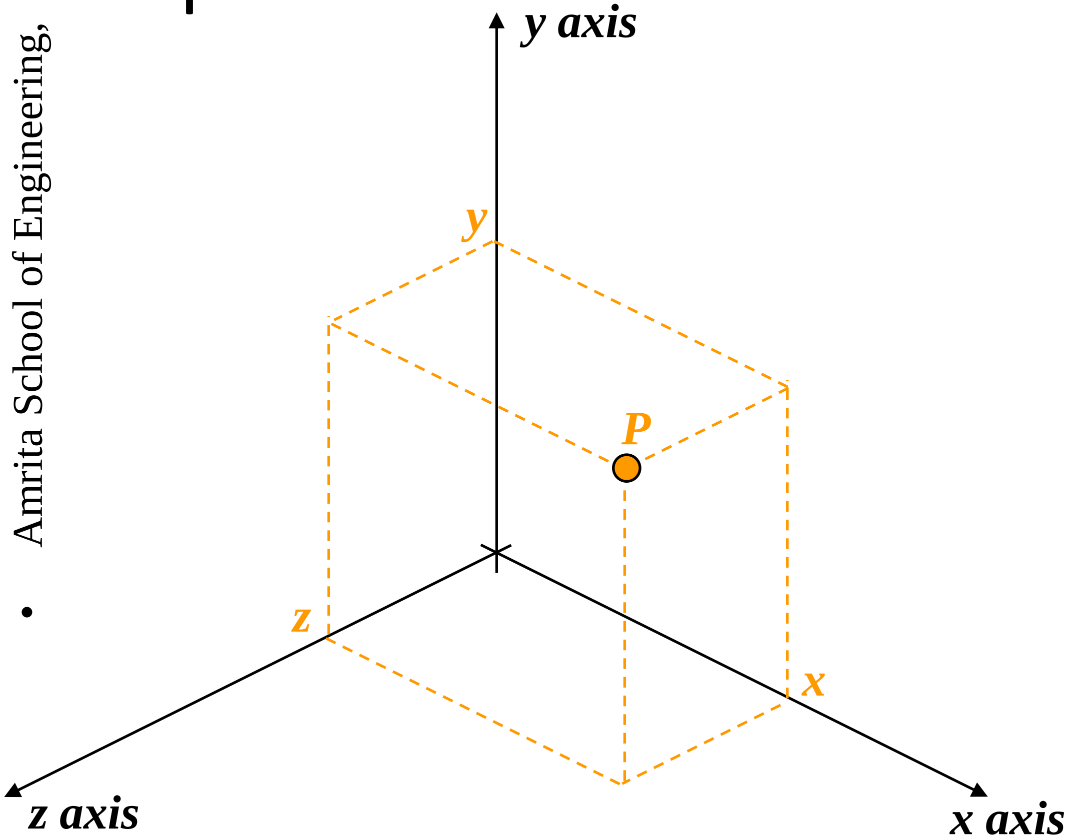
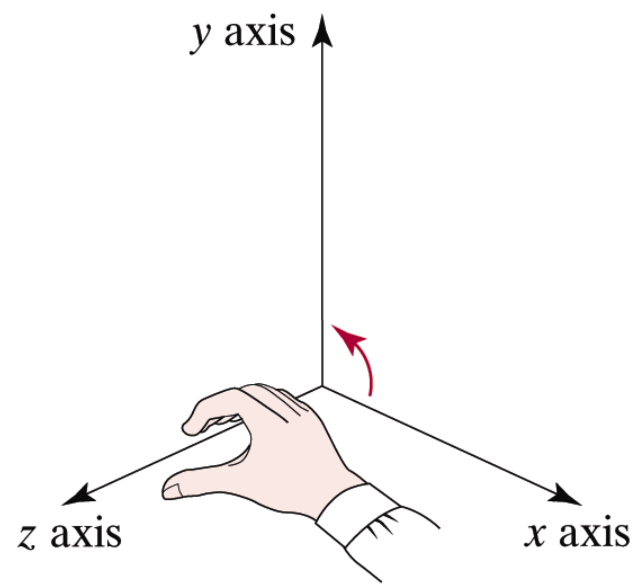
Contents

- In today's lecture we are going to have a look at:
  - Transformations in 3-D
    - How do transformations in 3-D work?
    - 3-D homogeneous coordinates and matrix based transformations
  - Projections
    - History
    - Geometrical Constructions
    - Types of Projection
    - Projection in Computer Graphics

3-D Coordinate Spaces

- Remember what we mean by a 3-D coordinate space

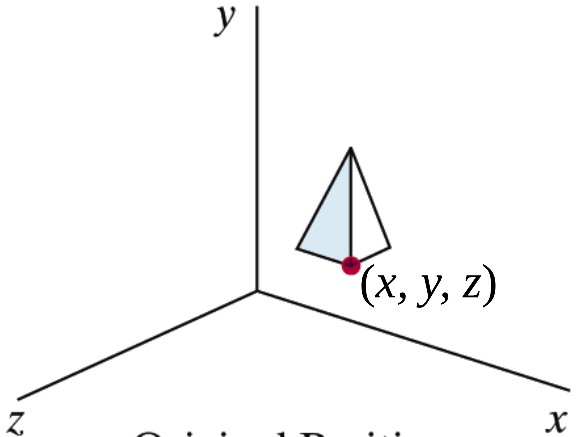




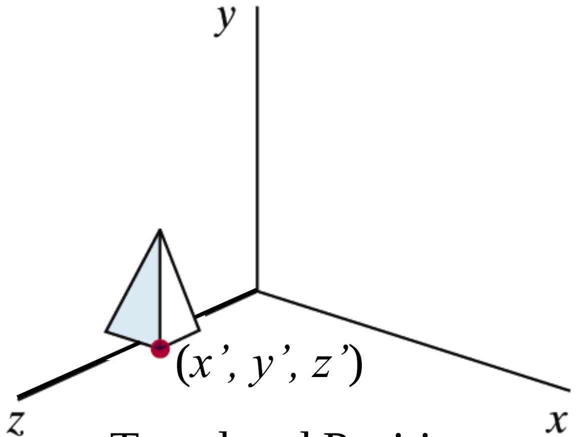
Right-Hand Reference System

Translations In 3-D

- To translate a point in three dimensions by tx, ty and tz simply calculate the new points as follows:
- $x' = x + tx \qquad y' = y + ty \qquad z' = z + tz$



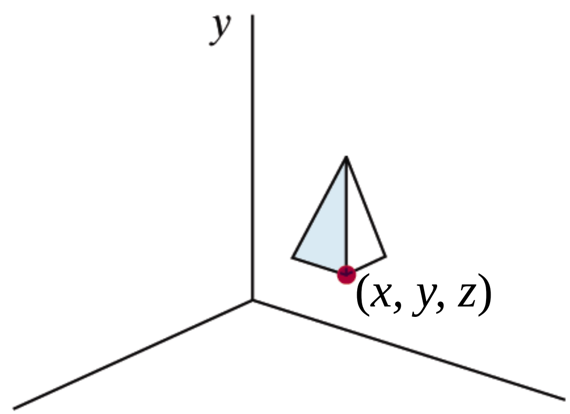
Original Position



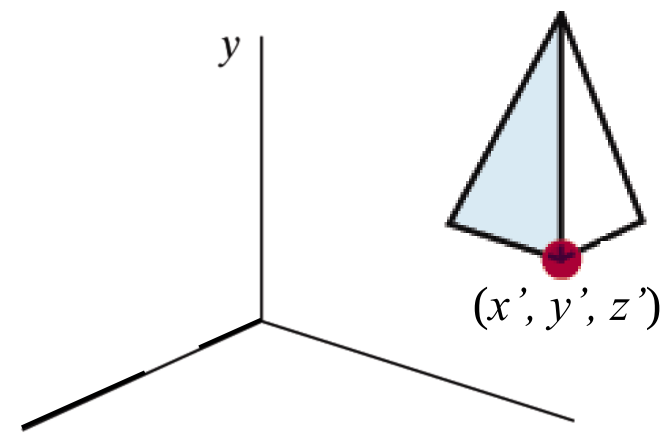
Translated Position

Scaling In 3-D

- To sale a point in three dimensions by *sx*, *sy* and *sz* simply calculate the new points as follows:
- $x' = sx*x$ $\quad$ $y' = sy*y$ $\quad$ $z' = sz*z$



Original Position



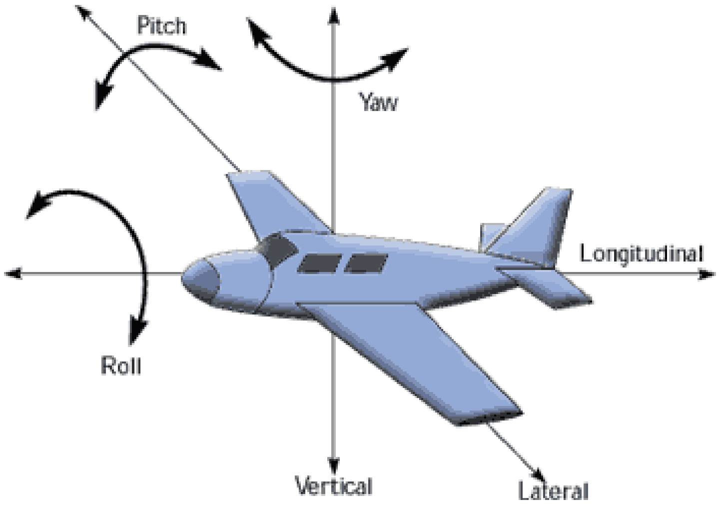
Scaled Position

Rotations In 3-D

- When we performed rotations in two dimensions we only had the choice of rotating about the z axis

