Amey Thakur B - 50

Terna Engineering College, Navi Mumbai

Q1. Derive matrix for Window to viewport Transformation

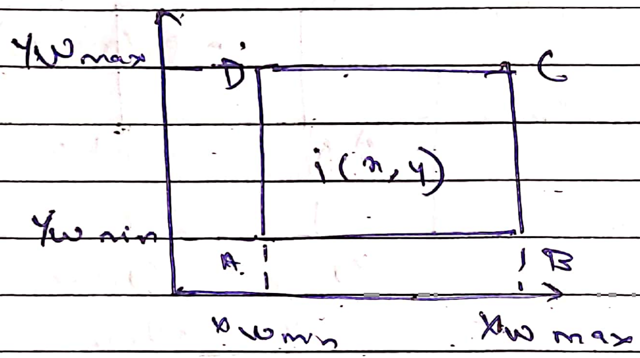
Ans:

- Translate the object so that lower left corner of the window is moved to the origin.

- Scale the object so that the window has the dimension of the viewport.

- Translate the object so that the scaled window area is positioned to viewport.

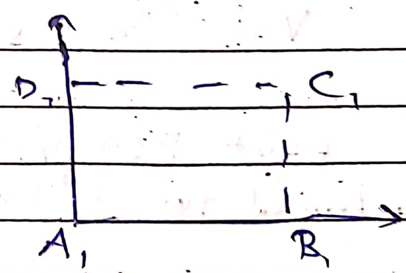
I: Initial position of window



II: Translate (xw min, yw min) to the origin.

The required translation matrix is,

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

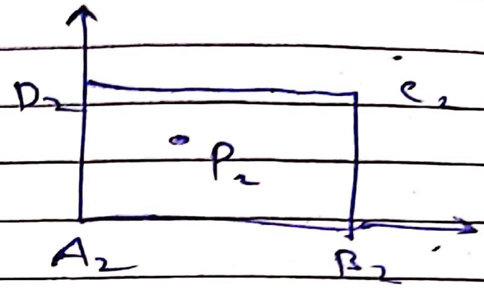


Amey Thakur B-50

III: Scale with respect to origin

Required scaling matrix is

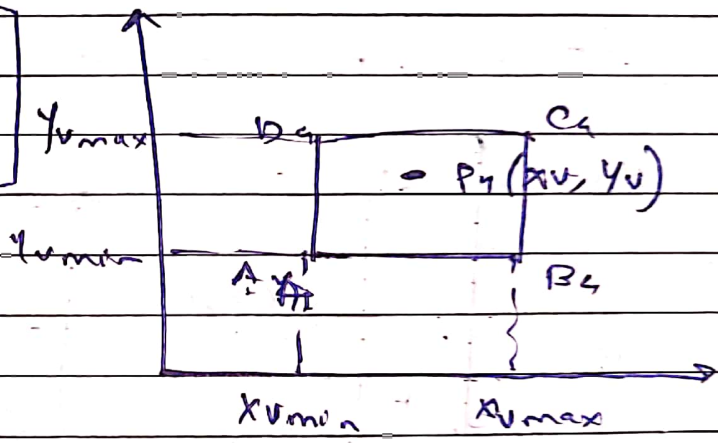
$$S = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



IV: Translate to (X_{vmin}, Y_{vmin})

The required translation matrix is

$$T_2 = \begin{bmatrix} 1 & 0 & X_{vmin} \\ 0 & 1 & Y_{vmin} \\ 0 & 0 & 1 \end{bmatrix}$$



$$\frac{X_v - X_{vmin}}{X_{vmax} - X_{vmin}}$$

The composite transformation matrix is,

$$V = T_2 \cdot S \cdot T_1$$

Any point $P(x, y)$ on the object will get transformed to $P'(x', y')$ such that

$$P' = V \cdot P$$

Amey Thakur B-50

Terna Engineering College, Navi Mumbai

Q2. Cohen Sutherland Line Clipping algorithm with example

Ans:

Cohen - Sutherland Line Clipping Algorithm.

