CSo -351

Assignment -4

Name - Jatin Garg

Class - CSE

Roll - 12075037

Ans 1. Projection from 3D to 2D is defined by Straight projection rays (Projectors) emanating from the 'center of projection', passing through each point of the object and intersecting the Projection plane to form a projection.

→ Two types of projections:

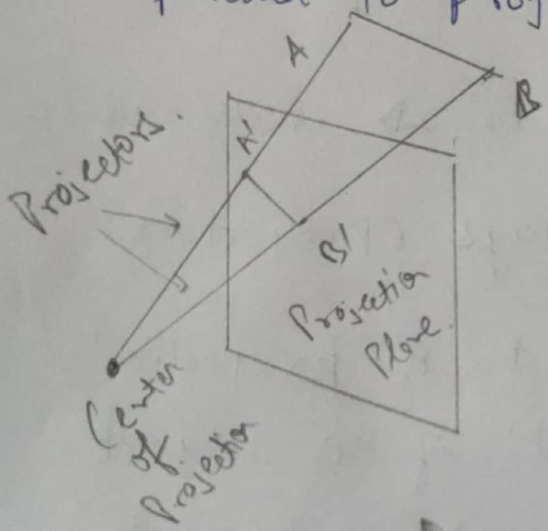
① Perspective : If the distance to Center of Projection is finite, then it is Perspective projection

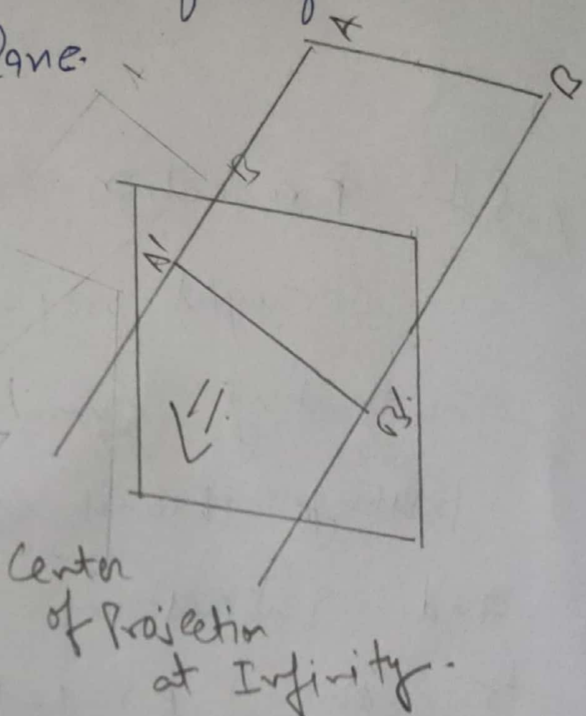
→ Visual effect is similar to human visual system.

→ Size of object varies inversely with distance from Center of projection.

→ Parallel lines do not in general project to parallel lines.

→ Angles only remain intact for faces parallel to projection plane.





(2) Parallel → If distance to center of projection is Infinite then it is Parallel projection.

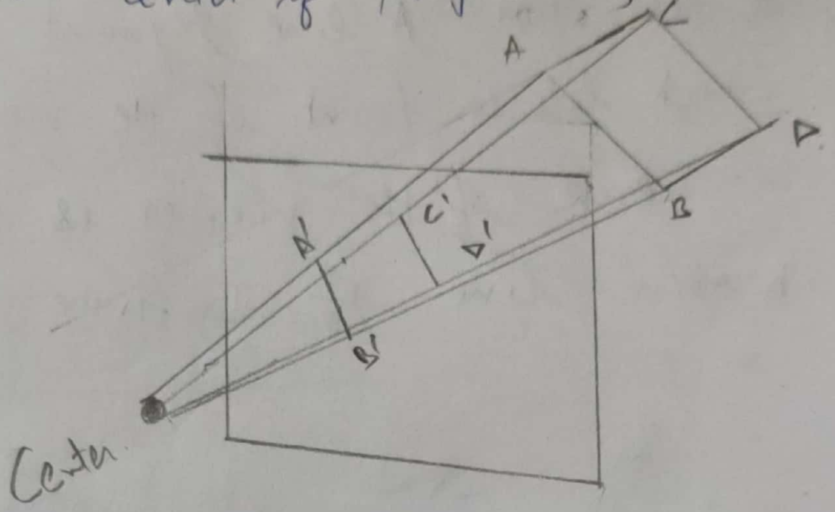
→ less realistic view because of no foreshortening.

→ However, parallel lines remains parallel.

