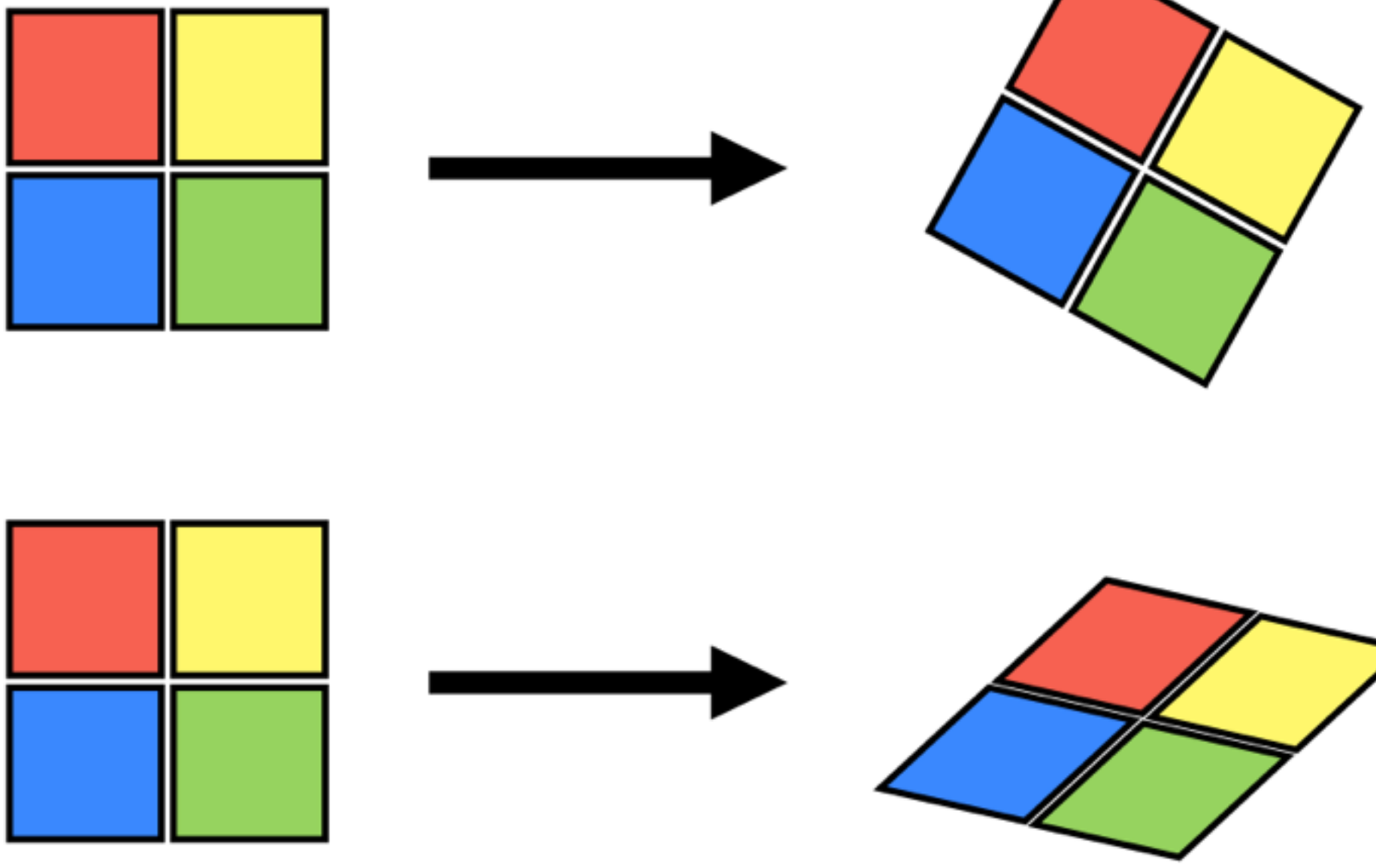
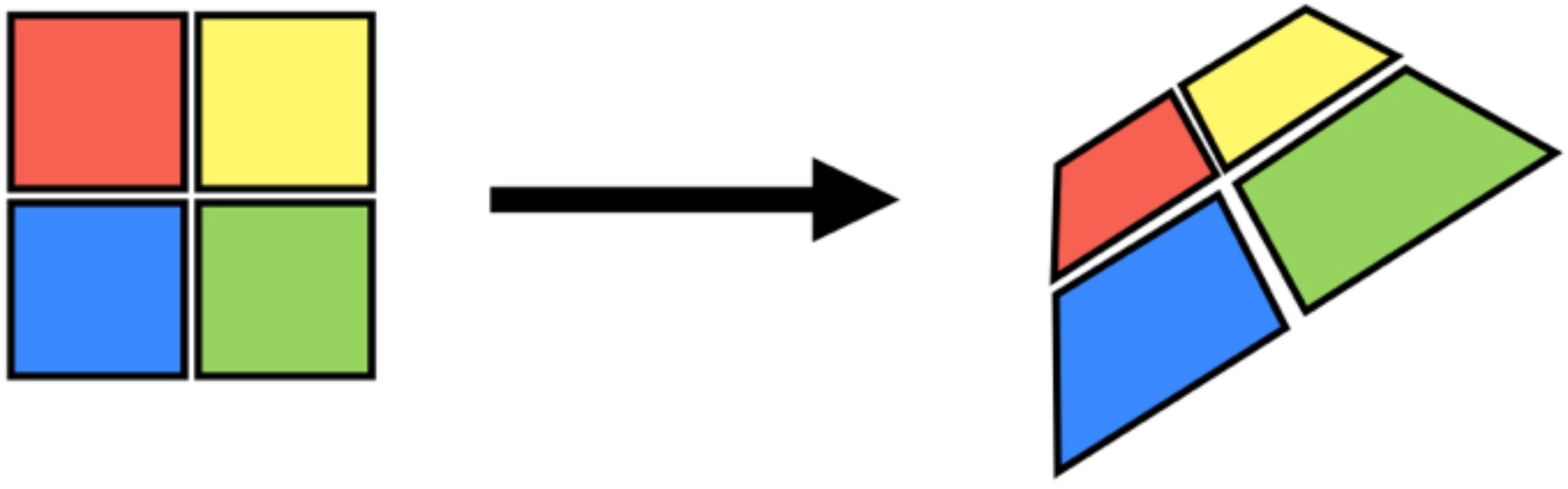
Perspective Viewing



