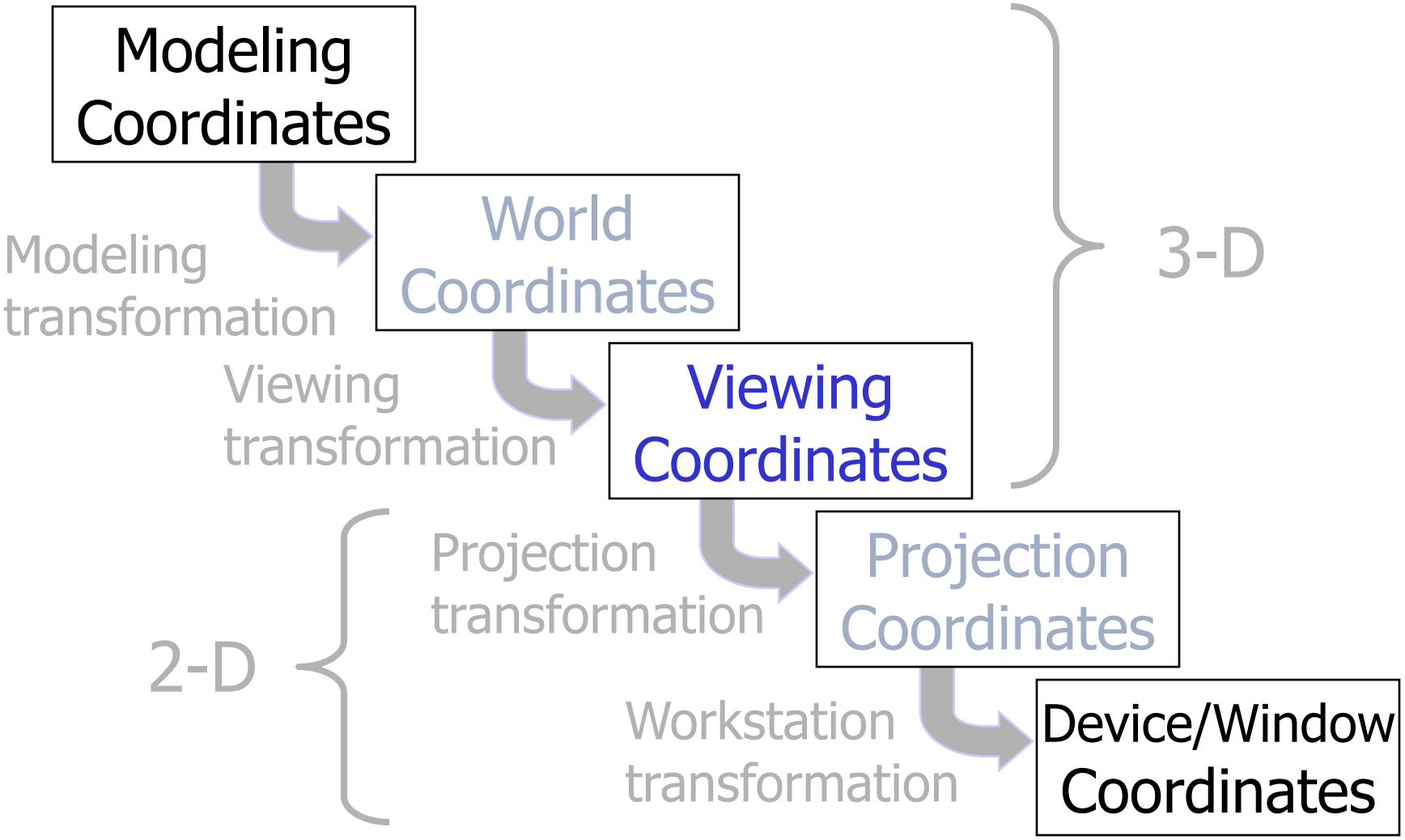
Viewing in 3D

Dr. Ayatullah F. Mollah
Department of Computer Science and Engineering,
Aliah University