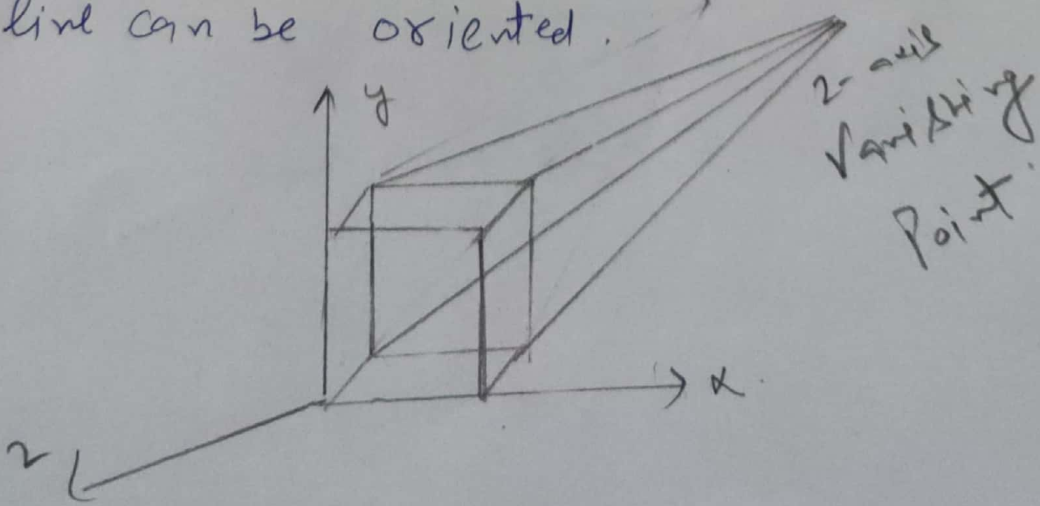
→ Angles only Remain Intact for faces Parallel to Projection Plane.

☆ Perspective Projection - anomalies.

① Perspective foreshortening: The farther an object is from center of projection, smaller it appears.