rigid

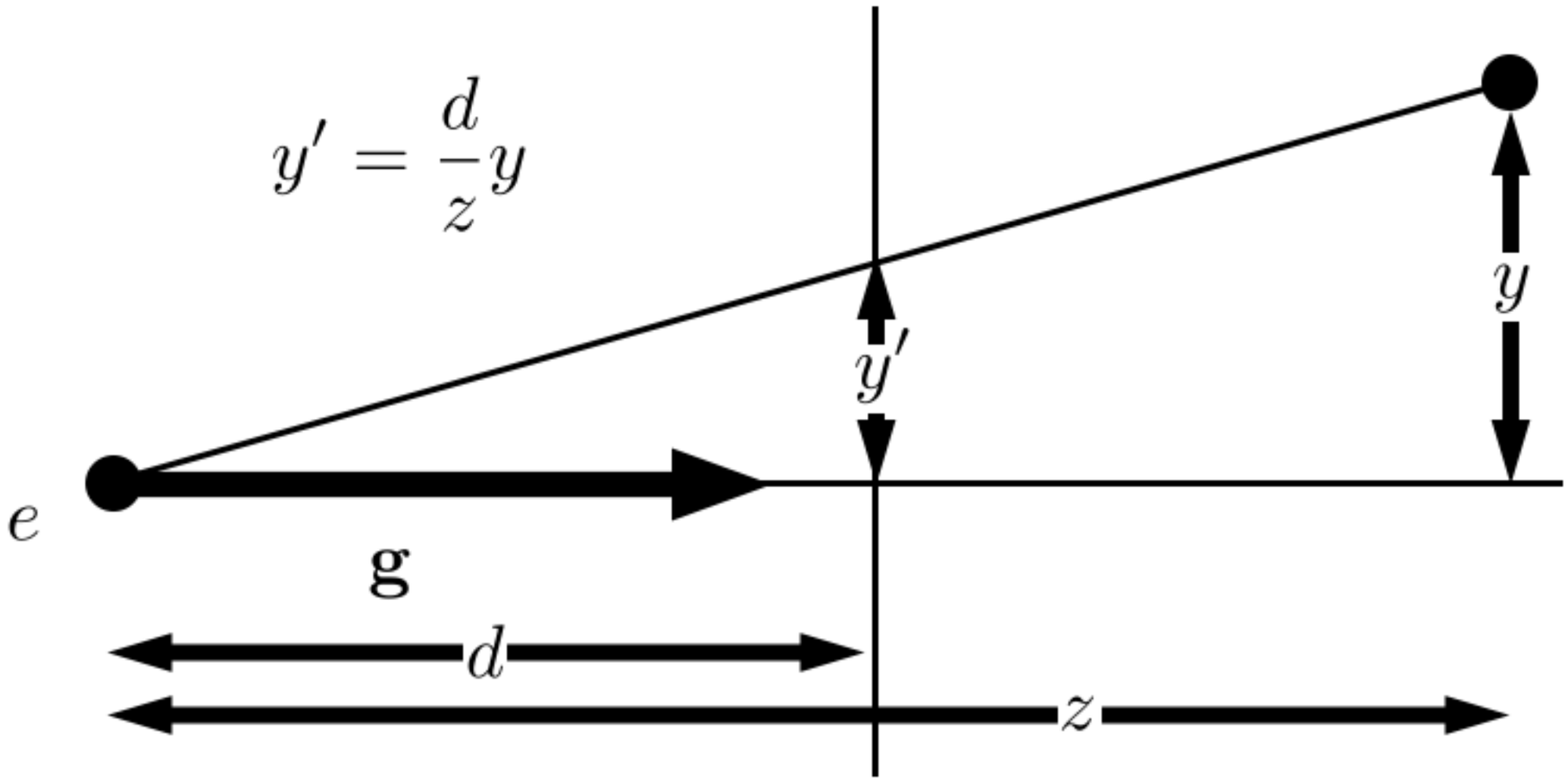
affine



projective

Projective Transformations

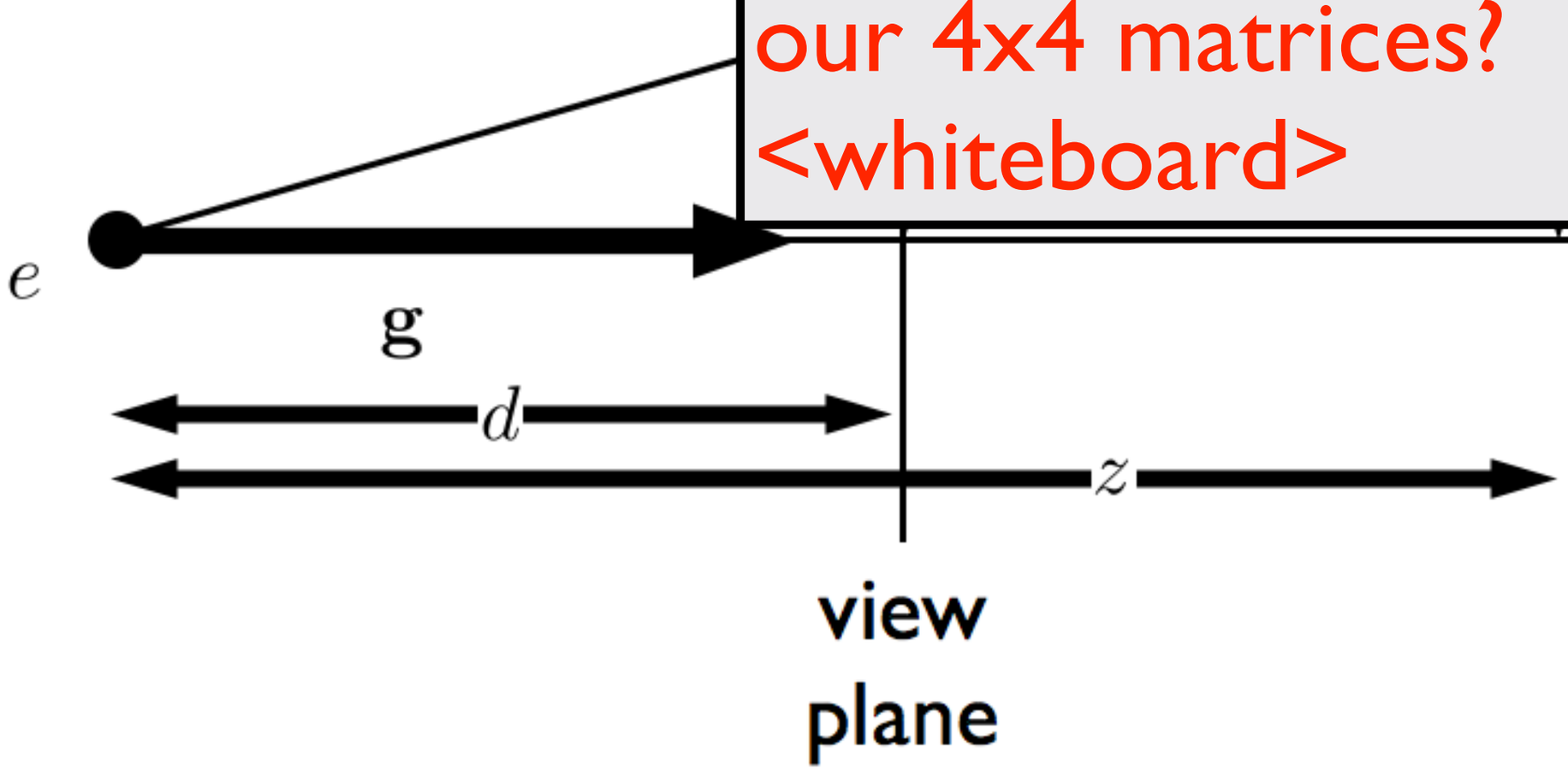
$$y' = \frac{d}{z} y$$



Projective Transformations

$$y' = \frac{d}{z} y$$

How can we represent this with our 4x4 matrices?
<whiteboard>

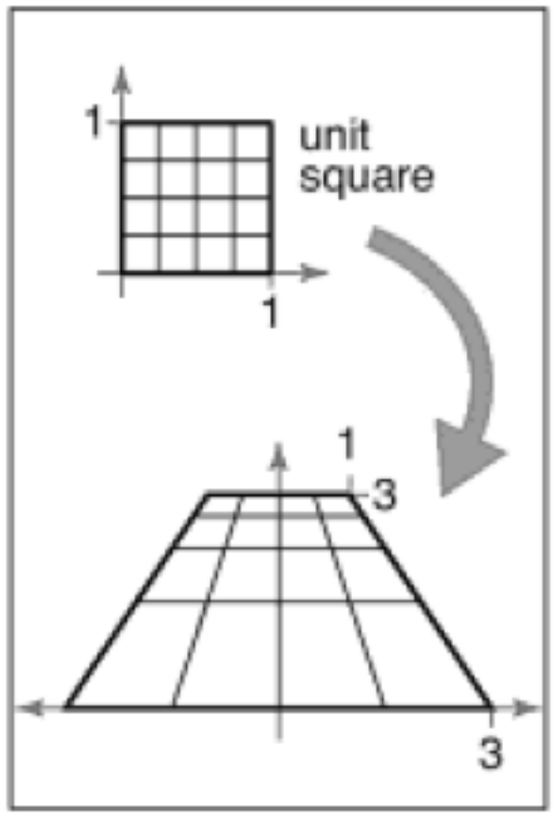


Projective Transformations

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \\ w \end{pmatrix} \rightarrow \begin{aligned} x &= \frac{\tilde{x}}{w} \\ y &= \frac{\tilde{y}}{w} \\ z &= \frac{\tilde{z}}{w} \end{aligned}$$

Example:

$$M = \begin{pmatrix} 2 & 0 & -1 \\ 0 & 3 & 0 \\ 0 & \frac{2}{3} & \frac{1}{3} \end{pmatrix}$$



[Shirley, Marschner]

