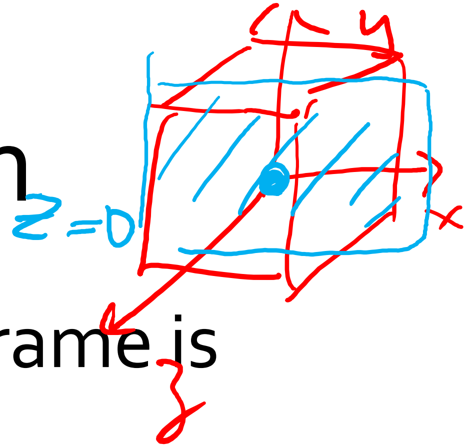
# Projections and Normalization




The default projection in the eye (camera) frame is orthogonal

For points within the default view volume

3D pt

$(x, y, z)$

$$x_p = x$$
$$y_p = y$$
$$z_p = 0$$

Projection plane at z = 0

Most graphics systems use view normalization

- All other views are converted to the canonical view by transformations that determine the projection matrix
- Allows use of the same pipeline for all views

# Default orthographic projection

$$\mathbf{p}_\mathrm{p} = \mathbf{M}\mathbf{p}$$

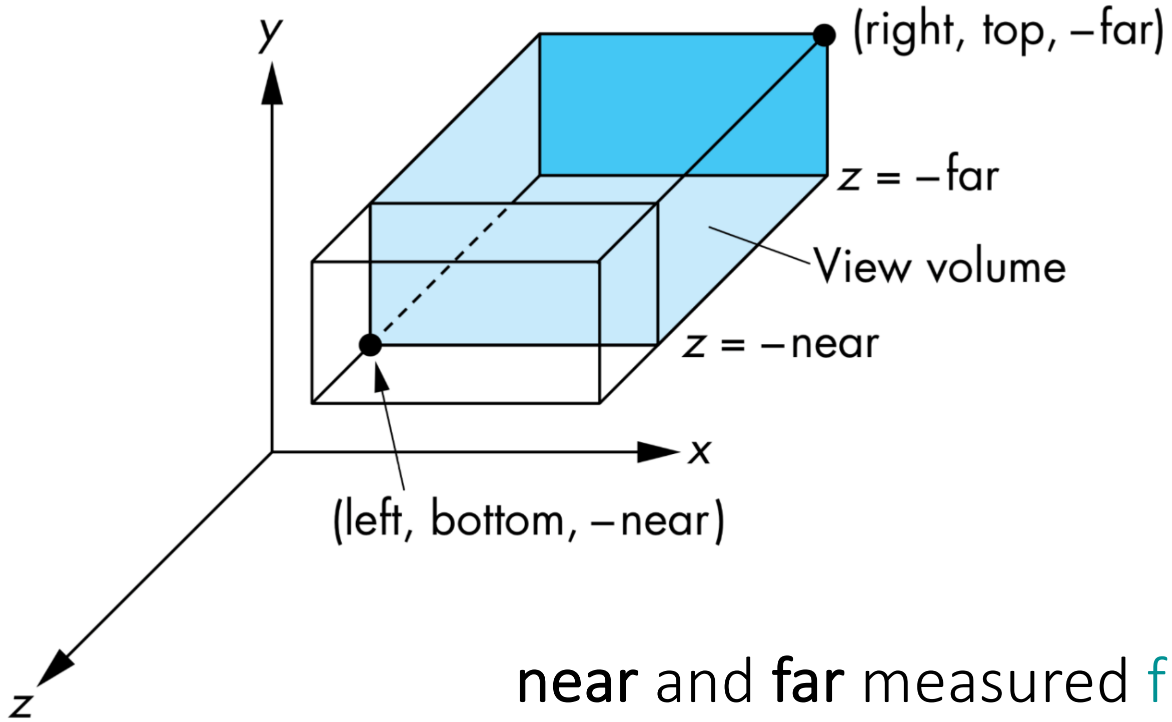
$$\mathbf{M}_{\mathbf{orth}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{p} = (x, y, z, 1)^\mathrm{T}$$