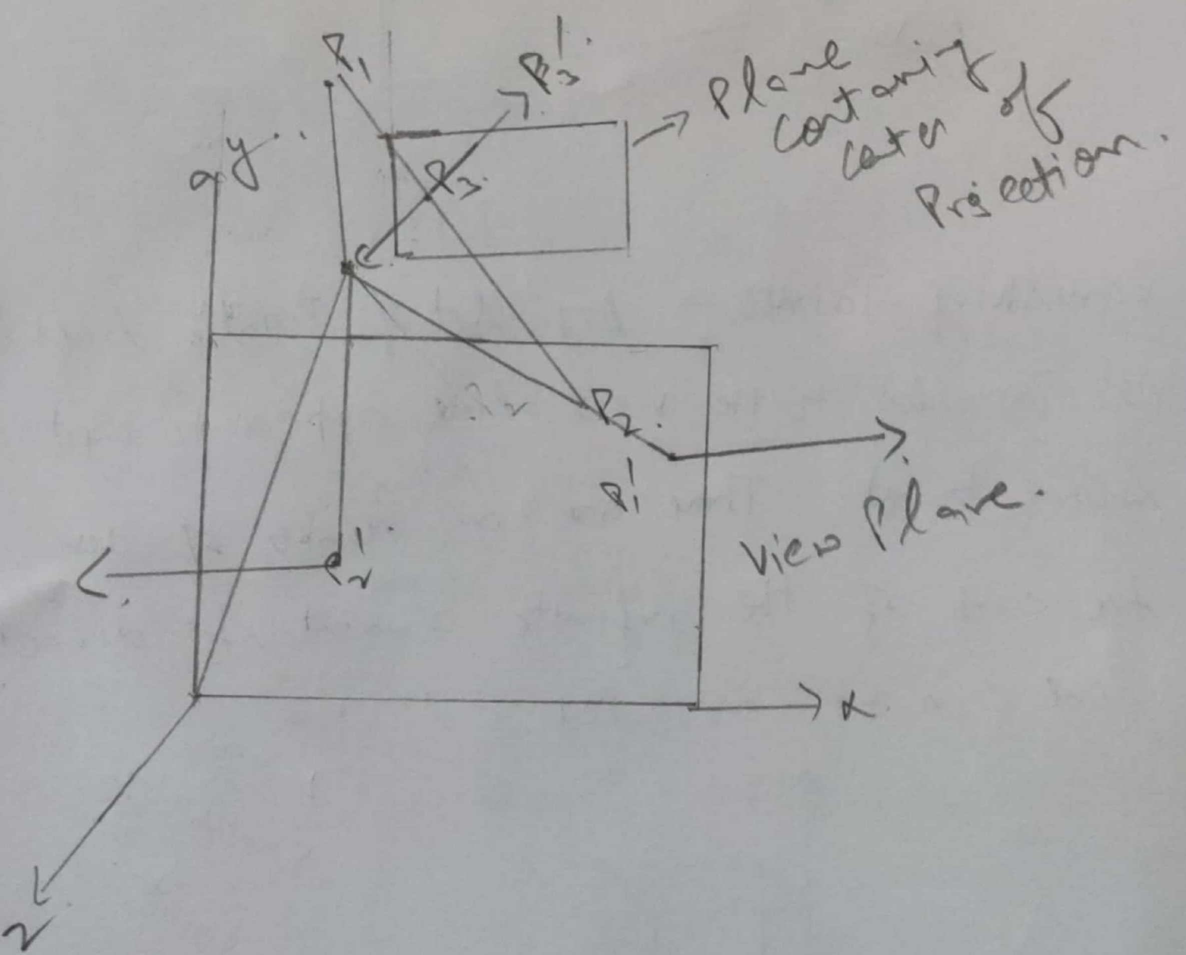


② Vanishing Points → Any set of parallel lines not parallel to the view plane appear to meet at some point. There are ∞ number of these, 1 for each of the infinite amount of directions line can be oriented.

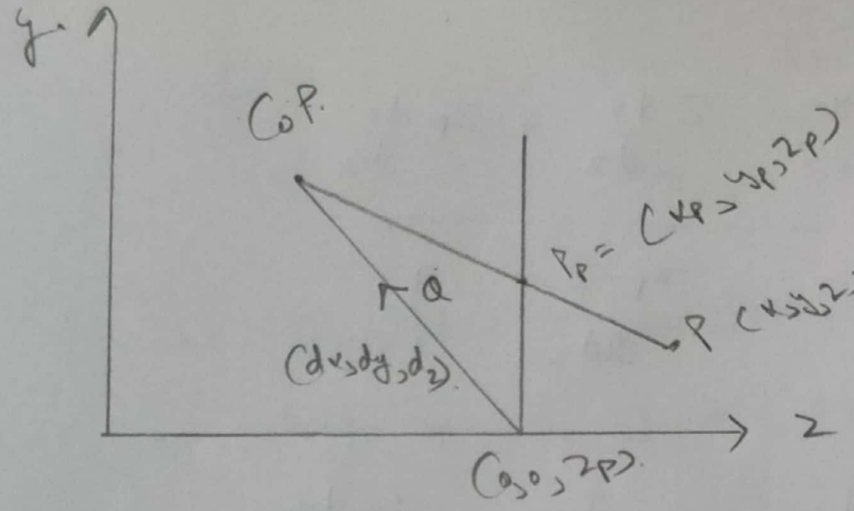


③ View-Confusion - objects behind the center of Projection upside down and back ward onto the view-plane.

④ Topological distortion: A line segment joining a point which lies in front of the viewer to a point in back of the viewer is projected to a broken line of Infinite extent.



Ans 2. First let us derive Generalized Projection Matrix from which we can obtain Perspective and Parallel Projection Matrices.



→ from the above diagram:

$$P_p = CoP + t(P - CoP) \quad , \quad t \in [0, 1]$$

$$CoP = (0, 0, z_p) + Q\,(d_x, d_y, d_z)$$

$$P' = (x', y', z')$$

$$x' = Q\,d_x + t(x - Q\,d_x)$$

$$y' = Q\,d_y + t(y - Q\,d_y)$$

$$z' = (z_p + Q\,d_z) + t(z - (z_p + Q\,d_z)).$$

→ upon solving.

$$t = \frac{Z_p - (Z_p + Q d_z)}{Z - (Z_p + Q d_z)}$$

$$x_p = \frac{x - Z\frac{dx}{dz} + Z_p\frac{dx}{dz}}{\frac{Z_p - Z}{Q d_z} + 1}.$$

$$y_p = \frac{y - Z\frac{dy}{dz} + Z_p\frac{dy}{dz}.}{\frac{Z_p - Z}{Q d_z} + 1}$$

$$Z_p = \frac{-Z\frac{Z_p}{Q d_z} + \frac{Z_p^2 + Z_p Q d_z}{Q d_z}}{\frac{Z_p - Z}{Q d_z} + 1}.$$

Scanned with CamScanner

The eq^n can be written in Matrix form as

$$M_G = \begin{bmatrix} 1 & 0 & -\frac{dx}{dz} & \frac{z_p dx}{d_z} \\ 0 & 1 & -\frac{dy}{dz} & \frac{z_p dy}{dz} \\ 0 & 0 & \frac{-z_p}{Q d_z} & \frac{z_p^2}{Q d_z} + z_p \\ 0 & 0 & \frac{-1}{Q d_z} & \frac{z_p}{Q d_z} + 1 \end{bmatrix}$$

→ M_G can be converted to Perspective Projection matrix by taking $z_p = -d$, $Q = d$, $dx = 0$, $dy = 0$, $d_z = 1$.

