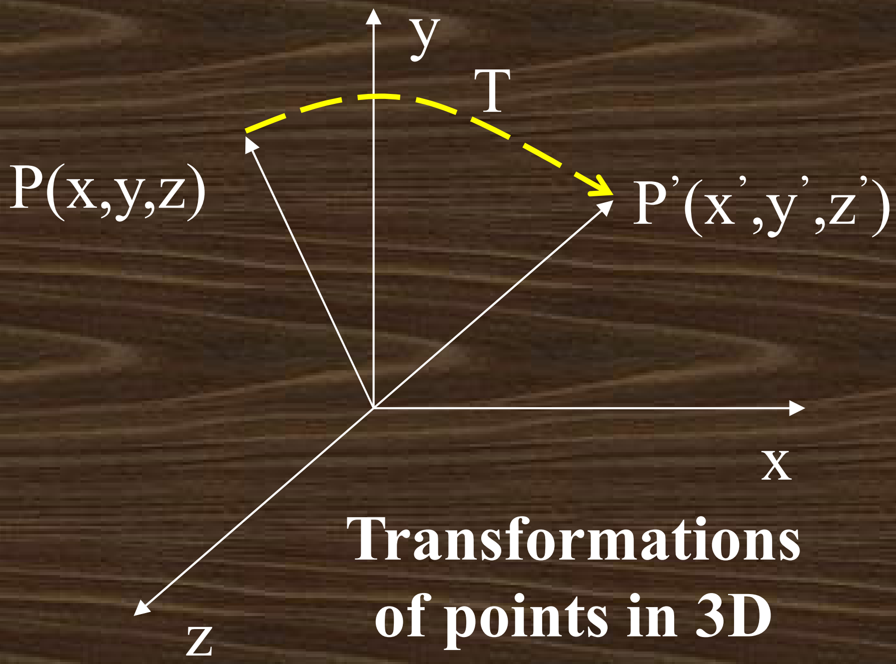
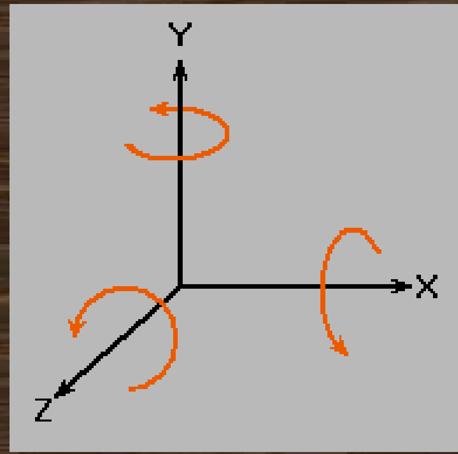
BASICS

Representation of Points in the 3D world: a vector of length 3

$$\bar{X} = [x \ \mathrm{y} \ \mathrm{z}]^T$$



Transformations of points in 3D



Right handed coordinate system

4 basic transformations

- **Translation**
- **Rotation**
- **Scaling**
- **Shear**

$\Big\}$ **Affine transformations**

Basics 3D Transformation equations

- Translation : P' = P + ΔP

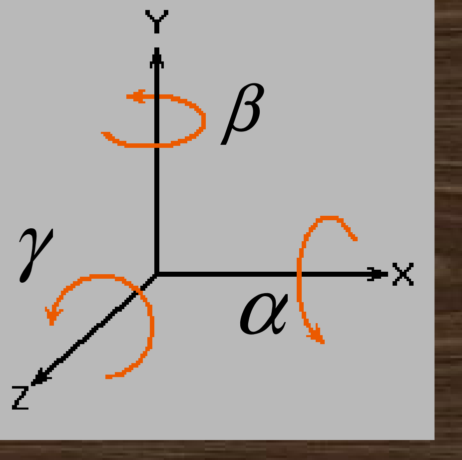
$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix}$$

