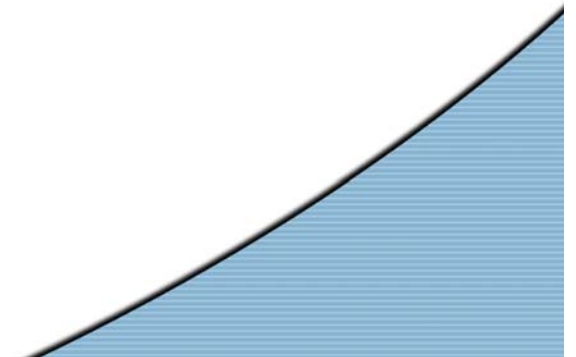
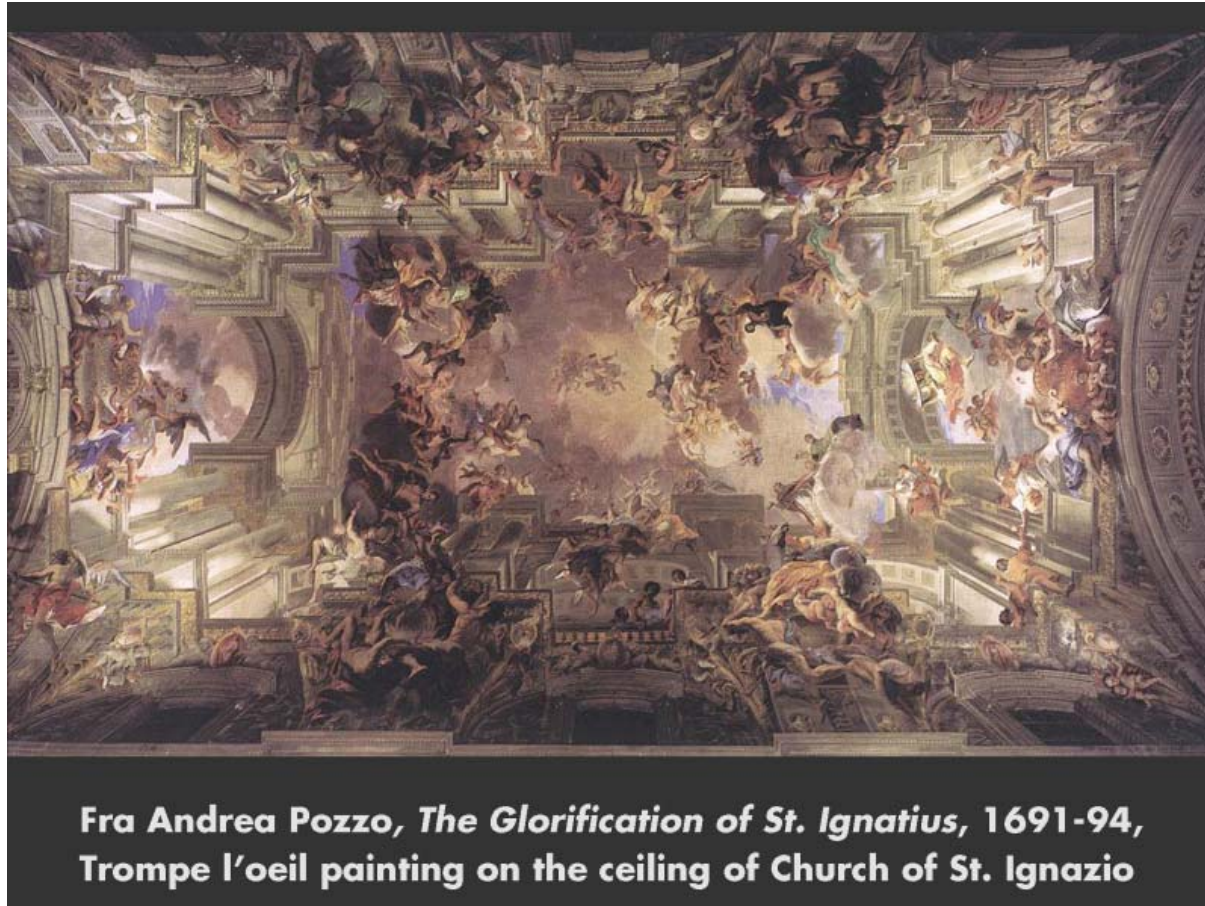


4. Projections





Perspective in Art

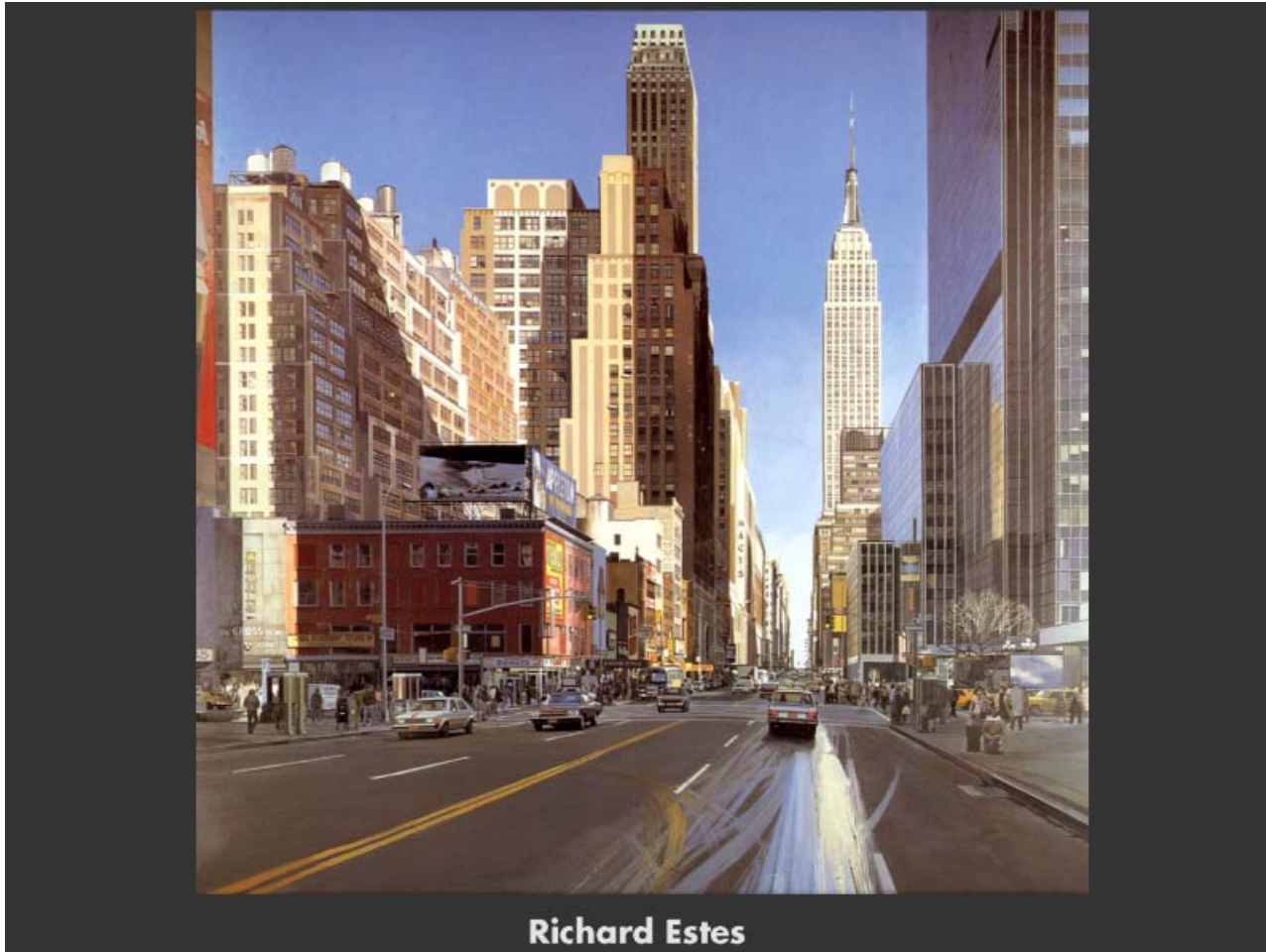


**Fra Andrea Pozzo, *The Glorification of St. Ignatius*, 1691-94,
Trompe l'oeil painting on the ceiling of Church of St. Ignazio**