Substitute in M_G

$$M_{per} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{d} & 0 \end{bmatrix}$$

Perspective ~~Parallel~~ Projection matrix

→ M_G can be converted to Parallel Projection matrix by taking $z_p = 0$, $Q = \infty$, $dx = 0$, $dy = 0$, $dz = 1$

$$M_{par} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Parallel Projection Matrix.

Arsa

Scanned with CamScanner

Ans 3.

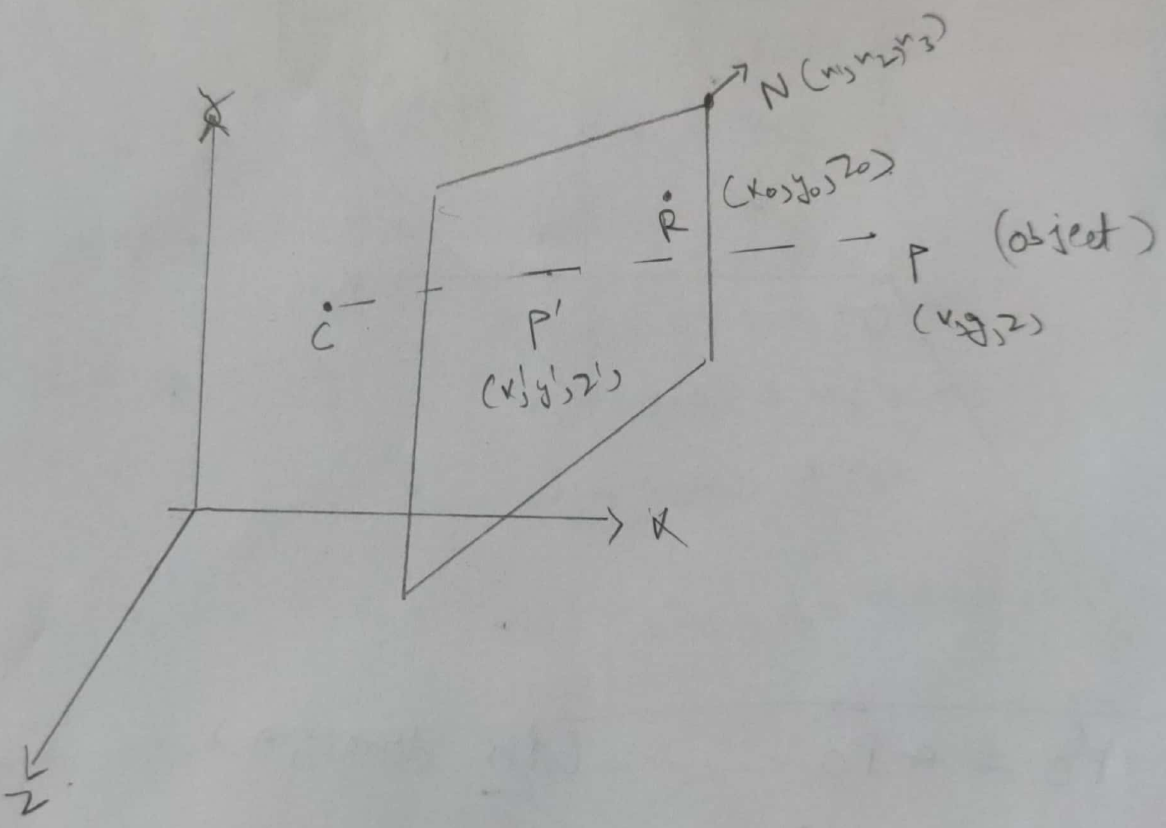
Center : $C(a, b, c)$

Refrence-Point : $R(x_0, y_0, z_0)$

Normal (N) : ~~$(n_1\hat{i}, n_2\hat{j}, n_3)$~~ $n_1\hat{i} + n_2\hat{j} + n_3\hat{k}$

→ Let $P(x, y, z)$ be the Point on object.

and $P'(x', y', z')$ be the Point on screen.