- Scaling: P'= SP

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

- Rotation : about an axis,

  P' = RP

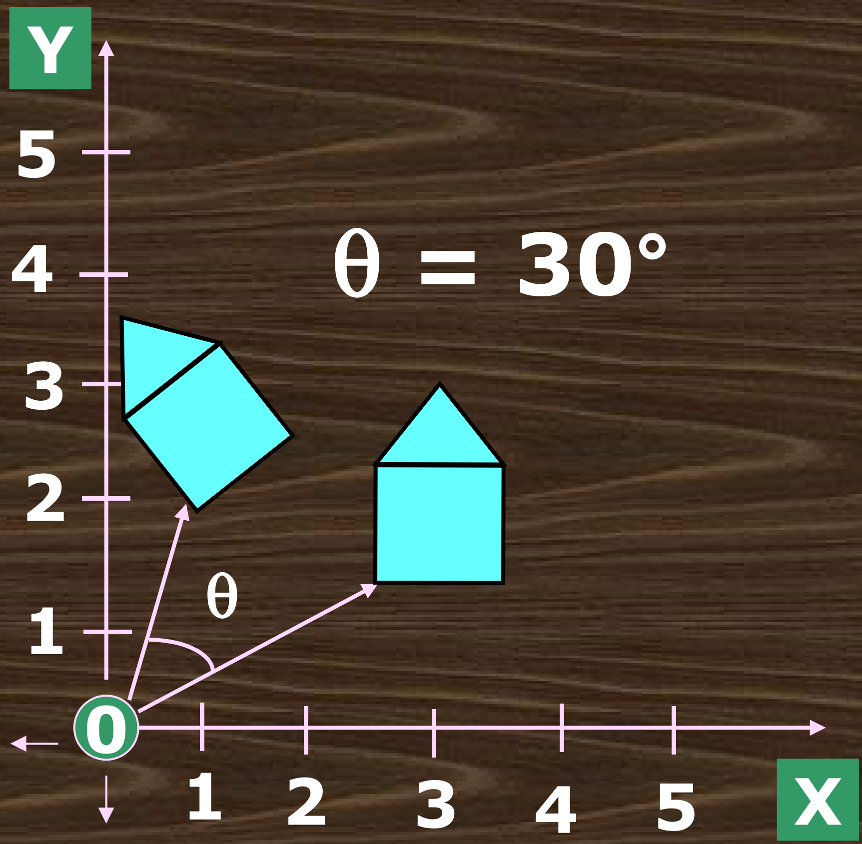


ROTATION - 2D

$$x' = x\cos(\theta) - y\sin(\theta)$$
$$y' = x\sin(\theta) + y\cos(\theta)$$

In matrix form, this is :

$$R = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$



Positive Rotations: counter clockwise about the origin

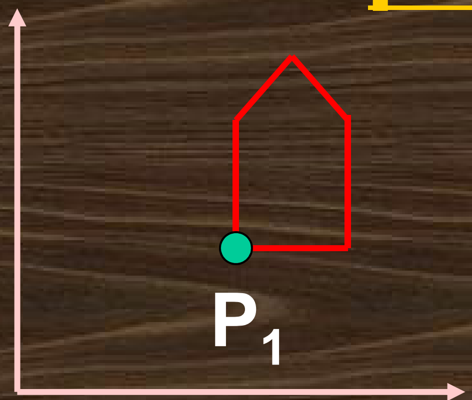
For rotations, $|R| = 1$ and $[R]^T = [R]^{-1}$. Rotation matrices are orthogonal.

Rotation about an arbitrary point P in space

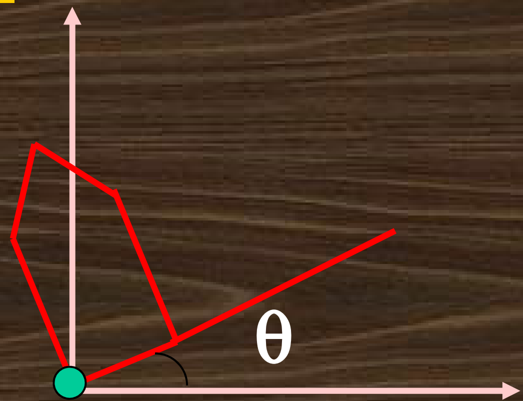
As we mentioned before, rotations are applied about the origin. So to rotate about any arbitrary point P in space, translate so that P coincides with the origin, then rotate, then translate back. Steps are:

- Translate by $(-P_x, -P_y)$
- Rotate
- Translate by (P_x, P_y)