$$\mathbf{p}_\mathrm{p} = (x, y, 0, 1)^\mathrm{T}$$

# OpenGL Orthogonal Viewing

```
Ortho(left,right,bottom,top,near,far)
```

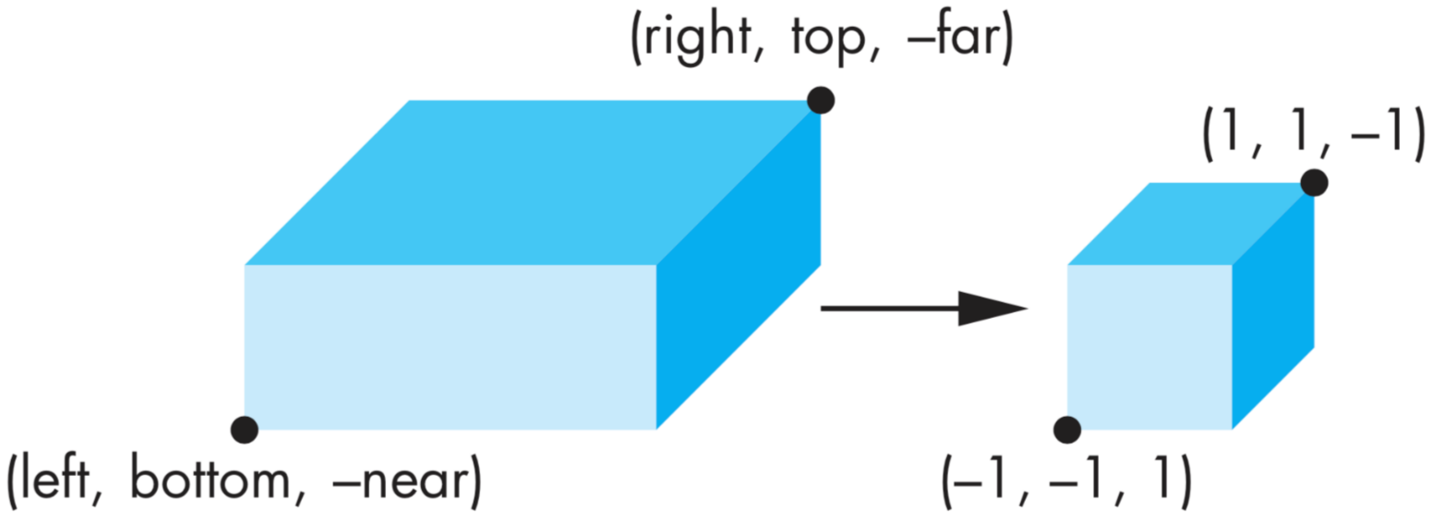



**near** and **far** measured from camera

How to normalize this into the canonical view?

# Orthogonal Normalization

**normalization** $\Rightarrow$ find transformation to convert specified clipping volume to canonical volume



# Orthogonal Matrix

Two steps

- Move center to origin
  - T(-(left+right)/2, -(bottom+top)/2, (near+far)/2))
- Scale to have sides of length 2
  - S(2/(left-right), 2/(top-bottom), 2/(near-far))

$$\mathbf{P} = \mathbf{ST} = \begin{bmatrix} \frac{2}{right - left} & 0 & 0 & -\frac{right + left}{right - left} \\ 0 & \frac{2}{top - bottom} & 0 & -\frac{top + bottom}{top - bottom} \\ 0 & 0 & \frac{2}{near - far} & \frac{far + near}{far - near} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

# Final Projection

Set $z = 0$

Equivalent to the homogeneous coordinate transformation

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

Hence, general orthogonal projection in 4D is

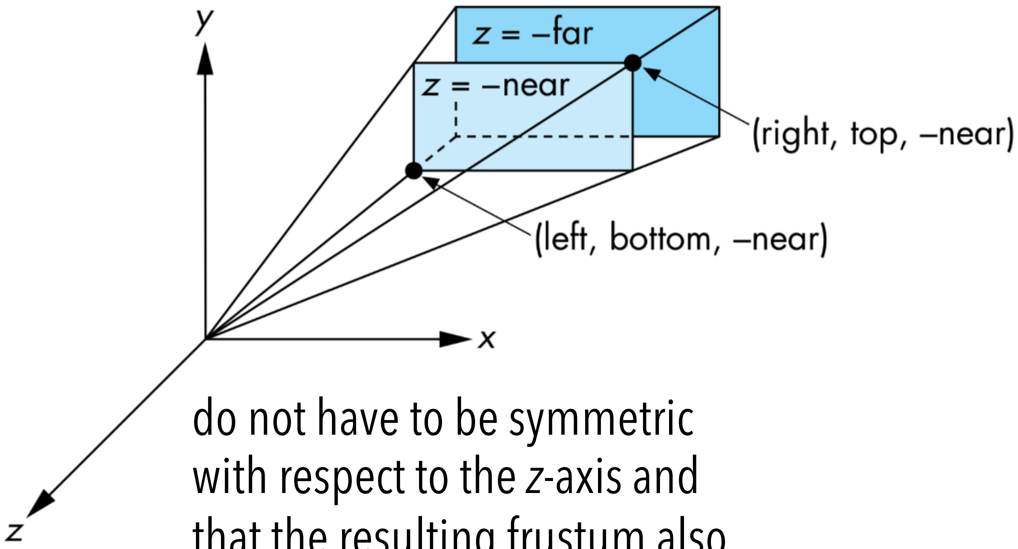
$$\mathbf{P} = \mathbf{M}_{\text{orth}}\mathbf{ST}$$