→ First translate the center to the origin.

for that translation Matrix M_T

$$M_T = \begin{bmatrix} 1 & 0 & 0 & -a \\ 0 & 1 & 0 & -b \\ 0 & 0 & 1 & -c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\vec{P'C} = \alpha\,\vec{PC}$$

→ $x' = \alpha(x-a)+a$

→ $y' = \alpha(y-b)+b$

→ $z' = \alpha(z-c)+c.$

→ $\vec{R_0P'}\cdot\vec{N} = 0$ {dot Product of Perpendicular vectors is 0}

→ $n_1x_0 + n_2y_0 + n_3z_0 = n_1x' - n_1a + n_2y' - n_2b + n_3z' - n_3c$

$-n_1a - n_2b - n_3c$

$$n_1x_0 + n_2y_0 + n_3z_0 = n_1\alpha(x-a) + n_1a + n_2\alpha(y-b) + n_2b + n_3\alpha(z-c) + n_3c$$

→ let $d_0 = n_1x_0 + n_2y_0 + n_3z_0$, $d_1 = n_1a + n_2b + n_3c.$

$d = d_0 - n_1a - n_2b - n_3c = d_0 - d_1.$

→ $d = \alpha[n_1(x-a) + n_2(y-b) + n_3(z-c)]$

$$\alpha = \frac{d}{n_1(x-a)+n_2(y-b)+n_3(z-c)}$$

$$M_{per} = \begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & d & 0 \\ n_1 & n_2 & n_3 & 0 \end{bmatrix}.$$

③ Translate Center back to $C(a, b, c)$

$$M_T^{-1} = \begin{bmatrix} 1 & 0 & 0 & +a \\ 0 & 1 & 0 & +b \\ 0 & 0 & 1 & +c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

→ Final Matrix $M_G = M_T^{-1} \, M_{per} \, M_T$.

$$M_T^{-1} \, M_{per} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & d & 0 \\ n_1 & n_2 & n_3 & 0 \end{bmatrix}$$

$$M_T^{-1} \cdot M_{per} = \begin{bmatrix} n_1 a + d & n_2 a & n_3 a & 0 \\ n_1 b & n_2 b + d & n_3 b & 0 \\ n_1 c & n_2 c & n_3 c + d & 0 \\ n_1 & n_2 & n_3 & 0 \end{bmatrix}.$$

$$M_T^{-1} \cdot M_{per} \cdot M_T = \begin{bmatrix} a n_1 + d & a n_2 & a n_3 & -(a d_0) \\ n_1 b & a n_2 + d & n_3 b & -(b d_0) \\ n_1 c & n_2 c & c n_3 + d & -(c d_0) \\ n_1 & n_2 & n_3 & -d_1 \end{bmatrix}.$$

Ans

Ans 4. It is a type of Parallel Projection where Projecting rays emerge Parallely from surface of the polygon and incident at an angle other than 90° on the plance.

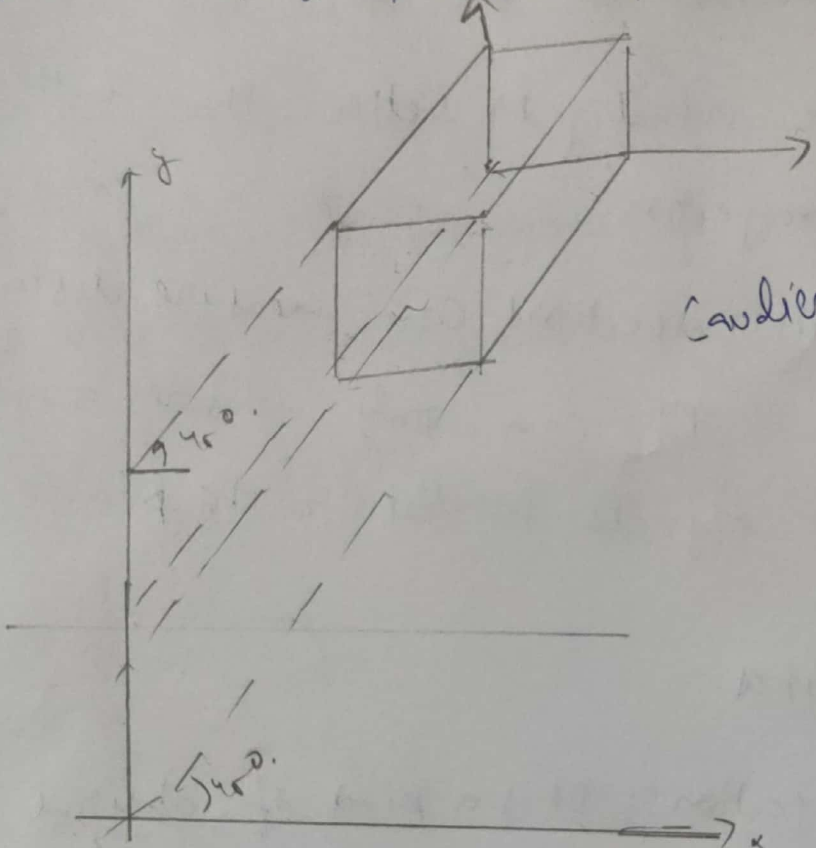
→ objects can be visualized better than with orthographic projection.