3-D Viewing Process

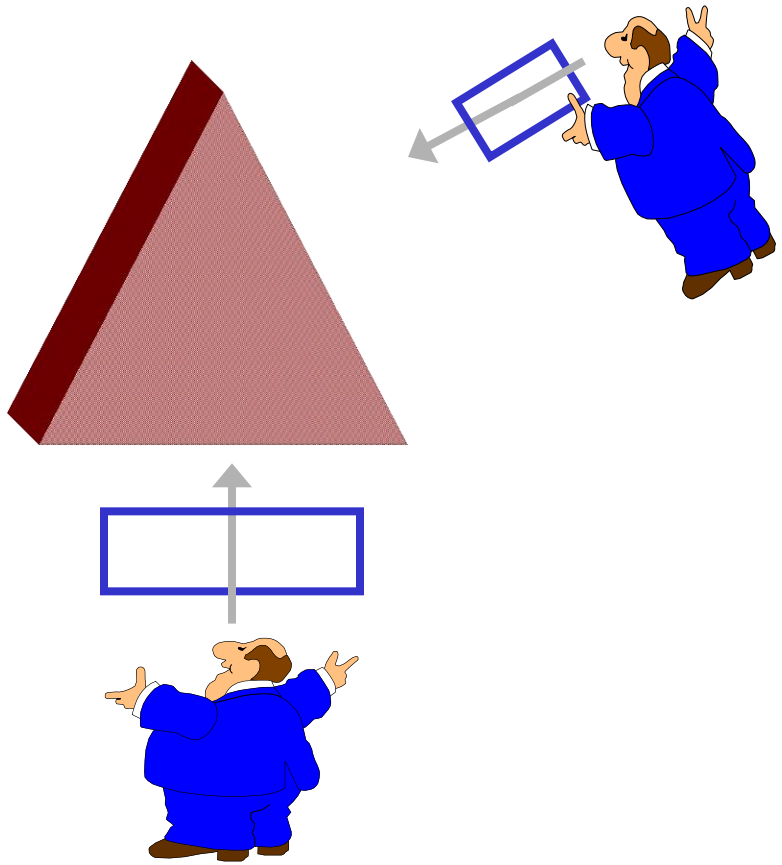


Viewing Coordinate System

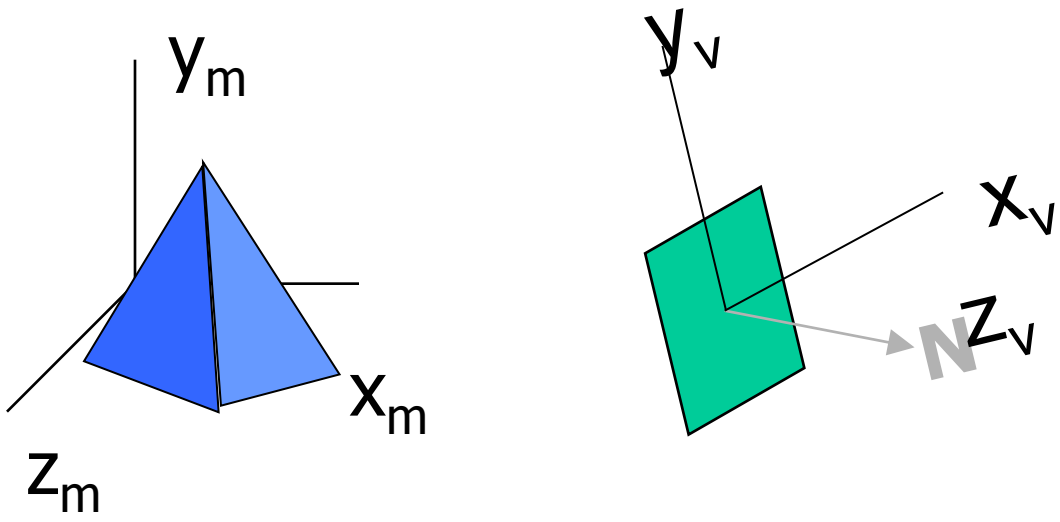
Identify viewer position relative to scene

Viewer "looks through" a window

Must specify position and view direction

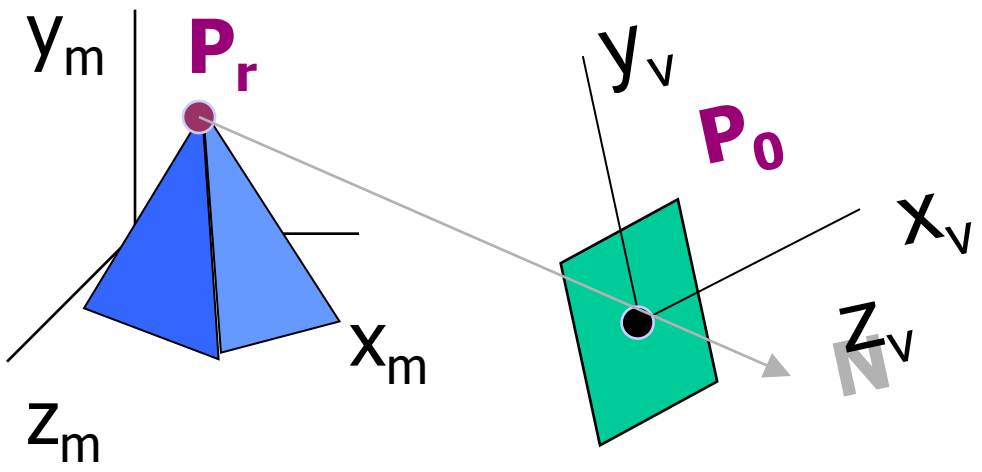


View Plane



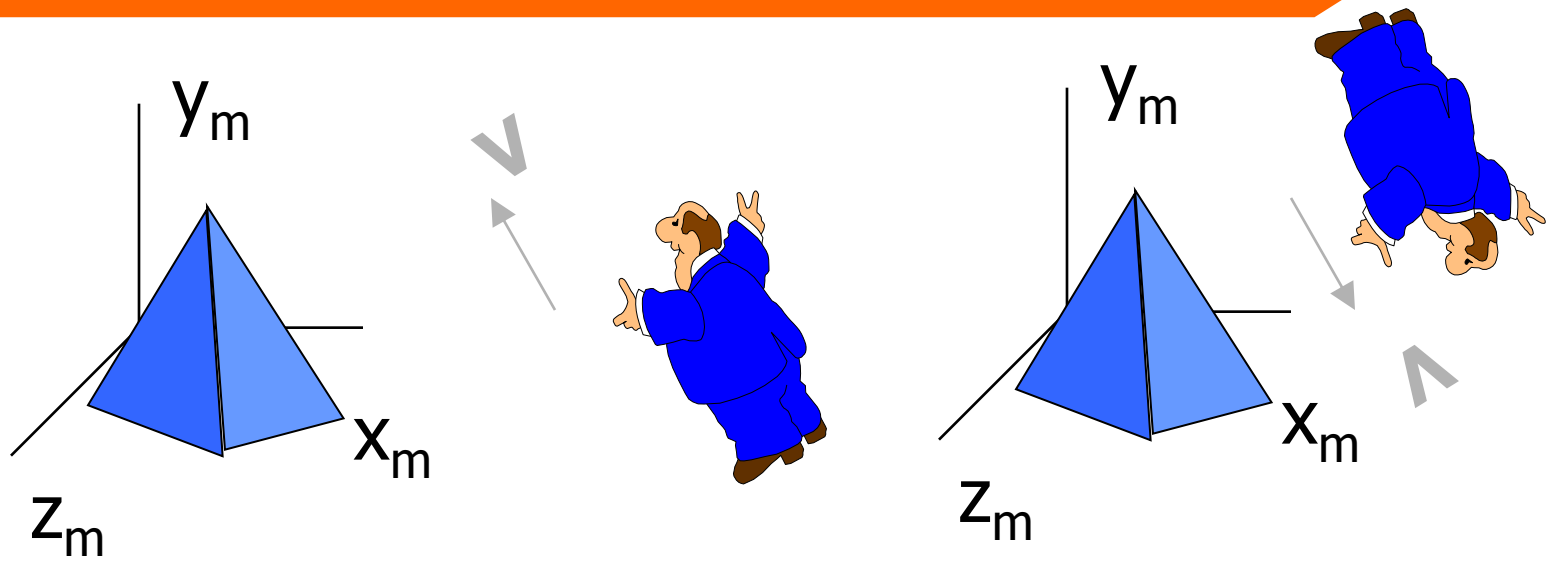
View plane defined by normal vector (**N**)

View Reference Points



- P_r: a point in the scene we are looking at
- P_0: a distant point from which we're looking
- Note P_r, P_0, and N are expressed in $x_m y_m z_m$

