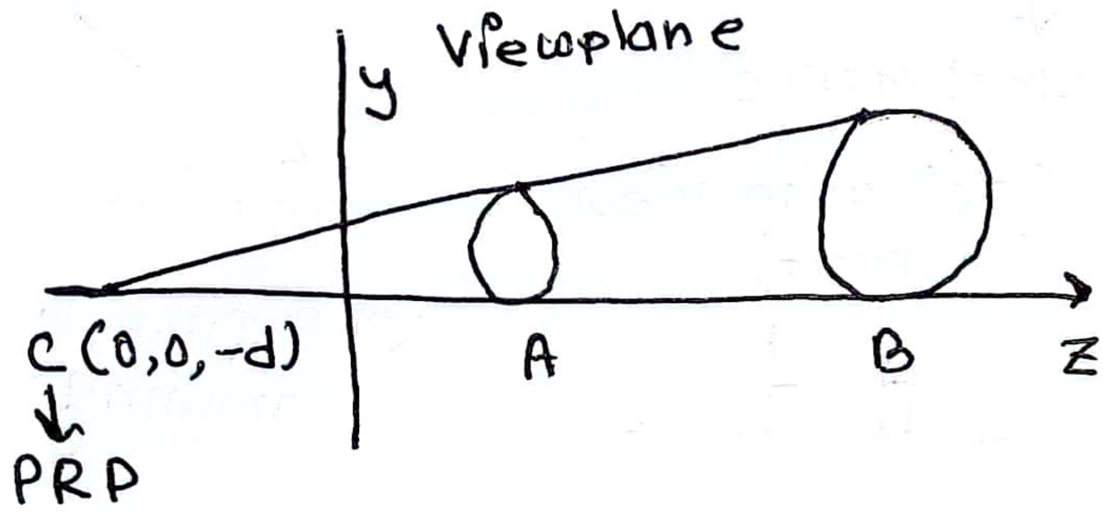
⊛ Mathematical Description of parallel projection:

Projection vector V, View plane reference point R_0, View plane normal N, The object point P is located at (x, y, z) in world ~~coro~~ coordinate. determine the image point coordinated $P(x', y', z')$



Perspective anomalies :

1. • Projection lines are drawn from the objet points towards view plane
   • PRP or center projection point where projection line converge
   • Perspective = PRP finite point



2. **Vanishing plane :** Parallel lines that are not parallel to the viewing plane, converge to vanishing point.

8.8 Three dimentional Cohen-Sutherlang algo

a) For the canonical parallel view volume, each bit is set (1) true or false(0)

Bit 1 = endpoint is above view volume = sign(y-1)

Bit2 = endpoint is below view volume = sign(-y)

Bit 3 = endpoint right to view volume = sign(x-1)

Bit 4 = endpoint left to view volume = sign(-x)

Bit 5 = endpoint behind view volume = sign(z-1)

Bit6 = endpoint infront of view volume = sign(-z)

Recall that sign(a) = 1 if a is positive, 0

b) For cannonical perspective view volume

Bit 1 = sign (y-z)

Bit2 = sign (-z-y)

Bit3 = sign (x-z)

Bit4 = sign (-z-x)

Bit5 = sign (z-1)

Bit6 = sign (z_f - z)

→ category ① visible if both region code are 000000 ② not visible if logical AND of the region code is not 000000 ③ logical AND is 000000 → clipping candidate

Chapter 10

10.1

Given points $P_1(1,2,0)$, $P_2(3,6,20)$, $P_3(2,4,6)$ and viewpoint $C(0,0,-10)$

The projection line from C to P_1 is $x=t$, $y=2t$, $z=-10+10t$

P_2 lies on line ($x=3$ when $t=3$).

compare the parameters values P_1 (at $t=1$) is in front of P_2 (at $t=3$). P_1 blocks P_2 view

P_3 does not lie on the projection line. So P_3 not block or hidden by P_1 and P_2

10.4: Why Hidden-surface algo needded?