Rotation about an arbitrary point P in space

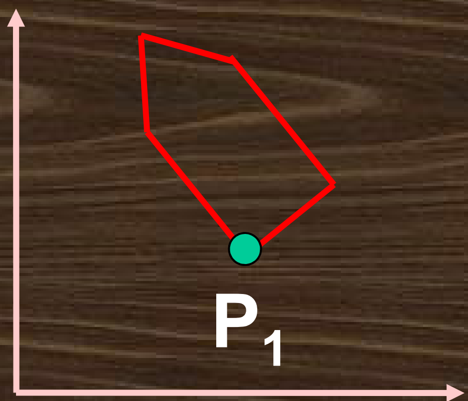


House at P_1



Rotation by θ

Translation of P_1 to Origin



Translation back to P_1

2D Transformation equations (revisited)

- Translation : P' = P + ΔP

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \Rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \end{bmatrix}^T \begin{bmatrix} x \\ y \end{bmatrix} ??$$

- Rotation : about an axis,

  P' = RP

$$\boldsymbol{R} = \begin{bmatrix} \mathbf{cos}(\boldsymbol{\theta}) & \mathbf{-sin}(\boldsymbol{\theta}) \\ \mathbf{sin}(\boldsymbol{\theta}) & \mathbf{cos}(\boldsymbol{\theta}) \end{bmatrix}$$

$$\begin{bmatrix} x'' \\ y'' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix}$$

Rotation about an arbitrary point P in space

$R_{gen} = T_1(-P_x, -P_y) * R_2(\theta) * T_3(P_x, P_y)$

$$= \begin{bmatrix} 1 & 0 & -P_x \\ 0 & 1 & -P_y \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & P_x \\ 0 & 1 & P_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos(\theta) & -\sin(\theta) & P_x*(\cos(\theta)-1)-P_y*(\sin(\theta) \\ \sin(\theta) & \cos(\theta) & P_y*(\cos(\theta)-1)+P_x*\sin(\theta) \\ 0 & 0 & 1 \end{bmatrix}$$

Using Homogeneous system

Homogeneous representation of a point in 3D space:

$$P = | \, x \; y \; z \; w \, |^{T}$$

$$(w = 1, \text{for a 3D point})$$

Transformations will thus be represented by 4x4 matrices:

P′ = A.P

Homogenous Coordinate systems

- In order to Apply a sequence of transformations to produce composite transformations we introduce the fourth coordinate
- Homogeneous representation of 3D point:

$|x\ y\ z\ h|^T$ (h=1 for a 3D point, dummy coordinate)

- Transformations will be represented by 4x4 matrices.

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

Homogenous Translation matrix

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

Homogenous Scaling matrix

$$R_\alpha = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation about x axis by angle α

$$R_\beta = \begin{bmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation about y axis by angle β

Change of sign?

$$R_\gamma = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 & 0 \\ \sin\gamma & \cos\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation about z axis by angle γ

How can one do a Rotation about an arbitrary Axis in Space?

3D Transformation equations (3)

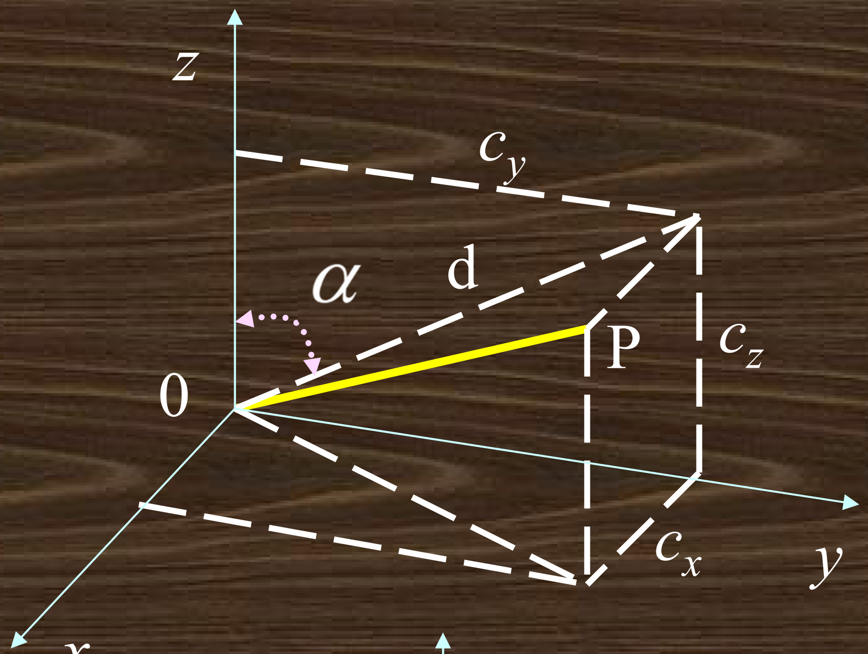
Rotation About an Arbitrary Axis in Space

