Three - Dimensional Graphics

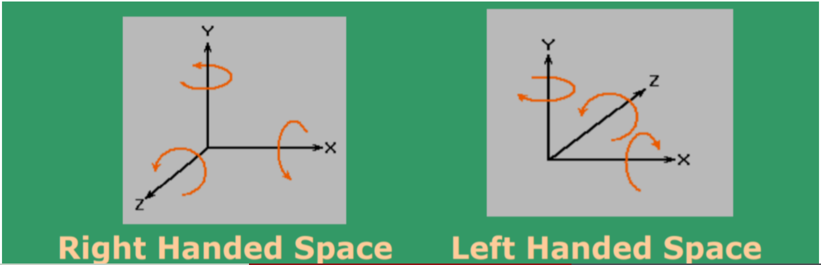
Dr. V Masilamani
masila@iiitdm.ac.in

Department of Computer Science and Engineering
IIITDM Kancheepuram
Chennai-127

Overview

Representation of Points

- Right-handed coordinate system: +ve X is right of origin; +ve Y is above origin; +ve Z axis comes towards the viewer
- Left-hand coordinate system: +ve X is right of origin; +ve Y above is origin; +ve Z axis goes away from the the viewer
- To transform from right handed coordinate system to left handed coordinate system, negate the z values.
- A 3D-point is represented in right handed coordinate system by default as (a, b, c) or $a\vec{i} + b\vec{j} + c\vec{k}$, where $\vec{i}, \vec{j}, \vec{k}$ are unit vectors in X, Y, Z directions respectively



Homogeneous representation of a 3D point

- A 3D point in homogeneous coordinate system: $P = (x, y, z, w)^T$
- Linear transformation of 3D point in homogeneous coordinate system:
  $P' = A.P$ where
  $$A = \begin{bmatrix} a & b & c & p \\ d & e & f & q \\ g & i & j & r \\ l & m & n & s \end{bmatrix} = \begin{bmatrix} T & K \\ \tau & \Theta \end{bmatrix}$$ where,
  - $T = \begin{bmatrix} a & b & c \\ d & d & f \\ g & i & j \end{bmatrix}$
    Produces linear transformations: *scaling, shearing, reflection and rotation*
  - $K = [\mathrm{p\ q\ r}]^T$, produces translation
  - $\tau = [\mathrm{l\ m\ n}]^T$, yields perspective transformation
  - $\Theta = \mathrm{s}$, is responsible for uniform scaling

Transformation Matrix in 3D:

Translation:

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

Scaling:

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

Shear:

$$\begin{bmatrix} 1 & 0 & Sh_x & 0 \\ 0 & 1 & Sh_y & 0 \\ 0 & Sh_z & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Origin is unaffected by scale and shear.

3D Reflection

The following matrices:

$$T_{XY} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_{YZ} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_{ZX} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

produce reflection about:

XY-plane YZ-plane ZY-plane

respectively.

Rotation Matrices along an axis:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\alpha) & -sin(\alpha) & 0 \\ 0 & sin(\alpha) & cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

X - axis

$$\begin{bmatrix} cos(\beta) & 0 & sin(\beta) & 0 \\ 0 & 1 & 0 & 0 \\ -sin(\beta) & 0 & cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

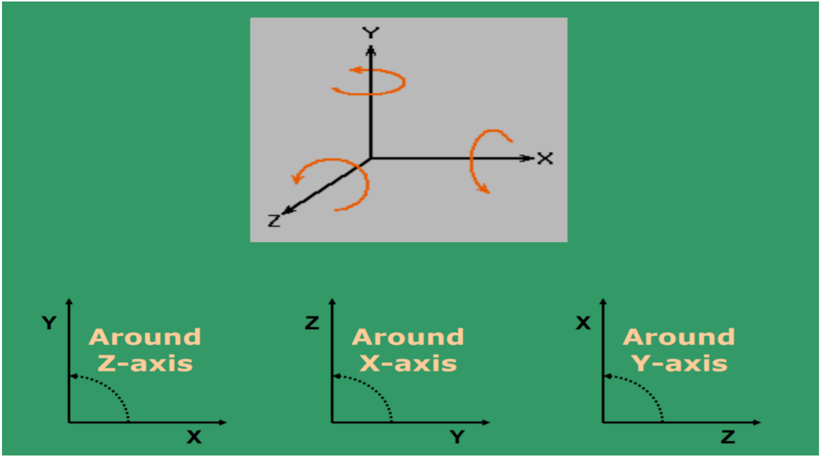
Y - axis

$$\begin{bmatrix} cos(\gamma) & -sin(\gamma) & 0 & 0 \\ sin(\gamma) & cos(\gamma) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

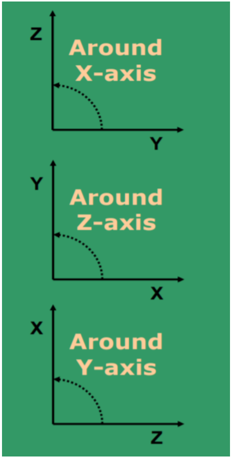
Z - axis

- HW: Derive the rotation matrix for rotating around Y-axis

Rotation Matrices along an axis: (cont.)