⇒ This algorithms are crucial because they determine which parts of objects or scenes should be visible, making images look realistic by hide objects that are behind other images

10.5: Two steps are required to determine

whether any point P_1 hides another point P_2 view

① whether two points lies on the same projection line

② if ~~to~~ they do, which point is in front of the other

10.6

In parallel projection, there are no vanishing points, P(a,b,z) any points with the same x and y coordinates (a,b) project onto the same line. To determine which point is closer, you only need to compare z coordinates.

10.7

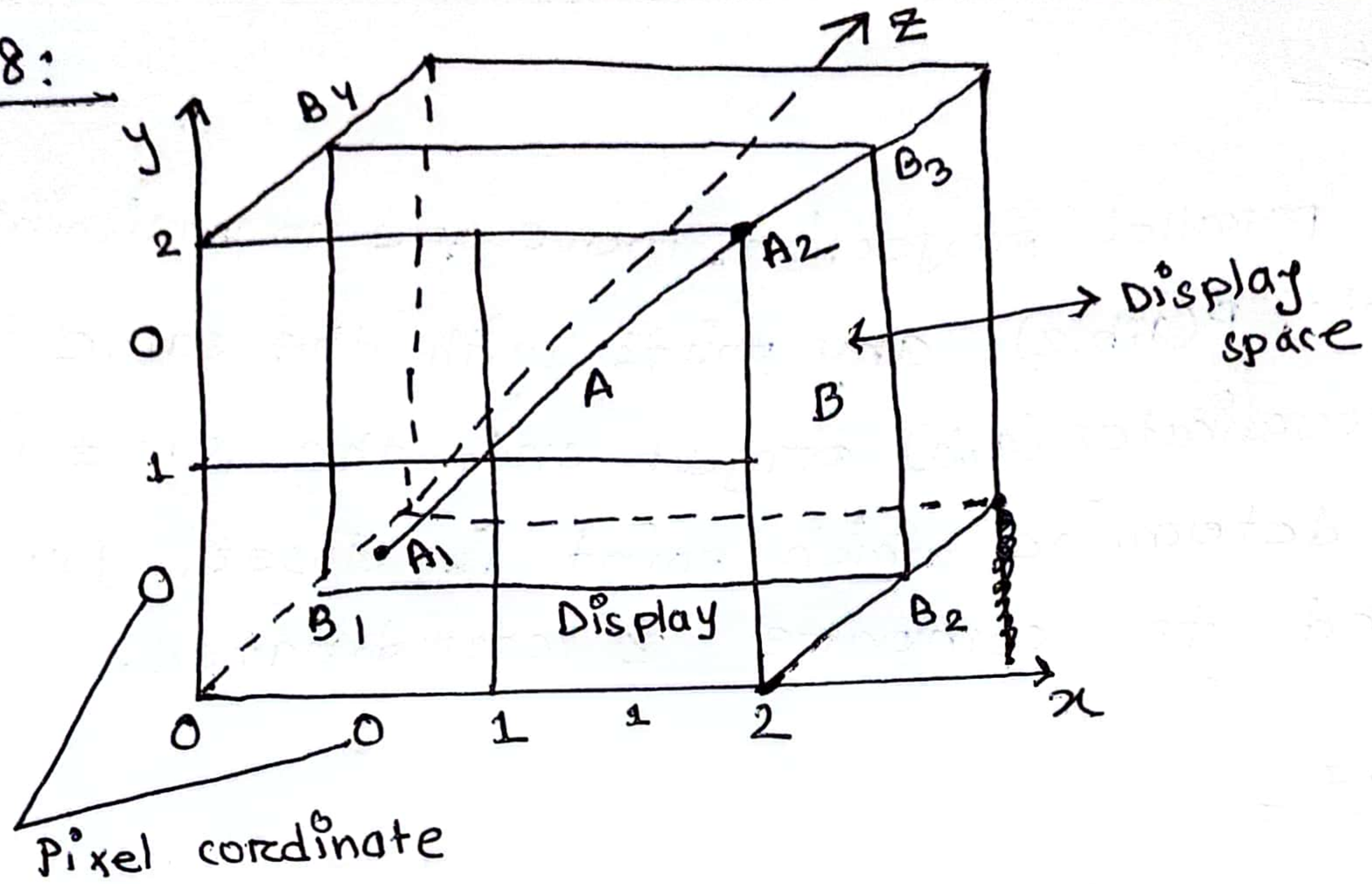
The z-buffer is like a depth map ~~de~~ behind your image. It keeps track of how far away the closest object is for each pixel (x,y) when drawing objects, the z-buffer compares the objects depth to the stored value. If the object is closer, it updates depth and replaces the color with new object. This creat a realistic images where closer objects block the view of distant ones.