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

1) **First of all, translate by: $-(x_0, y_0, z_0) = |T|$.**
2) **Next, we rotate the axis into one of the principle axes. Let's pick, Z ($|R_x|$, $|R_y|$).**
3) **We rotate next by θ degrees in Z ($|R_z(\theta)|$).**
4) **Then we undo the rotations to align the axis.**
5) **We undo the translation: translate by (x_0, y_0, z_0)**

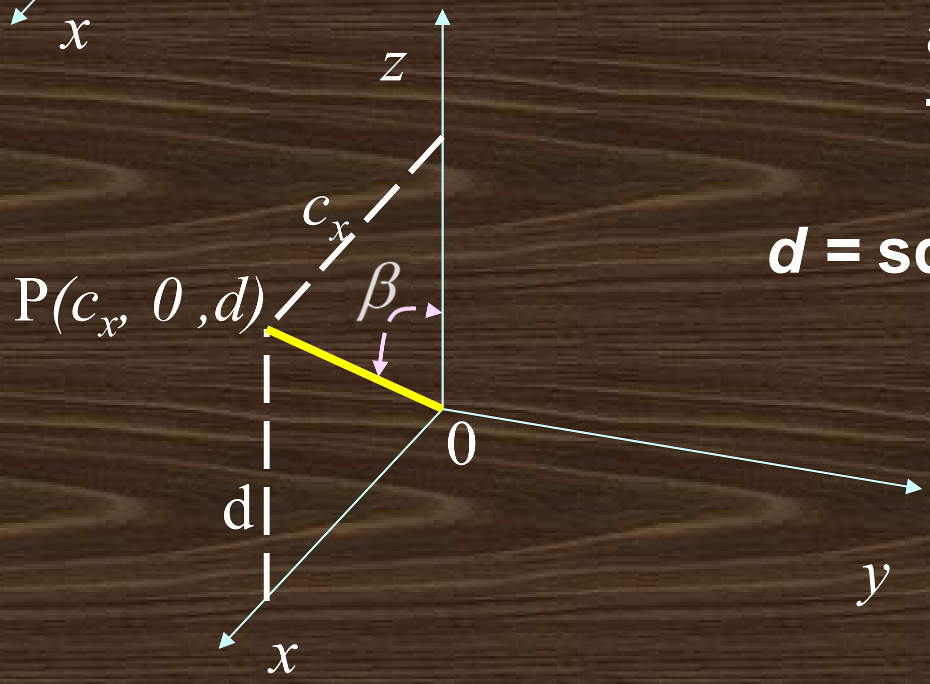
T*he tricky part is (2) above.*

This is going to take 2 rotations,
i) about x (to place the axis in the x-z plane)
and
ii) about y (to place the result coincident with the z axis).



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

Project the unit vector, along OP, into the y-z plane. The y and z components are c_y and c_z, the directions cosines of the unit vector along the arbitrary axis. It can be seen from the diagram above, that :



$$d = \mathrm{sqrt}(c_y^2 + c_z^2), \qquad \cos(\alpha) = c_z / d$$

$$\sin(\alpha) = c_y / d$$

Rotation by β about y:
How do we determine β?
Similar to above:

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 the unit vector)

$\sin(\beta) = c_x = c_x$ /(length of the unit vector).

Final Transformation:

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

If you are given 2 points instead, you can calculate the direction cosines as follows:

$$V = |\ (x_1 - x_0) \ \ (y_1 - y_0) \ \ (z_1 - z_0)\ |^T$$

$c_x = (x_1 - x_0) / |V|$

$c_y = (y_1 - y_0) / |V|$

$c_z = (z_1 - z_0) / |V|$,

where |V| is the length of the vector V.