Rotation Matrices along an axis: (cont.)



$$\begin{array}{cc} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\alpha) & -sin(\alpha) & 0 \\ 0 & sin(\alpha) & cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & \begin{array}{l} \leftarrow \boldsymbol{X} \\ \leftarrow \boldsymbol{Y} \\ \leftarrow \boldsymbol{Z} \\ \end{array} \end{array}$$

$$\begin{array}{cc} \begin{bmatrix} cos(\gamma) & -sin(\gamma) & 0 & 0 \\ sin(\gamma) & cos(\gamma) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & \begin{array}{l} \leftarrow \boldsymbol{X} \\ \leftarrow \boldsymbol{Y} \\ \leftarrow \boldsymbol{Z} \\ \end{array} \end{array}$$

$$\begin{array}{cc} \begin{bmatrix} cos(\beta) & 0 & sin(\beta) & 0 \\ 0 & 1 & 0 & 0 \\ -sin(\beta) & 0 & cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & \begin{array}{l} \leftarrow \boldsymbol{X} \\ \leftarrow \boldsymbol{Y} \\ \leftarrow \boldsymbol{Z} \\ \end{array} \end{array}$$

Rotation About an Arbitrary Axis in Space

Assume that we want to perform a rotation by θ degrees, about an axis in space passing through the point (x_0, y_0, z_0) with direction cosines (c_x, c_y, c_z).

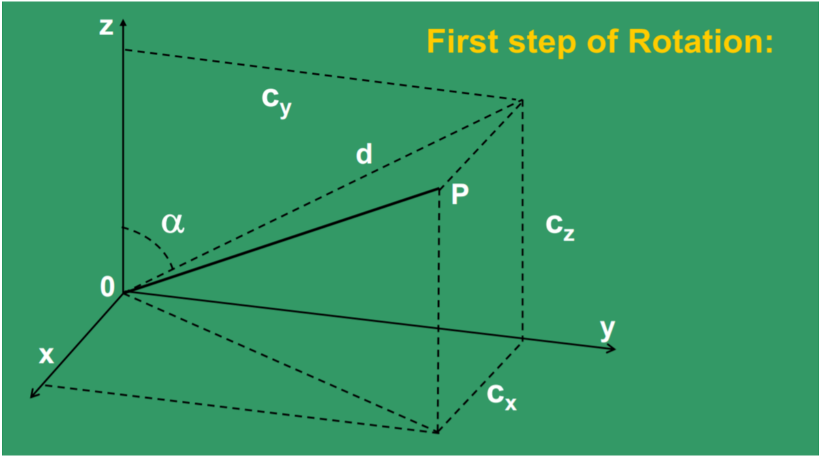
- First of all translate by:
  T = -(x_0, y_0, z_0)
- Next, we rotate the axis into one of the principle axes, let's say, Z(R_x, R_y).
- We rotate next by θ degrees about Z($| R_z(\theta) |$).
- Then we undo the rotations used to align the axis.
- We undo the translation: translate by
  $(-x_0, -y_0, -z_0)$

The tricky part of the algorithm is in step (2), as given before.

This is going to take 2 rotations:

- About x-axis
  (to place the axis in the xz plane)
- About y-axis
  (to place the result coincident with the z-axis)

Rotation About an Arbitrary Axis in Space (cont.)



Rotation about x by α:
How do we determine α?

Project the unit vector, along OP, into the yz plane.

The y and z components, c_y and c_z, are the direction cosines of the unit vector along the arbitrary axis.

It can be seen from the diagram, that :

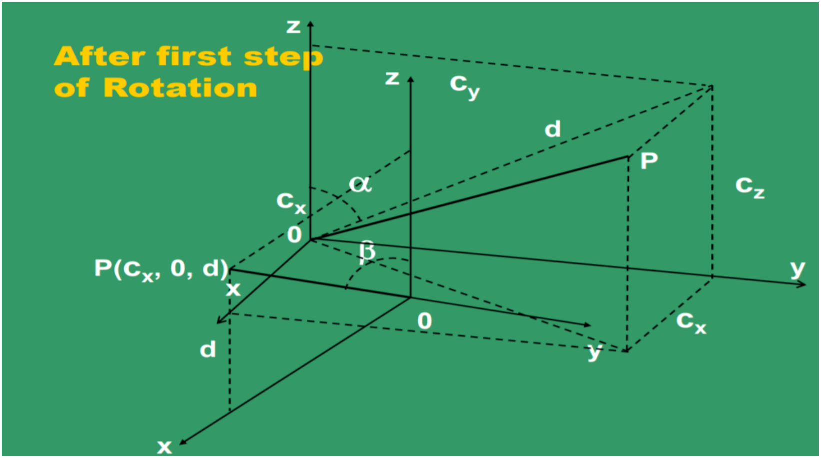
$\mathrm{d} = \mathrm{sqrt}(\mathrm{C}_y{}^2 + \mathrm{C}_z{}^2)$

$\cos(\alpha) = C_z/\mathrm{d}$

$\sin(\alpha) = C_y/\mathrm{d}$

$\alpha = sin^{-1}[C_y/\mathrm{sqrt}(C_y{}^2+\mathrm{C}_z{}^2)]$

Rotation About an Arbitrary Axis in Space (cont.)