→ oblique parallel projections can measure distances but not angle. It can only measure angles for faces of objects parallel to the plane.

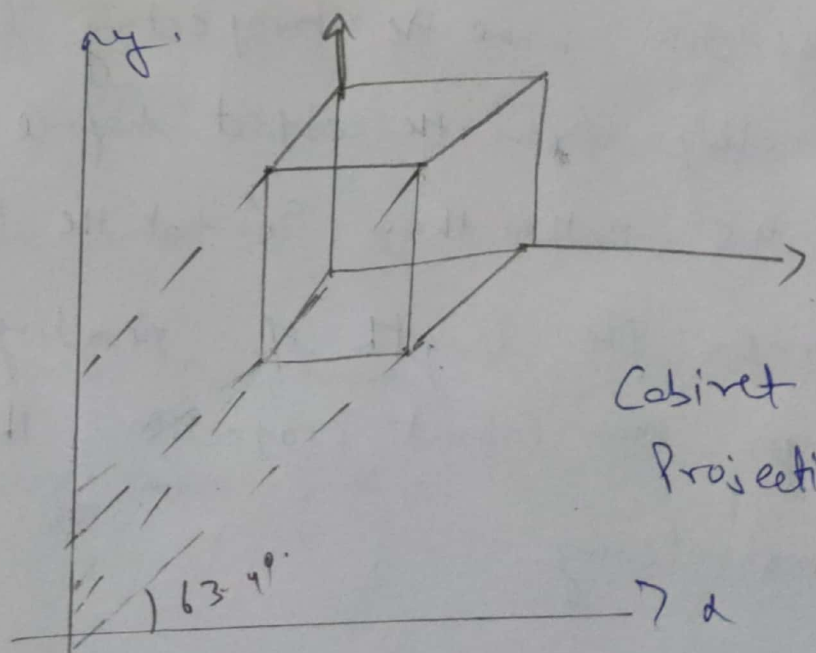
→ It is of 2 types.

(i) Cavalier Projection : It is a kind of oblique Projection where the projecting lines emerge Parallely from the object surface and Incident at 45° rather than 90° at the Projecting Plane. The length of reading axis is larger than Cabinet Projection. There is no foreshortening.

iii) Cabinet Projection : The direction of Projection makes 63.4° angle with the projection plane. This results in fore shortening of the Z-axis and provides a more "realistic" view.



Cavalier Projection.



Cabinet Projection

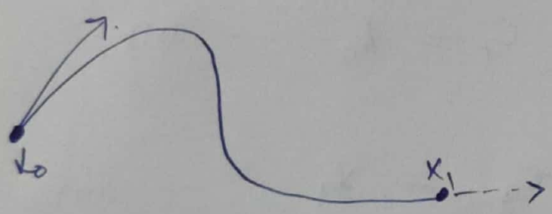
Curves and surfaces

Ans1. Polynomial functions can be used for generating Hermite curves. A Hermite curve is a curve for which the user provides:

→ The end point of the curve.

→ The parametric derivatives of curves at the end points (tangents with length) $\left(\frac{df_x}{dt}, \frac{df_y}{dt}\right)$.

where f_x and f_y are functions of t.



For x, we have following constraints:

→ The curve must pass through x_0 when $t=0$

→ The derivative must be x_0' when $t=0$.

→ The curve must pass through x_1, $t=1$

→ The derivative ~~must be~~ must be x_1', $t=1$.

→ Consider $f_x(t)$ for generating the hermite curve.

$$f_x(t) = C_0 + C_1 t + C_2 t^2 + C_3 t^3.$$

C_i are unknown, $t \in [0,1]$.

$$\frac{df_x}{dt} = C_1 + 2C_2 t + 3C_3 t^2$$

→ Constraints:

→ $f_x(0) = x_0 \Rightarrow C_0 = x_0$ --- (i)

→ $f'_x(0) = x'_0 \Rightarrow C_1 = x'_0$ --- (ii)

→ $f_x(1) = C_1 + C_2 + C_0 + C_3 = x_1$ --- (iii)

→ $f'_x(1) = C_1 + 2C_2 + 3C_3 = x'_1$ --- (iv)

Solving for C_0, C_1, C_2, C_3 in terms of x_0, x'_0, x_1, x'_1.