courtesy of Pat Hanrahan, Stanford University



Perspective in Art

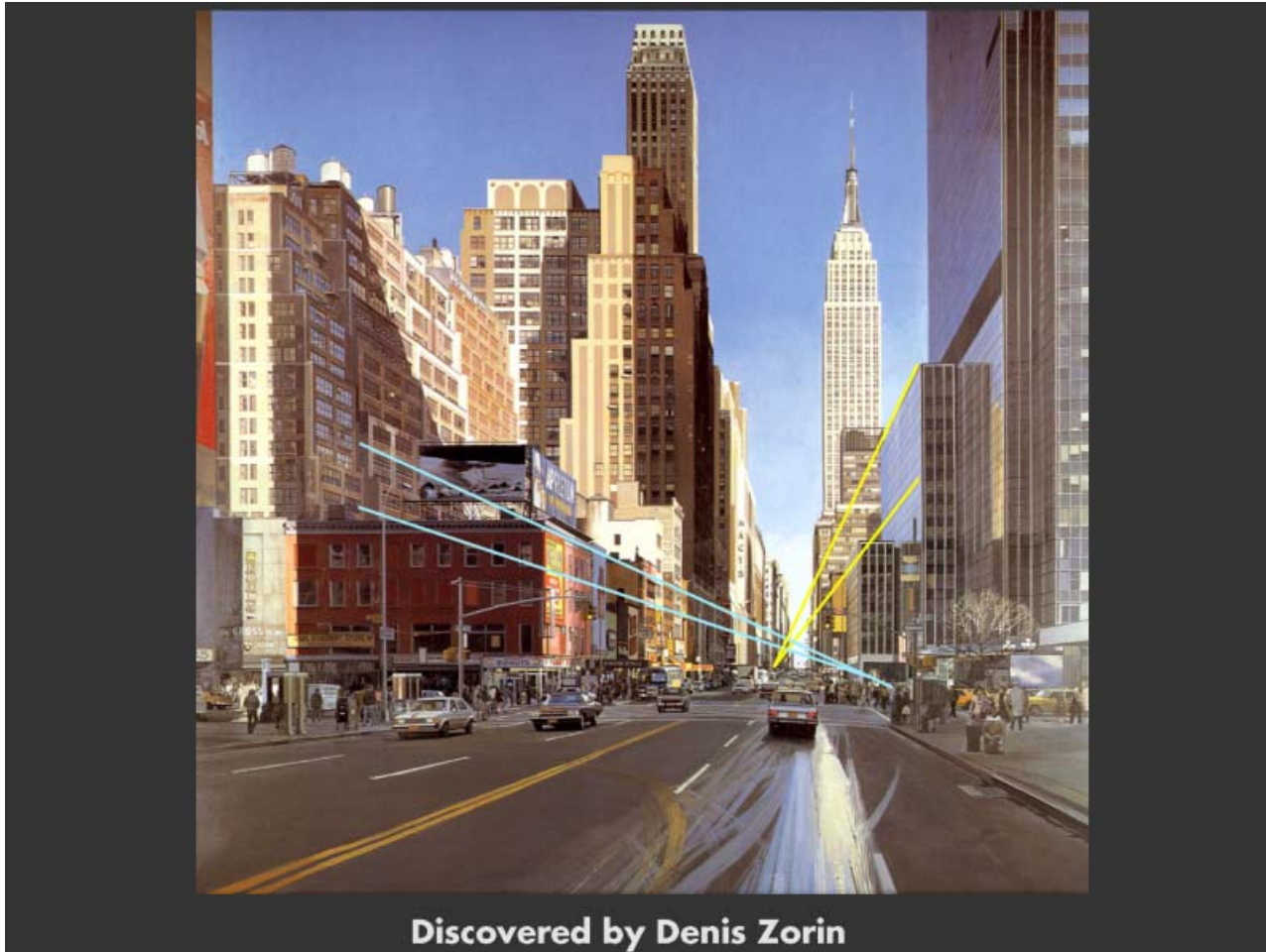


Richard Estes

courtesy of Pat Hanrahan, Stanford University



Perspective in Art



Discovered by Denis Zorin

courtesy of Pat Hanrahan, Stanford University



Perspective in Art



Giorgio de Chirico: *Mystery and Melancholy of a Street*



Perspective in Art



Raphael: School of Athens

from: Agrawala, Zorin, Munzner: Artistic Multiprojection Rendering, EGWR 2000



Perspective in Art

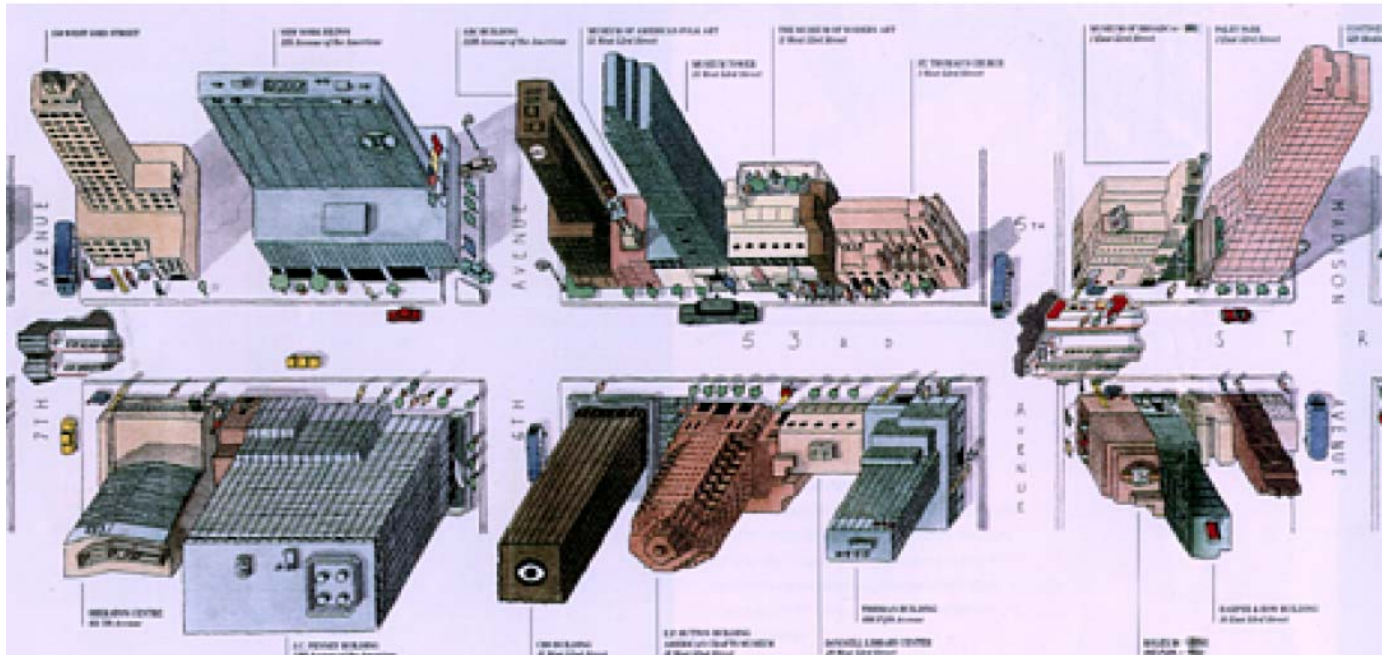


Cezanne: Still Life with Fruit Basked

from: Agrawala, Zorin, Munzner: Artistic Multiprojection Rendering, EGWR 2000



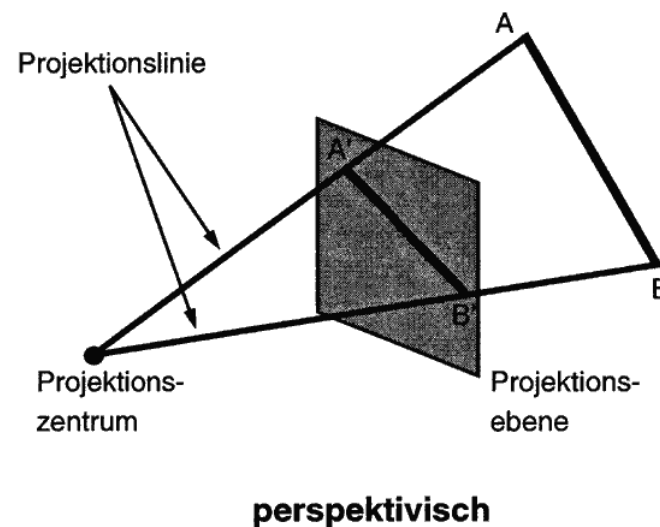
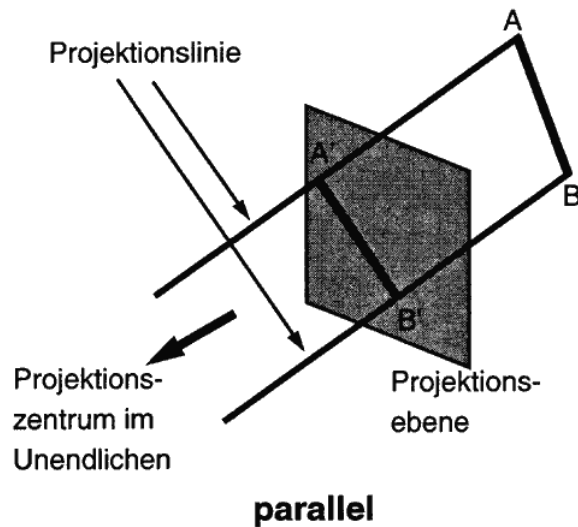
Perspective in Art





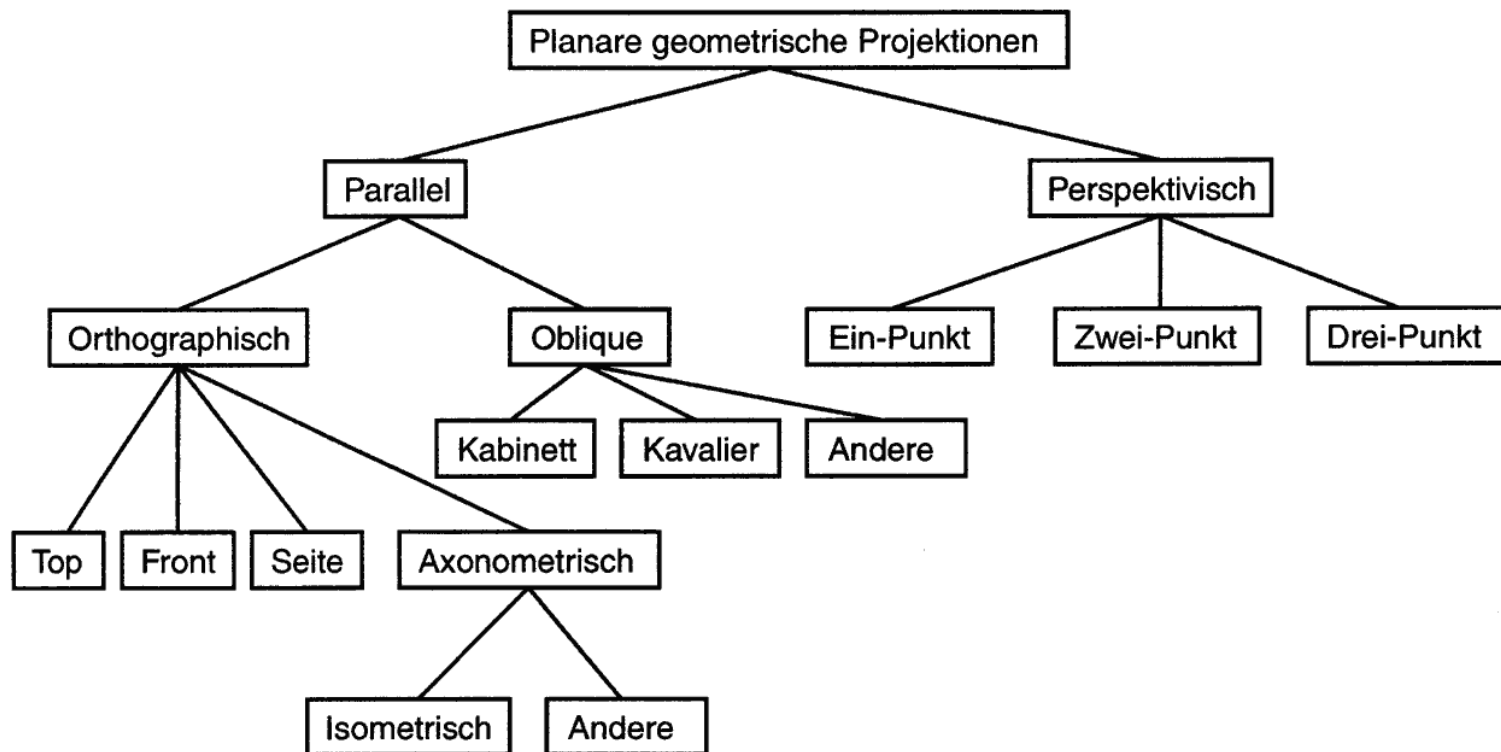
Parallel vs. Perspective Projection

- Rigorous mathematical treatment in *Projective Geometry*
- Planar Projections only