Look-Up Vector



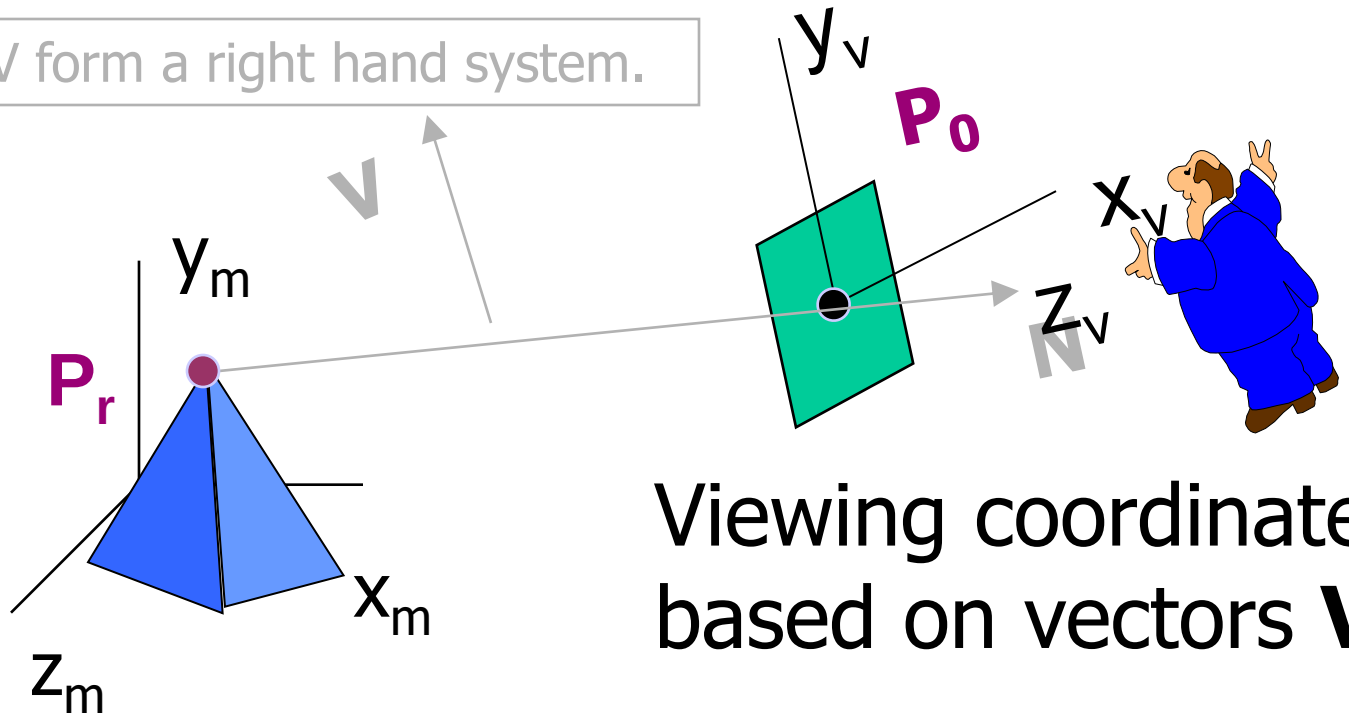
View-plane normal vector and reference point are not enough

We also need to specify orientation of view(er)

View-up vector (**V**) must be normal to **N**

Viewing Coordinates

N,V form a right hand system.