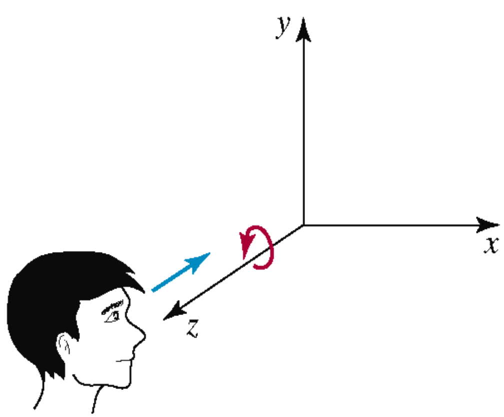
In the case of three dimensions we have more options

- Rotate about x – pitch
- Rotate about y – yaw
- Rotate about z - roll

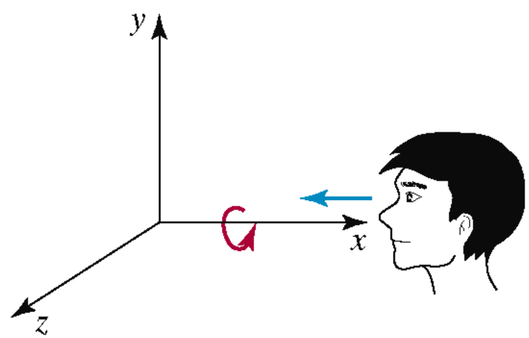


Rotations In 3-D (cont…)

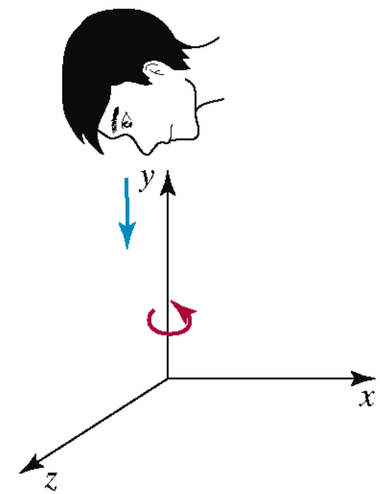
- The equations for the three kinds of rotations in 3-D are as follows:



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



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

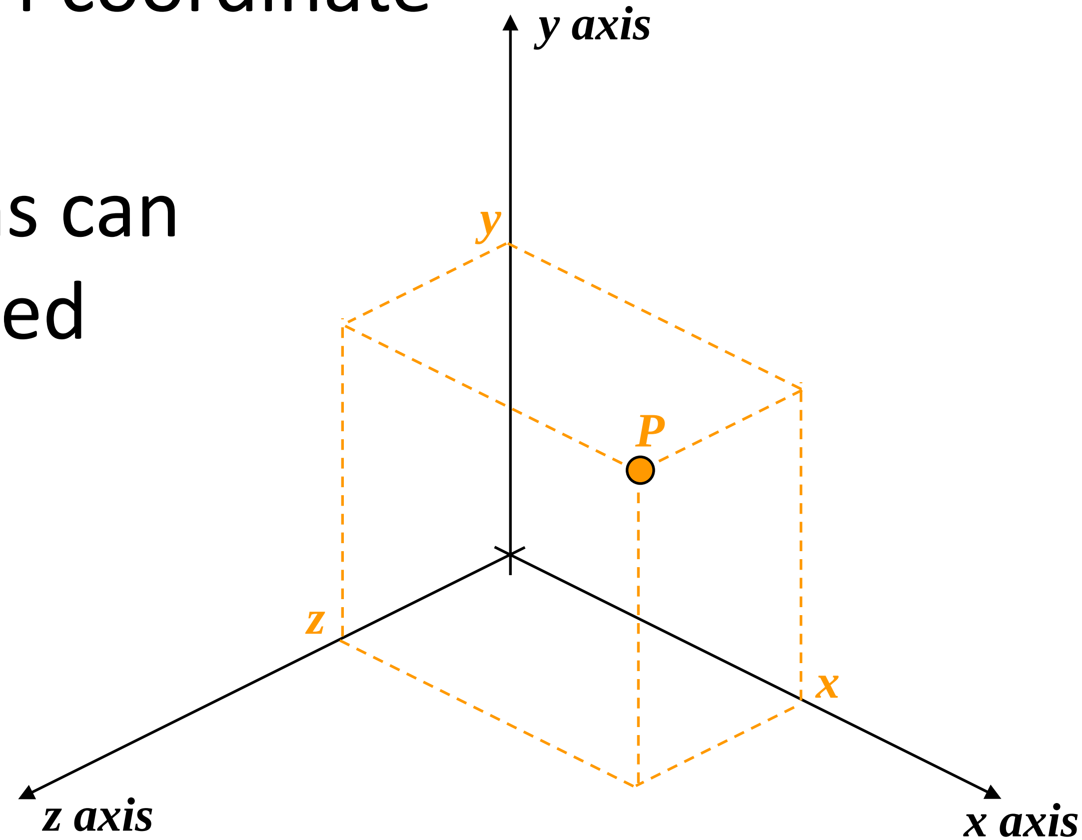


$$x' = z \cdot \sin\theta + x \cdot \cos\theta$$
$$y' = y$$
$$z' = z \cdot \cos\theta - x \cdot \sin\theta$$

Homogeneous Coordinates In 3-D

- Similar to the 2-D situation we can use homogeneous coordinates for 3-D transformations - 4 coordinate column vector
- All transformations can then be represented as matrices

$$P(x, y, z) = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$



3D Transformation Matrices

Translation by tx, ty, tz

$$\begin{bmatrix} 1 & 0 & 0 & tx \\ 0 & 1 & 0 & ty \\ 0 & 0 & 1 & tz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