Rotation About an Arbitrary Axis in Space (cont.)

Rotation by β about y:

How do we determine β?

Steps are similar to that done for β

- Determine the angle β to rotate the result into the Z axis:
- The x component is c_x and the z component is d

$\cos(\beta)$ = d = d/(length of unit vector)
$\sin(\beta) = c_x = c_x$/(length of unit vector)

Final Transformation for 3D rotation, about an arbitrary axis:

$$M = |T||R_x||R_y||R_z||R_y|^{-1}|R_x|^{-1}|T|^{-1}$$

Final Transformation matrix for 3D rotation, about an arbitrary axis:

$$M = |T||R_x||R_y||R_z||R_y|^{-1}|R_x|^{-1}|T|^{-1}$$

where:

$$T = \begin{bmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C_z/d & -C_y/d & 0 \\ 0 & C_y/d & C_z/d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

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

Final Transformation matrix for 3D rotation, about an arbitrary axis: (cont.)

$$M = |T||R_x||R_y||R_z||R_y|^{-1}|R_x|^{-1}|T|^{-1}$$

$$= [T\ R_x\ R_y][R_z][T\ R_x\ R_y]^{-1}$$

A special case of 3D rotation:

Rotation about an axis parallel to a coordinate axis (say, parallel to X-axis):

$$M = |T|\ |R_x|\ |T|^{-1}$$

Final Transformation matrix for 3D rotation, about an arbitrary axis: (cont.)

If you are given two points instead(on the axis of rotation), you can calculate the direction cosines of the axis as follows:

$$V = ((x_1\text{-}x_0),\ (y_1\text{-}y_0),\ (z_1\text{-}z_0))^T$$

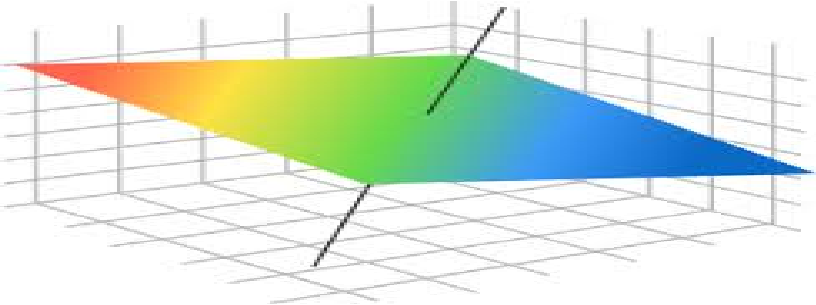
$$C_x = (x_1\text{-}x_0)/|V|$$

$$C_y = (y_1\text{-}y_0)/|V|$$

$$C_z = (z_1\text{-}z_0)/|V|$$

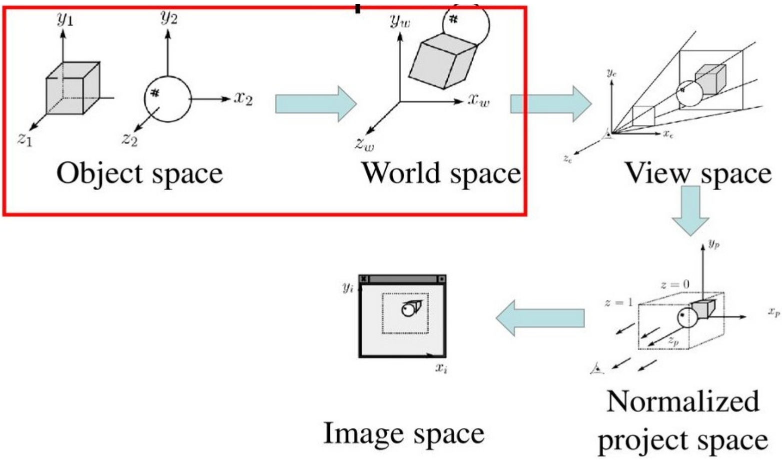
where |V| is length of the vector V.

Reflection through an arbitrary plane



- Method is similar to that of rotation about an arbitrary axis
- M $= |T||R_x||R_y||R_{fl}||R_y|^{-1}|R_x|^{-1}|T|^{-1}$
- T does the job of translating the origin to the plane.
- R_x and R_y will rotate the vector normal to the reflection plane (at the origin), until it is coincident with the $+Z$ axis.
- R_{fl} is the reflection matrix about XY plane or Z=0 plane.