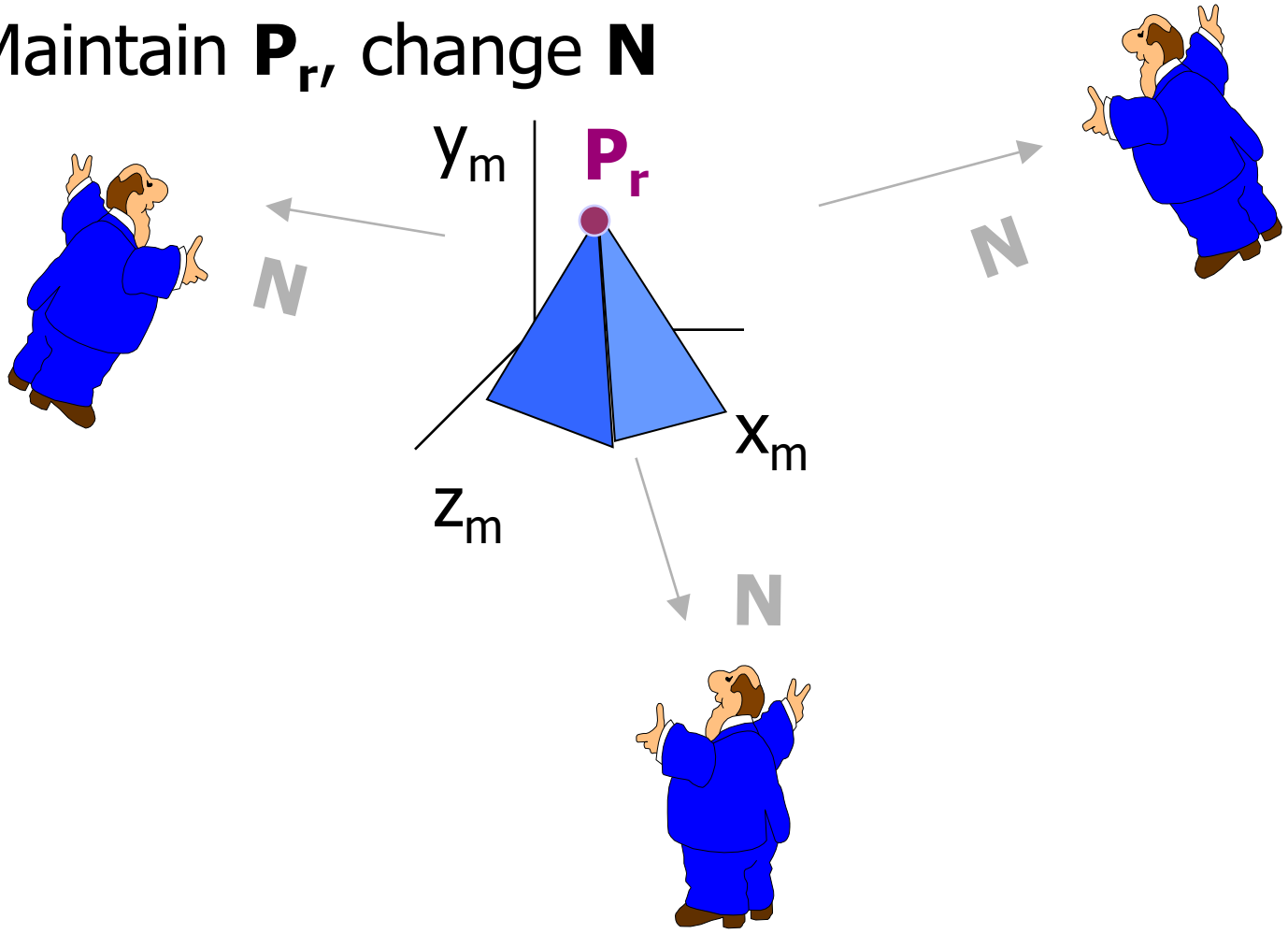


Viewing coordinate system based on vectors **V,N**

Forms the (x_v, y_v, z_v) coordinate axes

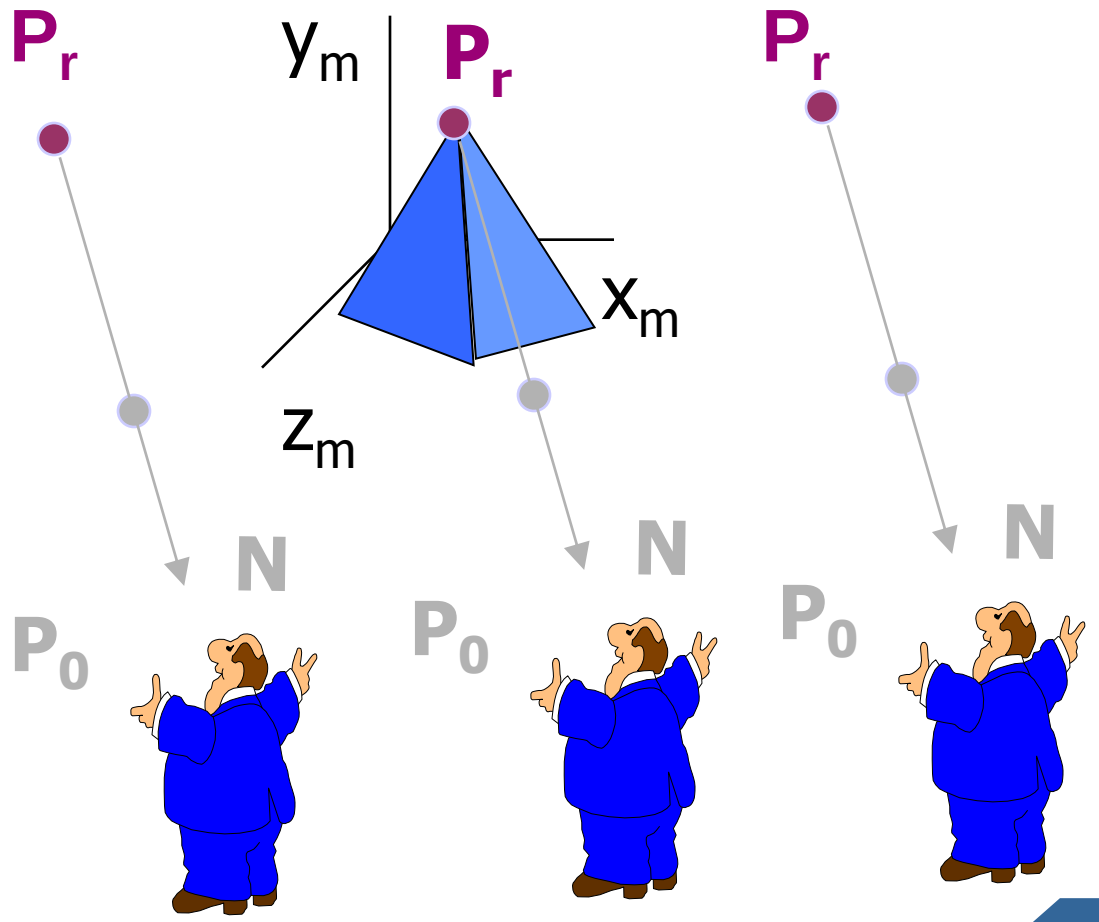
Changing Views (1)

Maintain $\mathbf{P_r}$, change **N**



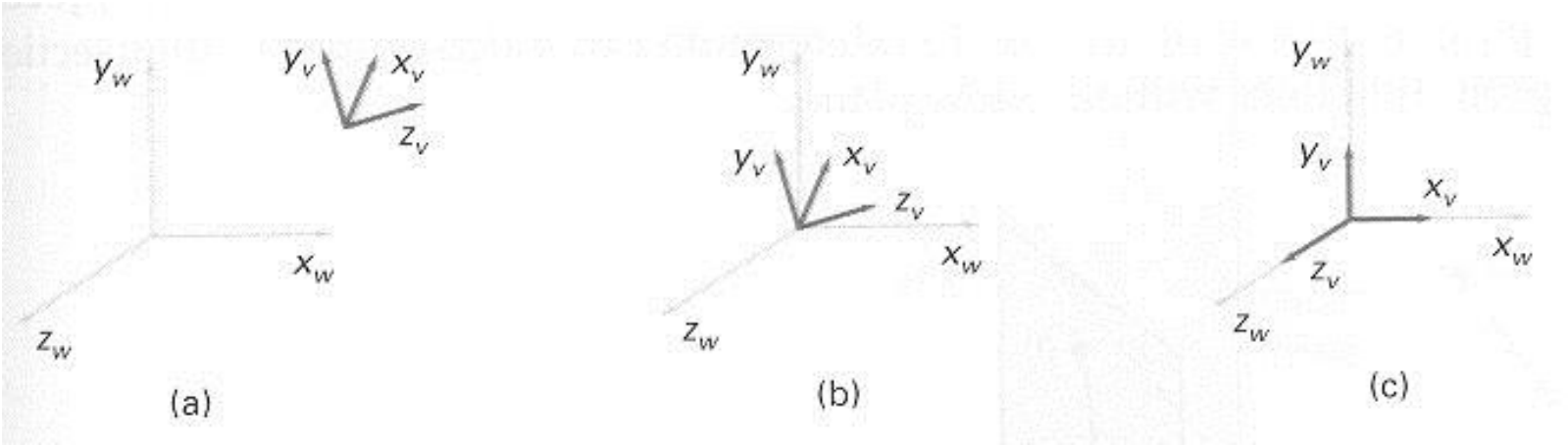
Changing Views (2)

Maintain **N**, change $\mathbf{P_r}$ and $\mathbf{P_0}$



Transformation from WC to VC

- Transformation sequences
  1. Translate the view reference point to the origin of the WC system
  2. Apply rotations to align the x_v, y_v, and z_v axes with the world axes



General sequence of translate-rotate transformation

Transformation from WC to VC (cont')

- Translation
  - view reference point(x_0, y_0, z_0)

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Rotation
  - rotate around the world x_w axis to bring z_v into the $x_w z_w$ plane
  - rotate around the world y_w axis to align the z_w and z_v axis
  - final rotation is about the z_w axis to align the y_w and y_v axis

Transformation from WC to VC (cont')