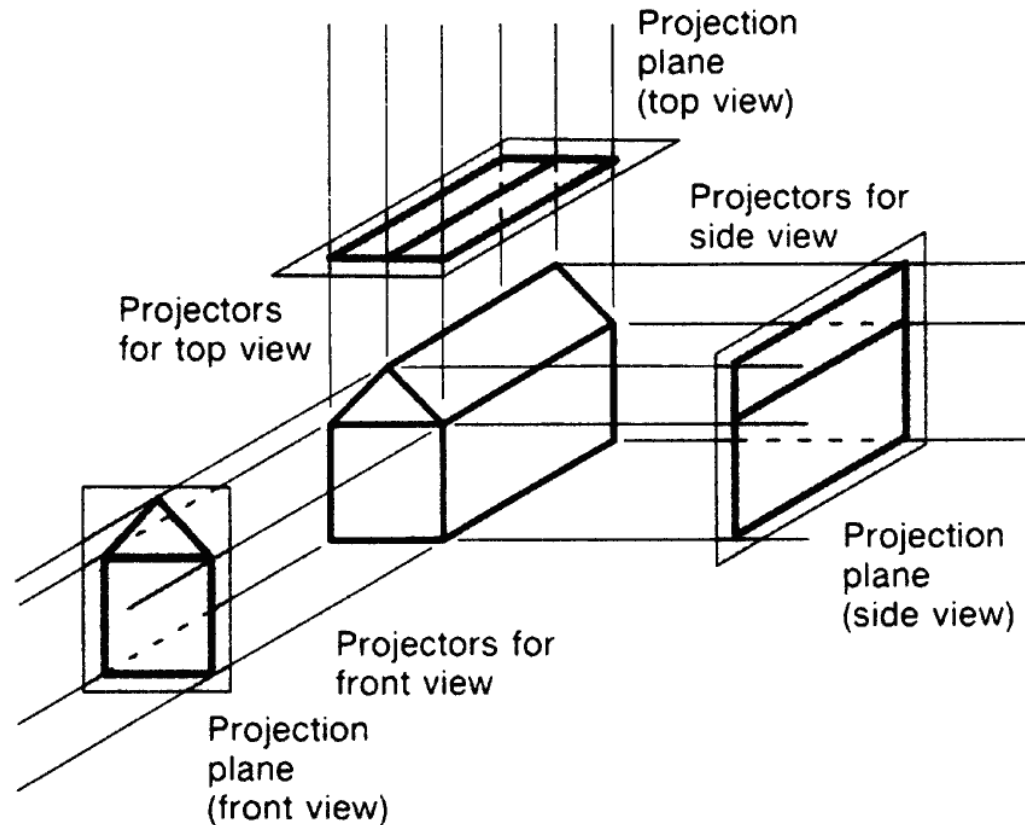
Classification





Orthographic Projection

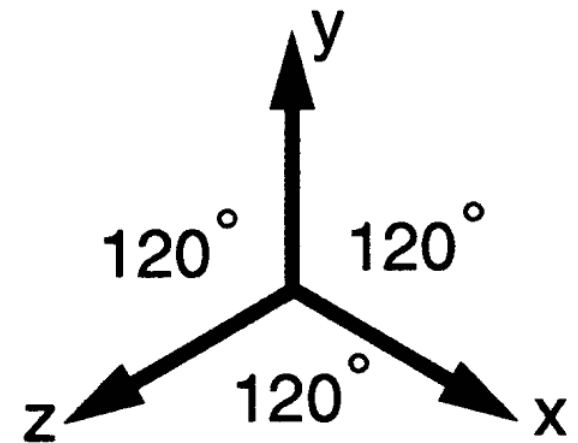
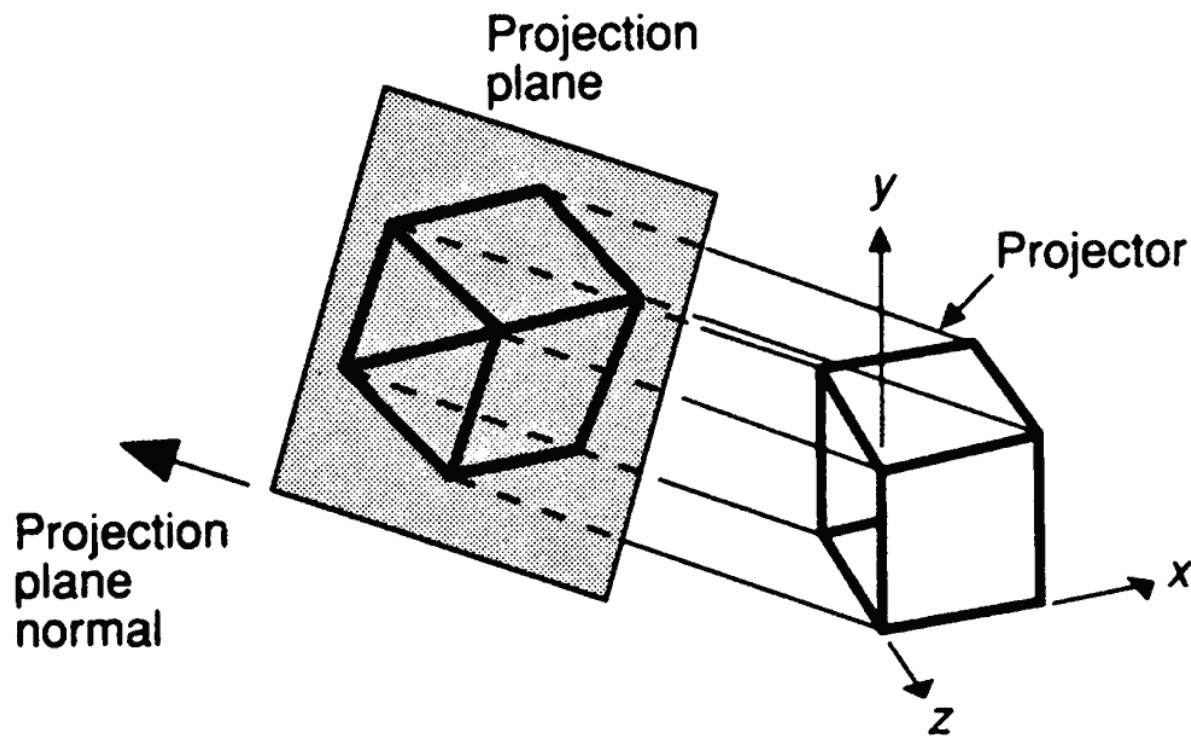
- Front-, top-, and side views