3D-Graphics pipeline



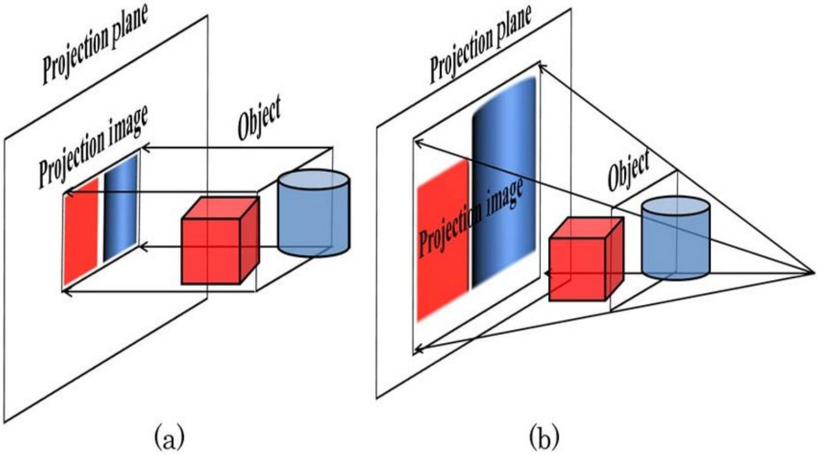
3D-Graphics pipeline (cont.)

- **Object Space:** Space where definition of objects are provided. Also called Modeling space
- **World Space:** Space where the scene and viewing specification is made
- **View space / Eyespace:** Space where eye point (COP) is at the origin looking down the Z axis
- **Normalized Viewing Space/3D Projective space :** Clipped portion of scene in view space, and nolrmalized with range: [-1:1] for X Y, [0:1] for Z.
- **Image Space:** 2D array of pixel values
- **Screen Space (2D):** Range of Coordinates: [0 : width] [0 : height]

Projections

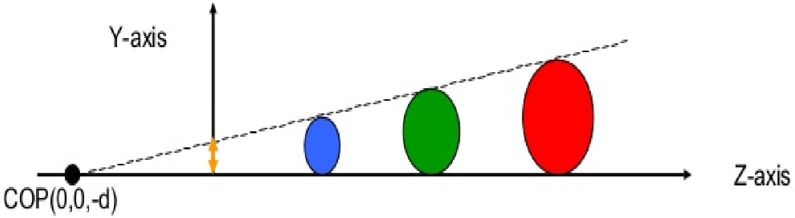
- **Need for projection:**
  - The scene(set of objects) is in 3D, but display system can display only 2D
  - Projection can be used to achieve the same
- **What is projection:** Any idempotent map is called as projection
- Projection of a 3D object is defined by projection rays (called projectors)
- The projection of a point P on the object is defined as P' if P' lies on a projector, and the projector hits the projection plane at P'
- Types of projection:
  - Parallel Projections: The projectors are parallel
  - Perspective: The projectors are emanating from a point(called centre of projection)

Projections (cont.)



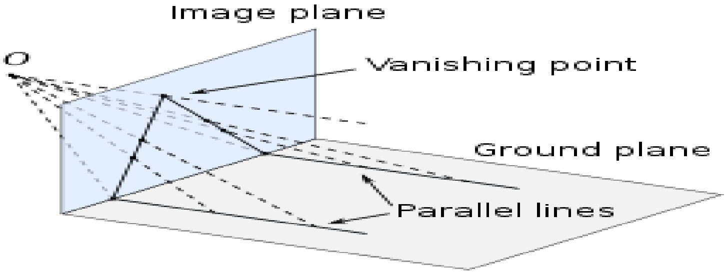
Perspective Projections

- **Perspective foreshortening:** The size of the perspective projection of the object varies inversely with the distance of the object from the center of projection.
- In the fig below, all three circular discs are mapped to the same line segment(in yellow)



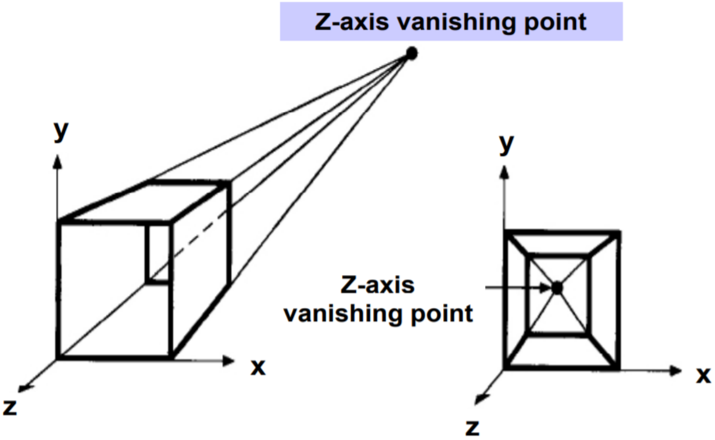
Perspective Projections (cont.)