Step I:

Accept the end points of line segment and window boundries i.e. (x_1, y_1) — (x_2, y_2)

(xw_{min}, yw_{min}) (xw_{max}, yw_{max})

Step II:

Assign a 4 bit code to each end of the line segment i.e.. $B_4\ B_3\ B_2\ B_1$

if $x_1 < xw_{min}$, then $B_1 = 1$ else $B_1 = 0$
if $x_2 > xw_{max}$, then $B_2 = 1$ else $B_2 = 0$
if $y_1 < yw_{min}$, then $B_3 = 1$ else $B_3 = 0$
if $y_2 > yw_{max}$, then $B_4 = 1$ else $B_4 = 0$

1001	1000	1010	
0001	0000	0010	TBRL
0101	0100	0110	

Step III:

a) If both the end point codes over 0000, the line is VISIBLE.
Display the line segment
STOP

Amey Thakur B-50

Terna Engineering College, Navi Mumbai

b) If logical AND of the end point code is not 0000 then line segment is not VISIBLE (INVISIBLE).
Discard the line segment.
STOP

c) If logical AND of the end point code is 0000, then the line segment is Clipping candidate.

Step IV:

Determine the intersecting boundry

If $B_1 = 1$, intersect with $x = x_{w min}$
If $B_2 = 1$, intersect with $x = x_{w max}$
If $B_3 = 1$, intersect with $y = y_{w min}$
If $B_4 = 1$, intersect with $y = y_{w max}$

Step V:

Determine the intersecting point co-ordinates (x', y').

The equations are -

$$x' = x_1 + \frac{1}{m}(y' - y_1) \quad \text{where } m = \frac{y_2 - y_1}{x_2 - x_1}$$

and here, $y' = y_{w min}$ or
$y' = y_{w max}$

$$y' = y_1 + m(x' - x_1)$$

here, $x' = x_{w min}$ or
$x' = x_{w max}$.

Amey Thakur. B-50

Step VI:

Goto step II until you get totally visible line segment.

Step VII:

STOP.

Example:

Find the clipping co-ordinate of line joining A (−1, 5) and B (3, 8)

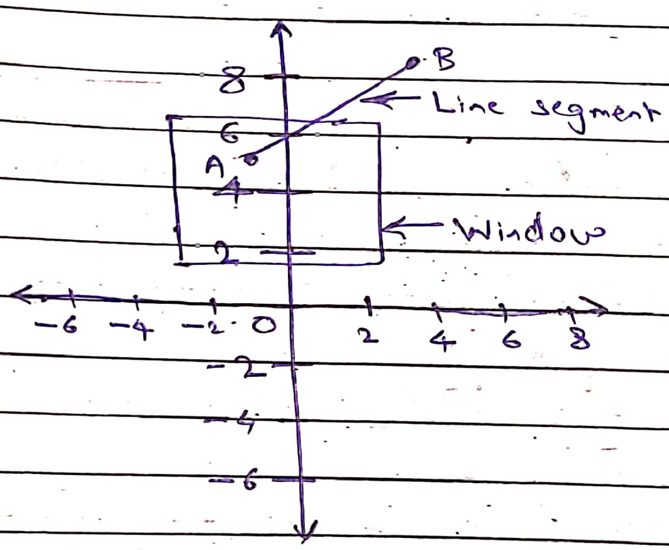
A (−2, 3) B (7, 2)
W (0, 0) (5, 5)

→ **Step I:**

A ($x_1 = -1$, $y_1 = 5$) B ($x_2 = 3$, $y_2 = 8$)

$x_{wmin} = -3$ $x_{wmax} = 2$

$y_{wmin} = 1$ $y_{wmax} = 6$



TBRL

1001	1000	1010 B
0001	0000 A	0010
0101	0100	0110

Step II:

For A (−1, 5) For B (3, 8)

Code = 0000 Code = 1010

Point A is visible Point B is not visible

0000
⊙ 1010
0000

∴ It is a clipping candidate

Amey Thakur B-50

Terna Engineering College, Navi Mumbai

Step III :

Intersecting boundary,

$B_2 = 1$, intersecting with $x = x_{w\,max}$

$B_4 = 1$, intersecting with $y = y_{w\,max}$

Select $x = X_{w\,max}$ as a clipping boundary.