- Rotation by *uvn* system
  - Calculate unit *uvn* vectors
    - N : view-plane normal vector
    - V : view-up vector
    - U : perpendicular to both N and V

$$\mathbf{n} = \frac{\mathbf{N}}{|\mathbf{N}|} = (n_1, n_2, n_3)$$

$$\mathbf{u} = \frac{\mathbf{V} \times \mathbf{N}}{|\mathbf{V} \times \mathbf{N}|} = (u_1, u_2, u_3)$$

$$\mathbf{v} = \mathbf{n} \times \mathbf{u} = (v_1, v_2, v_3)$$

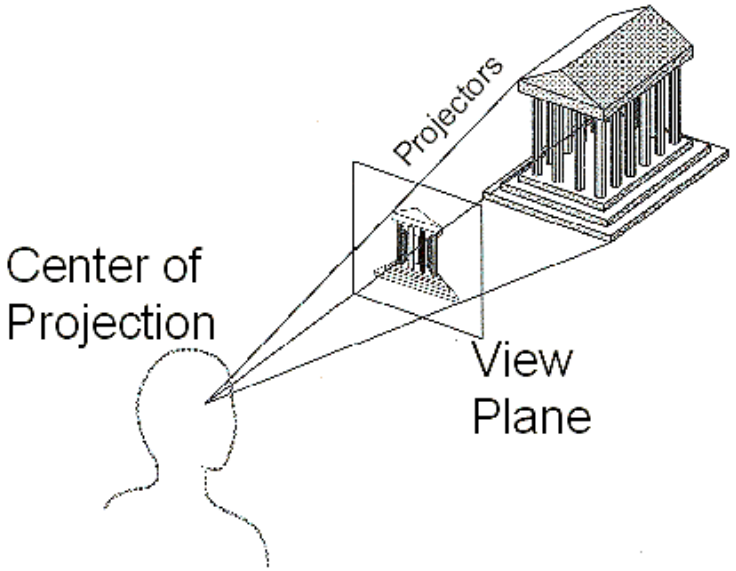
  - Form the composite rotation matrix

$$\mathbf{R} = \begin{bmatrix} u_1 & u_2 & u_3 & 0 \\ v_1 & v_2 & v_3 & 0 \\ n_1 & n_2 & n_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

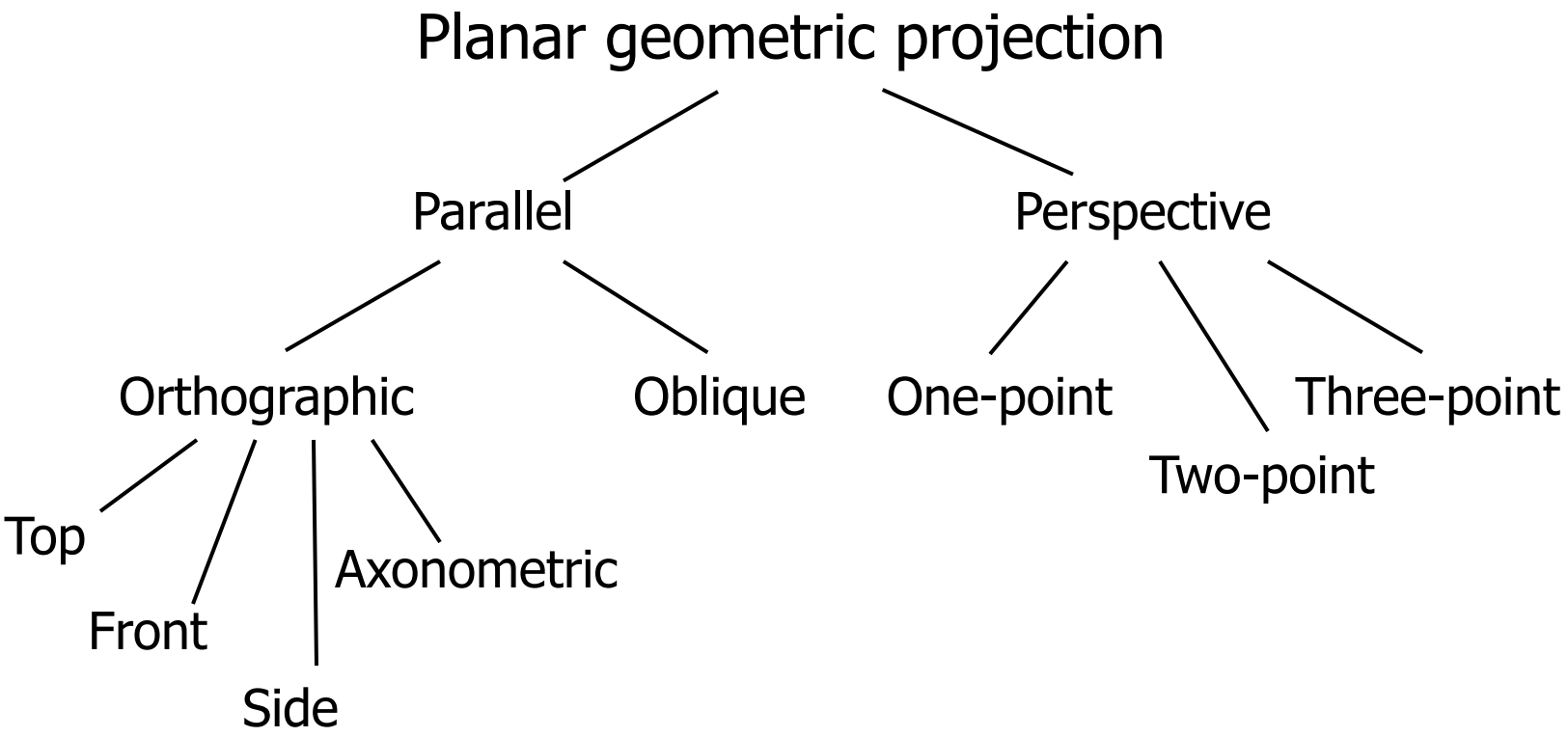
$$\mathbf{M}_{WC,VC} = \mathbf{R} \cdot \mathbf{T}$$

Projection

- General definition
  - Transform points in n-space to m-space(m<n)
- In computer graphics
  - Map viewing coordinates to 2D screen coordinates