<whiteboard>

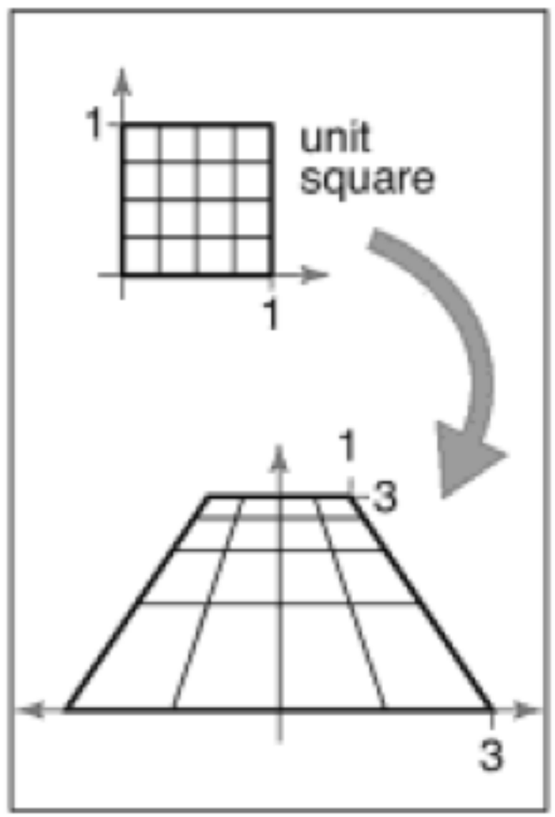
Projective Transformations

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \\ w \end{pmatrix} \rightarrow \begin{aligned} x &= \frac{\tilde{x}}{w} \\ y &= \frac{\tilde{y}}{w} \\ z &= \frac{\tilde{z}}{w} \end{aligned}$$

We can now implement perspective projection!

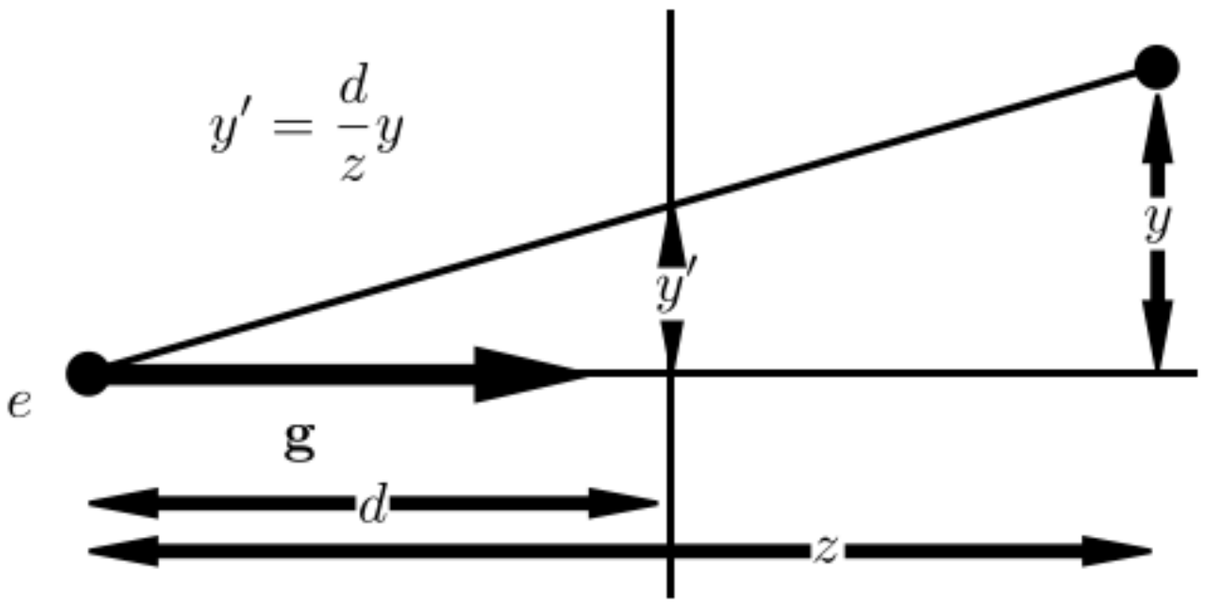
Example:

$$M = \begin{pmatrix} 2 & 0 & -1 \\ 0 & 3 & 0 \\ 0 & \frac{2}{3} & \frac{1}{3} \end{pmatrix}$$

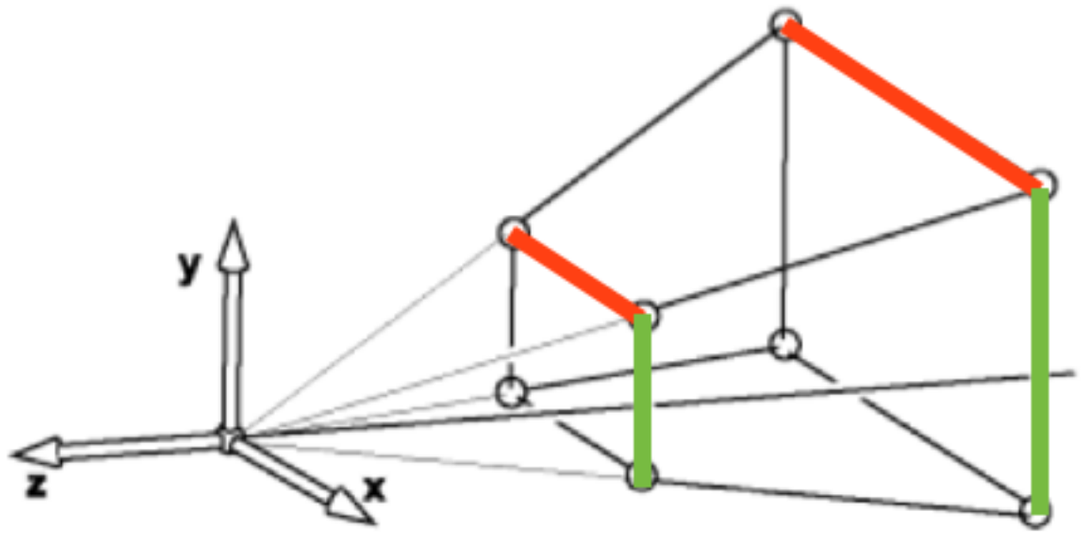


[Shirley, Marschner]

Perspective Projection



view plane



both x and y get multiplied by d/z

[Shirley, Marschner]