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

Scaling by sx, sy, sz

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

Rotate About X-Axis

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

Rotate About Y-Axis

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

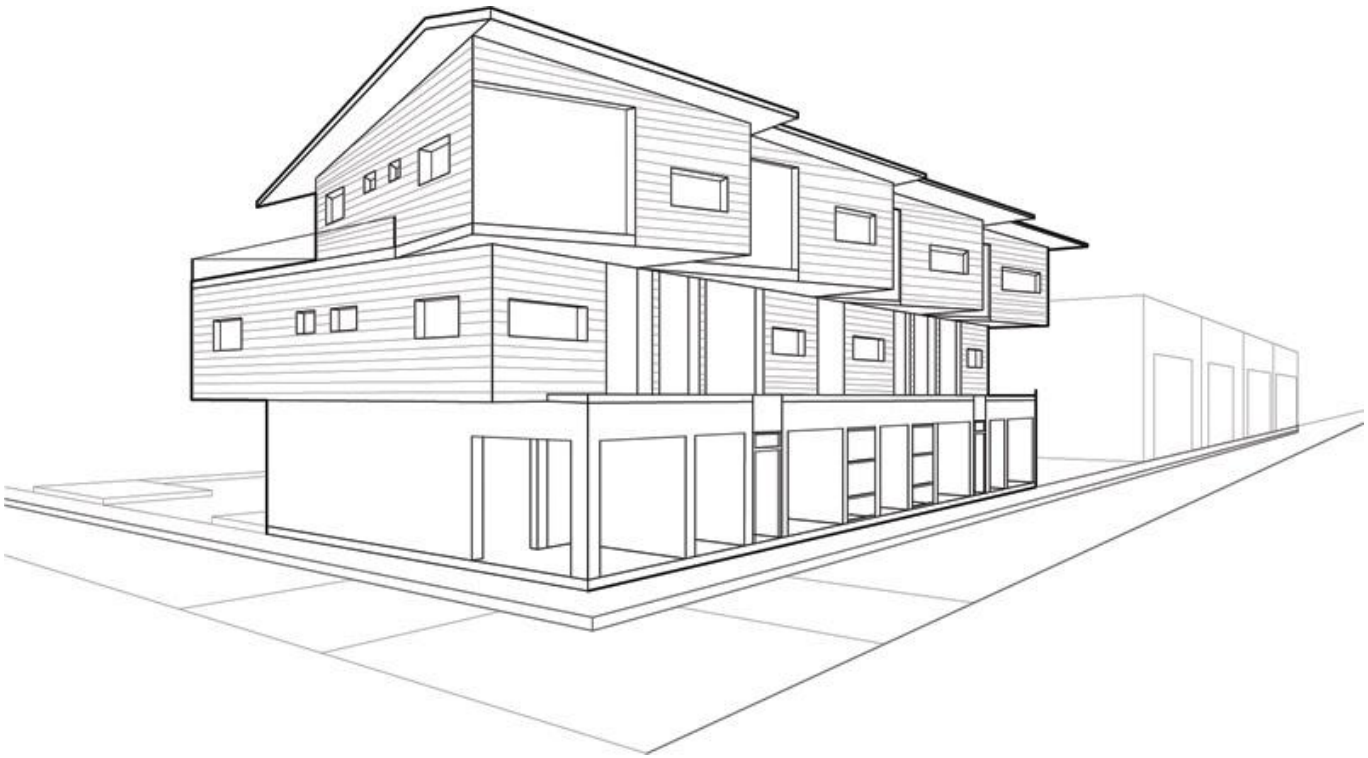
Rotate About Z-Axis

Computer Graphics : Viewing in 3-D

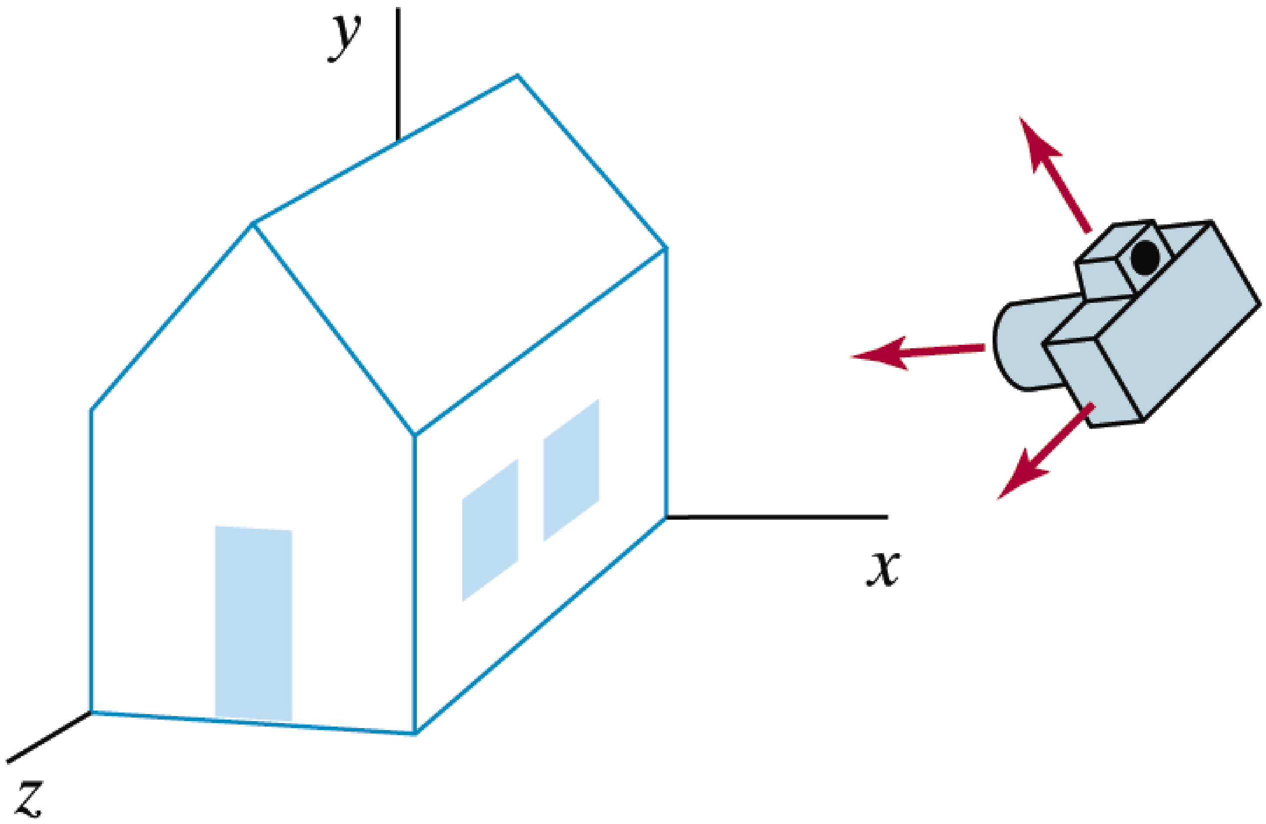
Chapter 12

•

Viewing in 3D



Remember The Big Idea



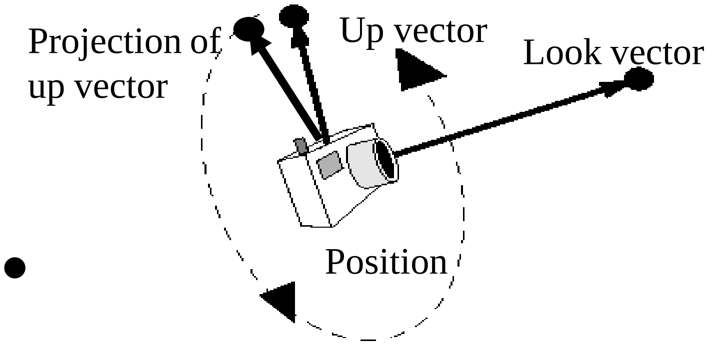
3D Transformation Pipeline

- Modeling Coordinates
- Modeling Transformation
  - World coordinates
- Viewing Transformation
  - Viewing coordinates
- Projection Transformation
  - Projection coordinates
- Workstation Transformation
  - Device coordinates

Viewing Coordinates

- Specifying the view plane
  - Establish the viewing-coordinate system/view reference coordinate system
  - Pick up a world coordinate position called the view reference point
  - This point is the origin of the viewing-coordinate system
  - The view reference point is often chosen to be close to or on the surface of some object
  - View plane/projection plane is then set up perpendicular to the viewing z axis

The Up And Look Vectors



- The *look vector* indicates the direction in which the camera is pointing
- The *up vector* determines how the camera is rotated
- For example, is the camera held vertically or horizontally

Viewing coordinate system

- View-plane normal vector N
- look-at point
- positive direction for Zv axis
- View-up vector V
- positive direction for Yv axis
- twist angle θ about the Zv axis
- V can be choosen in any direction except parallel to N
- Vector U
- using V and N , U vector perpendicular to V and N is defined in the direction of Xv axis

World to Viewing coordinates

- Translate the view reference point to the origin of the world-coordinate system

- Apply rotations to align the Xv, Yv and Zv axes with the world Xw, Yw and Zw axes
- - based on the direction of N we may require maximum of three rotation

Projections

- Display device (a screen) is 2D…
  - How do we map 3D objects to 2D space?
- 2D to 2D is straight forward…
  - 2D window to world.. and a viewport on the 2D surface.
  - Clip what won't be shown in the 2D window, and map the remainder to the viewport.
- 3D to 2D is more complicated…
  - Solution : Transform 3D objects on to a 2D plane using ***projections***

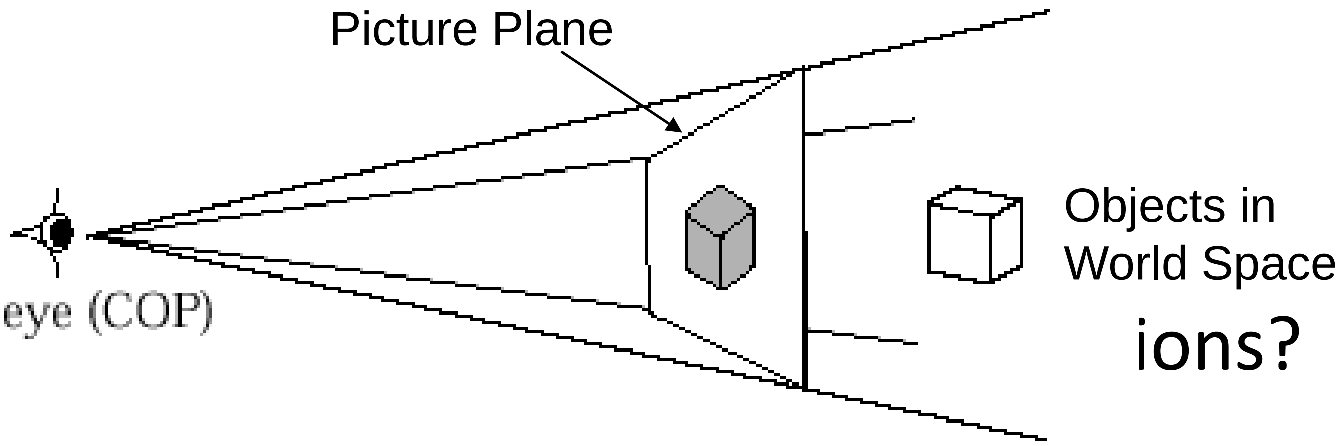
Projections

- In 3D…
  - View volume in the world
  - Projection onto the 2D projection plane
  - A viewport to the view surface
- Process…
  - 1… clip against the view volume,
  - 2… project to 2D plane, or window,
  - 3… map to viewport.

•

What Are Projections?

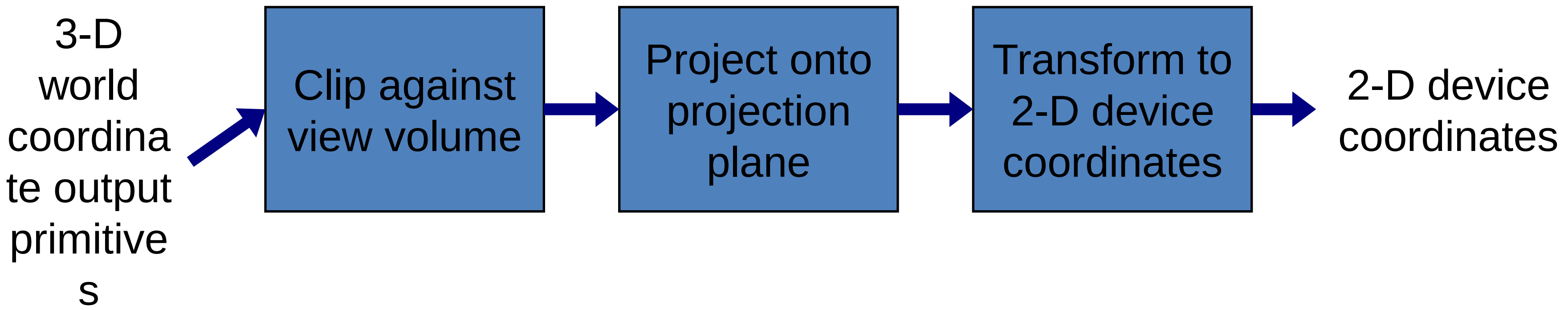
- Our 3-D scenes are all specified in 3-D world coordinates
- To display these we need to generate a 2-D image - *project* objects onto a *picture plane*



- So ions?