Inverse transformations

$$T^{-1} = \begin{bmatrix} 1 & 0 & 0 & -\Delta x \\ 0 & 1 & 0 & -\Delta y \\ 0 & 0 & 1 & -\Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Inverse Translation

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

Inverse scaling

Inverse Rotation

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha & 0 \\ 0 & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} \cos\beta & 0 & -\sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} \cos\gamma & \sin\gamma & 0 & 0 \\ -\sin\gamma & \cos\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

R_α^{-1} $\qquad$ R_β^{-1} $\qquad$ R_γ^{-1}

Concatenation of transformations

- The 4 X 4 representation is used to perform a sequence of transformations.
- Thus application of several transformations in a particular sequence can be presented by a single transformation matrix

$$v^* = R_\theta(S(Tv)) = Av; \ \mathrm{A} = \mathrm{R}_\theta.S.T$$

- The order of application is important… the multiplication may not be commutable.

Commutivity of Transformations

If we scale, then translate to the origin, and then translate back, is that equivalent to translate to origin, scale, translate back?

When is the order of matrix multiplication unimportant?

When does $T_1 * T_2 = T_2 * T_1$?

Cases where $T_1 * T_2 = T_2 * T_1$:

T_1	T_2
translation	translation
scale	scale
rotation	rotation
Scale (uniform)	rotation

COMPOSITE TRANSFORMATIONS

If we want to apply a series of transformations T_1, T_2, T_3 to a set of points, We can do it in two ways:

1) We can calculate $p'=T_1*p$, $p''= T_2*p'$, $p'''=T_3*p''$
2) Calculate $T= T_1*T_2*T_3$, then $p'''= T*p$.

Method 2, saves large number of additions and multiplications (computational time) – needs approximately 1/3 of as many operations. Therefore, we concatenate or compose the matrices into one final transformation matrix, and then apply that to the points.

Spaces

Object Space

definition of objects. Also called Modeling space.

World Space

where the scene and viewing specification is made

Eye space (Normalized Viewing Space)

where eye point (COP) is at the origin looking down the Z axis.

3D Image Space

A 3D Perspected space.
Dimensions: -1:1 in x & y, 0:1 in Z.
Where Image space hidden surface algorithms work.

Screen Space (2D)

Coordinates 0:width, 0:height

Projections

We will look at several planar geometric 3D to 2D projection:

-Parallel Projections
  Orthographic
  Oblique

-Perspective

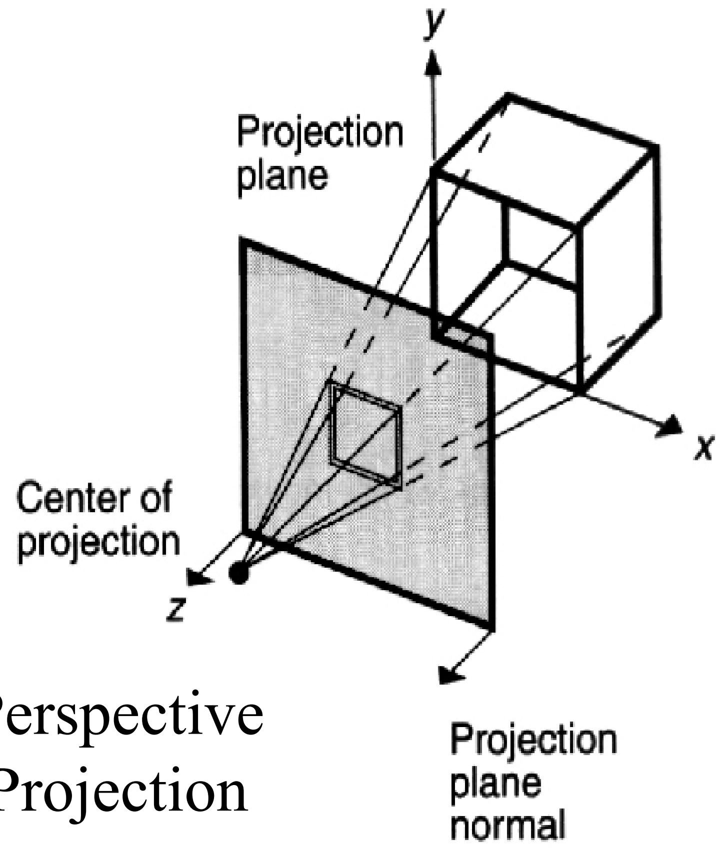
Projection of a 3D object is defined by straight projection rays (projectors) emanating from the center of projection (COP) passing through each point of the object and intersecting the projection plane.