- Vanishing Point: The perspective projections of any set of parallel lines that are not parallel parallel to the projection plane converge to a point, and such point is called as vanishing point.

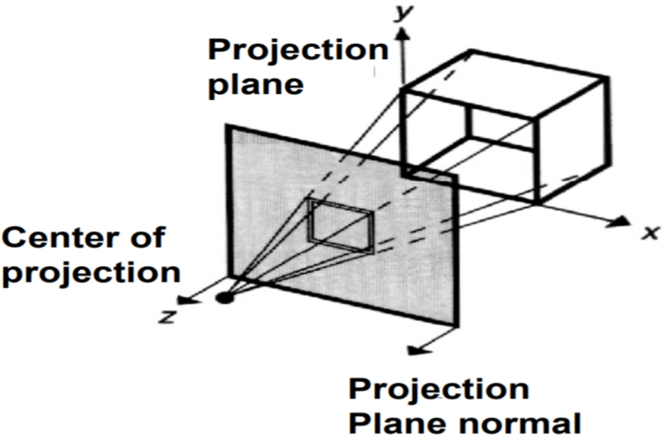


- Two parallel lines are mapped to non-parallel lines by perspective projection, and hence perspective projection is not affine transform

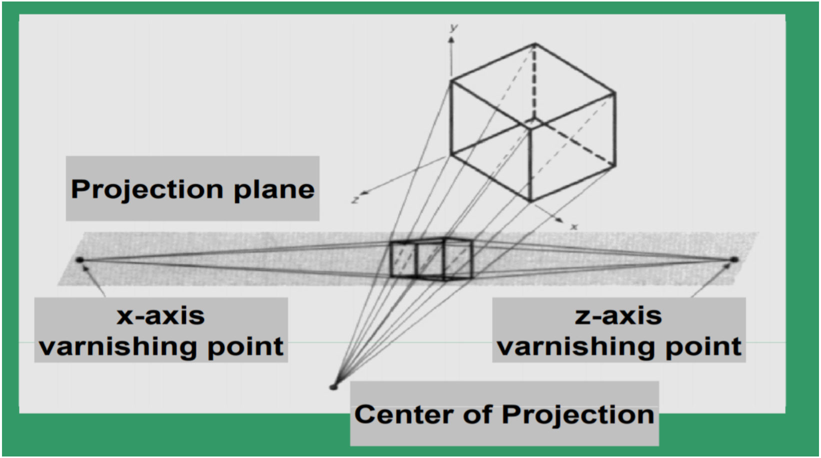
Perspective Projections (cont.)



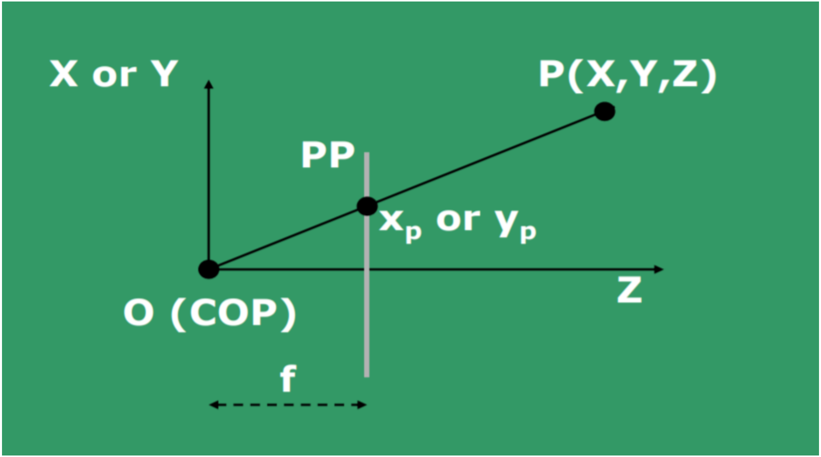
Perspective Projections (cont.)



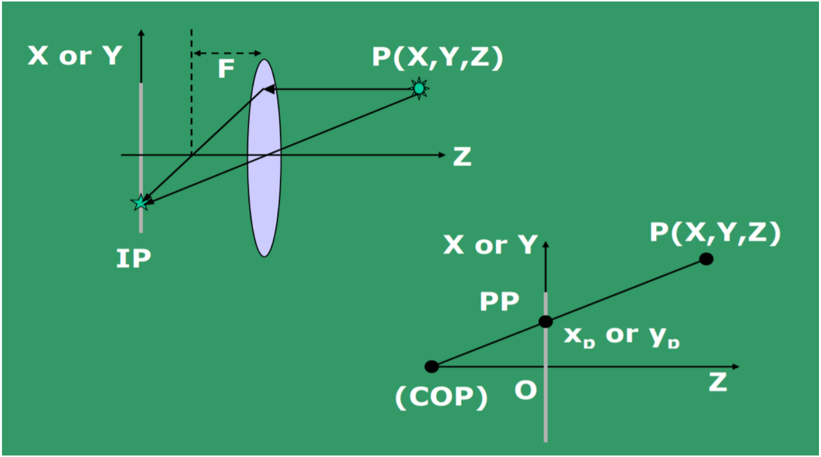
Perspective Projections (cont.)