Converting From 3-D To 2-D

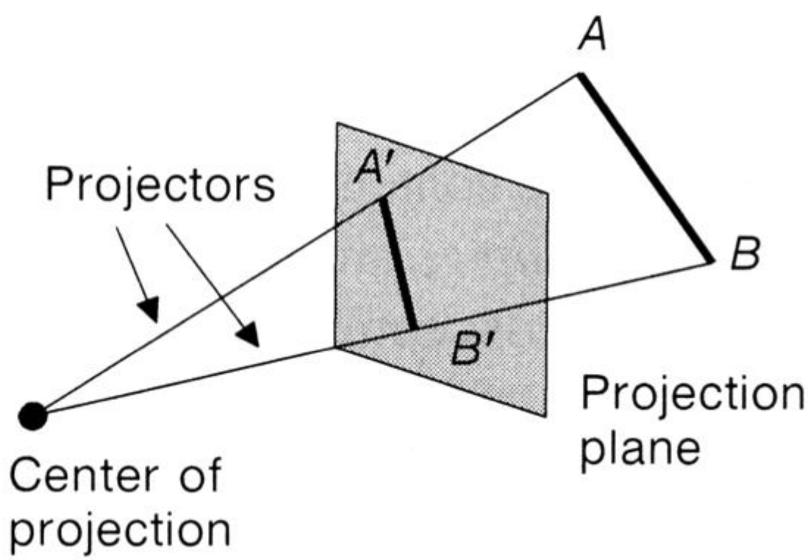
- Projection is just one part of the process of converting from 3-D world coordinates to a 2-D image



•

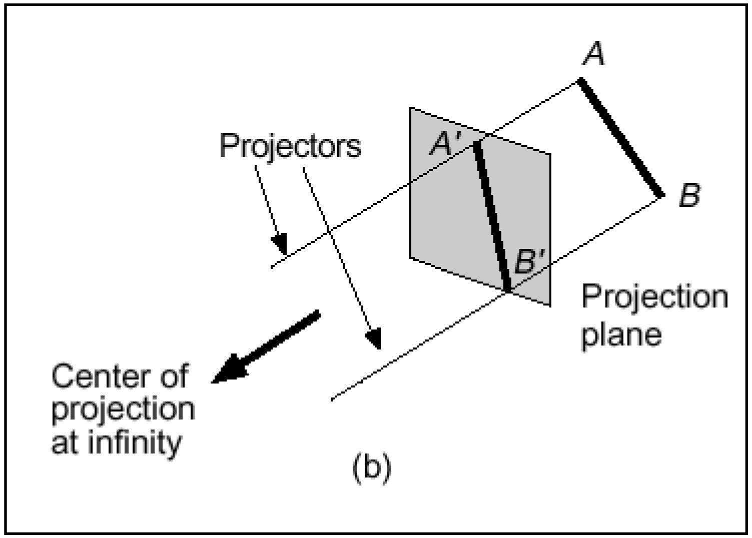
Projections

- Projections: key terms…
  - *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.

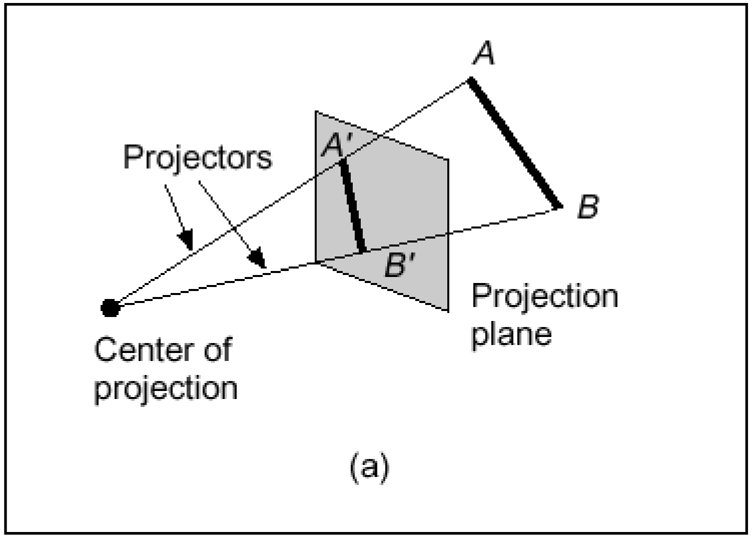


Types Of Projections

- There are two broad classes of projection:
  - Parallel: Typically used for architectural and engineering drawings
  - Perspective: Realistic looking and used in computer graphics



Parallel Projection



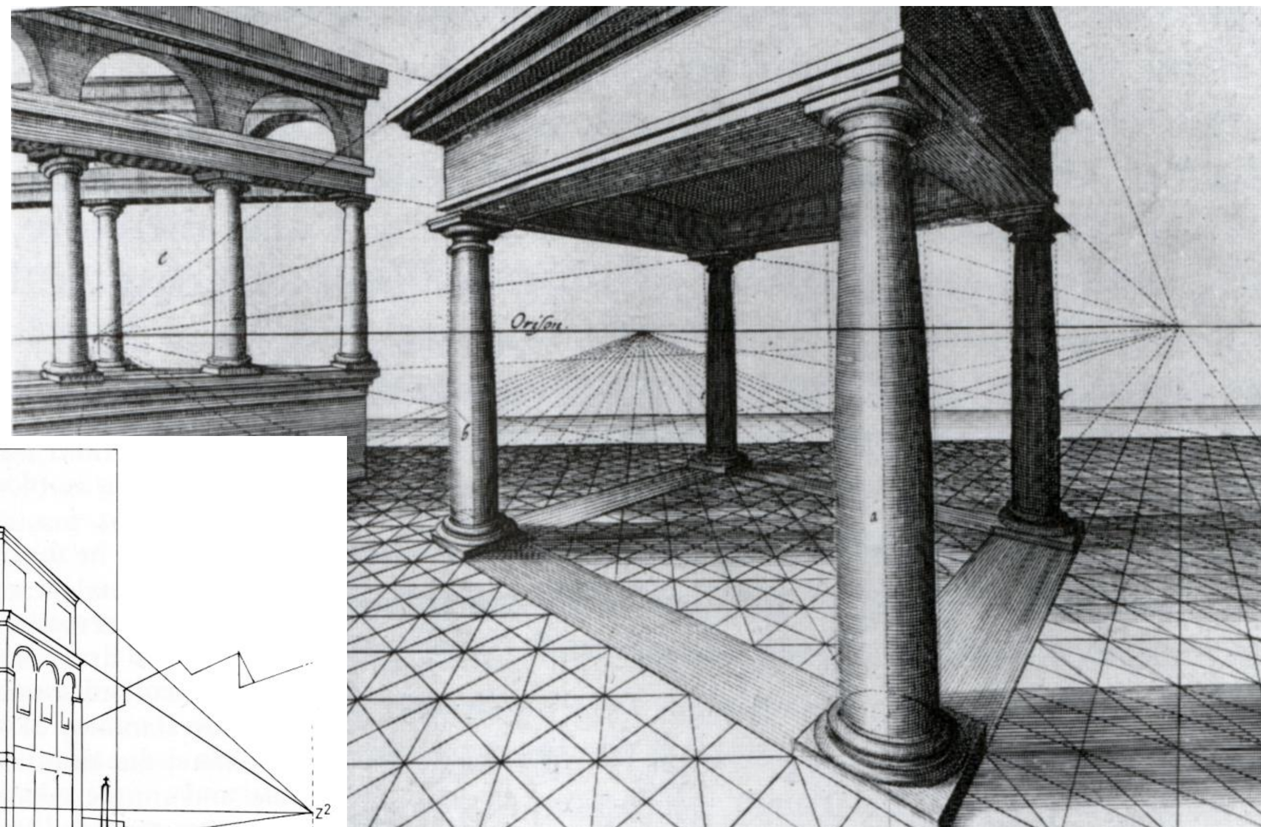
Perspective Projection

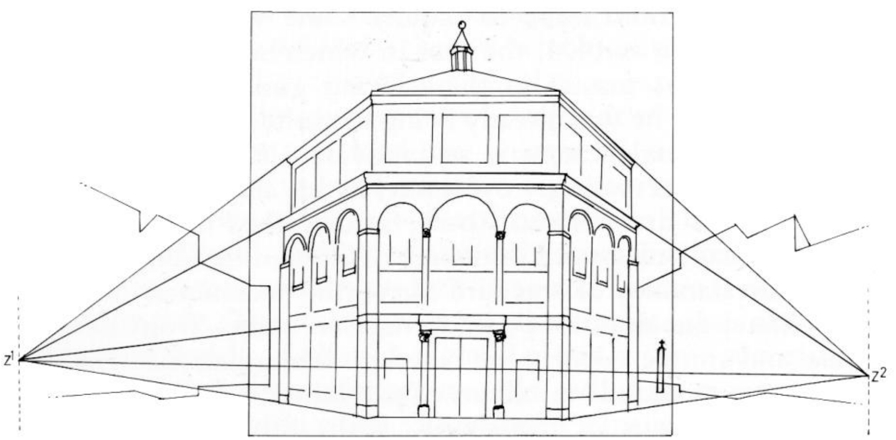
Perspective v Parallel

- Perspective:
  - visual effect is similar to human visual system...
  - has 'perspective foreshortening'
    - size of object varies inversely with distance from the center of projection.
  - angles only remain intact for faces parallel to projection plane.
- Parallel:
  - less realistic view because of no foreshortening
  - however, parallel lines remain parallel.
  - angles only remain intact for faces parallel to projection plane.