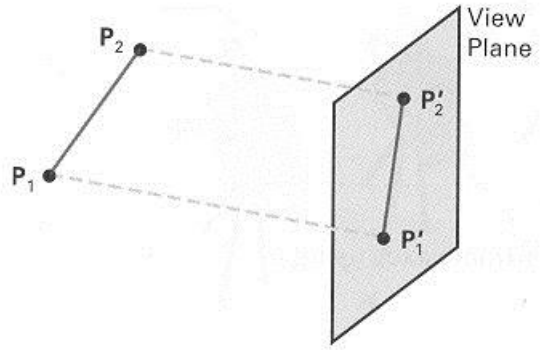


Taxonomy of Projections

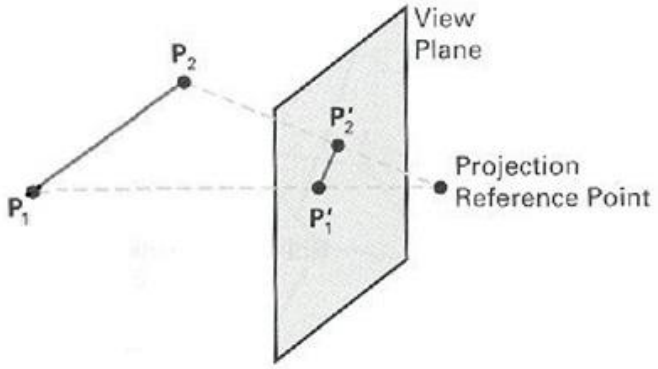


Parallel & Perspective

Parallel Projection

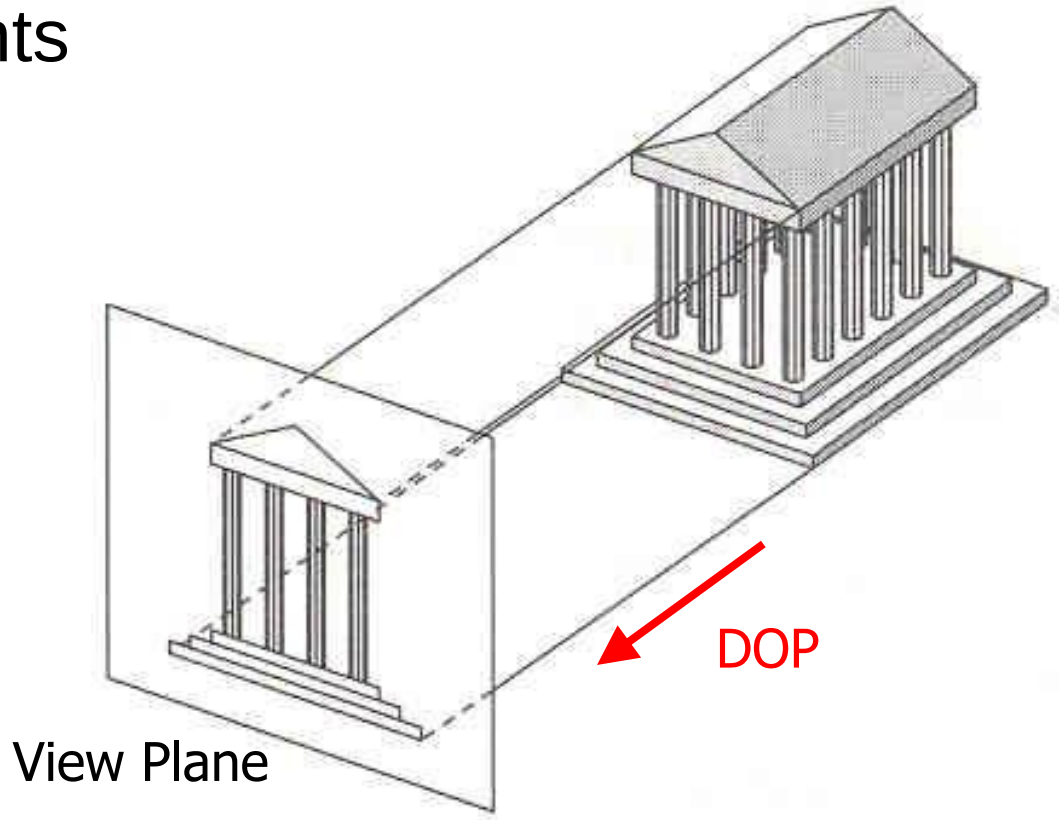


Perspective Projection



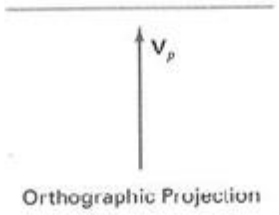
Parallel Projection

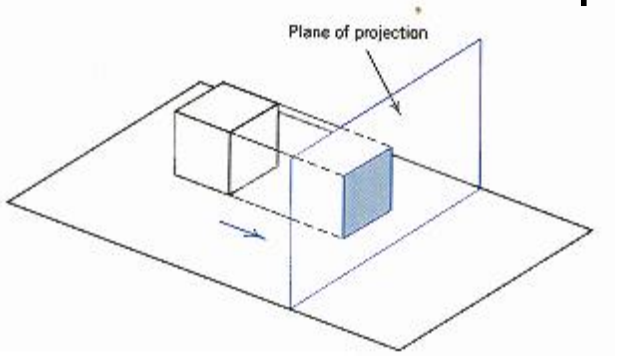
- Center of projection is at infinity
  - Direction of projection (DOP) same for all points



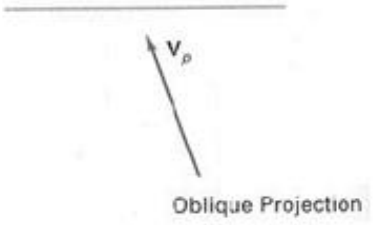
Orthographic & Oblique

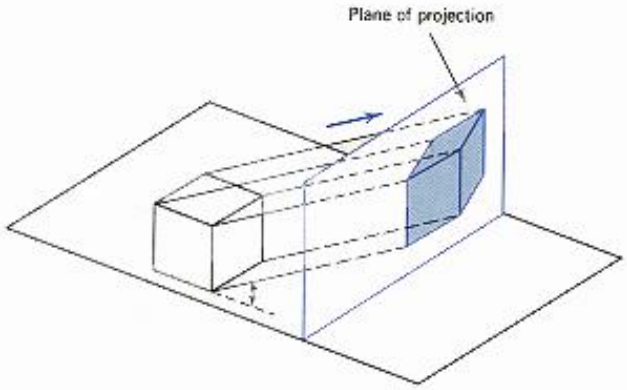
- Orthographic parallel projection
  - the projection is perpendicular to the view plane