Step IV :

Intersecting point co-ordinates

$x' = X_{w\,max} = 2$

$y' = y_1 + m(x' - x_1)$

$m = \dfrac{y_2 - y_1}{x_2 - x_1} = \dfrac{8-5}{3-(-1)} = \dfrac{3}{4}$

$y' = 5 + \dfrac{3}{4}(2-(-1))$

$= \dfrac{29}{4}$

$\therefore I_1 = \left(2, \dfrac{29}{4}\right)$ AI_1 is clipped line segment

Step V:

Code for $A(-1, 5) = 0000$

code for $I_1\left(2, \dfrac{29}{4}\right) = 1000$

$\therefore$ Point A is visible

Point I_1 is invisible

A AND I_1 = 0000 $\therefore$ AI is clipping candidate.

1000

0000

Amey Thakur B-50

Terna Engineering College, Navi Mumbai

Step VI :

Determine intersecting boundry B4 = 1,
intersect with $y = y_{w} max$.

Step VII :

Determine intersecting point co-ordinates.

Amey Thakur B-50

Terna Engineering College, Navi Mumbai

Q3. Explain 3D Translation, Rotation, Scaling and Reflection with matrix representation

Ans:

① Translation:

A point $P(x, y, z)$ is translated to $P'(x', y', z')$ by translation vector such that

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

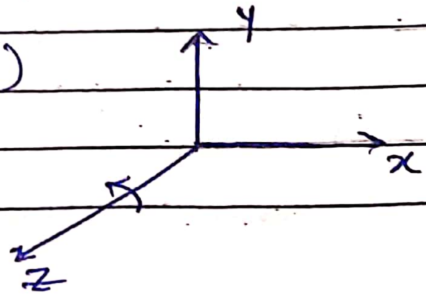
The homogeneous 3D co-ordinate transformation matrix T is given by

$$T = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T^{-1} = \begin{bmatrix} 1 & 0 & 0 & -t_x \\ 0 & 1 & 0 & -t_y \\ 0 & 0 & 1 & -t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

② Rotation: ⓘ Rotation about Z axis

Rotation about Z axis by an angle θ in anti-clockwise direction will shift point $P(x, y, z)$ to $P'(x', y', z')$



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

Amey Thakur B-50

Terna Engineering College, Navi Mumbai

The rotation matrix is

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

Inverse rotation i.e. rotation in clockwise direction is obtained by replacing θ by $-\theta$.

The matrix for reverse rotation is

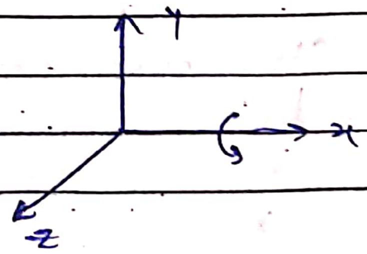
$$R_z^{-1} = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Transformation equation for rotation about x-axis and y-axis can be obtained with cyclic permutation x, y and z.

Replacing $x \rightarrow y \rightarrow z \rightarrow x$

(ii) Rotation about x-axis

Rotation about x axis by an angle θ in anticlockwise direction will shift point $P(x, y, z)$ to $P'(x', y', z')$.



All x-coordinates will remain unchanged

New co-ordinate,

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

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

$x' = x$

Amey Thakur B-50

Terna Engineering College, Navi Mumbai

Matrix representation is

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

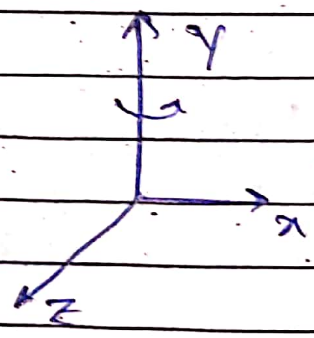
Inverse rotation about x-axis. i.e. rotation in clockwise direction is obtained by replacing θ by $-\theta$.

Matrix for inverse rotation is

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

(iii) Rotation about Y-axis

Rotation about Y axis by an angle θ in anticlockwise direction will shift point $p(x, y, z)$ to $p'(x', y', z')$



All z coordinates will remain unchanged

The new coordinate will then be

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

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

$y' = y$

Amey Thakur B-50

Matrix representation for y-axis rotation is,

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

Inverse rotation i.e. rotation in clockwise direction

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

③ Scaling :

A point $P(x, y, z)$ is scaled to $P'(x', y', z')$

such that,

$$x' = x \cdot s_x$$
$$y' = y \cdot s_y$$
$$z' = z \cdot s_z$$

The homogeneous 3D coordinate transformation of matrix S is given by

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

Amey Thakur B-50

Terna Engineering College, Navi Mumbai

(4) Reflection:

3D reflection about X-axis, Y-axis and Z axis is given by

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

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

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

Amey Thakur B-50

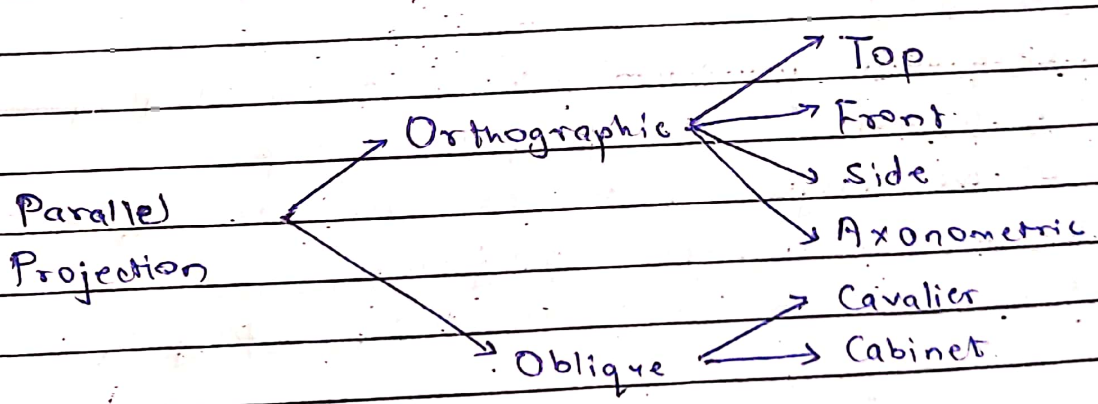
Terna Engineering College, Navi Mumbai

Q4. Illustrate perspective and parallel projection with different types.

Ans:

Parallel Projection

- Parallel projection discards z coordinate and parallel lines from each vertex on the object are extended until they intersect the view plane.
- In parallel projection, we specify a direction of projection instead of center of projection.
- In parallel projection, the distance from the center of projection to project plane is infinite.
- In this type of projection, we connect the projected vertices by line segments which corresponds to connections on the original object.
- Parallel projections ~~are~~ are less realistic, but they are good for exact measurements.
- In this type of projections, parallel lines remain parallel and angles are not preserved.
- Various types of parallel projections are shown →