Perspective Projections

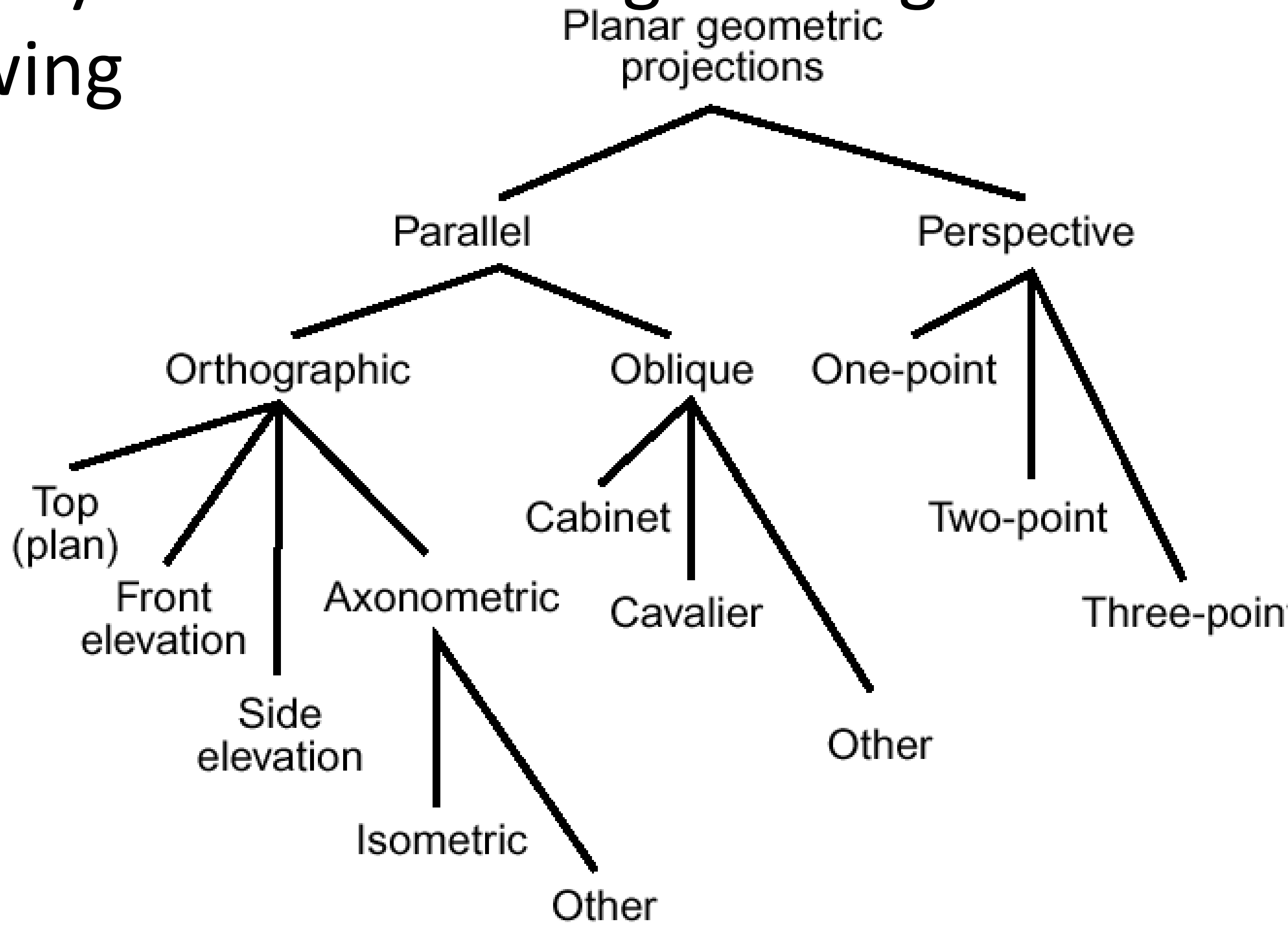
- Perspective projections are much more realistic than parallel projections





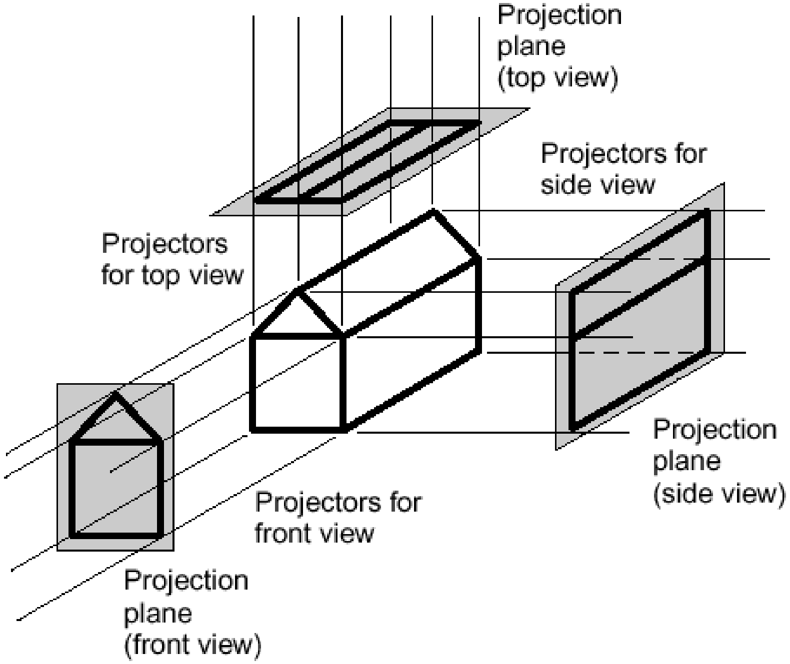
Types Of Projections (cont…)

- For anyone who did engineering or technical drawing

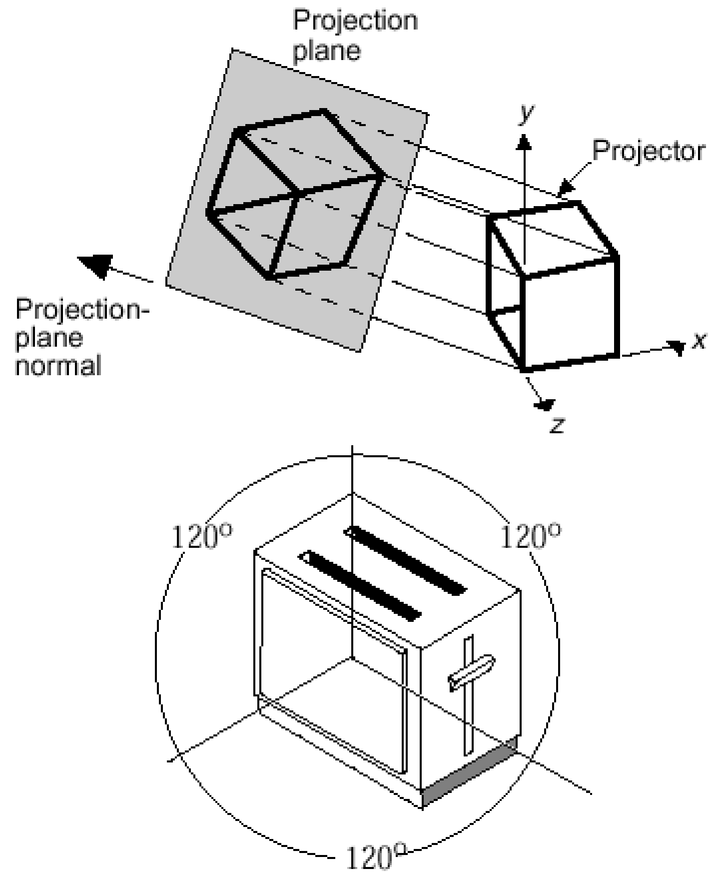


Parallel Projections

- Some examples of parallel projections



Orthographic Projection



Isometric Projection

Amrita School of Engineering, Bangalore

Isometric Projections

- Isometric projections have been used in computer games from the very early days of the industry up to today

Q*Bert **Sim City** **Virtual Magic Kingdom**

Perspective Projections

- Any parallel lines *not* parallel to the projection plane, converge at a vanishing point.
  - There are an infinite number of these, 1 for each of the infinite amount of directions line can be oriented.
- If a set of lines are parallel to one of the three principle axes, the vanishing point is called an *axis vanishing point*.
  - There are at most 3 such points, corresponding to the number of axes cut by the projection plane.

Perspective Projections

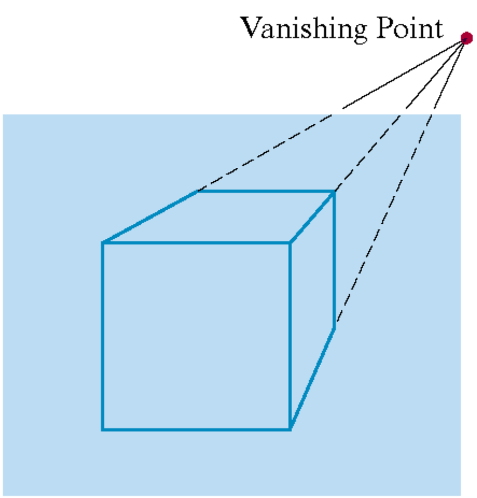
- Example:
  - if z projection plane cuts the z axis: normal to it, so only z has a principle vanishing point, as x and y are parallel and have none.

- Can categorise perspective projections by the number of principle vanishing points, and the number of axes the projection plane cuts.