fro arbitrary orthogonal view
to canonical view

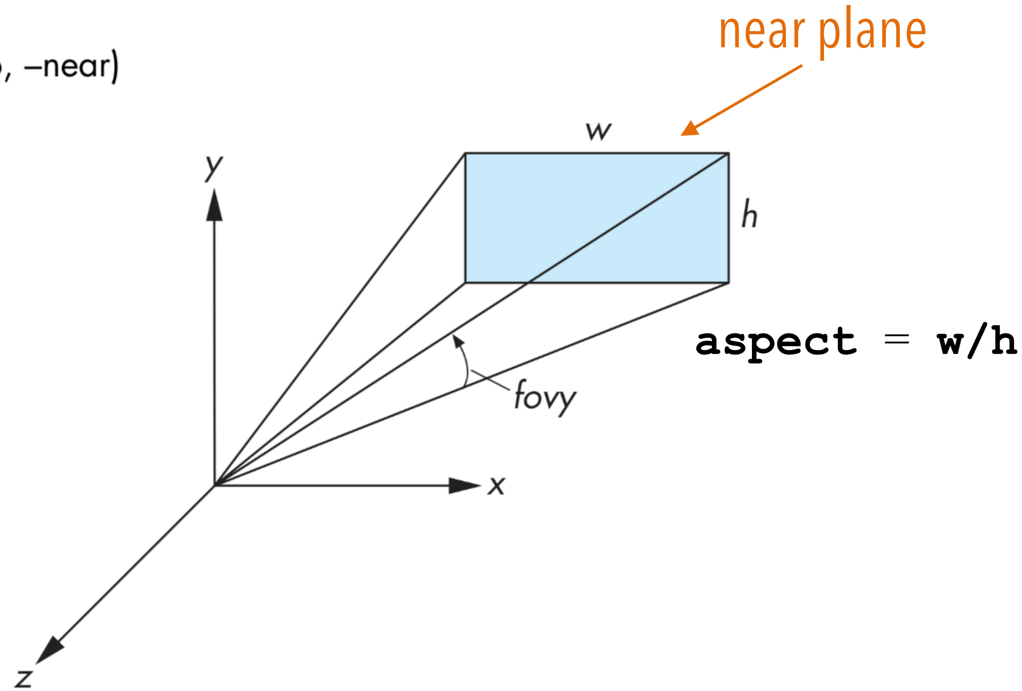
to project a 3D pt onto the view plane z = 0

# OpenGL Perspective Viewing

**`Frustum(left,right,bottom,top,near,far)`**




do not have to be symmetric
with respect to the *z*-axis and
that the resulting frustum also
does not have to be symmetric