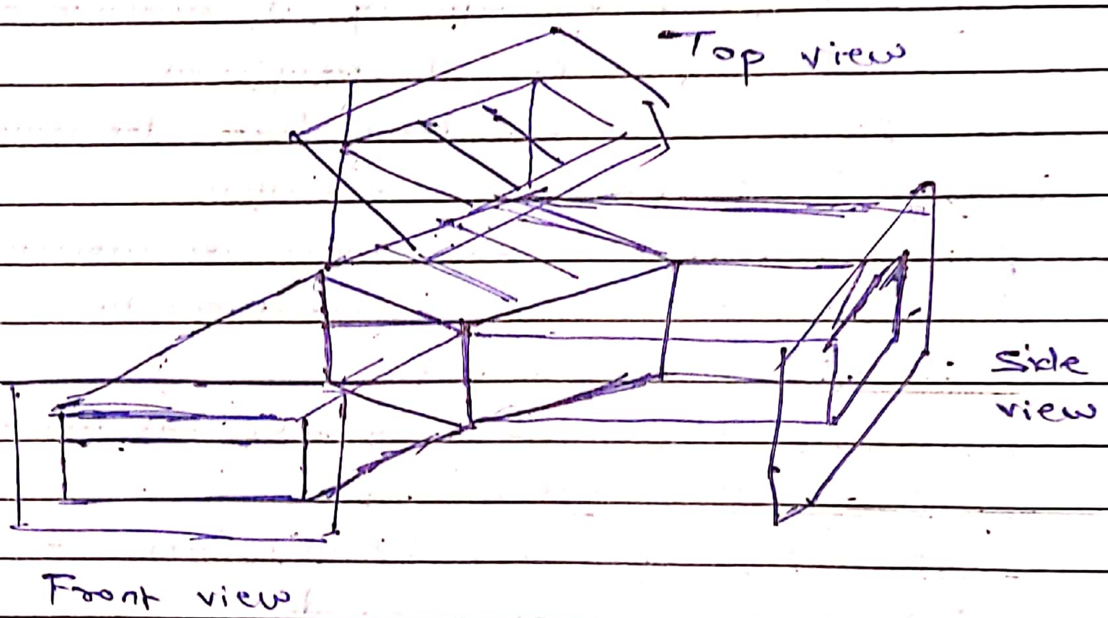


Amey Thakur B-50

Terna Engineering College, Navi Mumbai

(a) Orthographic Projection

- In orthographic projection the direction of projection is normal to the projection of the plane.
- There are 3 types of orthographic projections -
  - → Front Projection
  - → Top Projection
  - → Side Projection



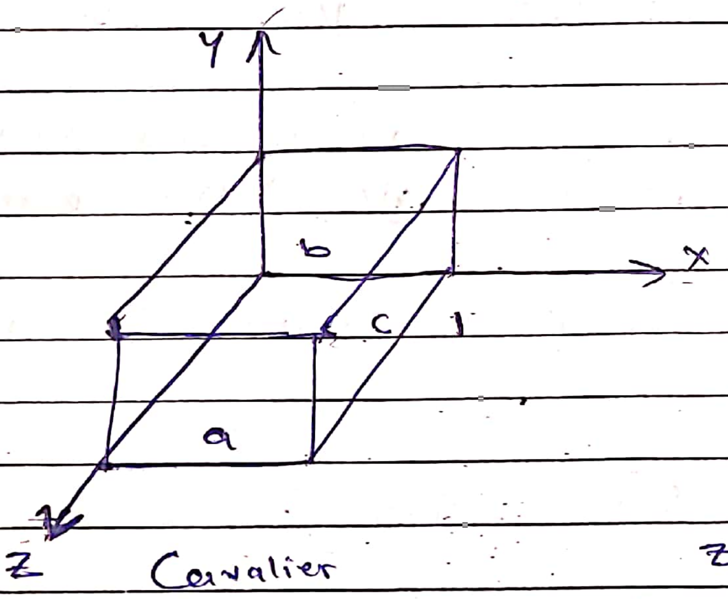
(b) Oblique Projection

- In orthographic projection, the direction of projection is not normal to the projection of plane.
- In oblique projection, we can view the object better than orthographic projection.
- There are 2 cases of oblique projection
  - → Cavalier
  - → Cabinet
- The cavalier projection makes 45° angle with the projection plane.

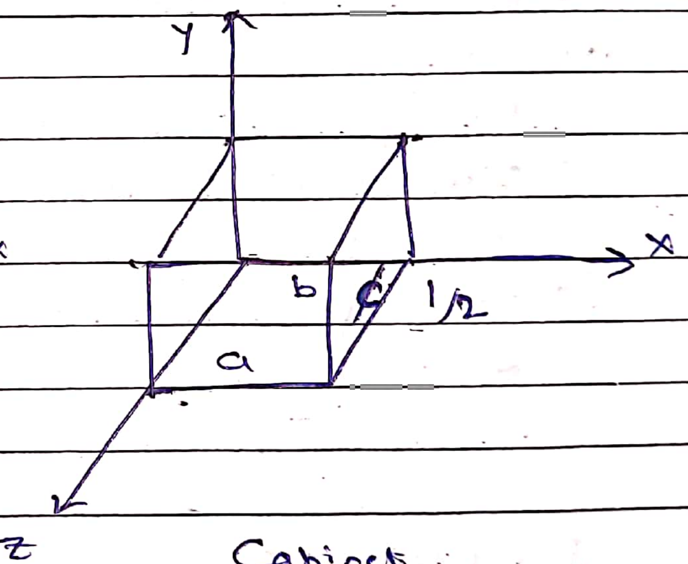
Amey . Thakur B-50

Terna Engineering College, Navi Mumbai

- The projection of a line perpendicular to the view plane has the same length as the line itself in Cavalier projection.
- In a Cavalier projection the forshortening factors for all 3 principle directions are equal.
- The cabinet projection makes 63.4° angle with the projection plane.
- In cabinet projection lines perpendicular to the viewing surface are projected at 1/2 their actual length.
- Both the projections are shown —