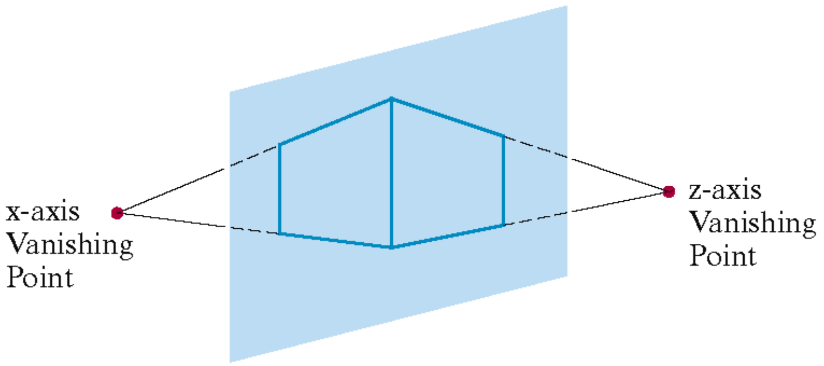
•

Perspective Projections

- There are a number of different kinds of perspective views
- The most common are one-point and two point perspectives



One Point Perspective Projection

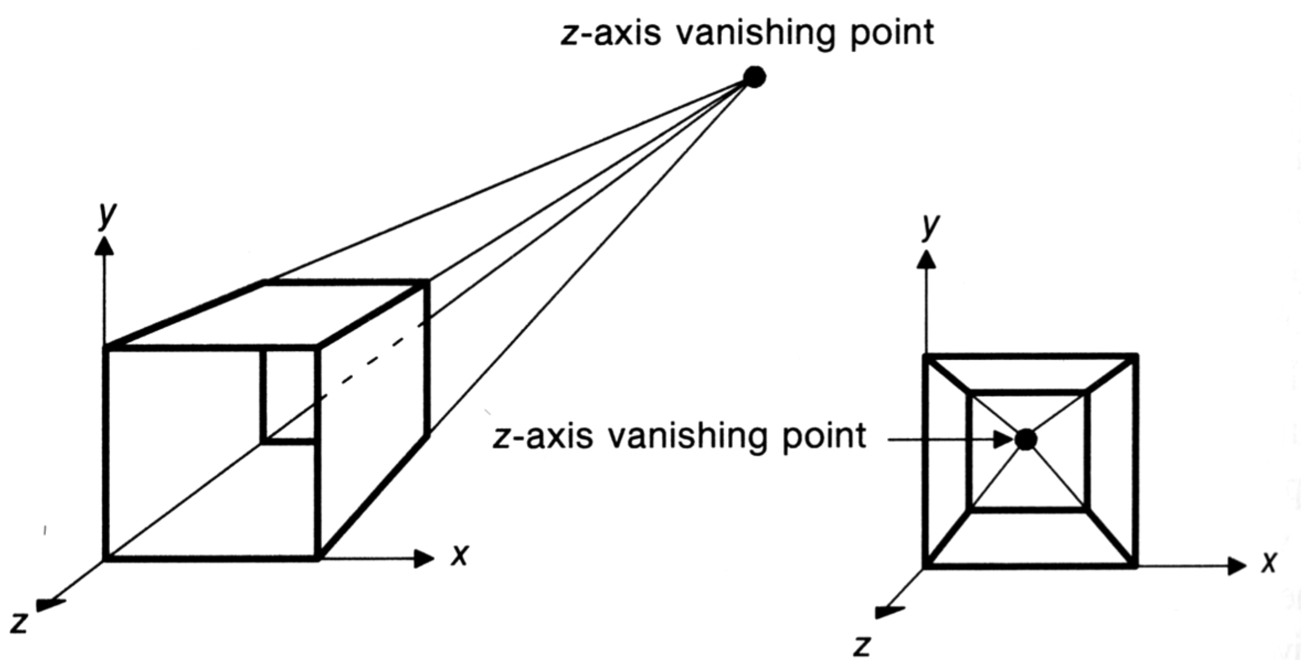


Two-Point Perspective Projection

Perspective Projections

- 2 different examples of a one-point perspective projection of a cube.

  (note: x and y parallel lines do not converge)



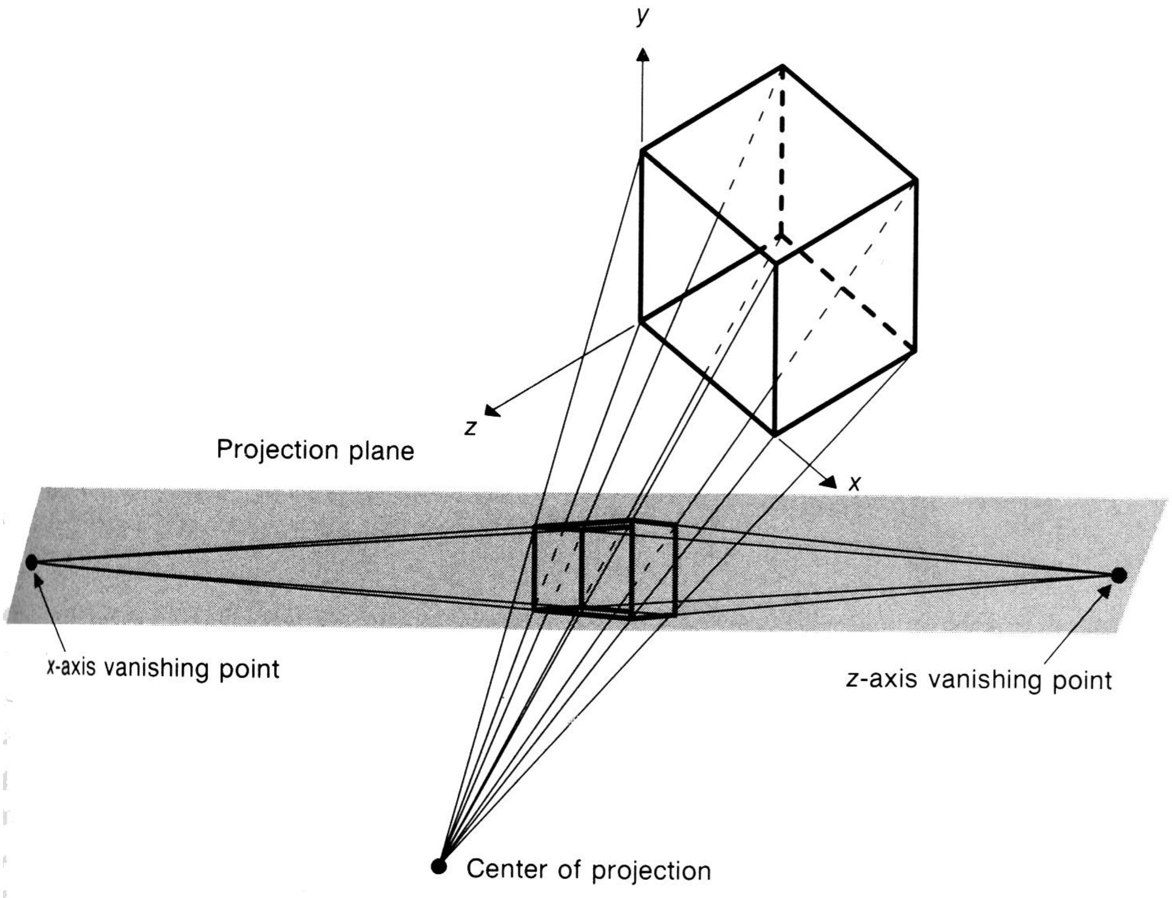
Perspective Projections

- Two-point perspective projection:
  - This is often used in architectural, engineering and industrial design drawings.
  - Three-point is used less frequently as it adds little extra realism to that offered by two-point perspective projection.

•

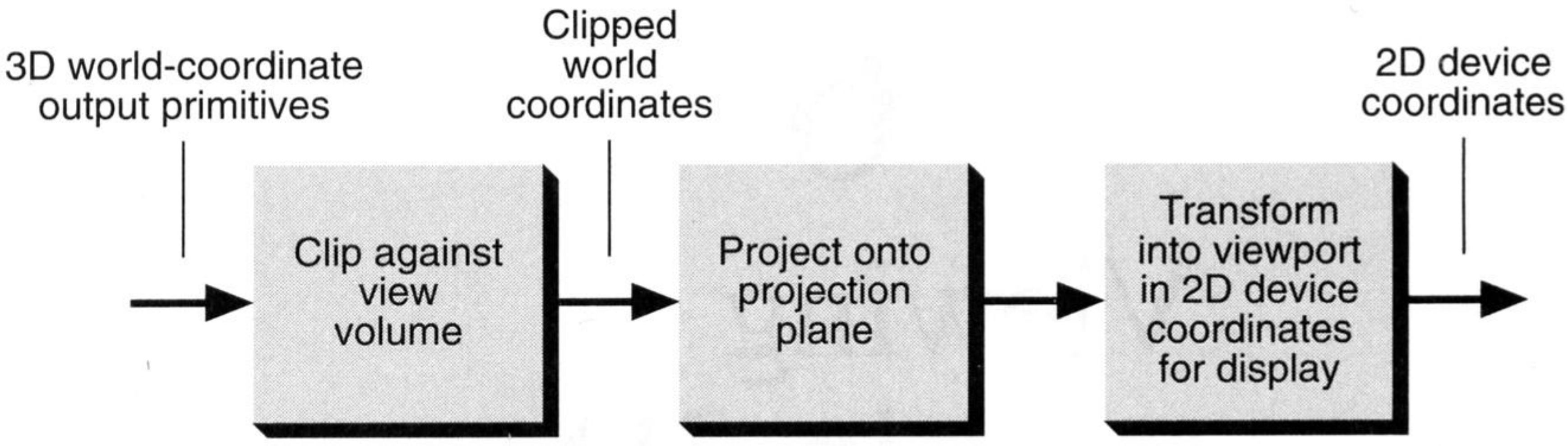
Perspective Projections

- Two-point perspective projection:



Projections

- Conceptual Model of the 3D viewing process



-

Parallel Projections

- 2 principle types:
  - *orthographic* and *oblique*.
- Orthographic :
  - direction of projection = normal to the projection plane.
- Oblique :
  - direction of projection != normal to the projection plane.
-

Parallel Projections

- Orthographic (or orthogonal) projections:
  - front elevation, top-elevation and side-elevation.
  - all have projection plane perpendicular to a principle axes.