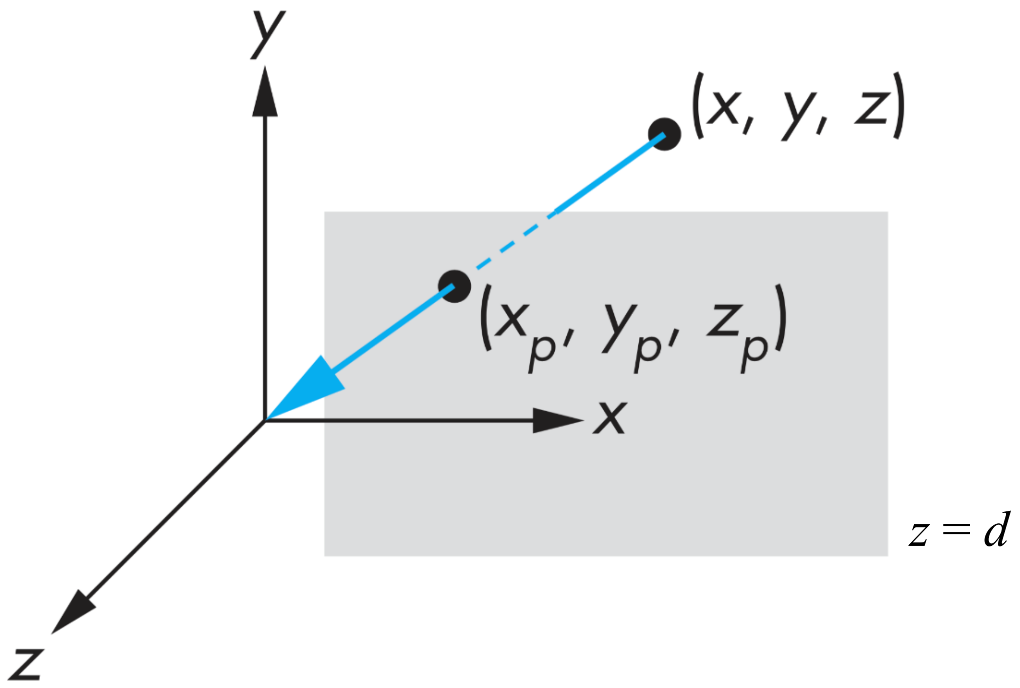

**`Perpective(fovy, aspect, near, far)`**

**`fovy`**: field of view in degrees in y direction
this often provides a better interface

# Simple Perspective

Center of projection at the origin

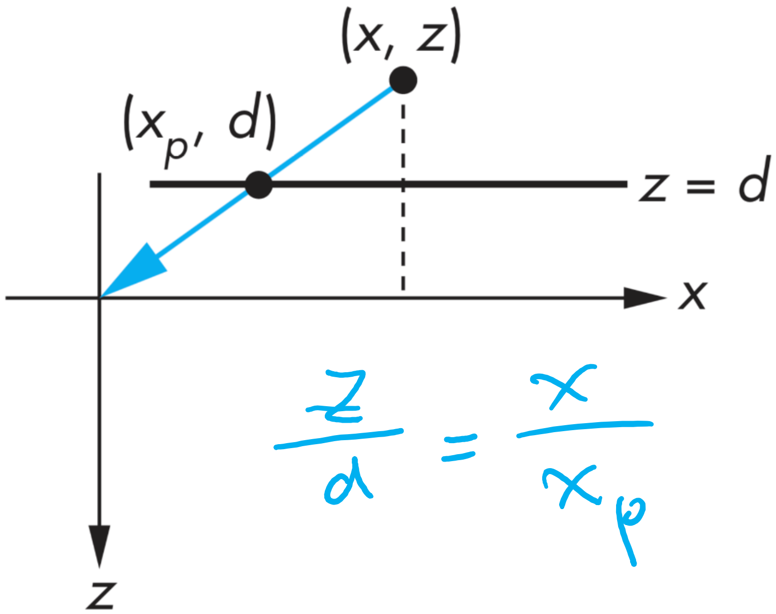
Projection plane $z = d,\ d < 0$



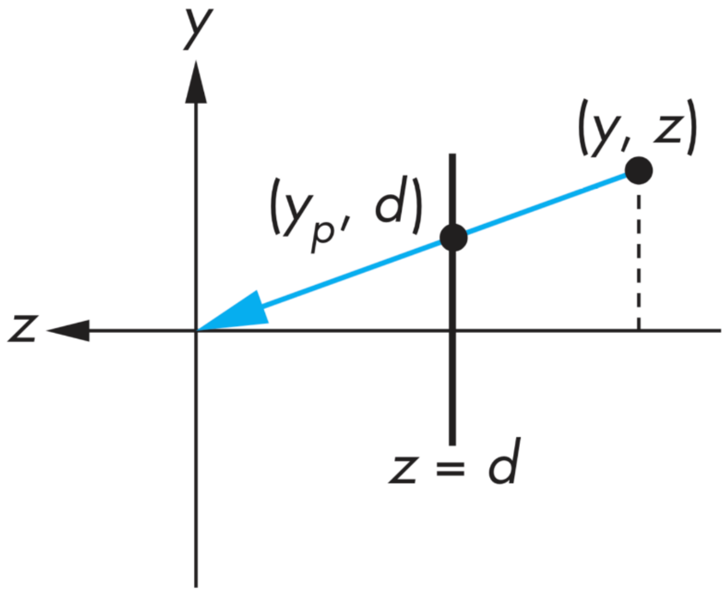
# Perspective Equations

$$\begin{pmatrix} x_p \\ y_p \\ z_p \\ 1 \end{pmatrix} = M \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Consider top and side views




$$\frac{z}{d} = \frac{x}{x_p}$$




$$x_p = \frac{x}{z/d} \qquad y_p = \frac{y}{z/d} \qquad z_p = d$$

# Homogeneous Coordinate Form

Consider $\mathbf{p} = \mathbf{Mq}$ where

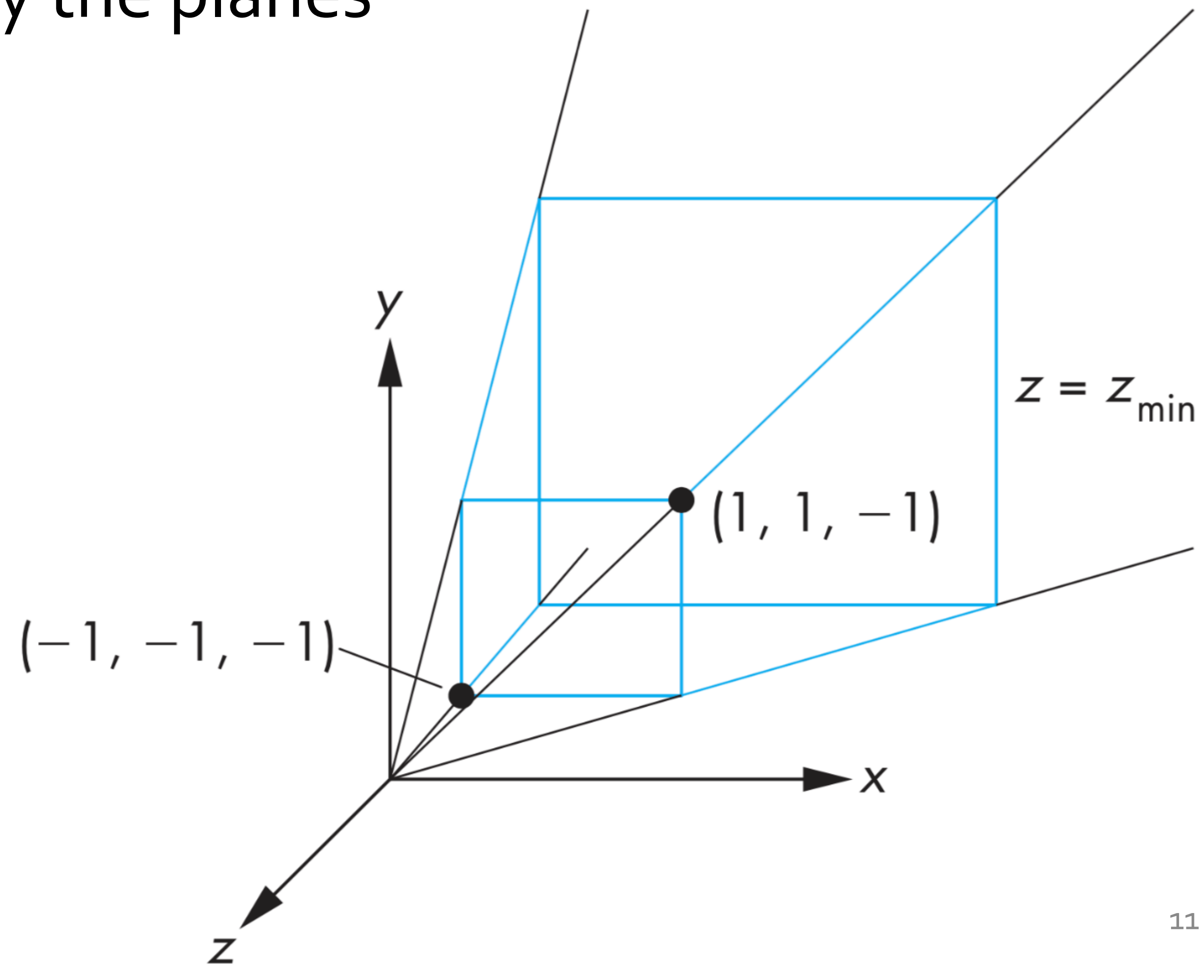
$$\mathbf{M} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \\ z/d \end{pmatrix}$$