⊛ z-buffer algorithum (10.2)

1. Initialize the screen and z-buffer
2. For each polygon:
   - Calculate depth at each pixel it covers.
   - If the depth is less than the z-buffer value, update it and set pixel color.

It keep track of closest objects's depth for each pixel. closer objet wins, gets shown, hides farther ones.

10.8:



Display space 2x2, region $0 \le x \le 2$, $0 \le y \le 2$ $0 \le z \le 1$, A is the ~~fore~~ front of square B, cordinate $A_1(1/2, 1/2, 0)$, $A_2(2, 2, 0)$ in B z buffer depth (1/2), $B_1(0, 0, 1/2)$, $B_2(2, 0, 1/2)$, $B_3(2, 2, 1/2)$ $B_4(0, 2, 1/2)$

1	y	a
0	a	y
	0	1

a = A color, y = background color. The image of B display

1	b	b
0	b	b
	0	1

b = B color, we apply z buffer algo object A & B Z buffer depth Z=1, background color = y

Frame buffer

1	y	y
0	y	y
	0	1

Z buffer

1	1	1
0	1	1
	0	1

③ Apply the algo to object B

a) Depth value for B pixel (0,0) is ½, Zbuff(0,0)=0
pixel (0,0) is unchanged

1	y	a
0	a	y
	0	1

Frame buffer

1	1	0
0	0	1
	0	1

Z-buffer

b) depth value of B pixel (1,0) is ½. present Z value is 1. So Zbuff set to ½ at (1,0) pixel

1	y	a
0	a	b
	0	1

Frame buffer

1	1	0
0	0	½
	0	1

Z-buffer

c) Depth value of B at pixel (0,1) is ½. present Zbuff value is 1. So pixel (0,1) is set to B

1	b	a
0	a	b
	0	1

Frame buffer

1	1	0
0	0	½
	0	1

Z-buffer

d) depth value of B at pixel (1,1) is ½. present Zbuff value is 0, so color pixel unchanged

1	b	a
0	a	b
	0	1

Frame buffer

1	½	0
0	0	½
	0	1

Z-buffer

2) Apply the algo to object A

a) Depth of A pixel (0,0) is Z = 0, Zbuff (0,0) is changed to 0, pixel (0,0) → color of A

Frame buff

1	y	y
0	a	y
	0	1

Z buffer

1	1	1
0	0	1
	0	1

b) object A not seen from pixel (1,0), Z buff → unchanged

1	y	y
0	a	y
	0	1

1	1	1
0	0	1
	0	1

c) objet A not seen from pixel (0,1) → unchanged

Frame

1	y	y
0	a	y
	0	1

Z - buff

1	1	1
0	0	1
	0	1

d) Depth of A pixel (1,1) is 0, Zbuff value 1>0 pixel has the color of A (1,1)

Frame buffer

1	y	a
0	a	y
	0	1

Z-buffer

1	1	0
0	0	1
	0	1