Simple perspective projection

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \\ z/d \end{pmatrix} \Rightarrow \begin{cases} x' = \frac{d}{z}x \\ y' = \frac{d}{z}y \\ z' = \frac{d}{z}z = d \end{cases}$$

This achieves a simple perspective projection onto the view plane z = d

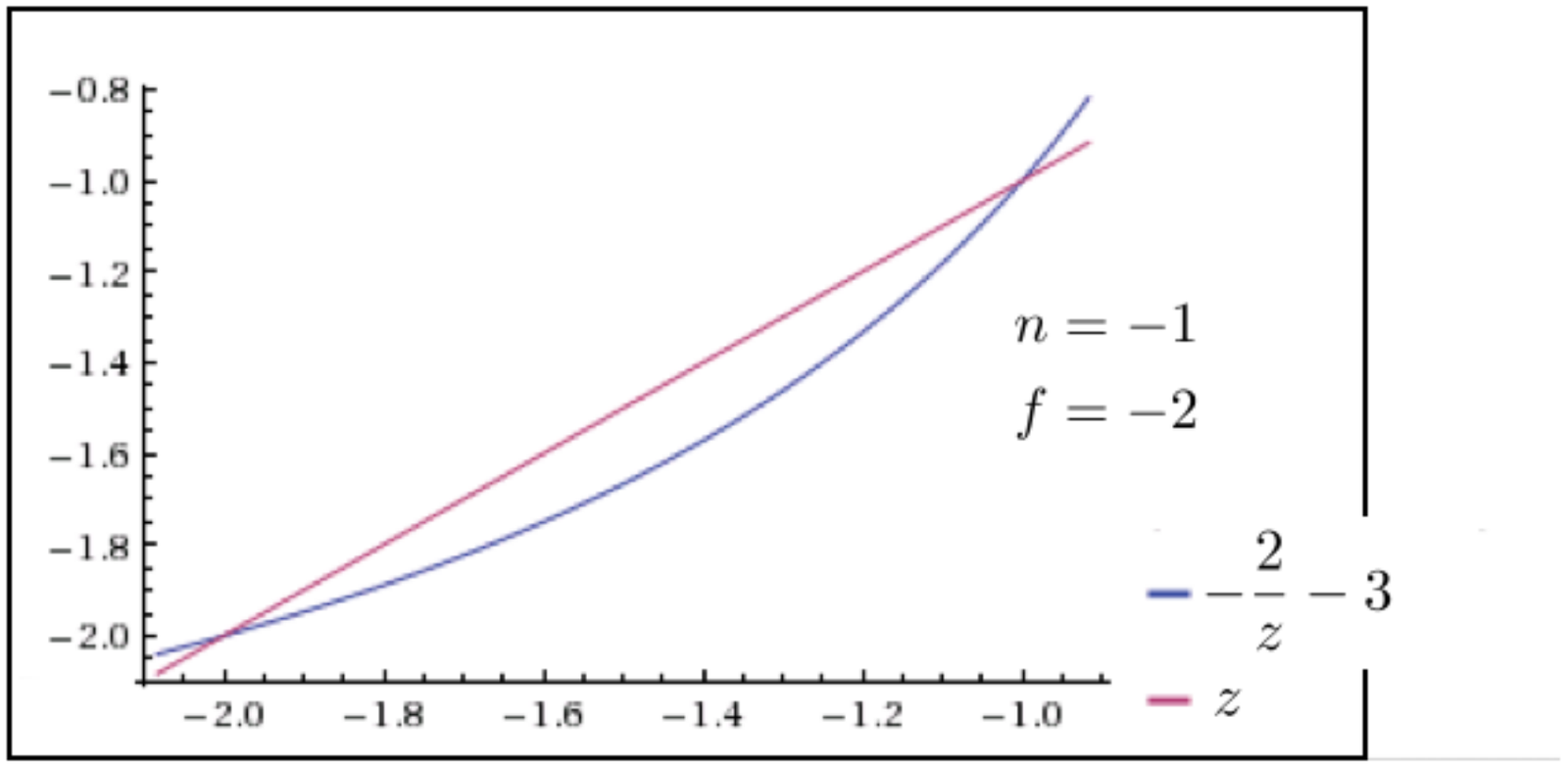
but we've lost all information about z!