$$\mathbf{q} = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \Rightarrow \mathbf{p} = \begin{pmatrix} x \\ y \\ z \\ z/d \end{pmatrix} = \begin{pmatrix} \frac{x}{z/d} \\ \frac{y}{z/d} \\ \frac{z}{z/d} = d \\ 1 \end{pmatrix}$$

# Simple Perspective

Consider a simple perspective with the COP at the origin, the near clipping plane at $z$ = -1, and a 90 degree field of view determined by the planes

$$x = \pm z, y = \pm z$$



# Perspective Matrices

Simple projection matrix in homogeneous coordinates

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

Note that this matrix is independent of the far clipping plane

# Generalization

$d = -1$

$$\mathbf{N} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \alpha & \beta \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

By this mapping, the point $(x, y, z, 1)$ goes to

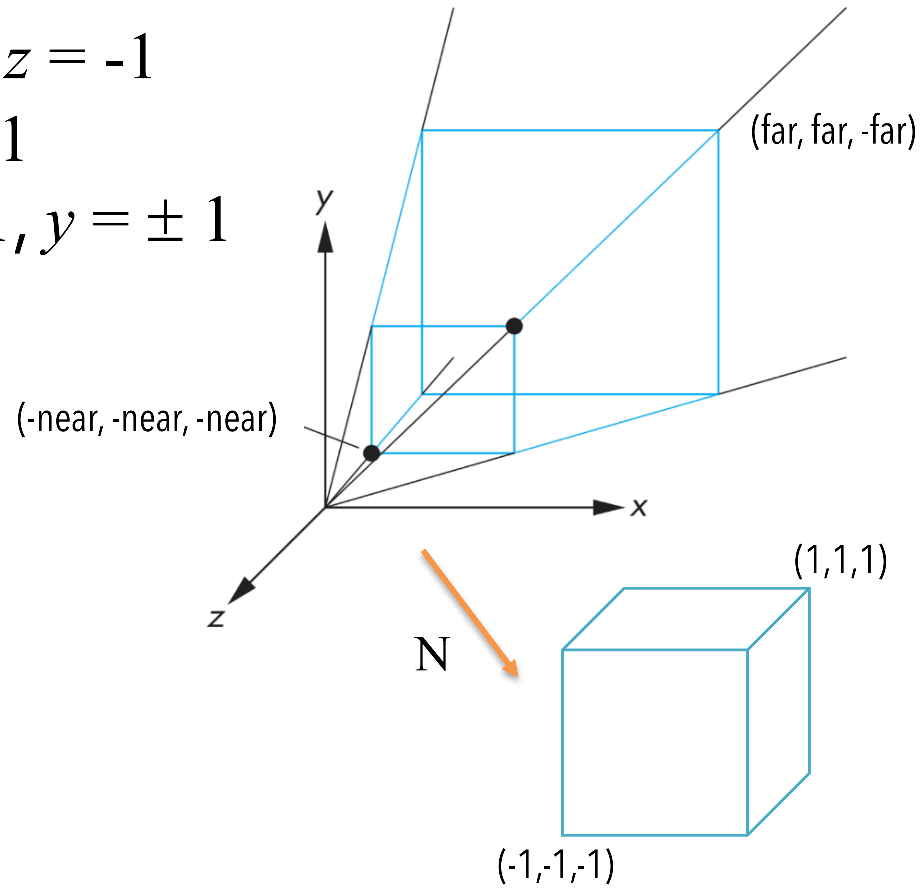
$$x'' = -x/z$$
$$y'' = -y/z$$
$$z'' = -(\alpha+\beta/z)$$

which projects orthogonally to the desired point regardless of $\alpha$ and $\beta$