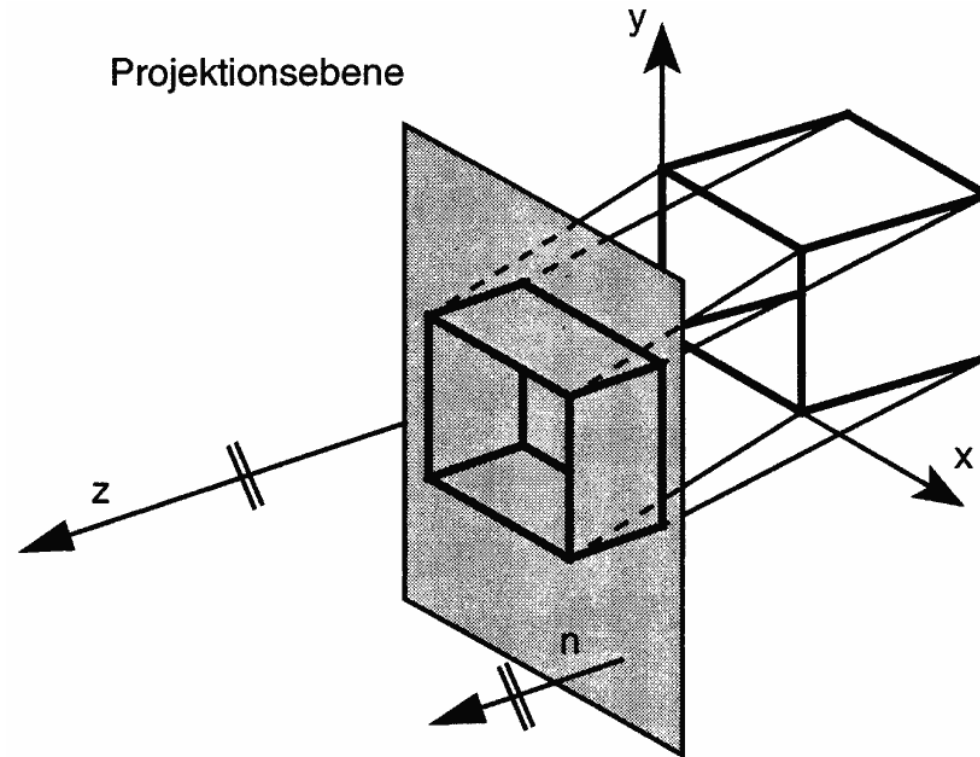
Isometric Projection

- Projection plane normal equals $(1, 1, 1)$



Oblique Projections

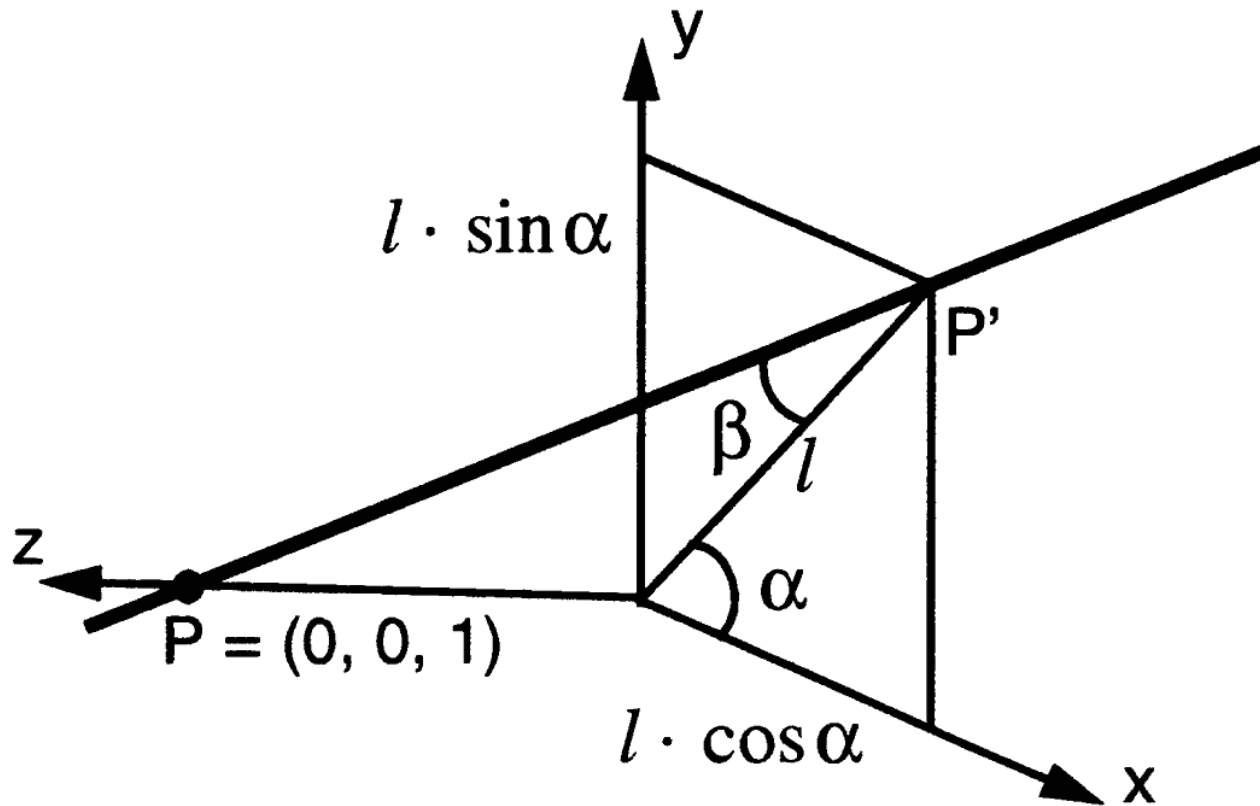
- Normal \neq Direction of Projection



Normale zur Projektionsebene



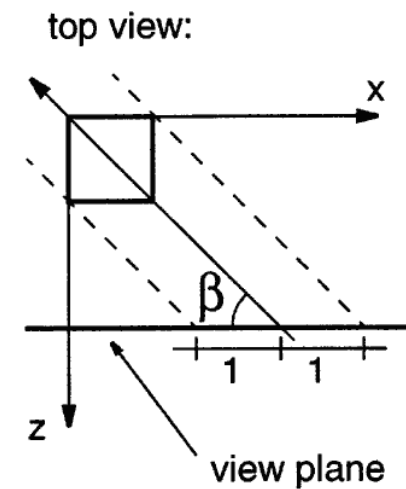
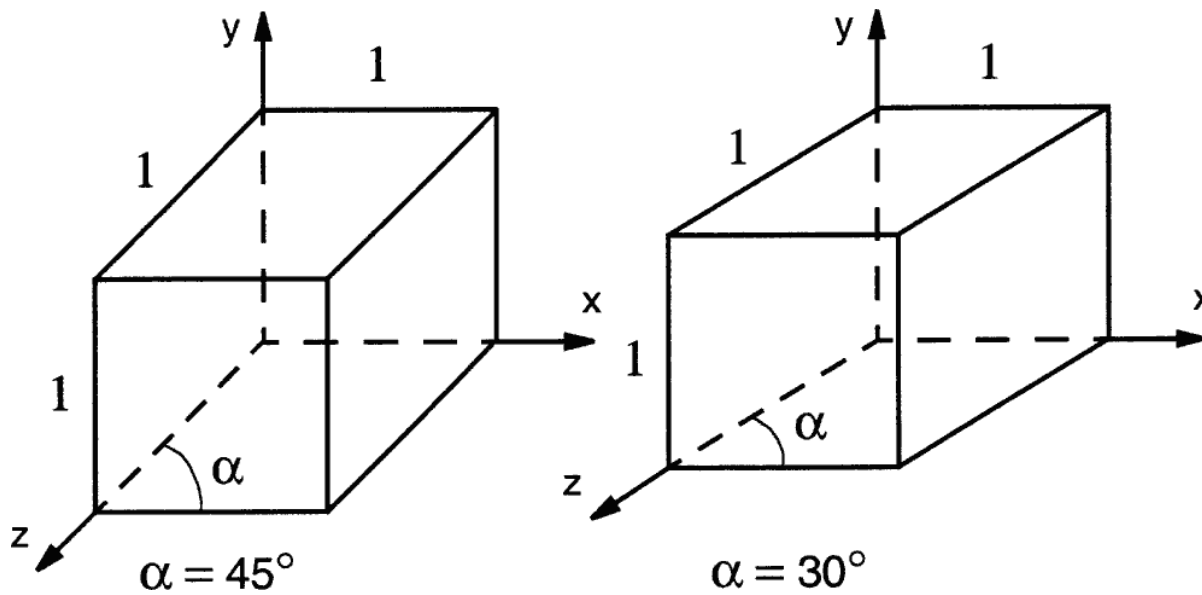
Direction of Projection





“Kavalier” Projection

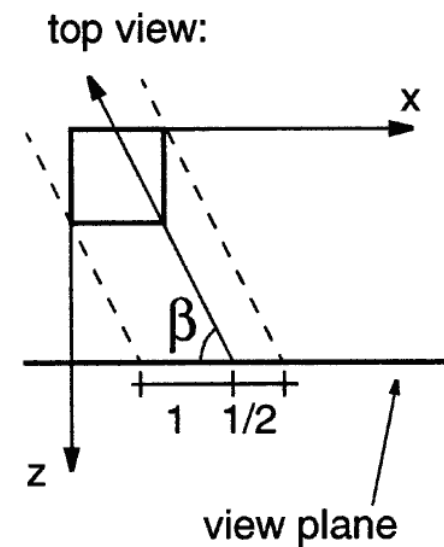
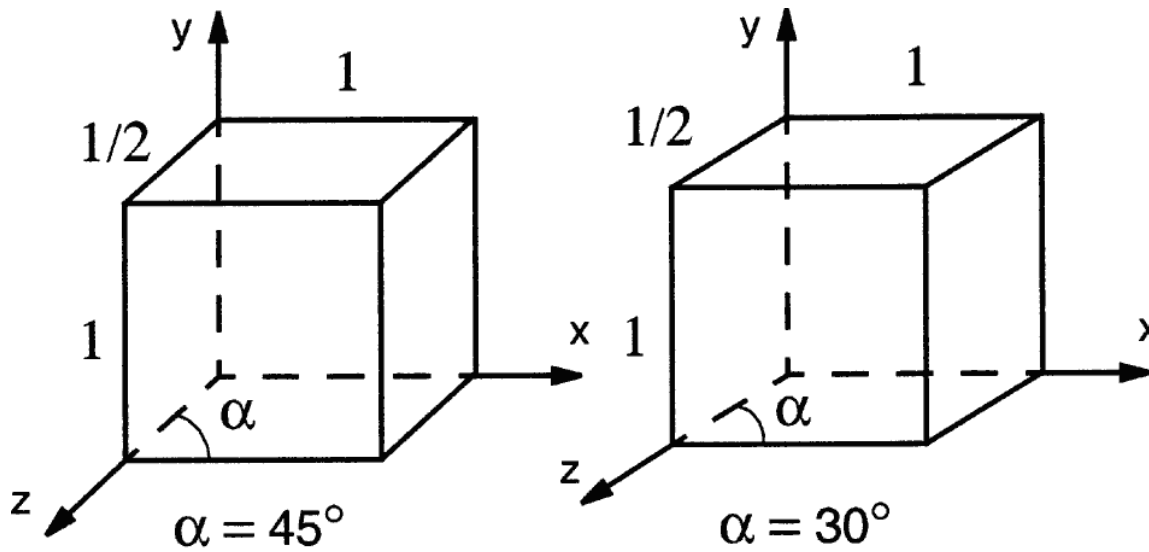
- $\beta = 45^\circ$