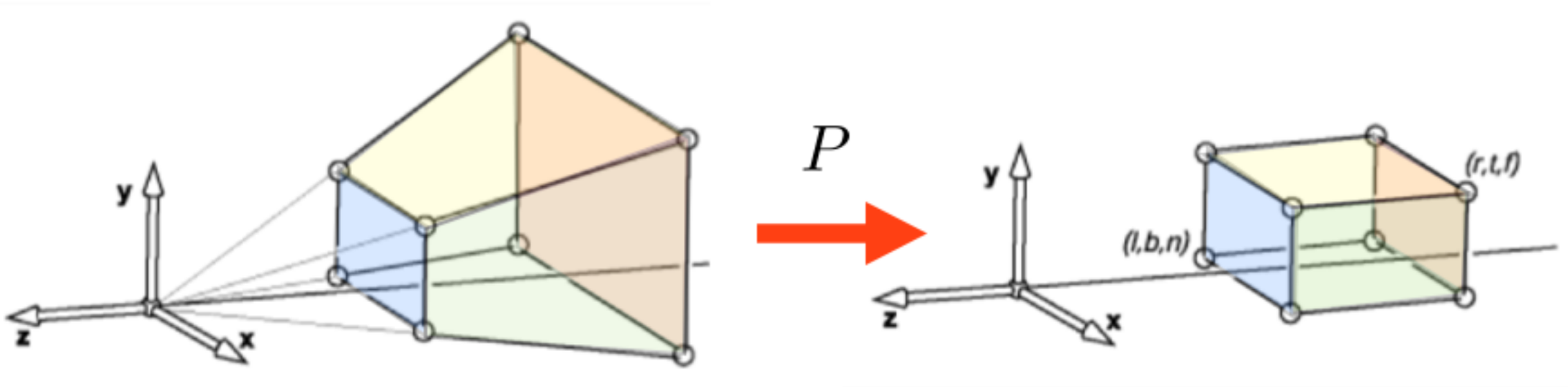
<whiteboard>

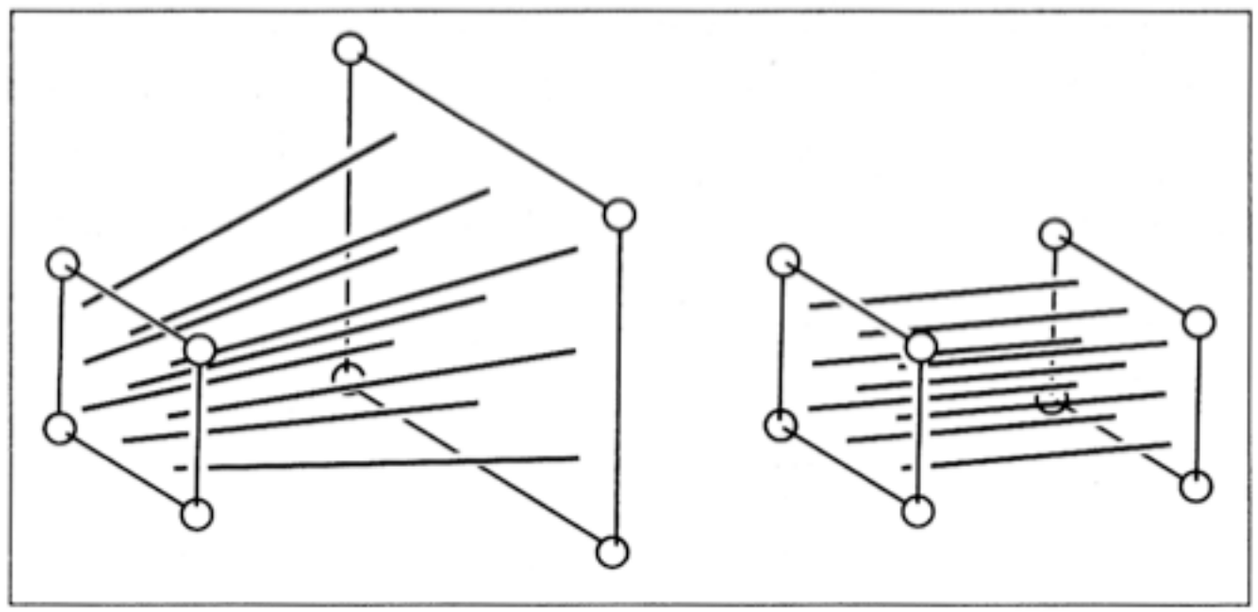
Perspective Projection

$$P = \begin{pmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & n+f & -fn \\ 0 & 0 & 1 & 0 \end{pmatrix} \qquad z' = (n+f) - \frac{nf}{z}$$

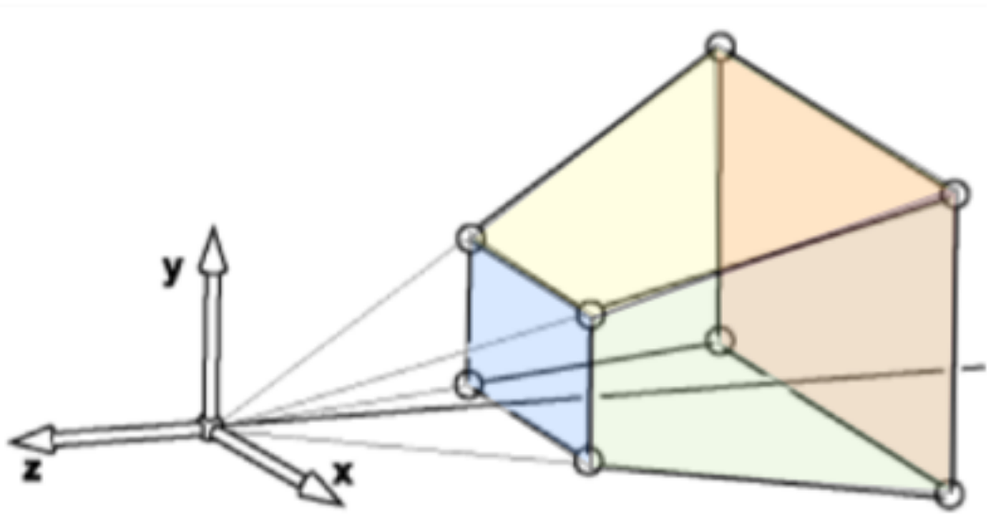
Example:



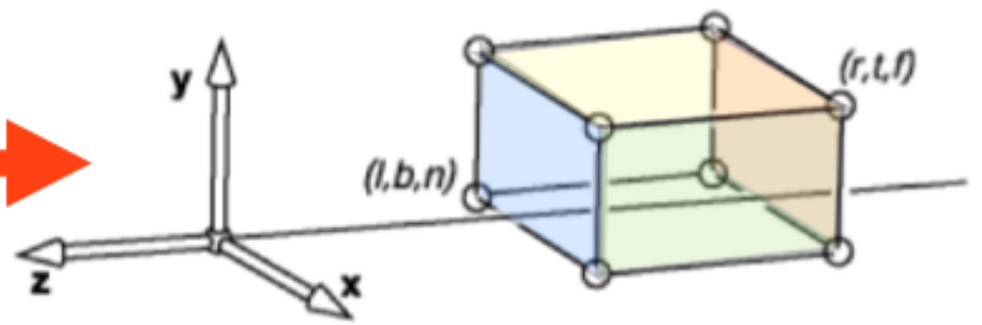


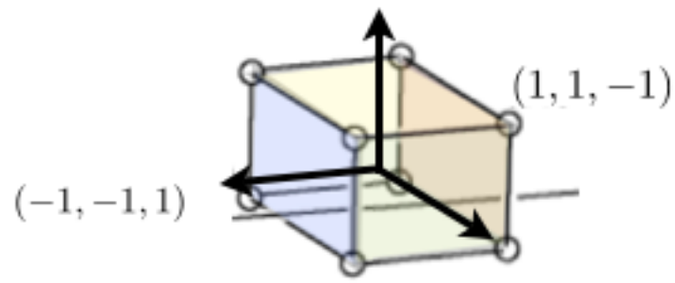


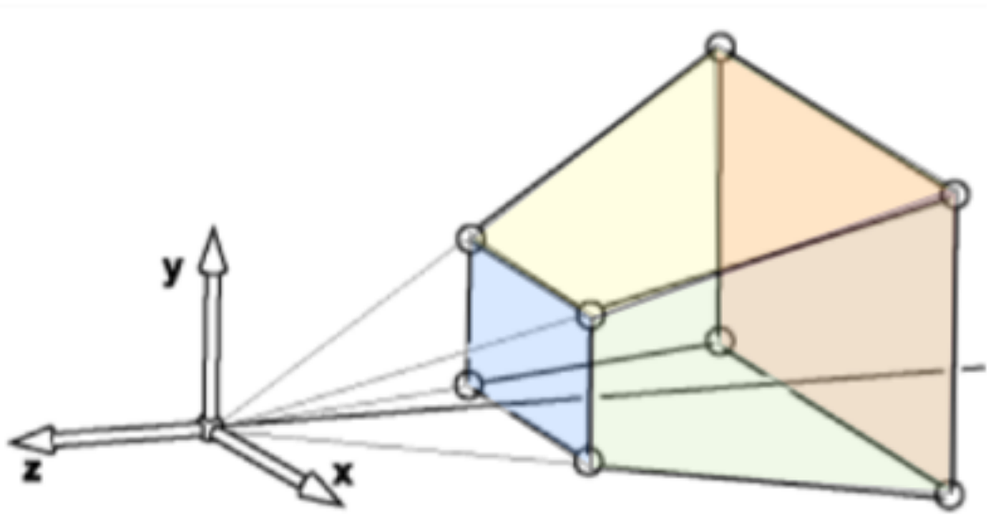
[Shirley, Marschner]