# Picking α and β

$$\begin{pmatrix} -1 \\ -1 \\ -1 \\ 1 \end{pmatrix} = N \cdot \begin{pmatrix} -n \\ -n \\ -n \\ 1 \end{pmatrix}$$

We want:
near plane $z$ = -near be mapped to $z$ = -1
far plane $z$ = -far be mapped to $z$ =1
and the sides be mapped to $x = \pm 1$, $y = \pm 1$



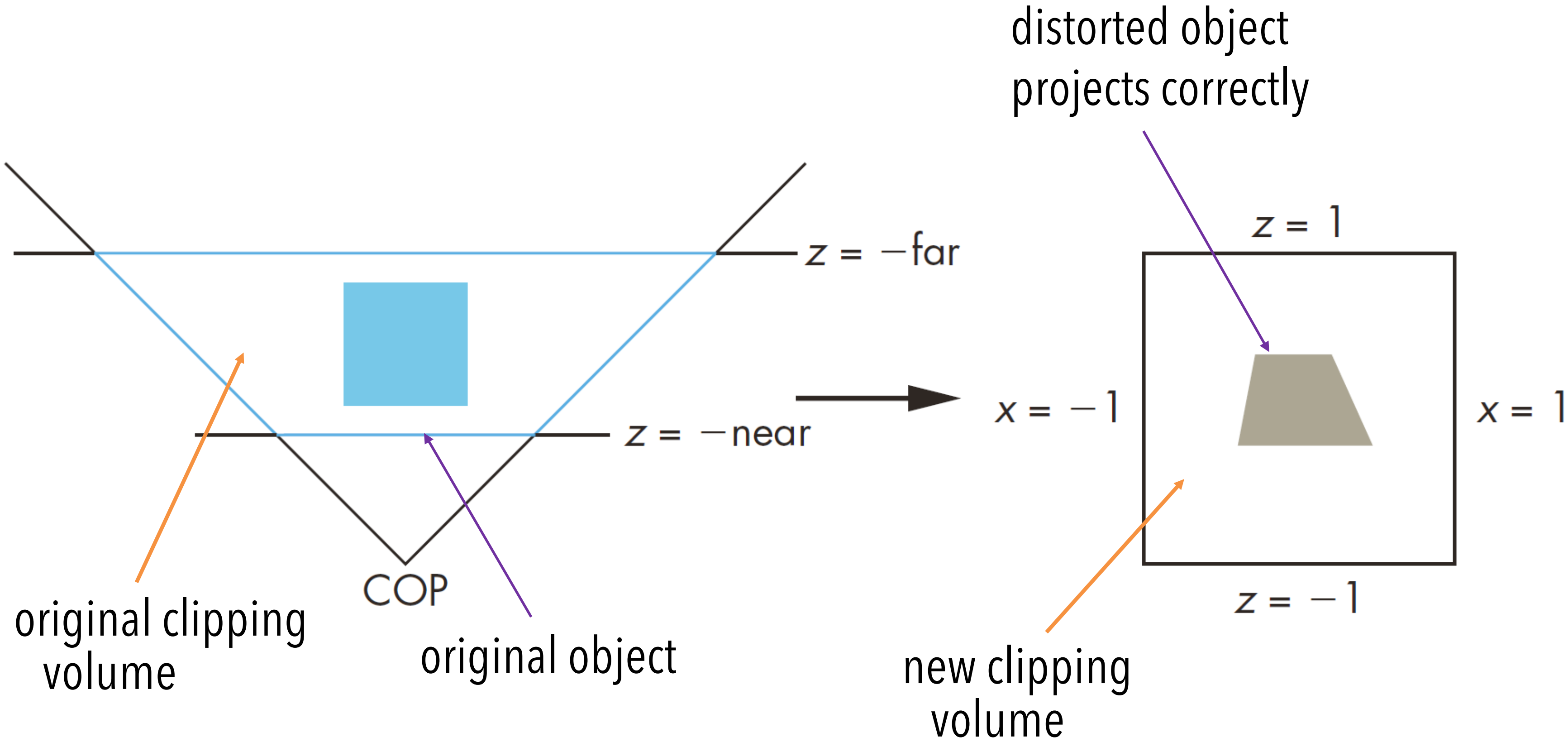

Solving two linear equations,
we have

$$\alpha = \frac{\text{near} + \text{far}}{\text{near} - \text{far}}$$

$$\beta = \frac{2 \times \text{near} \times \text{far}}{\text{near} - \text{far}}$$