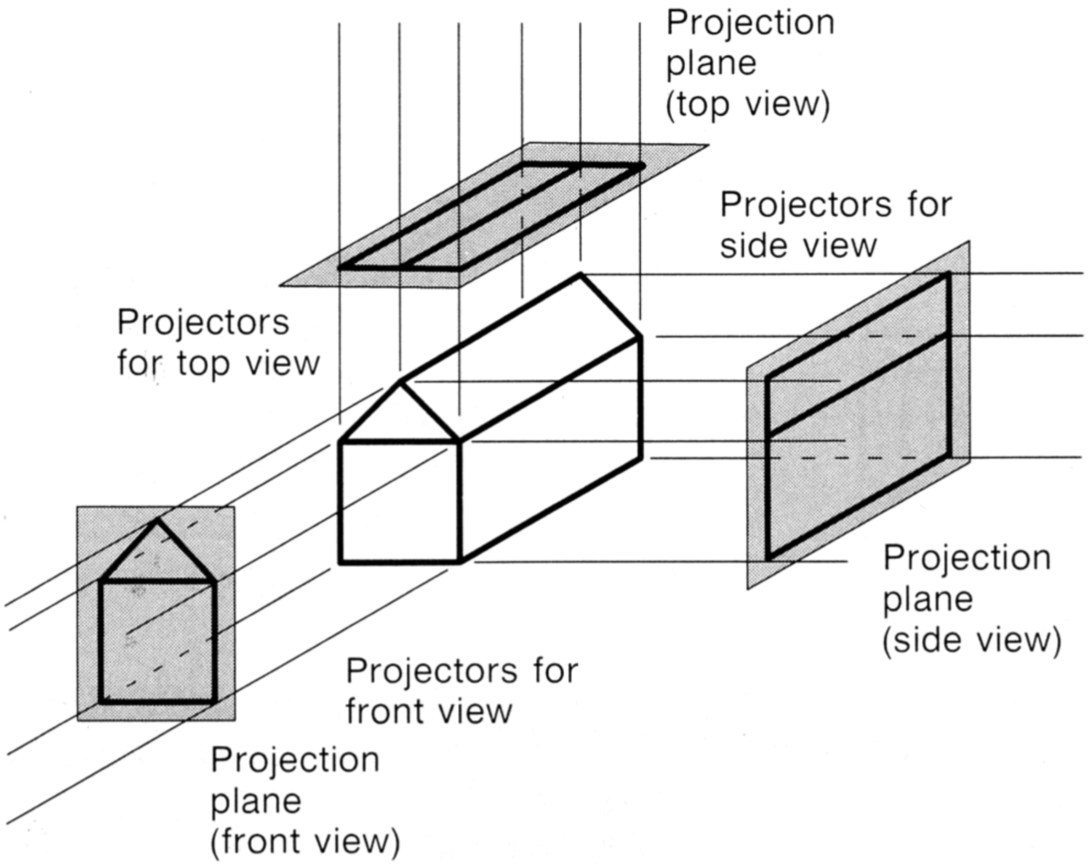
- Useful because angle and distance measurements can be made...

- However, As only one face of an object is shown, it can be hard to create a mental image of the object, even when several view are available.

-

Parallel Projections

- Orthogonal projections:

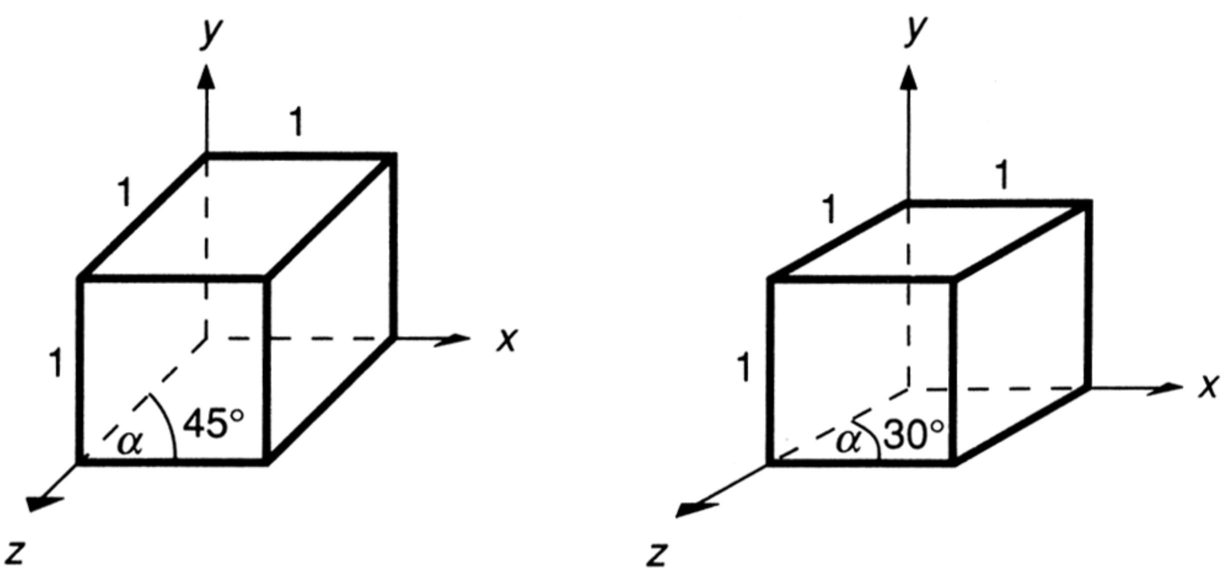


Parallel Projections

- Oblique parallel projections
  - Objects can be visualised better then with orthographic projections
  - Can measure distances, but not angles*
    - * Can only measure angles for faces of objects parallel to the plane
- 2 common oblique parallel projections:
  - *Cavalier* and *Cabinet*

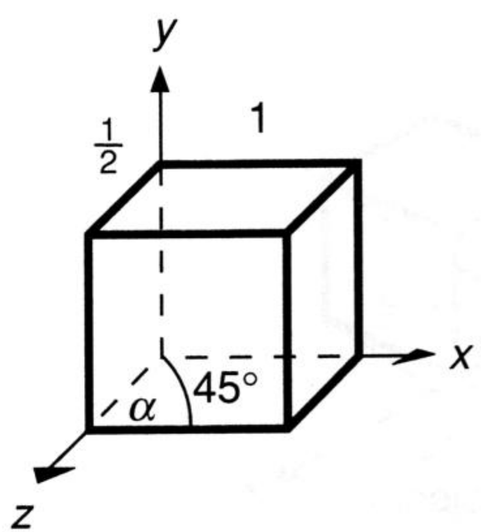
Parallel Projections

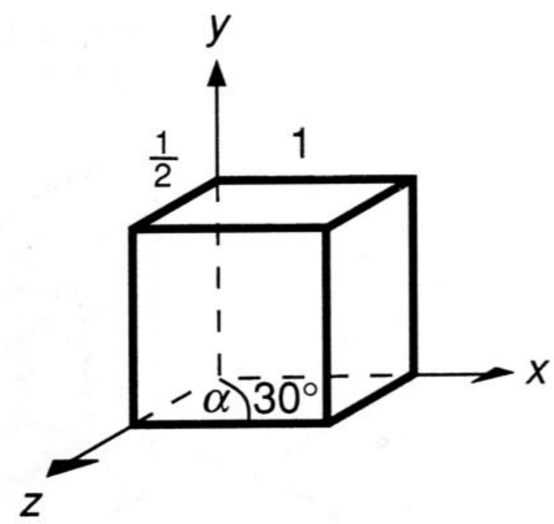
- Cavalier:
  - The direction of the projection makes a 45 degree angle with the projection plane.
  - Because there is no foreshortening, this causes an exaggeration of the z axes.



Parallel Projections

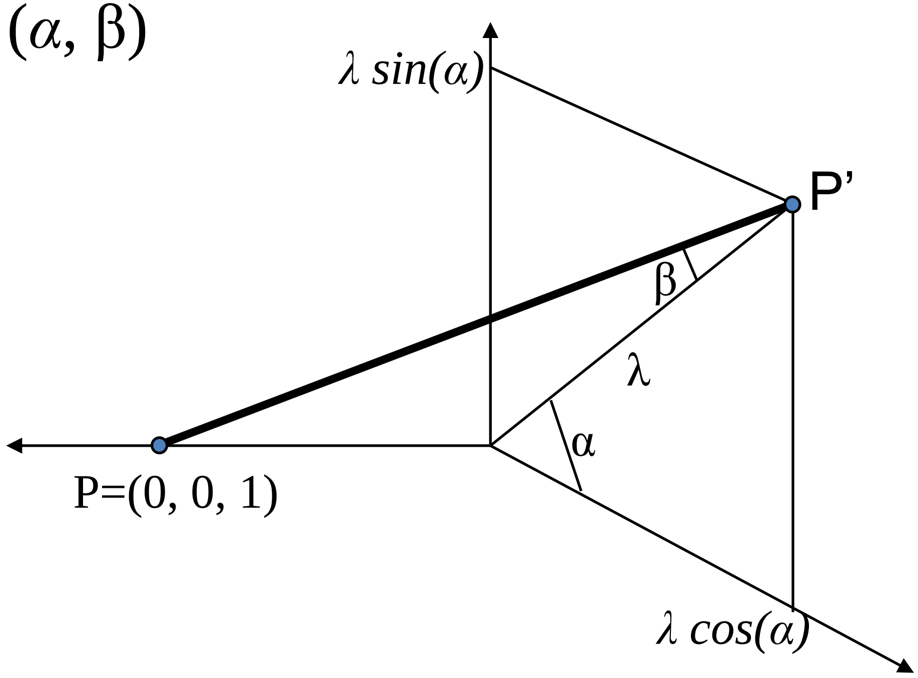
- Cabinet:
  - The direction of the projection makes a 63.4 degree angle with the projection plane. This results in foreshortening of the z axis, and provides a more "realistic" view.