Cavalier Projection



Cabinet Projection

Amey Thakur B - 50

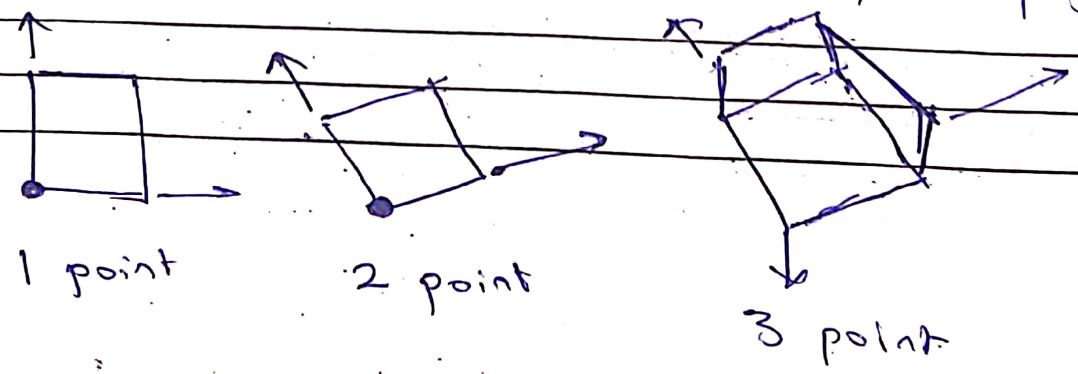
Terna Engineering College, Navi Mumbai

Perspective Projection

- In perspective projection, the distance from the center of projection to project plane is finite and the size of the object varies inversely with distance which looks more realistic.
- The distance and angles are not preserved and parallel lines do not remain parallel.
- Instead, they all converge at a single point called center of projection or projection reference point.
- There are 3 types of perspective projection which are -
  - → One point perspective projection is simple to draw.
  - → Two point perspective projection gives better impression of depth
  - → Three point perspective projection is most difficult to draw.

Perspective Projection → One point
Perspective Projection → Two point
Perspective Projection → Three point

- Figures shows all 3 types of perspective projection



Amey Thakur B-50

Q.5. Elaborate Back surface detection method and depth buffer method for visible surface detection in detail.

Ans:

Back Surface Detection

- Object surfaces that are oriented away from the viewer are called back surfaces or faces. The back surfaces of a cube are completely blocked by the cube itself and hidden from view.
- Therefore we can identify and remove these back surfaces.
- Eq^n. of plane is $Ax + By + Cz + D = 0$
- In object space method, the identification of back surfaces is based on above eq^n.

From the above eq^n, we can say, if a point (x, y, z) satisfies the eq^n then the point (x, y, z) is lying on the plane.

But,

if $Ax + By + Cz + D < 0$ — (1)

It means (x, y, z) lies on negative side.

and

if $Ax + By + Cz + D > 0$ — (2)

It means (x, y, z) lies on positive side.

- If we consider any point (x, y, z) as viewing point then any plane which satisfy the eq^n (1) must be a back surface.

