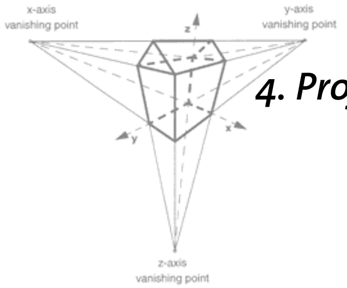
4. Projections

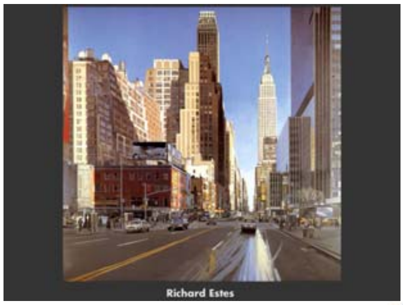


Perspective in Art

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 Univerity

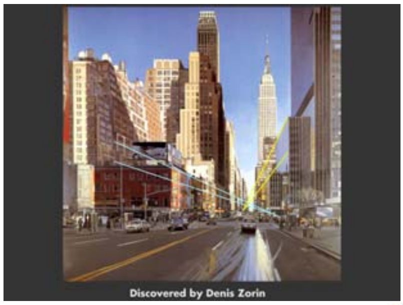
Perspective in Art



Richard Estes

courtesy of Pat Hanrahan, Stanford Univerity

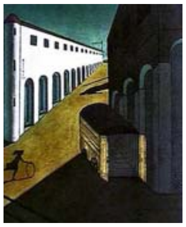
Perspective in Art



Discovered by Denis Zorin

courtesy of Pat Hanrahan, Stanford Univerity

Perspective in Art



Giorgio de Chrico: Mystery and Melancholy of a Street

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

Perspective in Art

Raphael: School of Athens

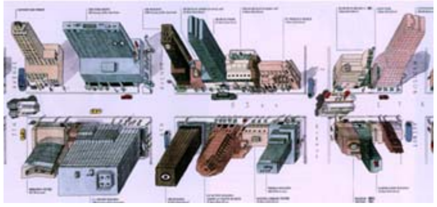
from: Agrawala, Zorin, Munzner: Artistic Multiprojection Rendering, EGWR 2000

Perspective in Art

Cezanne: Still Life with Fruit Basked

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

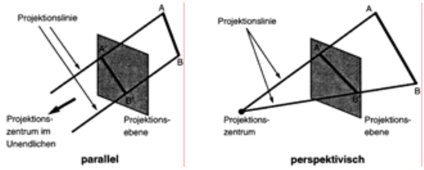
Perspective in Art



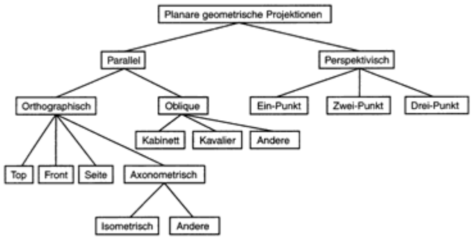
from: Agrawala, Zorin, Munzner: Artistic Multiprojection Rendering, EGWR 2000

Parallel vs. Perspective Projection

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

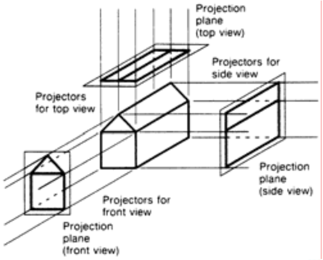


Classification



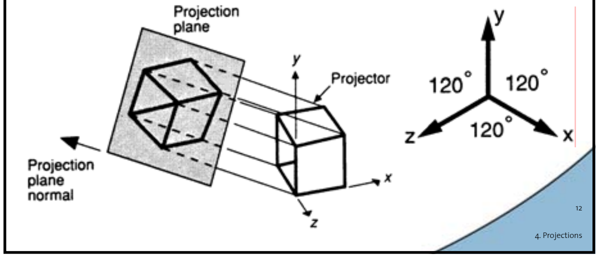
Orthographic Projection

- Front-, top-, and side views


Projection plane (top view)
Projectors for side view
Projectors for top view
Projection plane (side view)
Projectors for front view
Projection plane (front view)