Oblique Parallel Projections

- Cavalier, cabinet and orthogonal projections can all be specified in terms of (α, β) or (α, λ) since
  - $\tan(\beta) = 1/\lambda$



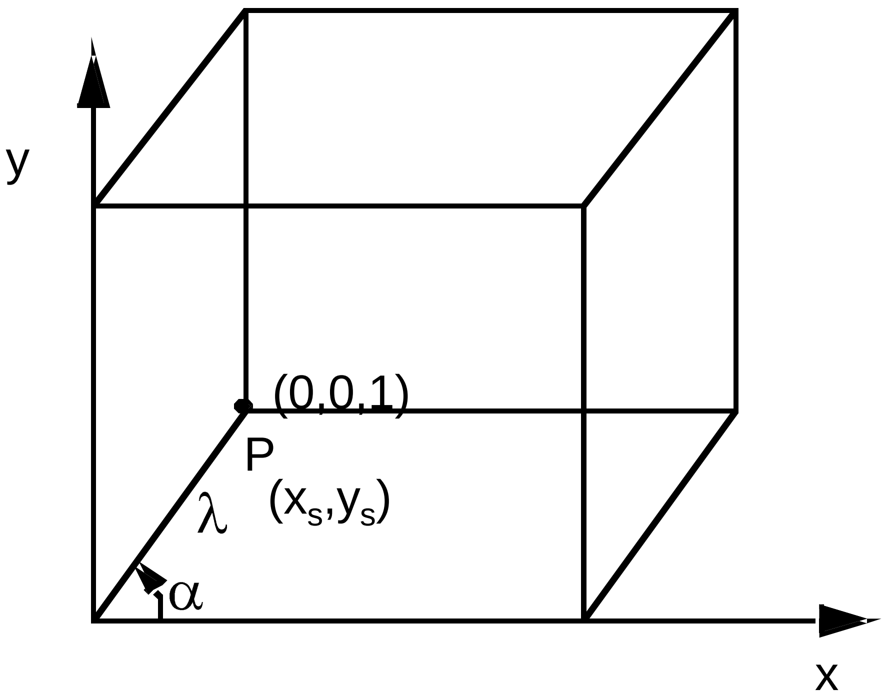
•

Oblique Parallel Projections

λ=1	β = 45	Cavalier projection	α = 0 - 360
λ=0.5	β = 63.4	Cabinet projection	α = 0 – 360
λ=0	β = 90	Orthogonal projection	α = 0 – 360

•

Oblique Parallel Projections



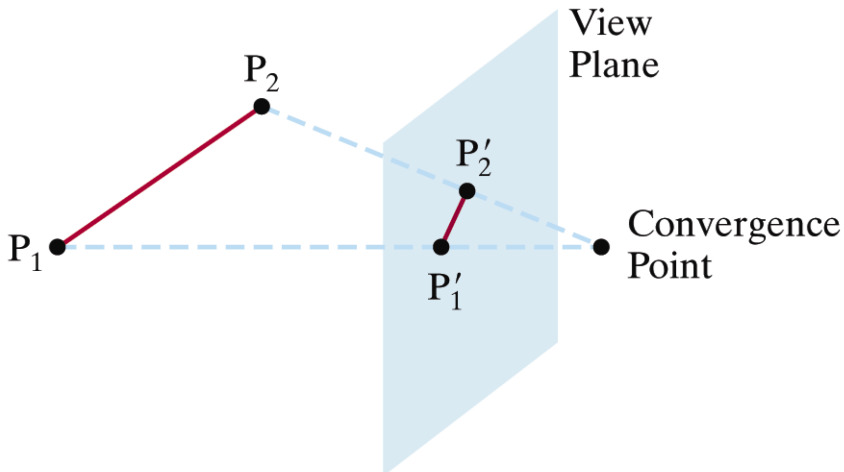
Consider the point P:

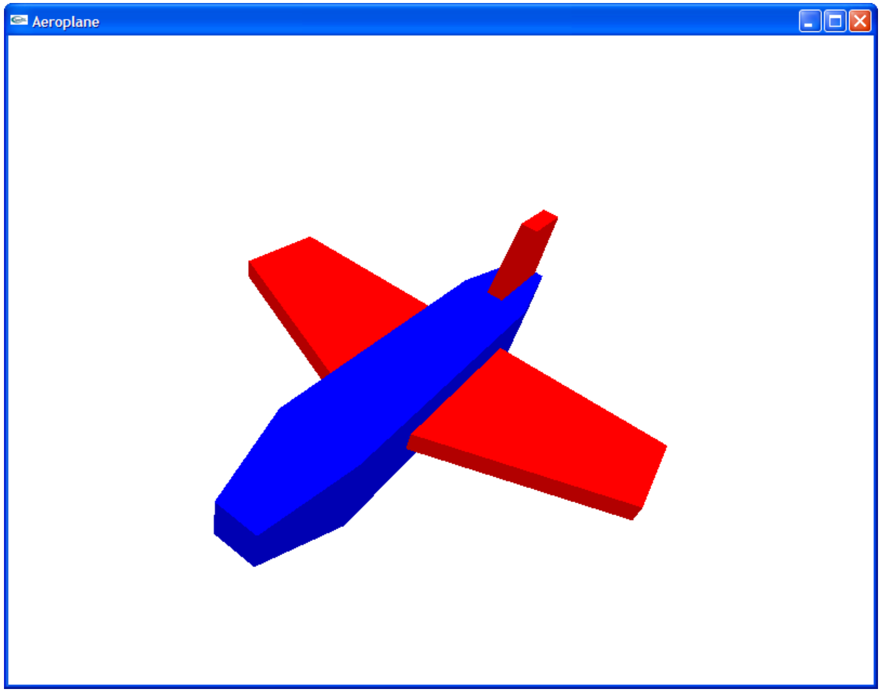
P can be represented in 3D space - (0,0,1)

P can be represented in 2D (screen coords) - (x_s,y_s)

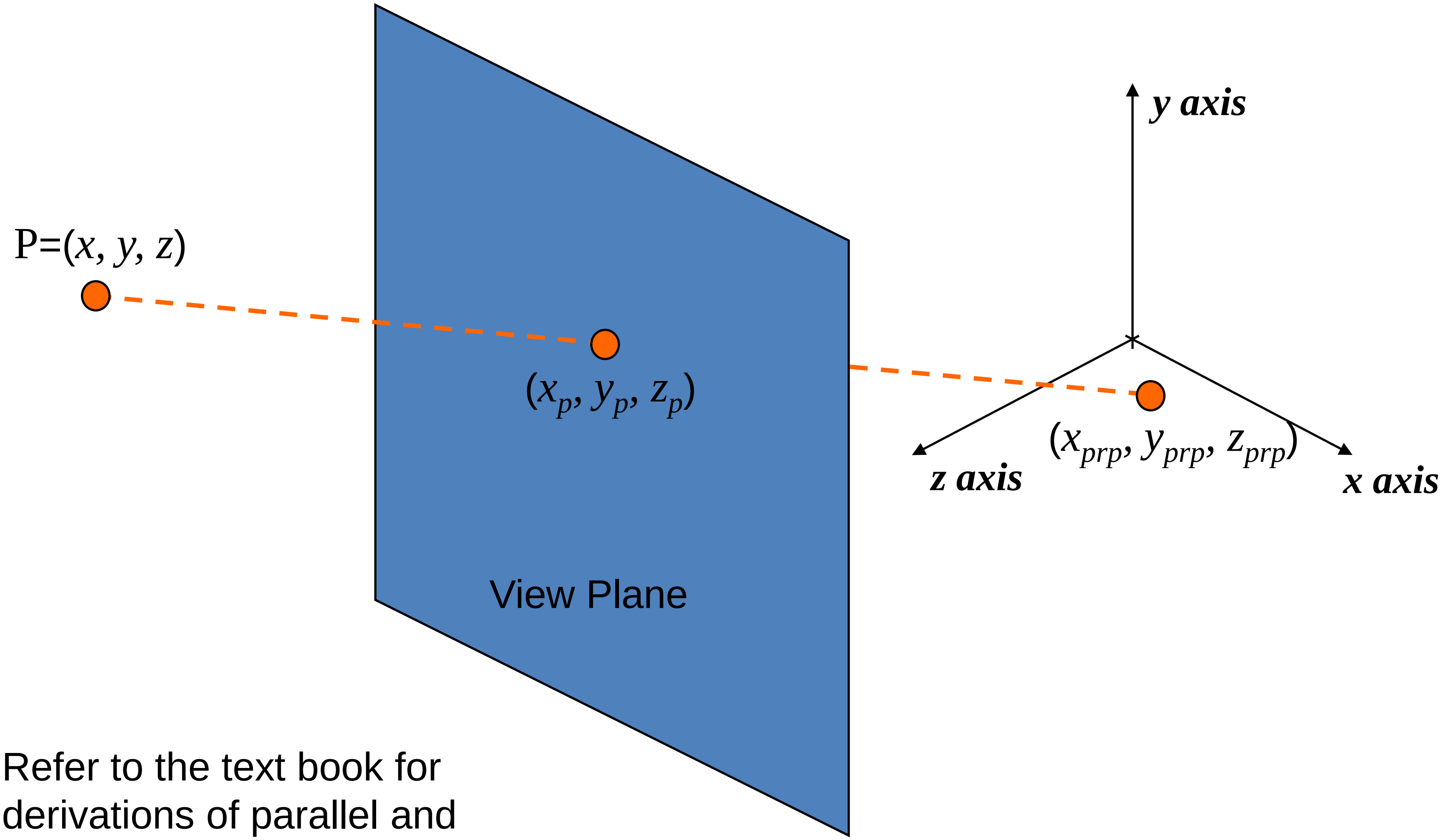
Perspective Projections

- Remember the whole point of perspective projections





Projection Calculations



Refer to the text book for derivations of parallel and perspective projections