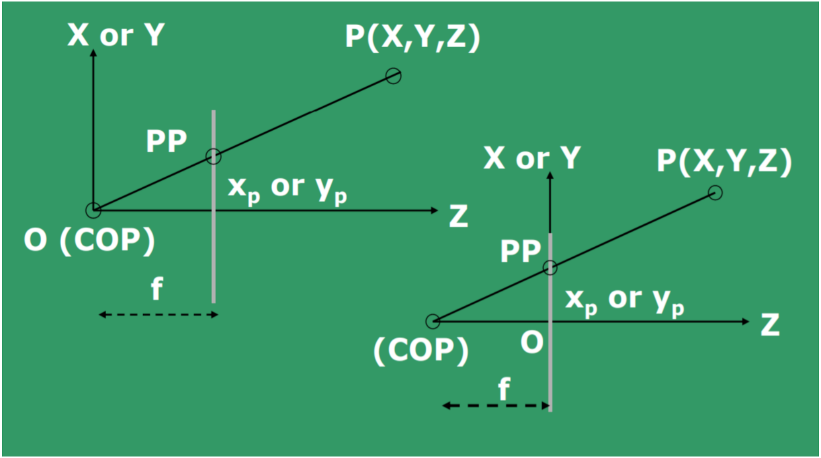
Perspective Geometry and Camera Models



Perspective Geometry and Camera Models (cont.)



Perspective Geometry and Camera Models (cont.)



Equations of Perspective geometry

$x_p/\mathrm{f} = \mathrm{X}/\mathrm{Z};$

$y_p/\mathrm{f} = \mathrm{Y}/\mathrm{Z};$

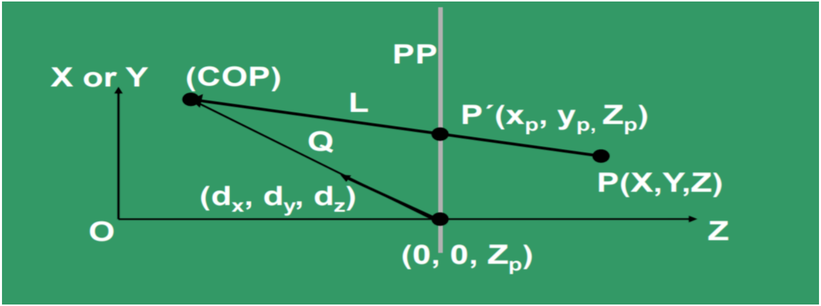
$$M_{per} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix}$$

$\mathrm{p}^{'}=\mathrm{M}_{per}.\mathrm{P}$
where $\mathrm{P} = [\mathrm{X\ Y\ Z\ 1}]^T$

$x_p/\mathrm{f} = \mathrm{X}/(\mathrm{Z}+\mathrm{f});$

$y_p/\mathrm{f}=\mathrm{Y}/(\mathrm{Z}+\mathrm{f});$

$$M_{per} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/f & 1 \end{bmatrix}$$



Parametric eqn. of the line L between COP and P:

COP + t(P-COP); $0 < t < 1$. ———————(1)

Generalized formulation of perspective projection: (cont.)

Let the direction vector from (0, 0, Z_p) to COP be (d_x, d_y, d_z), and Q be the distance from (0, 0, Z_p) to COP.

Then COP = (0, 0, Z_p) + Q(d_x, d_y, d_z).
Substitute COP in (1)
The coordinates of any point on line L is:

$$X^{'} = Qd_x + (X - Qd_x)t;$$

$$Y^{'} = Qd_y + (Y - Qd_y)t;$$

$$Z^{'} = (Z_p + Qd_z) + (Z - (Zp + Qd_z))t; \text{ ——(2)}$$

The objective is to find (x_p, y_p, z_p). when
$(X', Y', Z') = (x_p, y_p, Z_p)$, $Z' = Z_p$, substitute Z' in (2)
t = -Qd_z/(Z - (Z_p + Qd_z)
Now substitute t to obtain x_p and y_p.