Perspective Projections

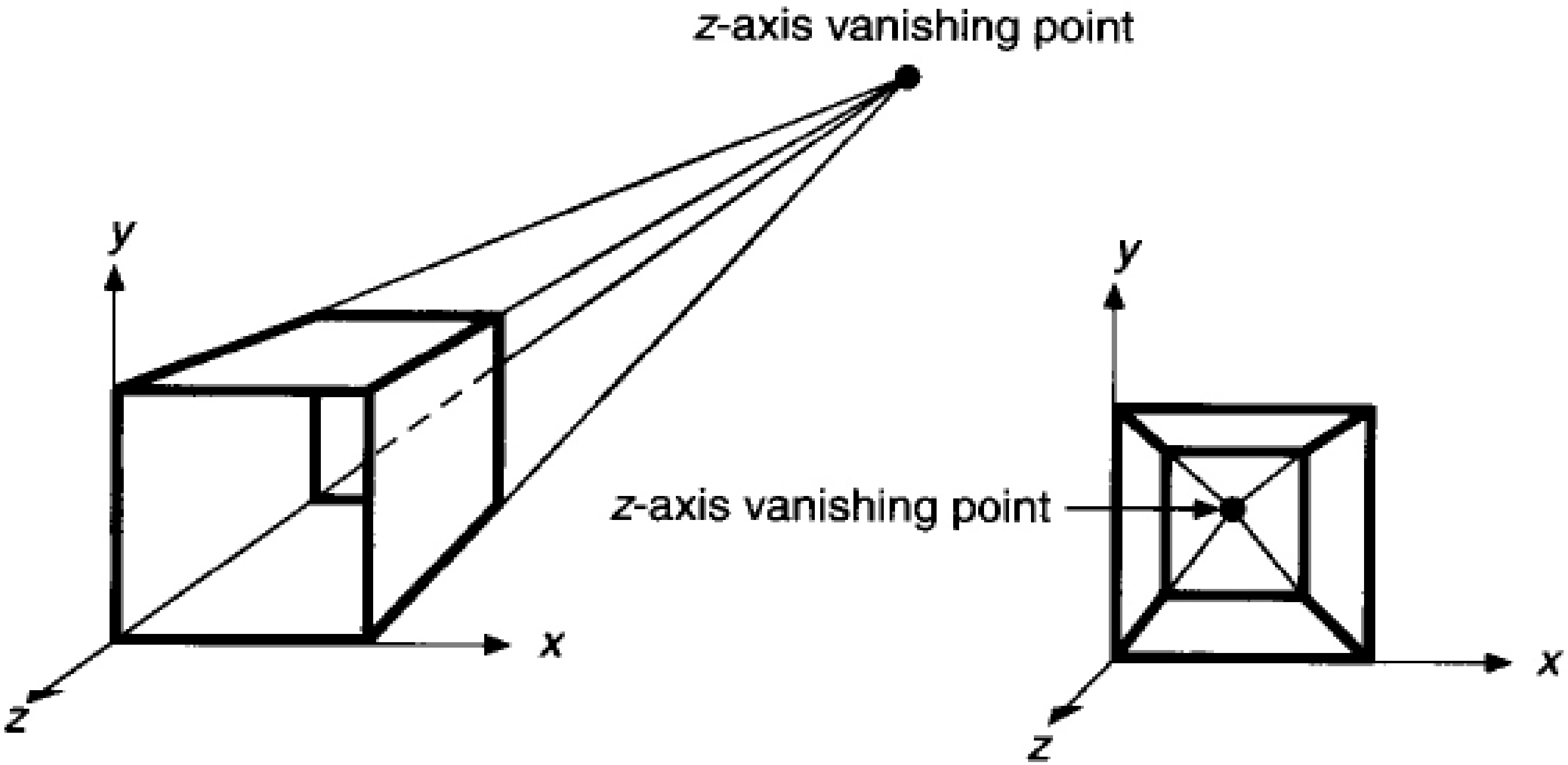
Distance from COP to projection plane is finite. The projectors are not parallel & we specify a center of projection.