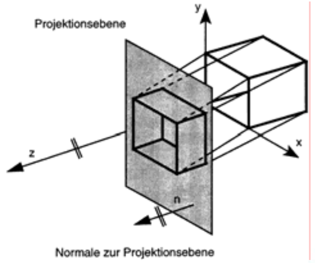
Isometric Projection

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

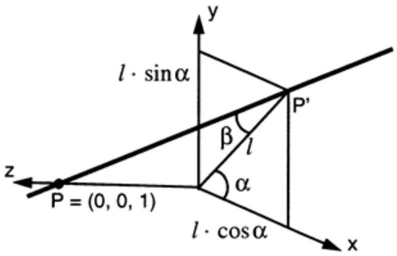


Oblique Projections

- Normal ≠ Direction of Projection

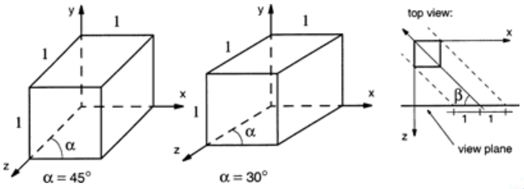


Direction of Projection



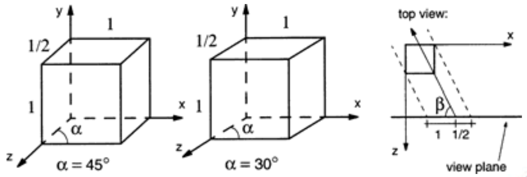
"Kavalier" Projection

- β = 45°



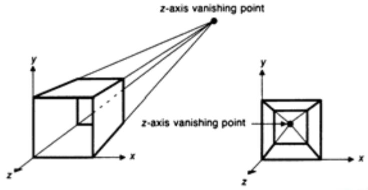
"Kabinett" Projection

- β = 63.43°

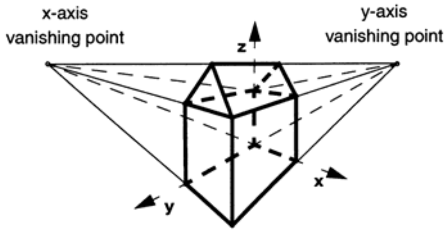


Perspective Projection

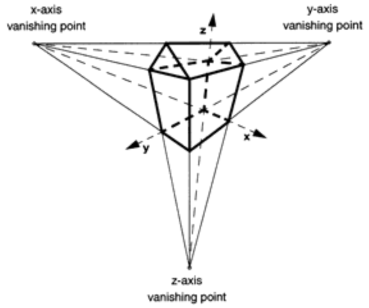
- 1, 2, or 3 vanishing points
- Defined by number of intersections between projection plane and coordinate axes



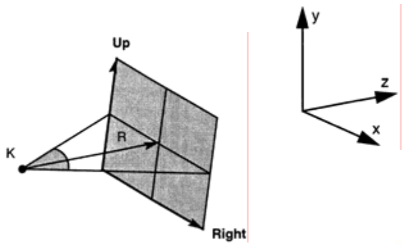
2 Vanishing Points



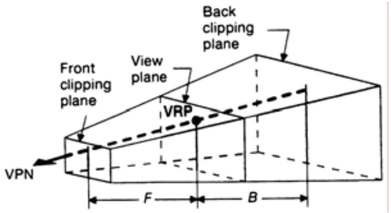
3 Vanishing Points



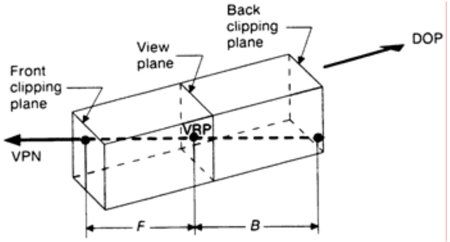
Left-Handed Camera Coordinates



Clipping Planes – Perspective

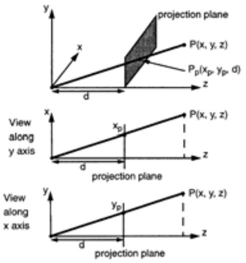


Clipping Planes – Parallel



Mathematics of Projection

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



Homogenous Coordinates

- Compute relations

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

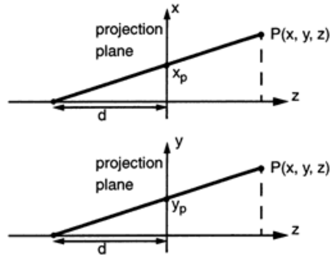
- Homogenous coordinates

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

- 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



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

Parallel Projection

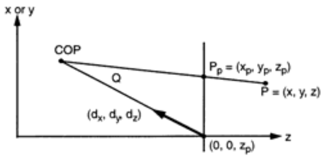
- $d \rightarrow \infty$: yields matrix for parallel projection

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

General Setting

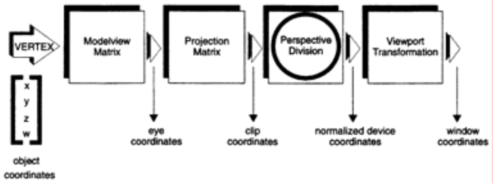
- Arbitrary center of projection (COP)

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



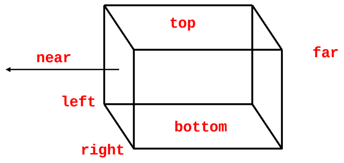
Projections in OpenGL

- Stages of Vertex Transformation



Projections in OpenGL

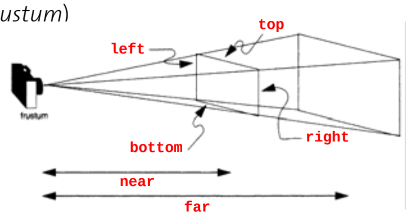
- Parallel Projection (*Orthographic Projection*)



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

Projections in OpenGL

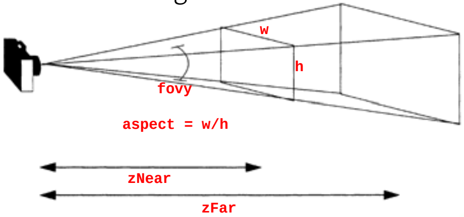
- Perspective Projection (*Definition of a Frustum*)



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

OpenGL-Utilitity Functions for Defining Projections

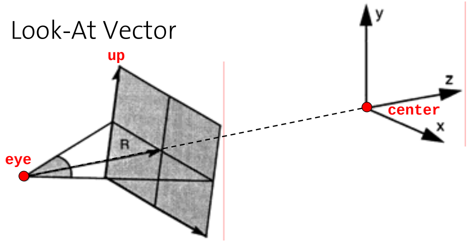
- Camera Analogon



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

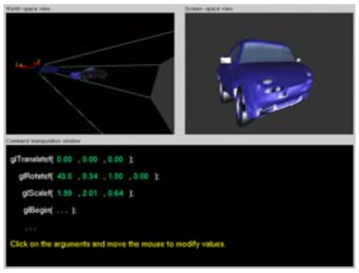
OpenGL-Utilitity Functions

- Look-At Vector



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

The OpenGL Camera



```
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.