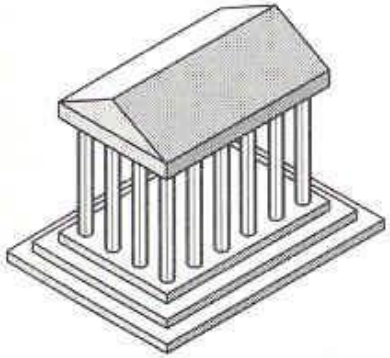
- Oblique parallel projection
  - The projectors are inclined with respect to the view plane

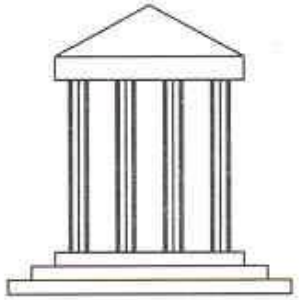




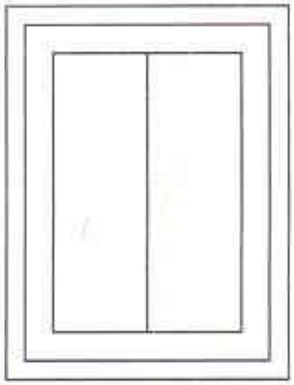
Orthographic Projections

- DOP perpendicular to view plane

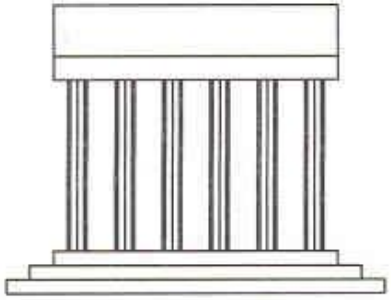




Front

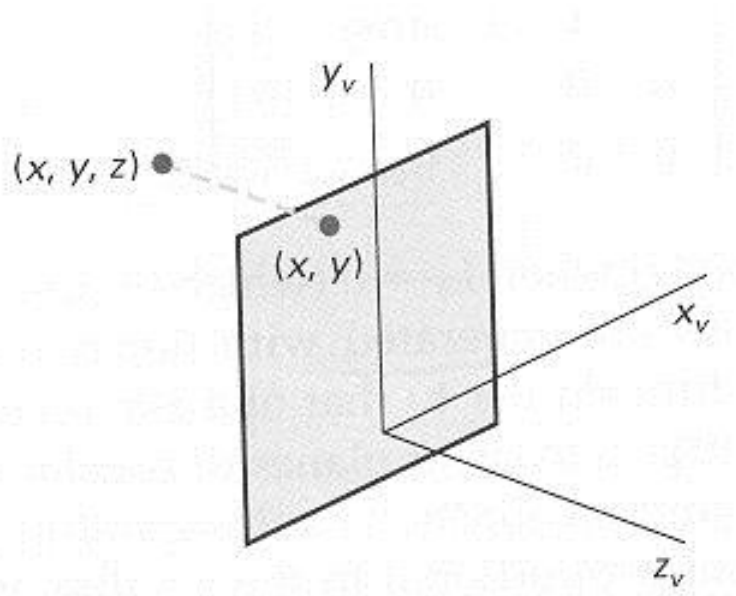


Top



Side

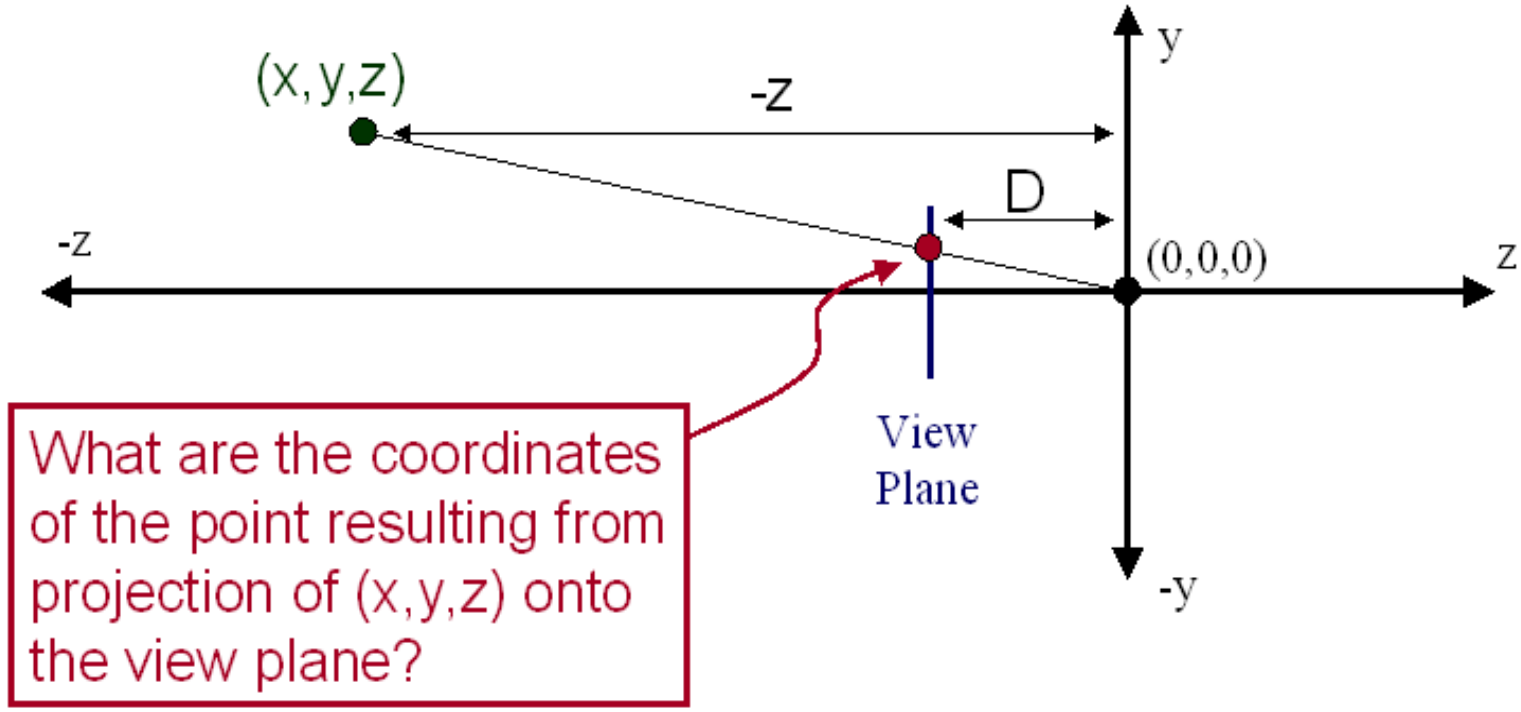
Orthographic Coordinates



$$x_p = x, \quad y_p = y$$

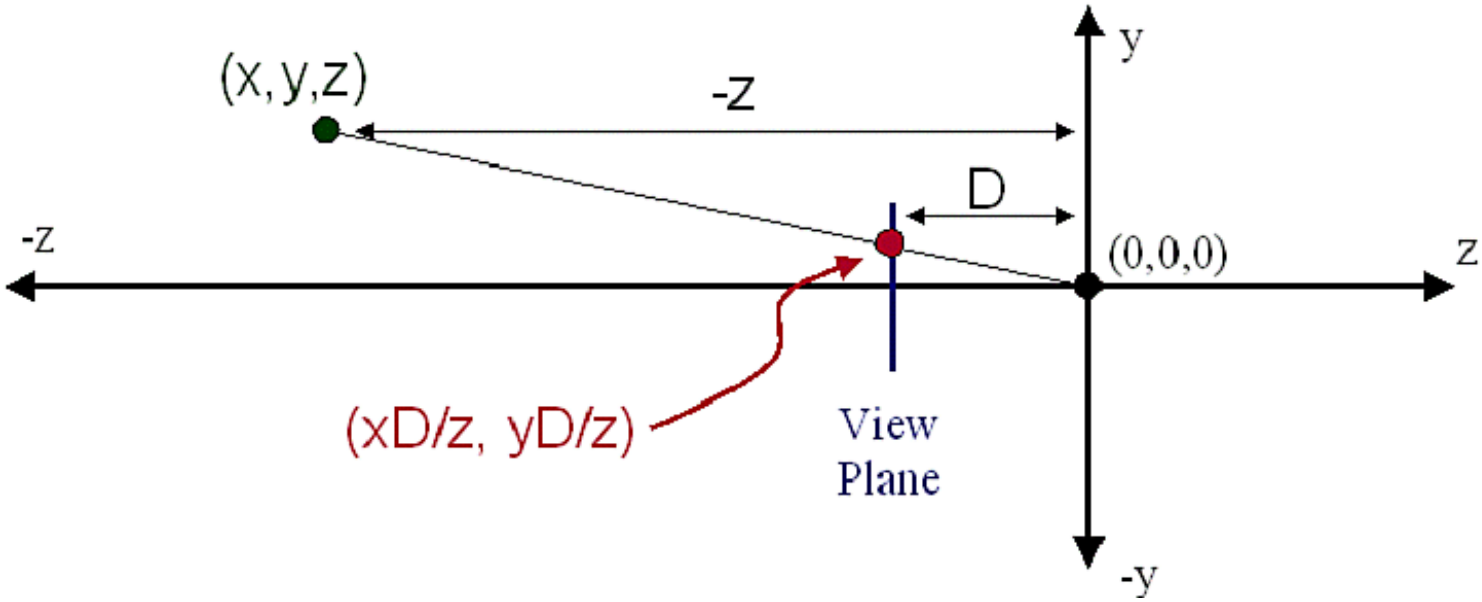
Perspective Projection

- Compute 2D coordinates from 3D coordinates with similar triangles



Perspective Projection

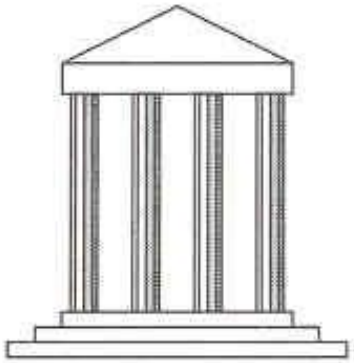
- Compute 2D coordinates from 3D coordinates with similar triangles