Center of Projection is also called the Perspective Reference Point

COP = PRP

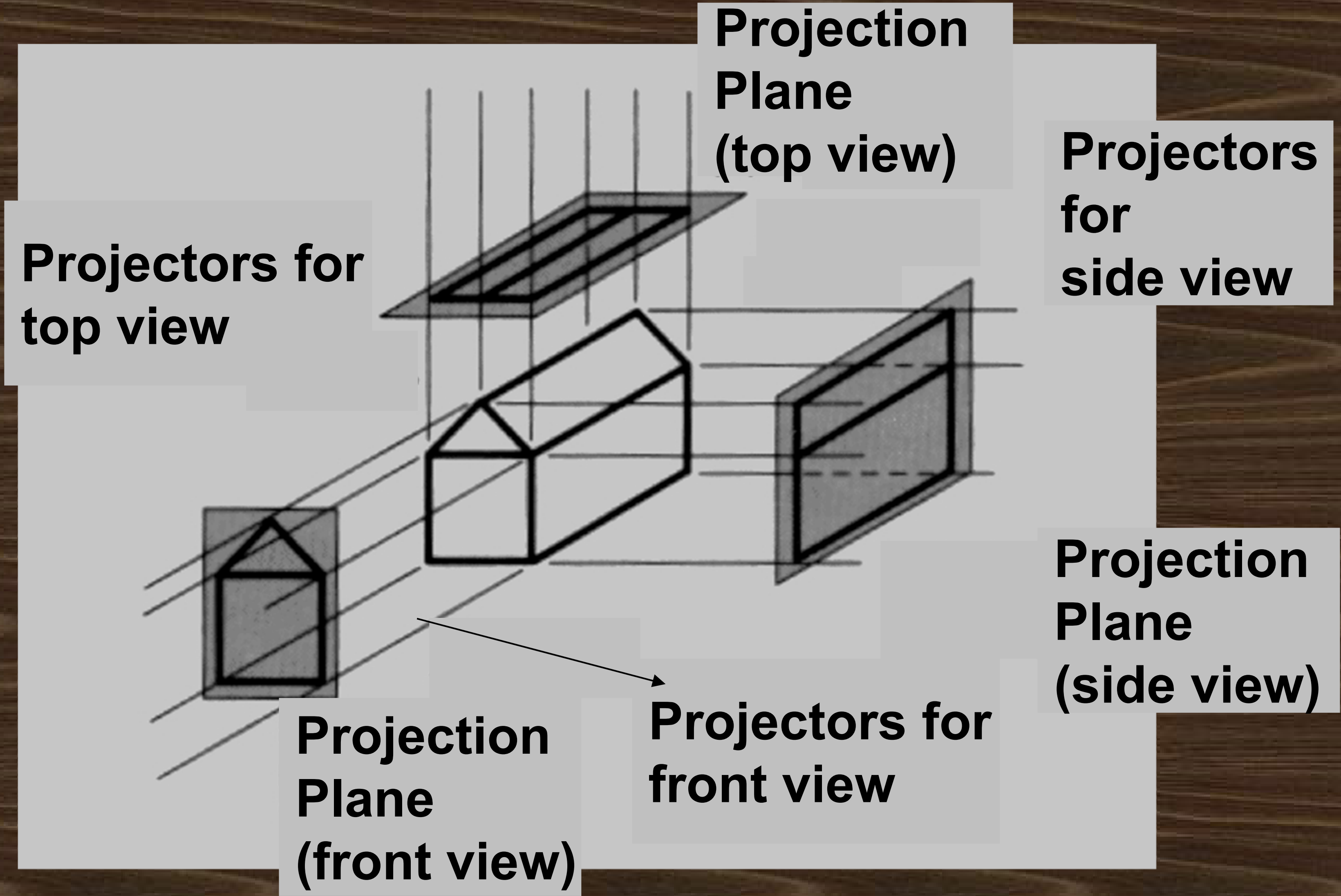


Perspective Projection



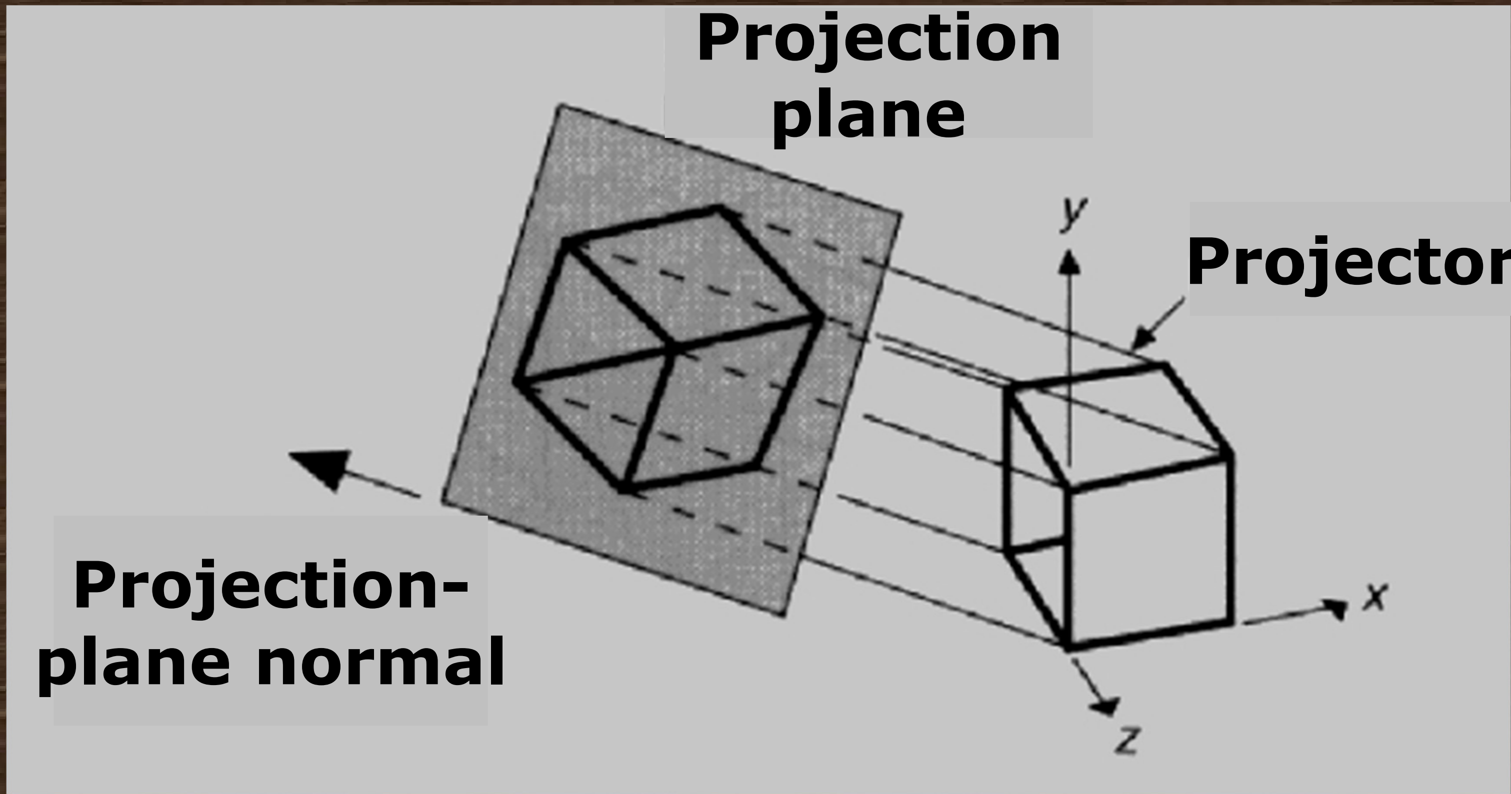
- **Perspective foreshortening:** the size of the perspective projection of the object varies inversely with the distance of the object from the center of projection.

- **Vanishing Point:** The perspective projections of any set of parallel lines that are not parallel to the projection plane converge to a vanishing point.



Example of Orthographic Projection

Example of Isometric Projection:



Example Oblique Projection