Then the new clipping volume is the canonical clipping volume

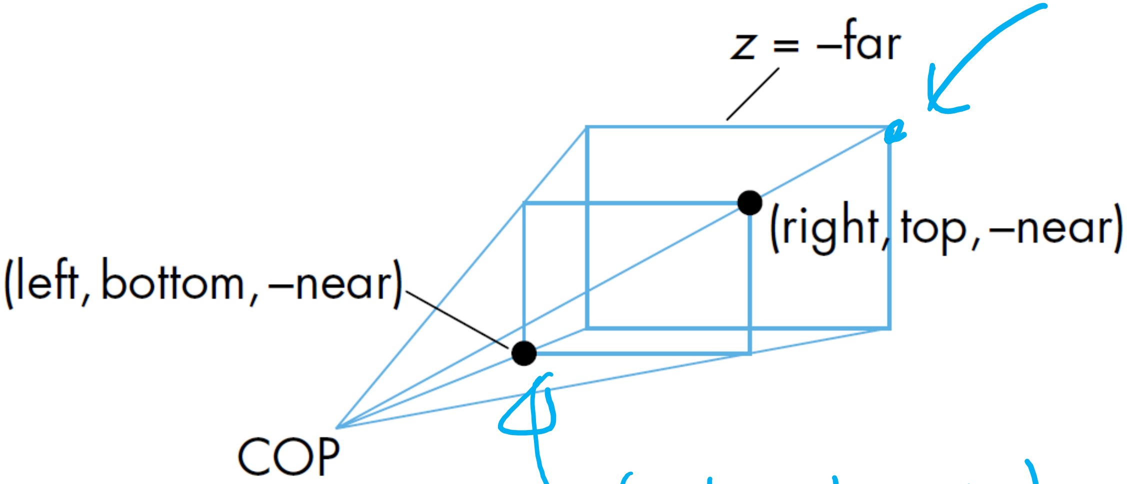
# Normalization Transformation



# OpenGL Perspective

glFrustum allows for an unsymmetric viewing frustum (although Perspective does not)

$(-f, -f, -f)$




$(-n, -n, -n)$

An unsymmetric viewing frustum can be normalized to the canonical view volume by first apply a shear and a scaling before applying N

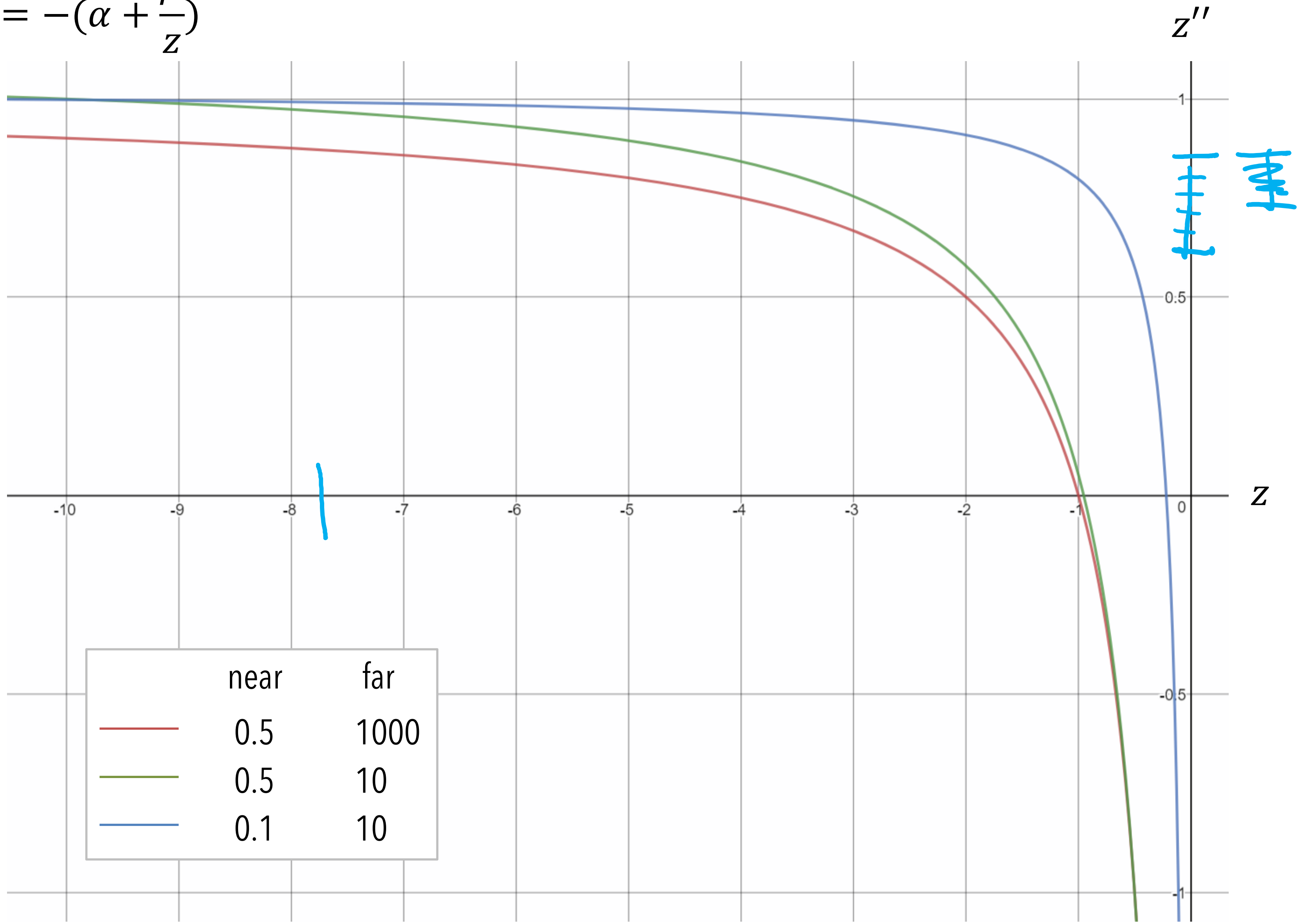
# Normalization and Hidden-Surface Removal