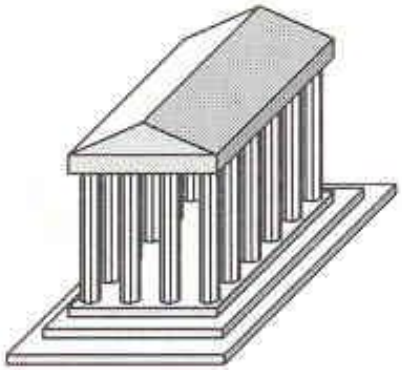
Perspective vs. Parallel

- Perspective projection
  - + Size varies inversely with distance – looks realistic
  - – Distance and angles are not(in general) preserved
  - – Parallel line do not (in general) remain parallel
- Parallel projection
  - + Good for exact measurements
  - + Parallel lines remain parallel
  - – Angles are not (in general) preserved
  - – Less realistic looking

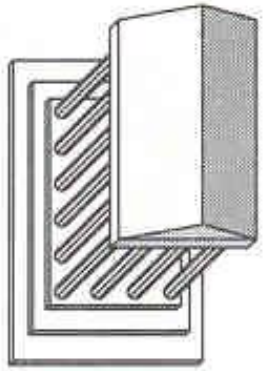
Classical Viewing



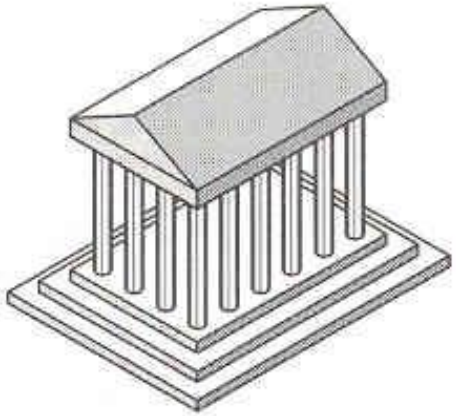
Front elevation



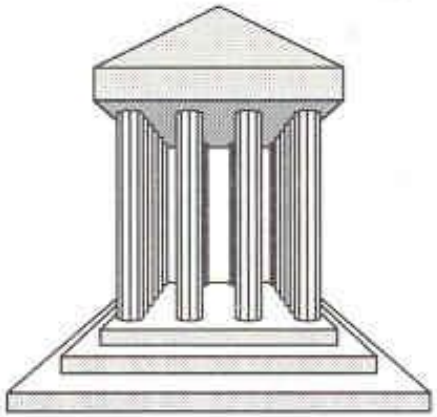
Elevation oblique



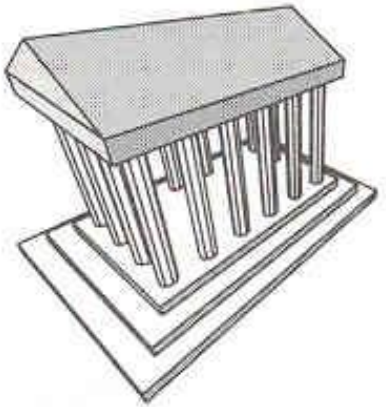
Plan oblique



Isometric



One-point perspective



Three-point perspective

(Thank You)