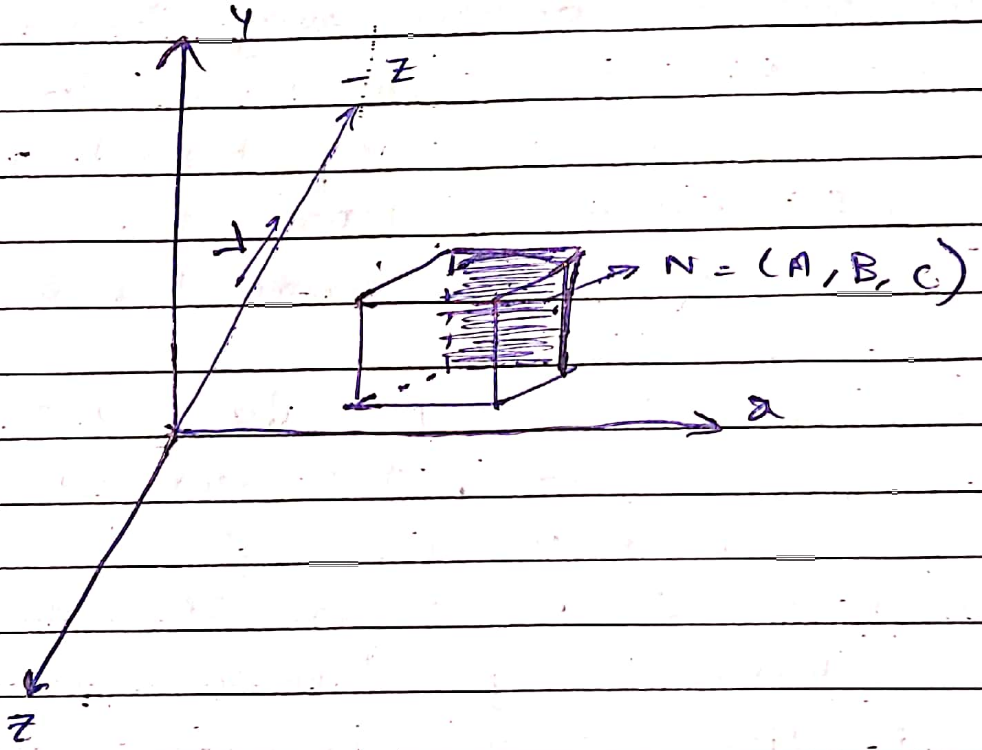
After finalizing the back face we have to remove it from the further visibility.

Amey Thakur B-50

Terna Engineering College, Navi Mumbai

Let N = (A, B, C) be the normal vector.

In right handed system with viewing direction along the negative Z axis, the polygon is back face if C < 0



- Also, we cannot see any face whose normal has z component C = 0.

Thus we can say any polygon is classified as a back face when its normal vector is negative or equal to zero. i.e $C \leq 0$.

Amey Thakur B-50

Terna Engineering College, Navi Mumbai

Depth Buffer (Z - Buffer)

- Another way to handle hidden lines and surfaces is Z buffers. It is also called as depth buffer algorithm. Here we are sorting the polygons according to their position in space. And then in frame buffer itself. We are sorting polygons which are closer to viewer. We know that frame buffer is used to store the images which we want to display on monitor. Here for visibility test we are making use of Z buffer along with frame buffer.

- Thus Z buffer is a large array to hold all the pixels of display. Z buffer is somewhat similar to frame buffer. In frame buffer we are having arrays to store x and y co-ordinates of an image; Similarly, Z buffer contains Z co-ordinate of pixels which we want to display. It helps to sort the polygons by comparing their Z position. In simple terms, it keeps track of nearest Z coordinate of those points which are seen from pixel (x, y)

- When there is nothing to display on monitor i.e frame buffer is empty, at that time we have to initialize Z buffer elements to a very large negative value. A large negative value of Z axis represents a point beyond which there is nothing i.e Setting background color.

$\therefore$ Z buffer $(x, y) = Z_{initial}$

Amey Thakur B-50

Terna Engineering College, Navi Mumbai

- Polygon will be entered one by one into frame buffer by the display file interpreter. During this process, for each pixel (x, y) that lies inside polygon. Then we have to calculate $Z(x, y)$ for that pixel (x, y). Then we have to compare Z position of the polygon point with $Zbuffer(x, y)$ value and decide whether the new surface is in front or behind the current contents of frame buffer.
- If the new surface has Z value greater than $Zbuffer$ value, then it lies in front. So we have to modify the contents of $Zbuffer(x, y)$ by new z value and set the pixel value at (x, y) to the color of the polygon at (x, y)

i.e. (Z (x, y) > Zbuffer (x, y))

```
{
    Zbuffer (x, y) = Z (x, y)
    put pixel (x, y, polygon-color)
}
```

- If the Z value of new surface is smaller than $Zbuffer(x, y)$ then it lies behind some polygon which was previously entered. So the new surface should be hidden and should not be displayed. The frame buffer and $Zbuffer$ should not be modified here. Here the comparison should be carried out by using pixel by pixel method.