Although our selection of the form of the perspective matrices may appear somewhat arbitrary, it was chosen so that if $z_1 > z_2$ in the original clipping volume then the for the transformed points $z_1' > z_2'$

Thus hidden surface removal works if we first apply the normalization transformation

However, note that the formula $z'' = -(\alpha+\beta/z)$ is nonlinear, which implies that the distances are distorted by the normalization which can cause numerical problems especially if the near distance is small

$$z'' = -(\alpha + \frac{\beta}{z})$$



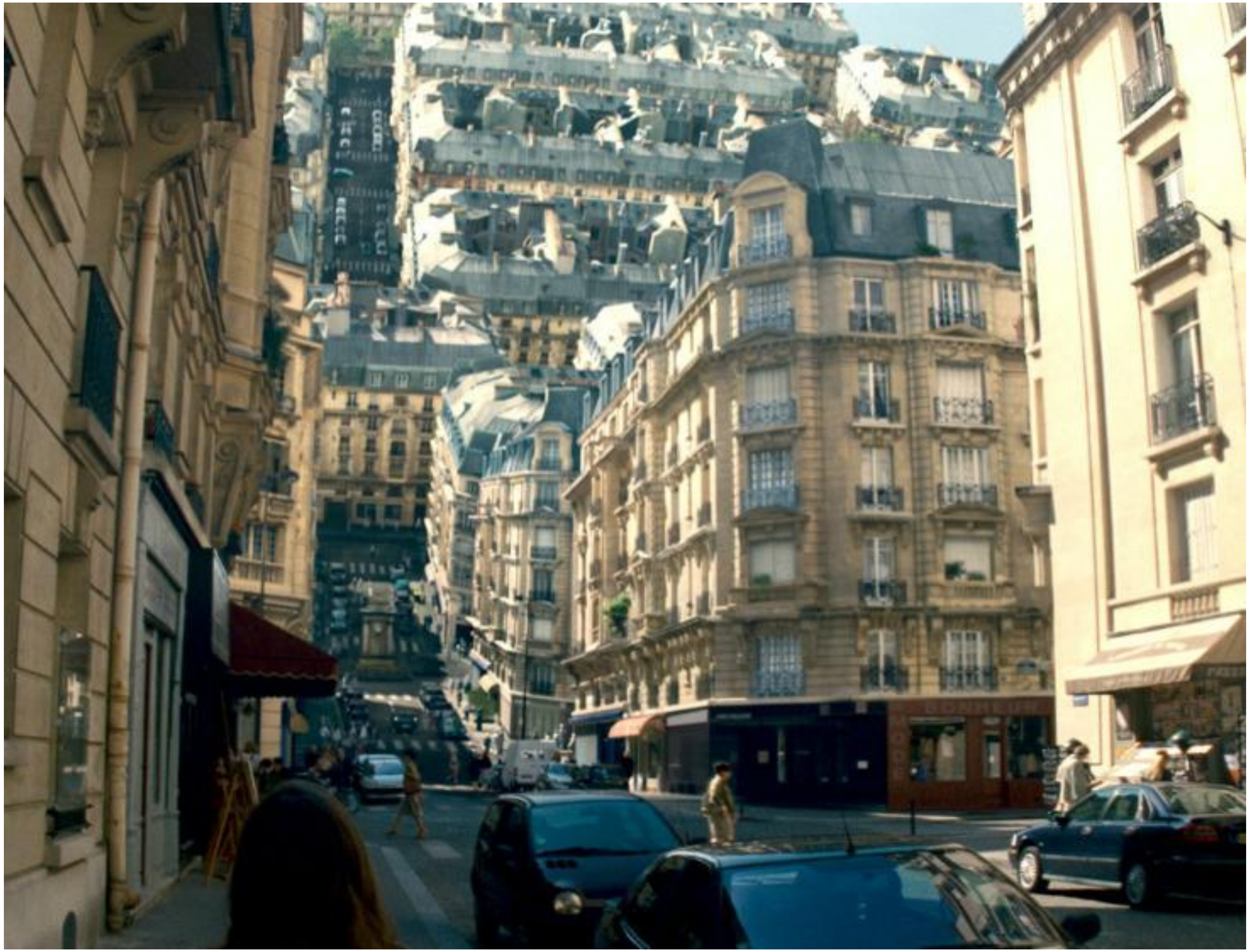
# Why do we do it this way?

Normalization allows for a single pipeline for both perspective and orthogonal viewing

We stay in four dimensional homogeneous coordinates as long as possible to retain three-dimensional information needed for hidden-surface removal and shading

We simplify clipping

# Special Projection Effects



The Movie “Inception”