“Kabinett” Projection

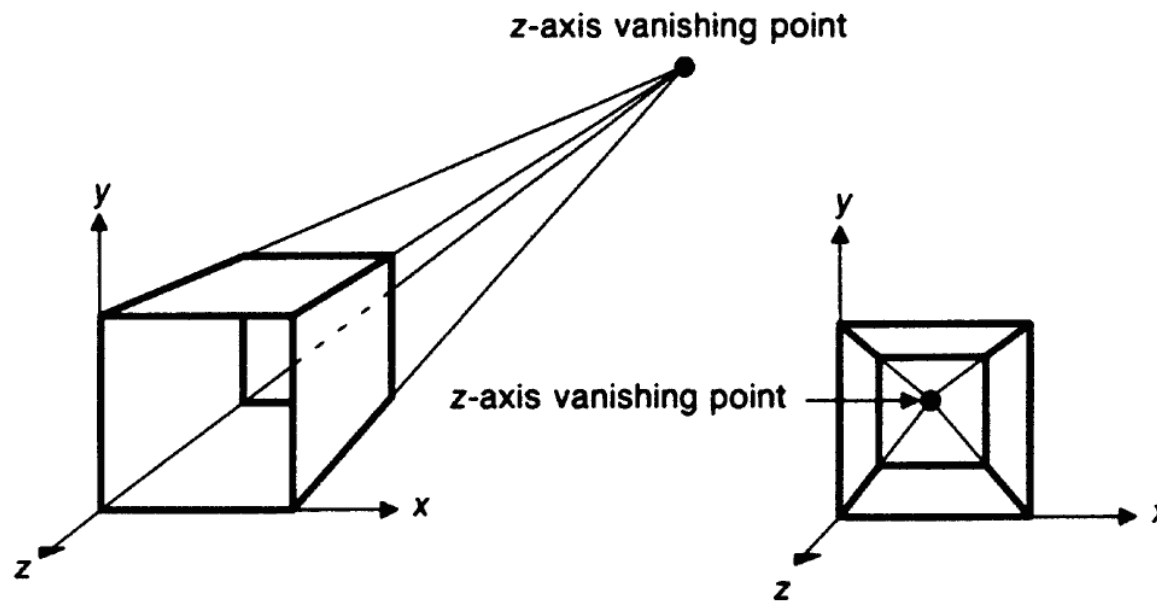
- $\beta = 63.43^\circ$





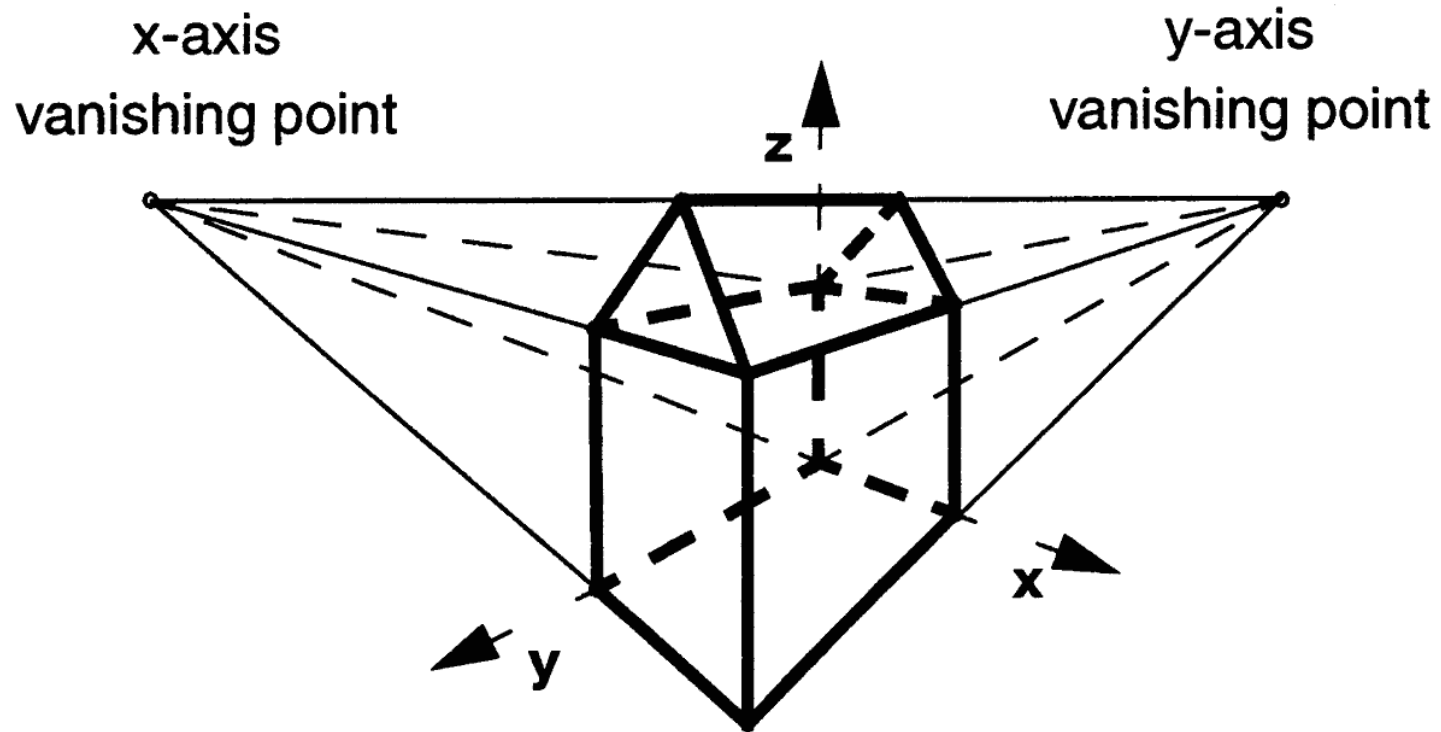
Perspective Projection

- 1, 2, or 3 vanishing points
- Defined by number of intersections between projection plane and coordinate axes



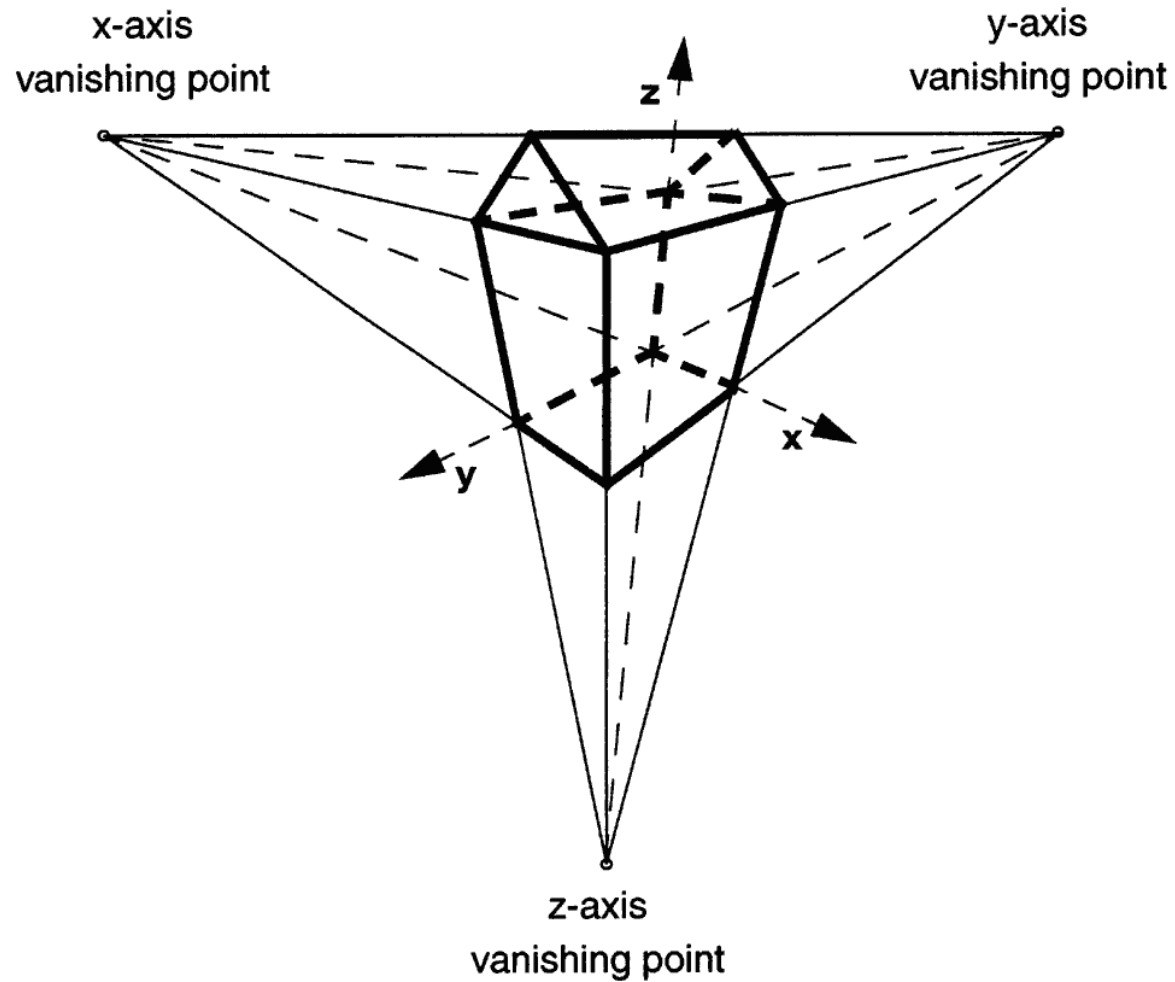


2 Vanishing Points



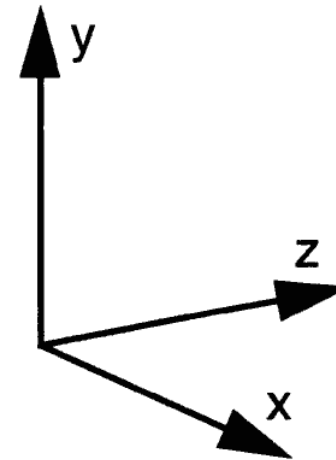
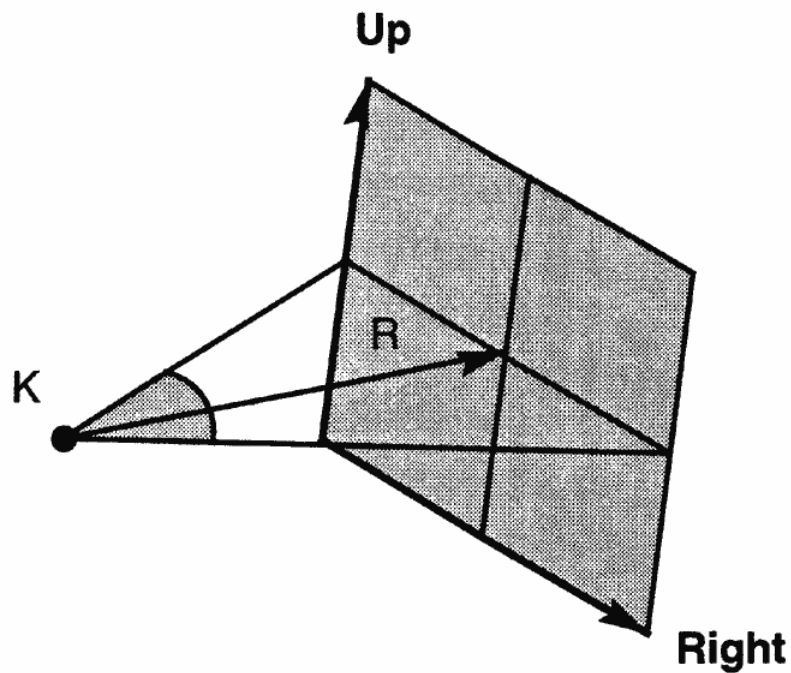