Generalized formulation of perspective projection: (cont.)

$$x_p = \frac{X - Z\frac{d_x}{d_z} + Z_p\frac{d_x}{d_z}}{\frac{Z_p - Z}{Qd_z} + 1}$$

$$y_p = \frac{Y - Z\frac{d_y}{d_z} + Z_p\frac{d_y}{d_z}}{\frac{Z_p - Z}{Qd_z} + 1}$$

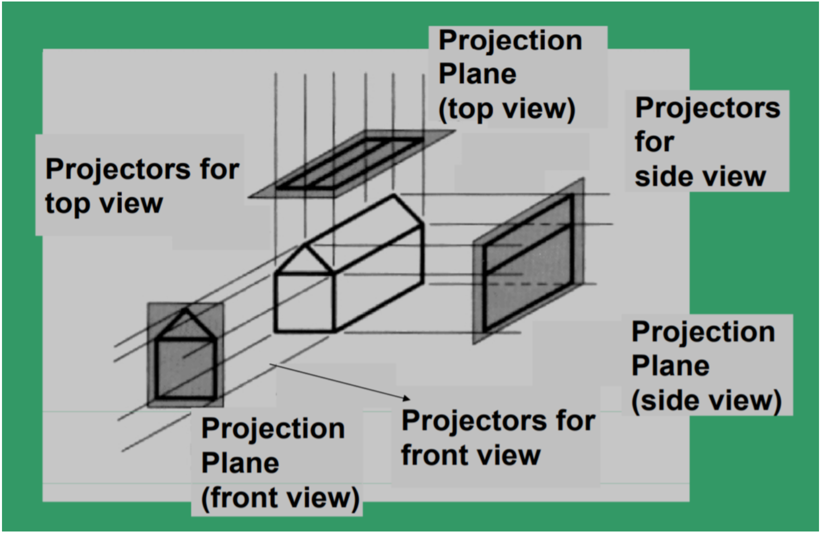
Generalized formula of perspective projection matrix:

$$M_{gen} = \begin{bmatrix} 1 & 0 & -\mathrm{d}_x/d_z & Z_p \; d_x/d_z \\ 0 & 1 & -\mathrm{d}_y/d_z & Z_p \; d_y/d_z \\ 0 & 0 & -Z_p/Qd_z & {Z_p}^2/Q\mathrm{d}_z + Z_p \\ 0 & 0 & -1/Q\mathrm{d}_z & Z_p/Qd_z + 1 \end{bmatrix}$$

Parallel Projection

- If the projectors are parallel lines, then the projection is called as parallel projection
- For Parallel projection, direction of projection (DOP) needs to be specified, not the COP
- **Orthographic Projection:** Direction of projection is normal to the projection plane

Example of Orthographic Projection



Transformation Orthographic Projection

The transformation matrix for orthographic projection when the projector is in Z axis direction

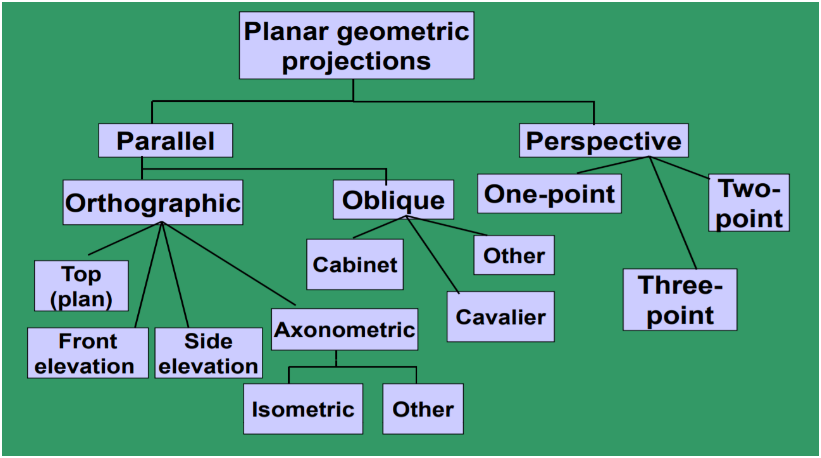
$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The transformation matrix for orthographic projection when the projector is in Y axis direction

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

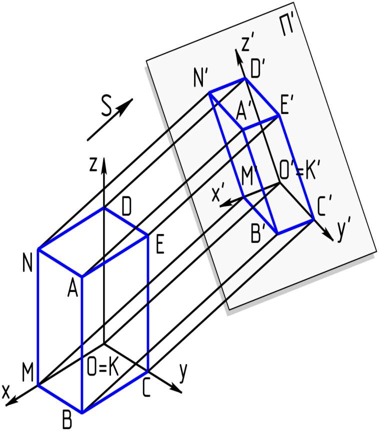
The transformation matrix for orthographic projection when the projector is in x axis direction