$\Rightarrow C_0 = x_0, \; C_1 = x'_0$

Scanned with CamScanner

from (i), (ii), (iii)

$$C_2 + C_3 = x_1 - x_0 - x_0' \quad \text{— (v)}$$

from (ii), (iv)

$$2C_2 + 3C_3 = x_1' - x_0' \quad \rightarrow \quad \text{(vi)}$$

from (v), (vi)

$$C_3 = x_1' - x_0' - 2(x_1 - x_0 - x_0')$$

$$= x_1' + 2x_0 + x_0' - 2x_1 \quad \text{--- (vii)}$$

from (vii), (v)

$$C_2 = \cancel{4x_1 - x_1' - 3x_0 - 3x_0'} \quad \text{--- (viii)}$$

$$= 3x_1 - 3x_0 - x_1' - 2x_0'$$

Hence

$$\boxed{\begin{aligned} C_0 &= x_0 \\ C_1 &= x_0' \\ C_2 &= 3x_1 - 3x_0 - x_1' - 2x_0' \\ C_3 &= 2x_0 - 2x_1 + x_1' + x_0' \end{aligned}}$$

Put value of C_0, C_1, C_2, C_3 in $f_x(t)$

$$f_x(t) = x_0 + t(x_0') + t^2(3x_1 - 3x_0 - x_1' - 2x_0')$$
$$+ t^3(2x_0 - 2x_1 + x_0' + x_1').$$

$$f_x(t) = x_0(2t^3 - 3t^2 + 1) + x_0'(t^3 - 2t^2 + t)$$
$$x_1(-2t^3 + 3t^2) + x_1'(t^3 - t^2).$$

Representing in Matrix Form.

$$x = [x_0\ x_1\ x_0'\ x_1'] \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$

→ extending this to 3-dimensions.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x_0\ x_1 & x_0' & x_1' \\ y_0\ y_1 & y_0' & y_1' \\ z_0\ z_1 & z_0' & z_1' \end{bmatrix} \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$

Ans.

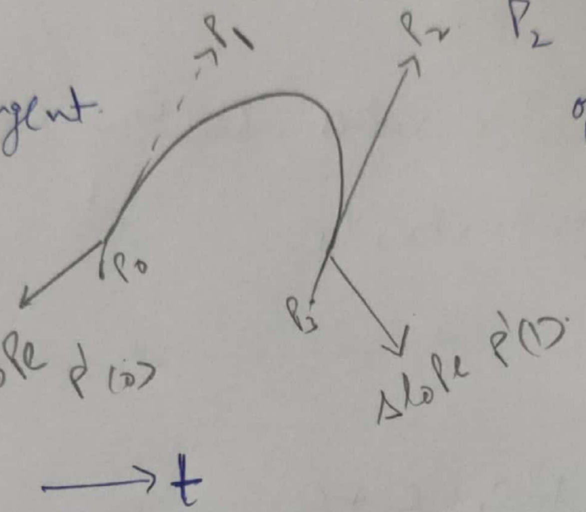
Ans 2. Let us consider a Beizer Curve.

Here the end points are defined by two control points and two points control the tangents.

P_1 located at $1/3$ start tangent.

P_2 located at $2/3$ of the end tangent vector.

$$P'(0) = \frac{P_1 - P_0}{1/3}$$



$$P'(1) = \frac{P_3 - P_2}{1/3}.$$

Considering the end site conditions for the curve:

$\Rightarrow$ $f_x(0) = x_0 = C_0$ —(i)

$\Rightarrow$ $f_x(1) = x_3 = C_0 + C_1 + C_2 + C_3$ — (ii)

$\Rightarrow$ Let the curve be: $f_x(t) = C_0 + C_1 t + C_2 t^2 + C_3 t^3$

$$\frac{df_x}{dt} = C_1 + 2C_2 t + 3C_3 t^2.$$

Scanned with CamScanner

→ Now Approximating the derivative conditions :-

→ $f_x'(0) = 3(x_1 - x_0) = c_1$ —— (3)

$f_x'(1) = 3(x_3 - x_2) = c_1 + 2c_2 + 3c_3$ – (4)

→ Solving For values c_0, c_1, c_2, c_3 in terms of x_0, x_1, x_2, x_3, x_4.

① $c_0 = x_0$

$\therefore x_3 = x_0 + c_1 + c_2 + c_3$

$\Rightarrow c_1 + c_2 + c_3 = x_3 - x_0$.

Scanned with CamScanner

$C_1 = 3(x_1 - x_0)$

$\therefore C_2 + C_3 = x_3 - x_0 - 3(x_1 - x_0)$ — (5)

$C_1 + 2C_2 + 3C_3 = 3(x_3 - x_2)$

$2C_2 + 3C_3 = 3(x_3 + x_0 - x_1 - x_2)$ — (6)

from (5) & (6).