3 Vanishing Points



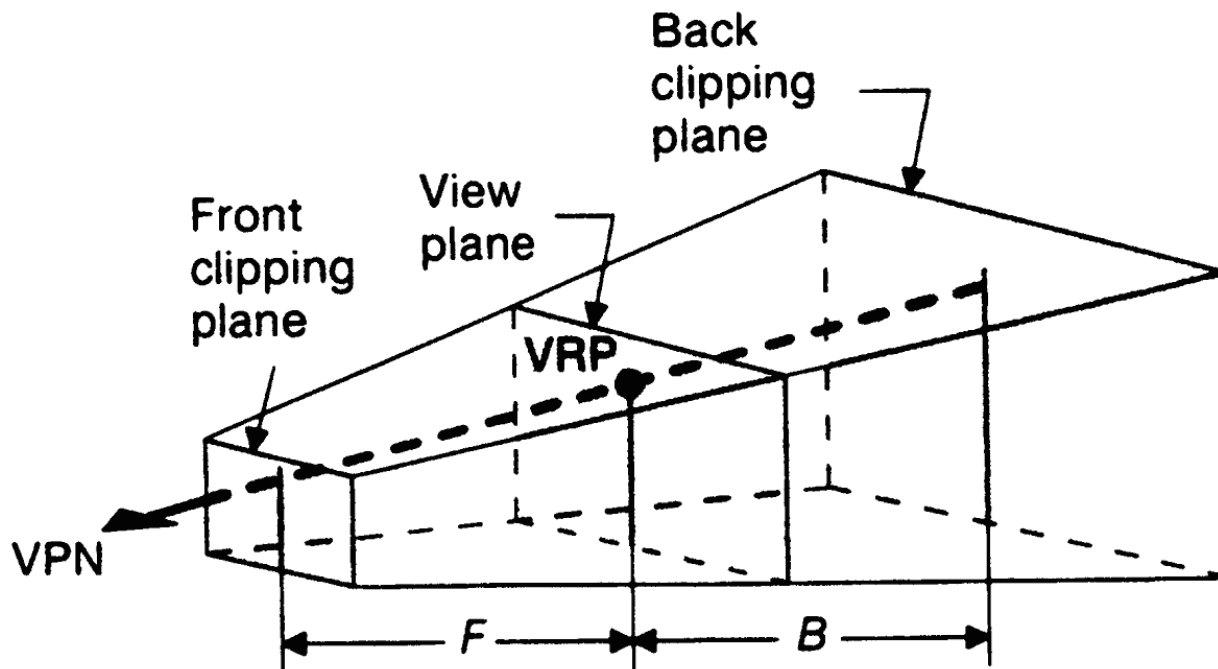


Left-Handed Camera Coordinates



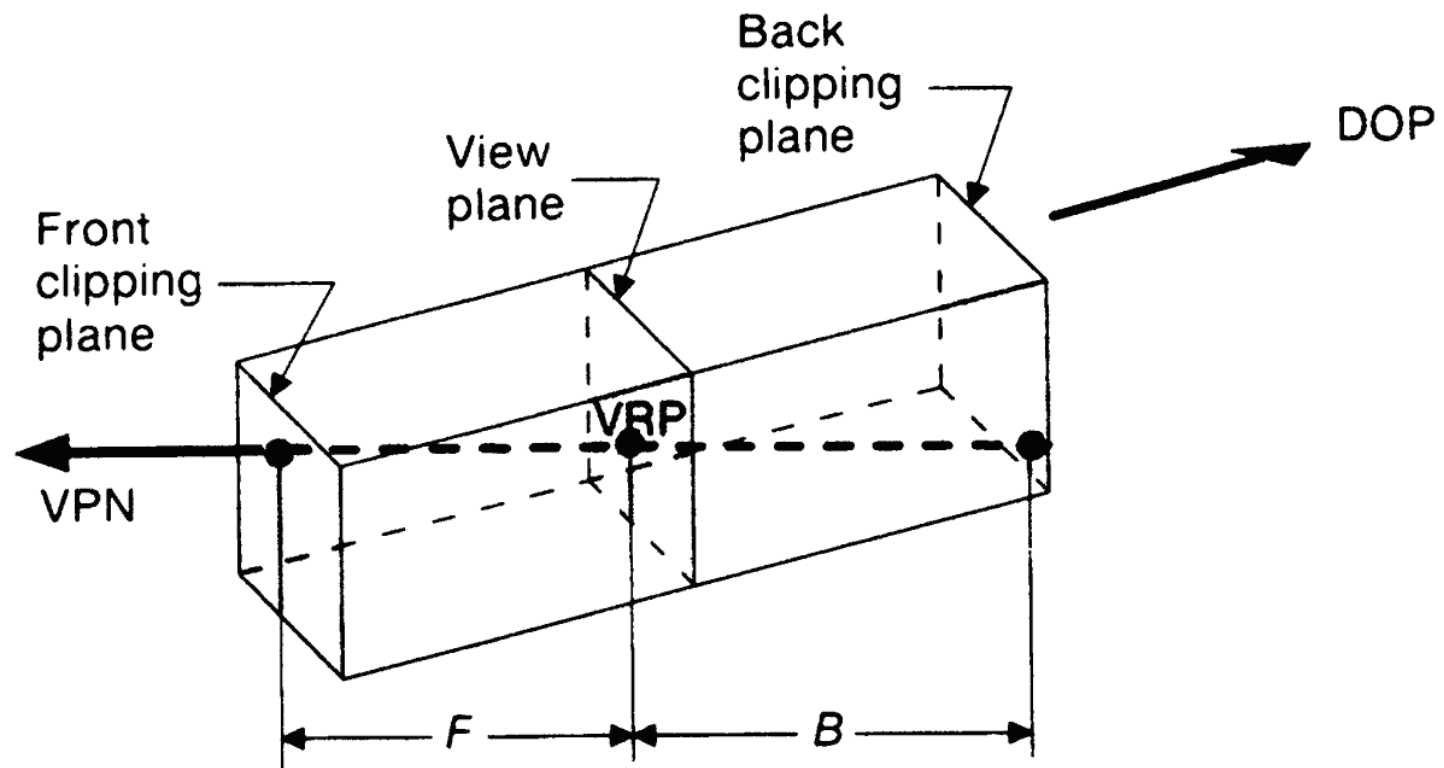


Clipping Planes – Perspective





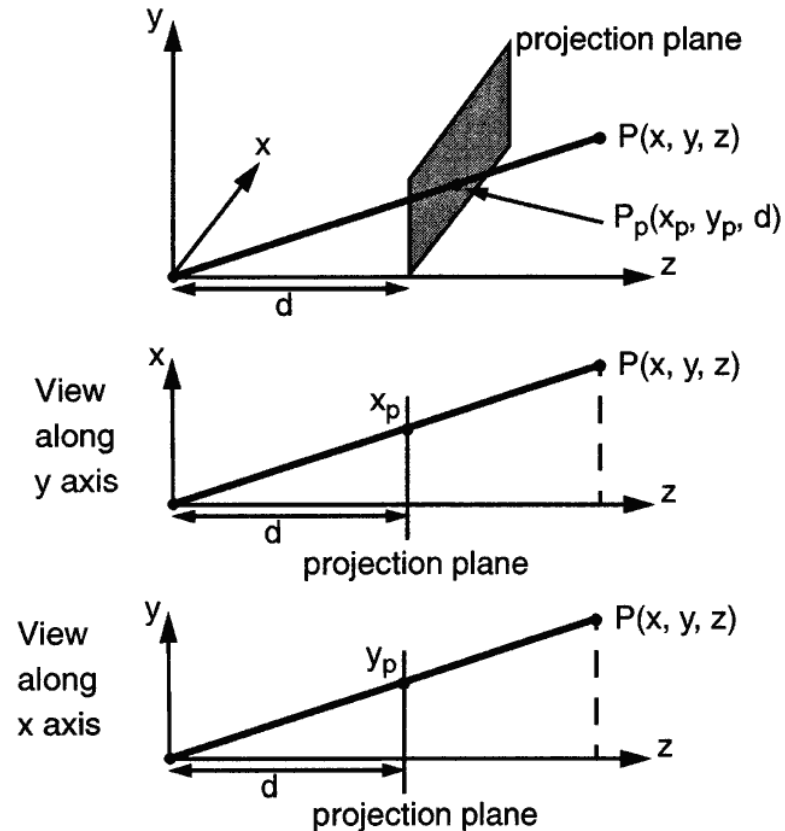
Clipping Planes – Parallel





Mathematics of Projection

- Projection plane parallel to xy -plane at $z=d$





Homogenous Coordinates

- Compute relations

$$x_p = dx/z \quad y_p = dy/z$$

- Homogenous coordinates

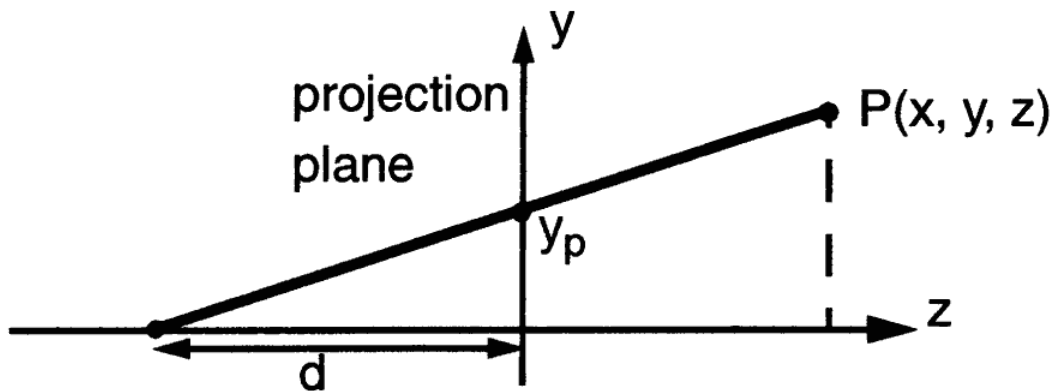
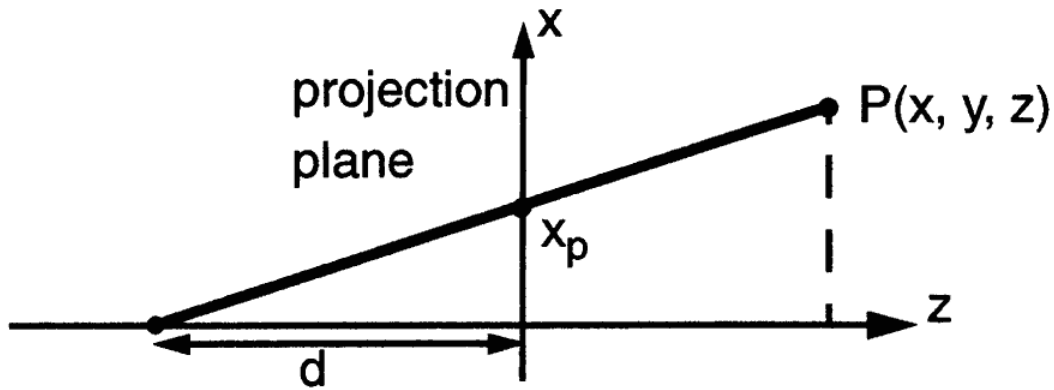
$$\mathbf{M}_{\text{per}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix}$$

- 3D coordinates

$$\left(\frac{X}{W}, \frac{Y}{W}, \frac{Z}{W} \right) = (x_p, y_p, z_p) = \left(\frac{x}{z/d}, \frac{y}{z/d}, d \right)$$



Z=0 Plane