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

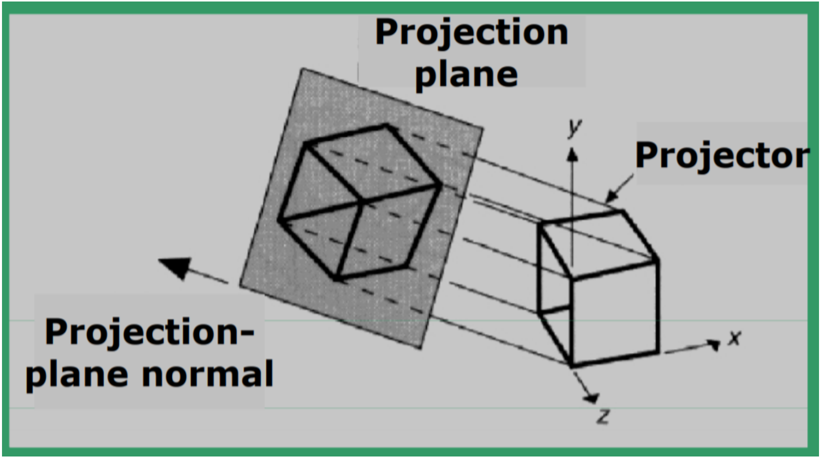

Planar geometric projections
Parallel
Perspective
Orthographic
Oblique
One-point
Two-point
Three-point
Cabinet
Other
Cavalier
Top (plan)
Front elevation
Side elevation
Axonometric
Isometric
Other

Axonometric orthographic projections

- An orthographic projection is said to be axonometric if
  - The projection plane is not normal to principal axes
- Types of axonometric
  - **Isometric projection:** Projection plane normal makes equal angles with each principle axis. DOP Vector: [1 1 1].
  - **Dimentric:** Projection plane normal makes equal angles with each two principle axis.
  - **Trimentric:** Projection plane normal makes unequal l angles with the principle axis.

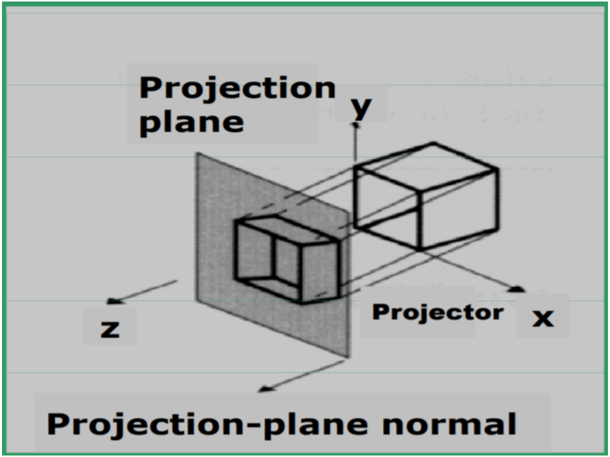


Isometric Projection

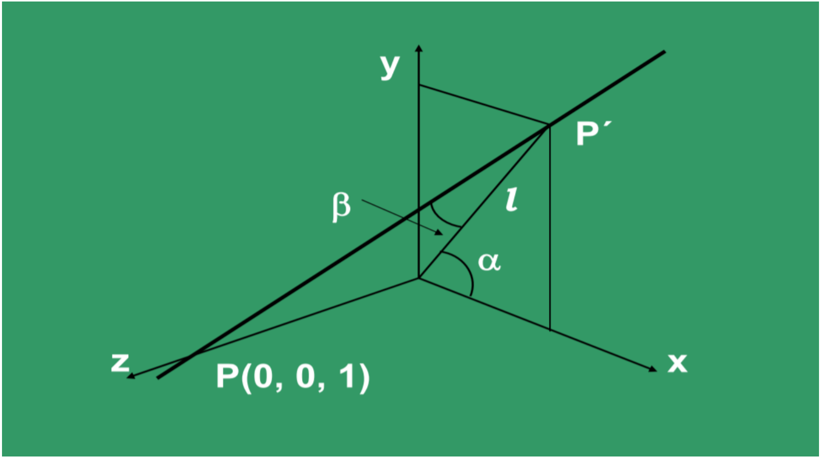


Oblique projections

- Plane of projection is normal to a Principle axis, but Projectors are not normal to the projection plane.



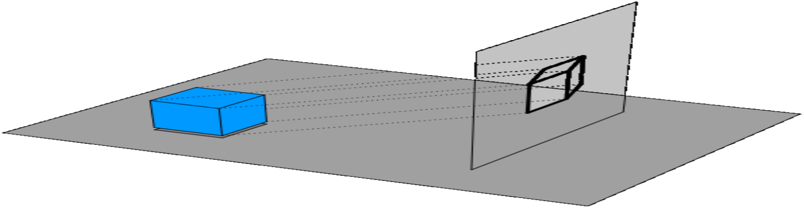
General oblique projection of a point/line:



General oblique projection of a point/line: (cont.)

- **Projection Plane:** x-y plane
- P' is the projection of P(0, 0, 1) onto x-y plane
- When DOP varies, both ‘l´ and α will vary

Cabinet Projection



- The Oblique projection is said to be **cavalier** if $tan(\alpha) = 1$, where the α is the angle between the projector and the projection plane
- The Oblique projection is said to be **cabinet** if $tan(\alpha) = 2$, where the α is the angle between the projector and the projection plane

Acknowledgements

- Some of the slides have been adopted from NPTEL and other internet sources

Thank You! :)