$C_3 = x_3 - 3x_2 + 3x_1 - x_0$ — (7)

$C_2 = 3x_2 - 6x_1 + 3x_0$ — (8)

Hence

$$\boxed{\begin{aligned} C_0 &= x_0 \\ C_1 &= 3x_1 - 3x_0 \\ C_2 &= 3x_2 - 6x_1 + 3x_0 \\ C_3 &= x_3 - 3x_2 + 3x_1 - x_0 \end{aligned}}$$

$\Rightarrow$ Now on replacing the original Hermite matrix.

$$\begin{bmatrix} x_s \\ x_e \\ x_s' \\ x_e' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -3 & 3 & 0 & 0 \\ 0 & 0 & -3 & 3 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$x(t) = [x_s\ x_e\ x_s'\ x_e'] \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 1 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}.$$

So the Beizer Curve becomes.

$$x(t) = [t^3\ t^2 + 1] \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}.$$

$$\cancel{x(t) = \begin{bmatrix} (1-t)^3 \\ 3t(1-t)^2 \\ 3t^2(1-t) \\ t^3 \end{bmatrix}^T \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}}$$

$$\cancel{x(t) = (1-t^3)x_0 + 3t(1-t^2)x_1}$$

$+$

$$x(t) = \begin{bmatrix} (1-t)^3 & 3t(1-t)^2 & 3t^2(1-t) & t^3 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

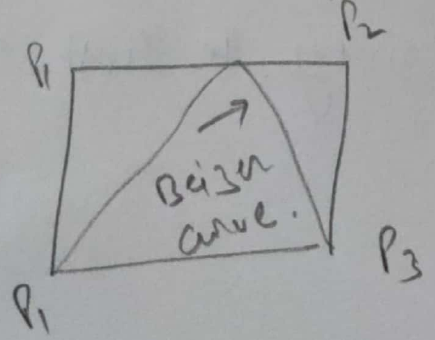
$$x(t) \Rightarrow x_0(1-t)^3 + x_1(3t)(1-t)^2 + x_2(3t^2)(1-t) + t^3 x_3$$

$$x(t) = \sum_{i=0}^{3} {}^3C_i \, t^i (1-t)^{3-i} x_i$$

Hence Proved.

★ Propeties of Beizer Curves:

→ Convex Hull Property: This ensures that all Beizer Curves lies in the convex hull of their control points.



→ Bernstein Polynomial:

The blending Function of Cubic beizer Curves are a special case of the Bernstein Polynomial.

(d - degree ; K: index)

$$B_{Kd}(t) = \frac{d!}{K!\,(d-K)!} \cdot t^{K} (1-t)^{d-K}$$

$$B(t) = \sum_{K=0}^{d} B_{Kd}(t) \cdot P_K \qquad t \in [0,1]$$

These Polynomial give the blending Polynomials for any degree of Beizer form., they all sum to 1. They all lie in range (0,1).

→ The First and last control points are Interpolated

→ The tangent to the Curve at the first Control point is along the line joining the First and second control points.

→ The tangent at last control point is along the line joining the second last and last control points.

→ The curve can be rendered in many ways Example: by converting to the line segments by a subdivision algorithm.

→ The degree of the Beizer curve depends on the number of control points.

→ The beizer curve lacks local control Changing the position of one control point affects the entire curve.

Ans 3. The Bezier surface is formed by moving Bezier curve through space. The surface is formed when each control point B_i sweeps a curve in the space.

The Bezier surface is extension of the Bezier curve in 2D directions, a tensor product of 2 curves.

It is given by

$$C(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} P_{i,j} \, B_{i,n}(u) \, B_{j,m}(v).$$

where $u,v \in [0,1]$

where the parametric directions have degrees n, m hence having $(n+1) \times (m+1)$ control points.

It is given as

$$C(u,v) = \frac{\sum_{i=0}^{n} \sum_{j=0}^{m} w_{ij} \cdot P_{ij} \cdot B_{i,n}(u) \cdot B_{j,m}(v)}{\sum_{i=0}^{n} \sum_{j=0}^{m} w_{ij} \, B_{i,n}(u) \cdot B_{j,m}(v)}.$$

Scanned with CamScanner

The Baizer Patches are generated ~~by~~ the same way Beizer curves are. If the patches share some control points along an edge, they will form at the Beizer curve defined by those control points.

If the patch is ~~to be joi~~ to be joined with regular polygon, the control points along the edge to be joined should be collinear, since that will make the patch flat at the edge.

The Patches can be connected in 3 ways :-

1) Planar : when the patches never connect back with themselves

2) Cylindrical : Last patch edge wraps around and connects back up to leading edge forming cylinderical shape.

3) Toroidal : If all edges wraps back on themselves a donut shape is formed known as torus.