$$\mathbf{M}'_{\text{per}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/d & 1 \end{bmatrix}$$



Parallel Projection

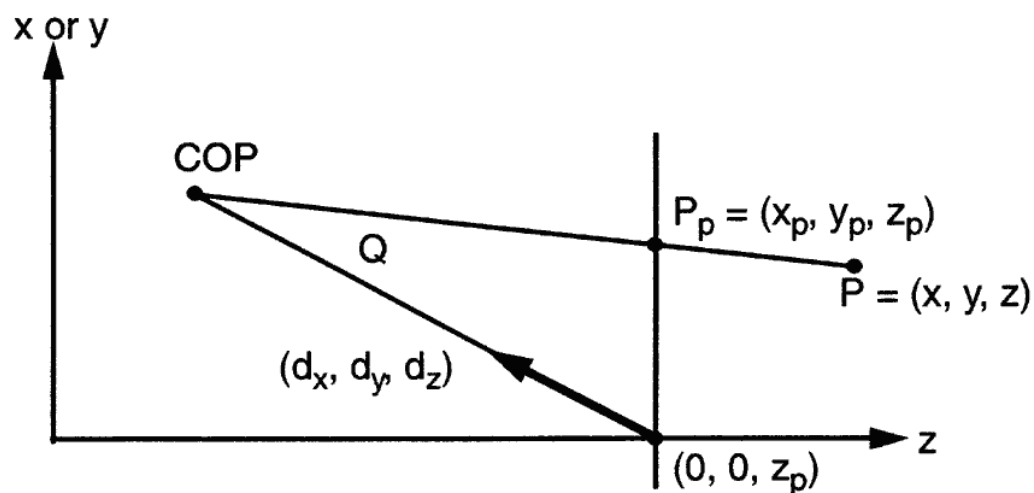
- $d \rightarrow \infty$: yields matrix for parallel projection

$$\mathbf{M}_{\text{ort}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



General Setting

- Arbitrary center of projection (COP)

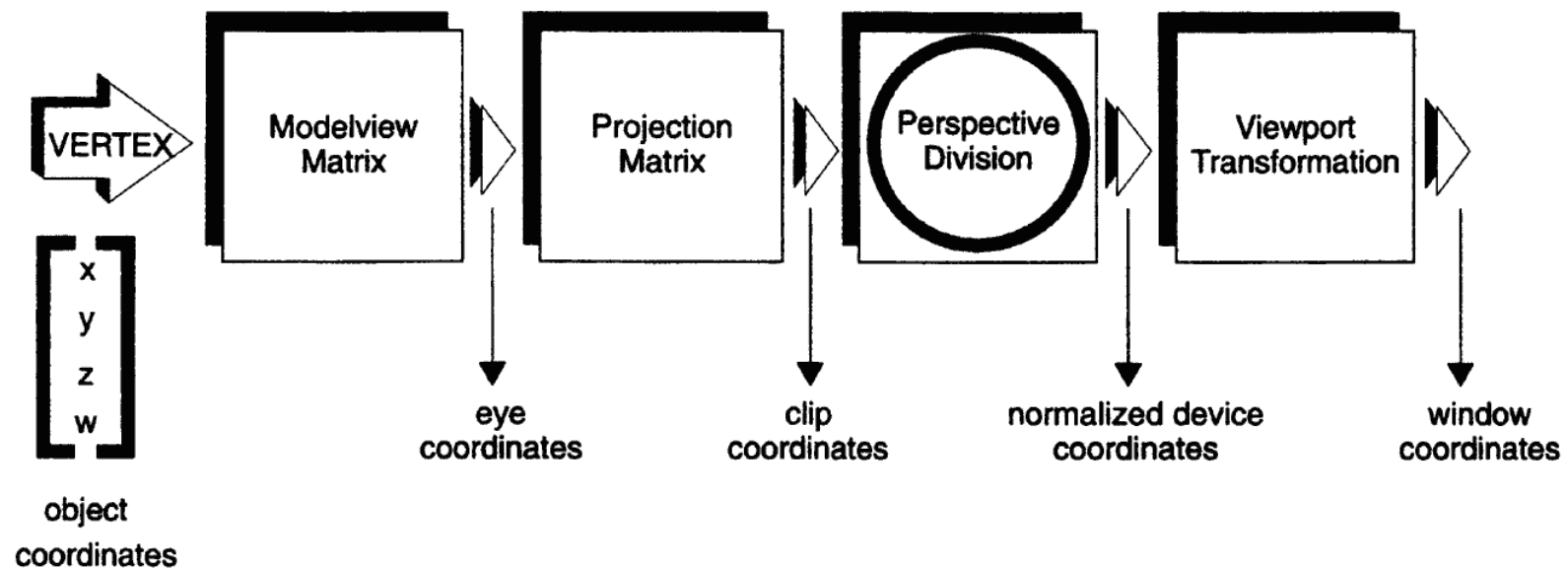


$$\mathbf{M}_{\text{general}} = \begin{bmatrix} 1 & 0 & -\frac{d_x}{d_z} & z_p \frac{d_x}{d_z} \\ 0 & 1 & -\frac{d_y}{d_z} & z_p \frac{d_y}{d_z} \\ 0 & 0 & -\frac{z_p}{Qd_z} & \frac{z_p^2}{Qd_z} + z_p \\ 0 & 0 & -\frac{1}{Qd_z} & \frac{z_p}{Qd_z} + 1 \end{bmatrix}$$



Projections in OpenGL

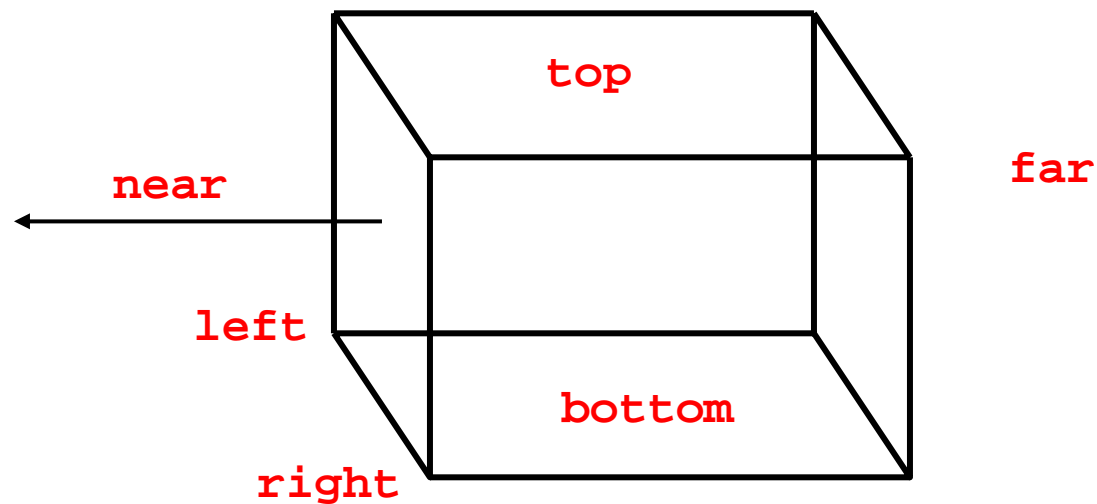
- Stages of Vertex Transformation





Projections in OpenGL

- Parallel Projection (*Orthographic Projection*)