Amey Thakur B-50

Terna Engineering College, Navi Mumbai

Advantages

1. Easy to implement
2. As Z buffer algorithm process one object at a time, total no. of objects can be large.
3. It does not require any additional technique

Disadvantages

1. It requires lot of memory as we are storing each pixel's Z value
2. Time consuming process. as we are comparing each and every pixel

Amey Thakur B-50

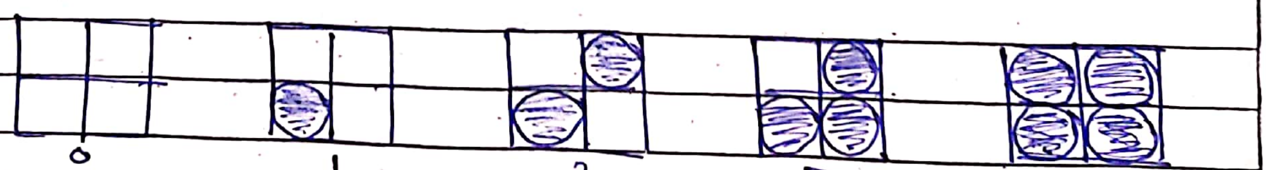
Terna Engineering College, Navi Mumbai

Q.6. Discuss Halftone and Dithering technique

Ans:

Half toning

- Many display and hardcopy devices are bi-level
- They can only produce two intensity levels.
- In such displays or hardcopy devices we can create an apparent increase in the number of available intensity value.
- When we view a very small area from a sufficient large viewing distance, our eyes average find details within the small area and record only the overall intensity of the area
- The phenomenon of apparent increase in the no. of available intensities by considering combine intensity of multiple pixels is known as Half toning
- The half toning is commonly used in printing black and white photographs in newspapers, magazines and books
- The pictures produced by half toning process are called half tones
- In computer graphics, half tone reproductions are approximated using rectangular pixel regions say 22/33 pixels.
- These regions are called halftone pattern or pixel pattern
- Figure shows half tone pattern to create no. of intensity levels



$0 \leq I \leq 0.2$ $0.2 \leq I < 0.4$ $0.4 \leq I < 0.6$ $0.6 \leq I < 0.8$ $0.8 \leq I < 1.0$

2×2 pixel patterns for creating 5 intensity levels

Amey Thakur B-50

Terna Engineering College, Navi Mumbai

Dithering Techniques.

- Dithering refers to technique for approximating halftones without reducing resolution, as pixel grid patterns do.
- The term dithering is also applied to half tone approximation method using pixel grid and something it is used to refer to color halftone approximations only
- Random values added to pixel intensities to break up contours are often referred as dither noise.
- No. of methods is used to generate intensity variations
- Ordered dither methods generate intensity variations with a one to one mapping of points in a scene to the display pixel.
- To obtain n^2 intensity levels, it is necessary to set up an $n * n$ dither matrix D_n whose element are distinct positive integers in the range of 0 to $n^2 - 1$.
- For example, it is possible to generate 4 intensity levels with

$D_2 = \begin{matrix} 3 & 1 \\ 0 & 2 \end{matrix}$ and it is possible to generate nine intensity levels with

$$D_3 = \begin{matrix} 7 & 2 & 6 \\ 4 & 0 & 1 \\ 3 & 8 & 5 \end{matrix}$$

Amey Thakur B-50

Terna Engineering College, Navi Mumbai

- The matrix element for D_2 & D_3 are in the same order as the pixel mask for setting up 22 & 33 pixel grid respectively.
- For bi-level system we have to determine display intensity values by comparing input intensities to the matrix elements
- Each input intensity is first scaled to the range $0 < I < n^2$.
- If the intensity I is to be applied to screen position (x, y) we have to calculate number for the either matrix as

$$i = (x \bmod n) + 1$$

$$j = (y \bmod n) + 1$$

- If $I > D_n(i, j)$ the pixel at position (x, y) is turned on; otherwise the pixel is not turned on.
- Typically no of intensity levels is taken to be a multiple of 2.
- High order dither matrices can be obtained from lower order matrices with the recurrence relation.

$$D_n = \begin{matrix} 4D_{n/2} + D_2(1,1)\,U_{n/2} & 4D_{n/2} + D_2(1,2)\,U_{n/2} \\ 4D_{n/2} + D_2(2,1)\,U_{n/2} & 4D_{n/2} + D_2(2,2)\,U_{n/2} \end{matrix}$$

- Another method for mapping a picture with m.n points to a display area with m.n pixels is error diffusion
- Here, the error between an input intensity value and the displayed pixel intensity at a given position is dispersed, or diffused to pixel position to the right and below the current pixel position.