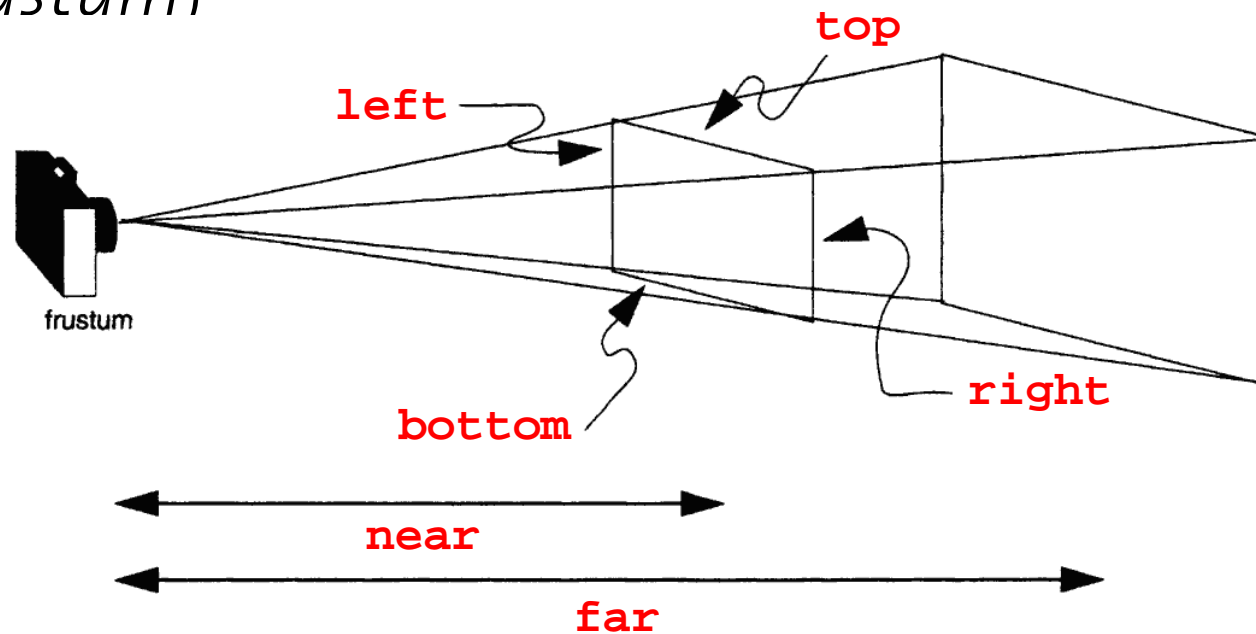


```
glOrtho(left, right, bottom, top, near, far);
```



Projections in OpenGL

- Perspective Projection (*Definition of a Frustum*)

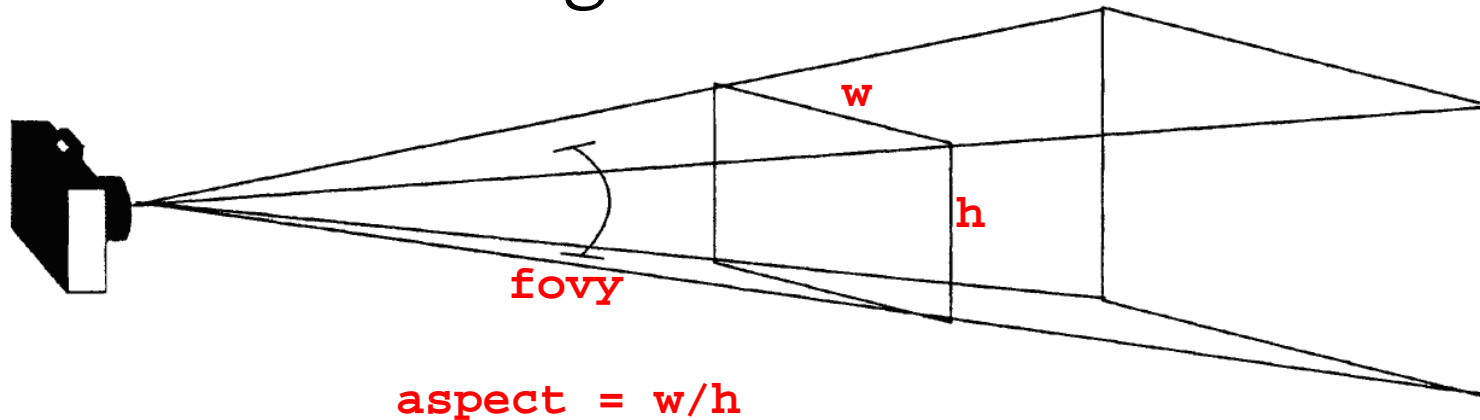


```
glFrustum(left, right, bottom, top, near, far);
```



OpenGL-Utility Functions for Defining Projections

- Camera Analogon

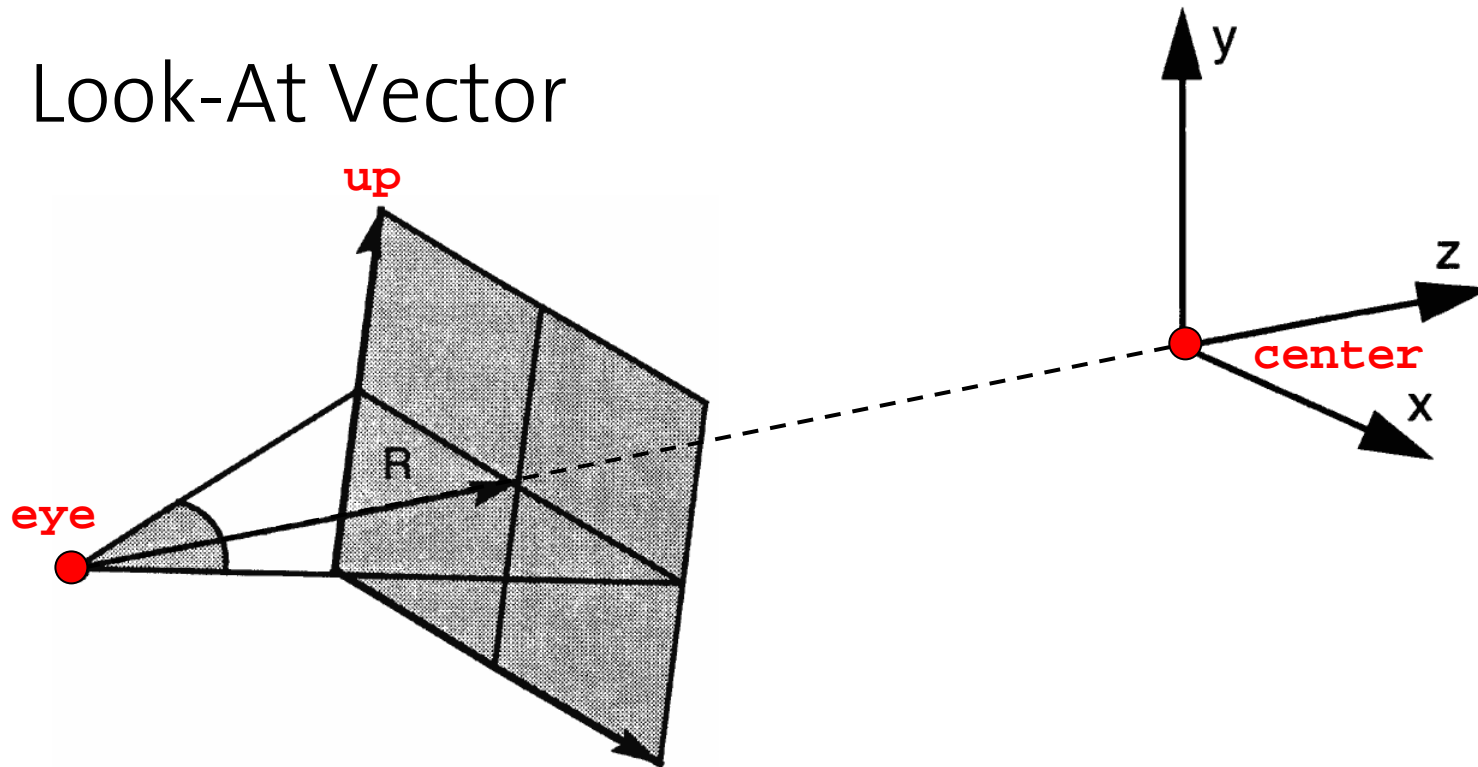


```
gluPerspective(fovy, aspect, zNear, zFar);
```



OpenGL-Utility Functions

- Look-At Vector

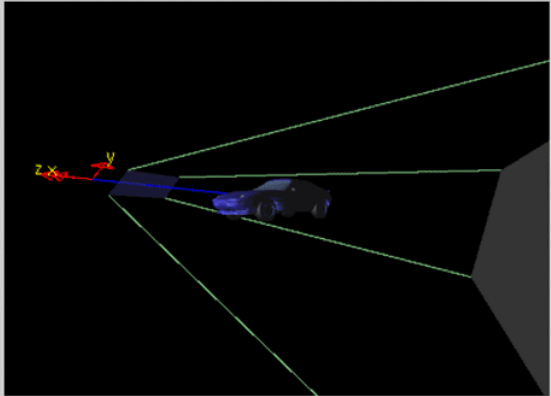


```
gluLookAt(eyeX, eyeY, eyeZ,  
          centerX, centerY, centerZ,  
          upX, upY, upZ);
```

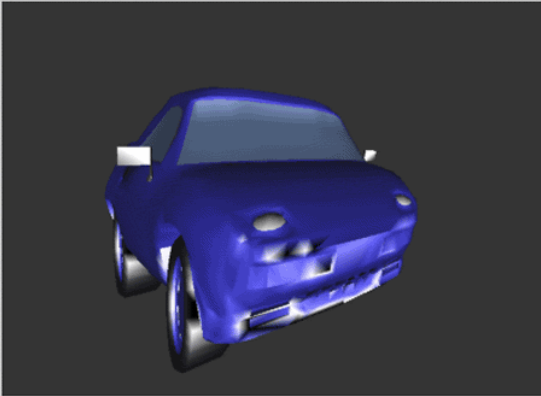



The OpenGL Camera

World-space view



Screen-space view



Command manipulation window

```
glTranslatef( 0.00 , 0.00 , 0.00 );  
glRotatef( 43.0 , 0.34 , 1.00 , 0.00 );  
glScalef( 1.99 , 2.01 , 0.64 );  
glBegin( ... );  
...  
Click on the arguments and move the mouse to modify values.
```

The image shows a software interface for an OpenGL camera. It is divided into three main sections. The top-left section, titled 'World-space view', shows a 3D scene with a blue car and a camera frustum represented by green lines. A coordinate system with x, y, and z axes is visible. The top-right section, titled 'Screen-space view', shows the rendered 2D image of the blue car. The bottom section, titled 'Command manipulation window', displays a list of OpenGL commands with numerical arguments that can be interactively modified. A yellow instruction at the bottom of this window says 'Click on the arguments and move the mouse to modify values.'