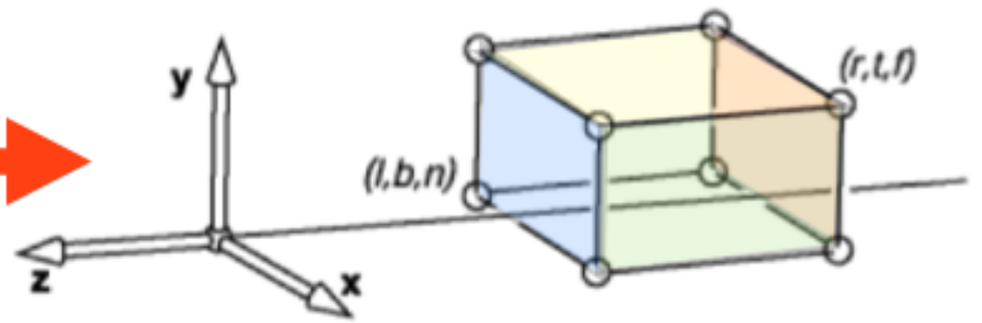
P





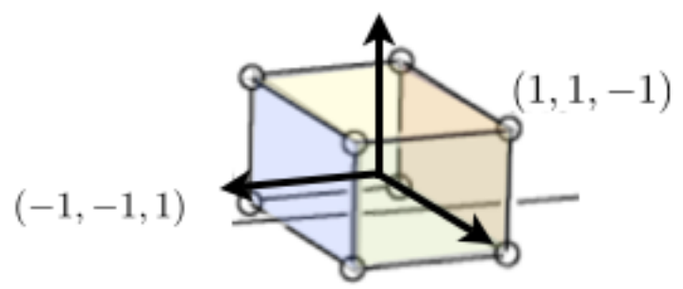


P



M_{orth}

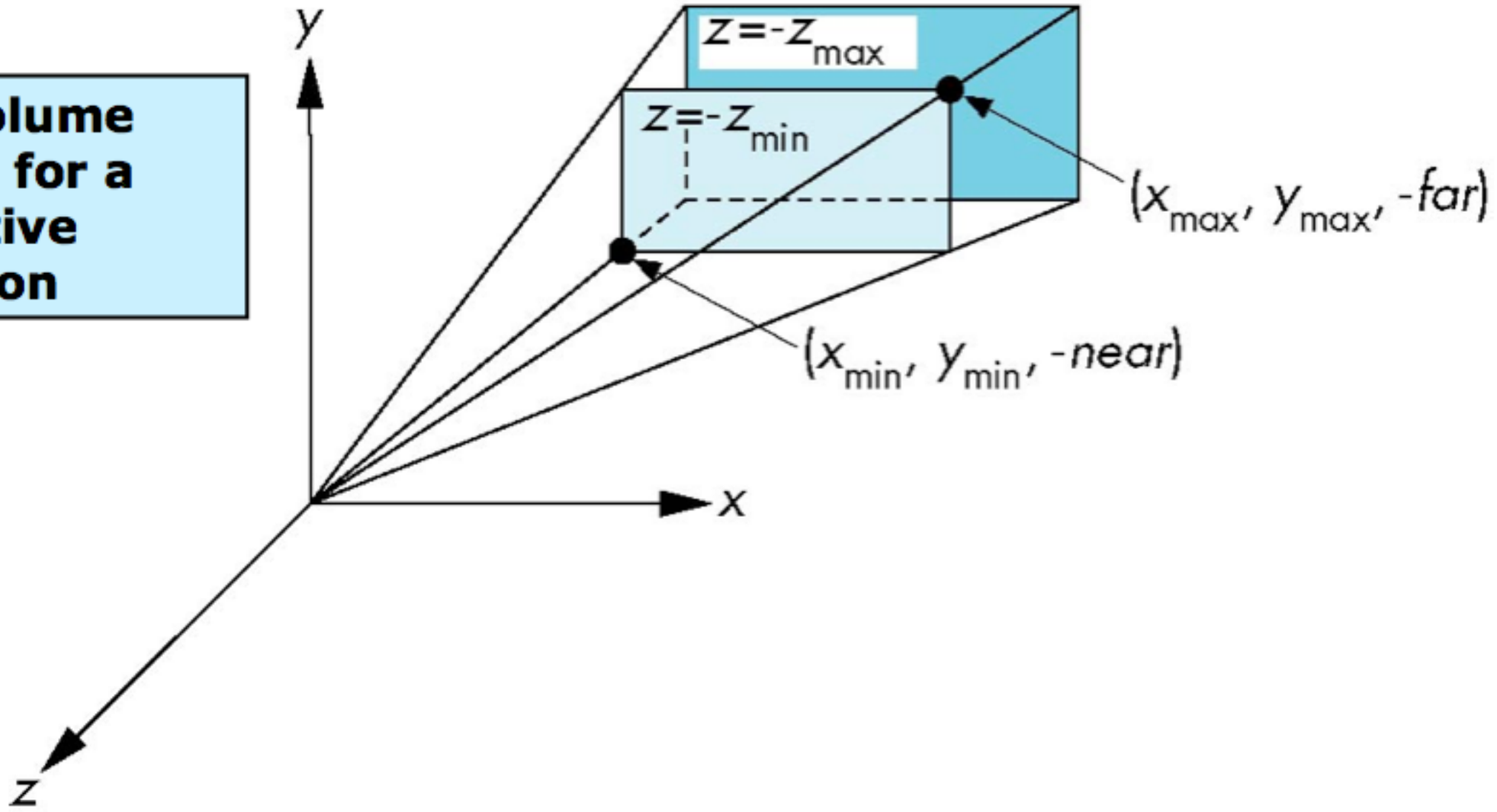
$$M_{\text{per}} = M_{\text{orth}} P$$



OpenGL Perspective Viewing

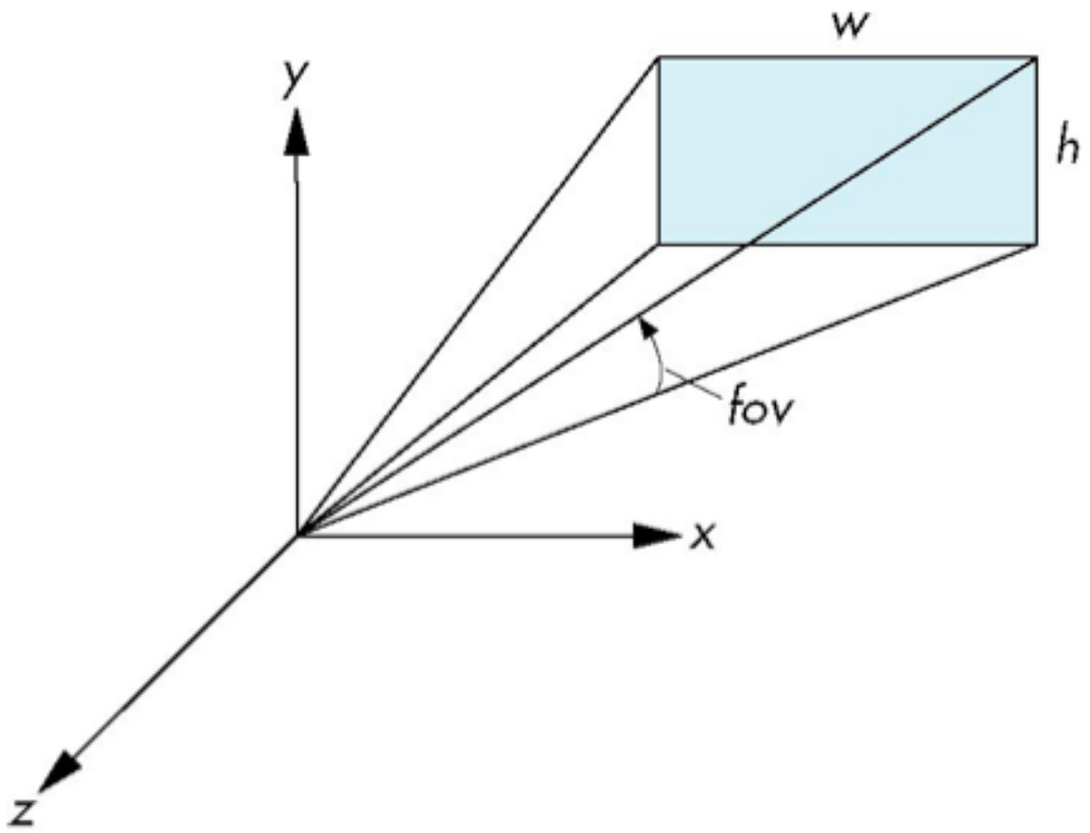
`glFrustum(xmin,xmax,ymin,ymax,near,far)`

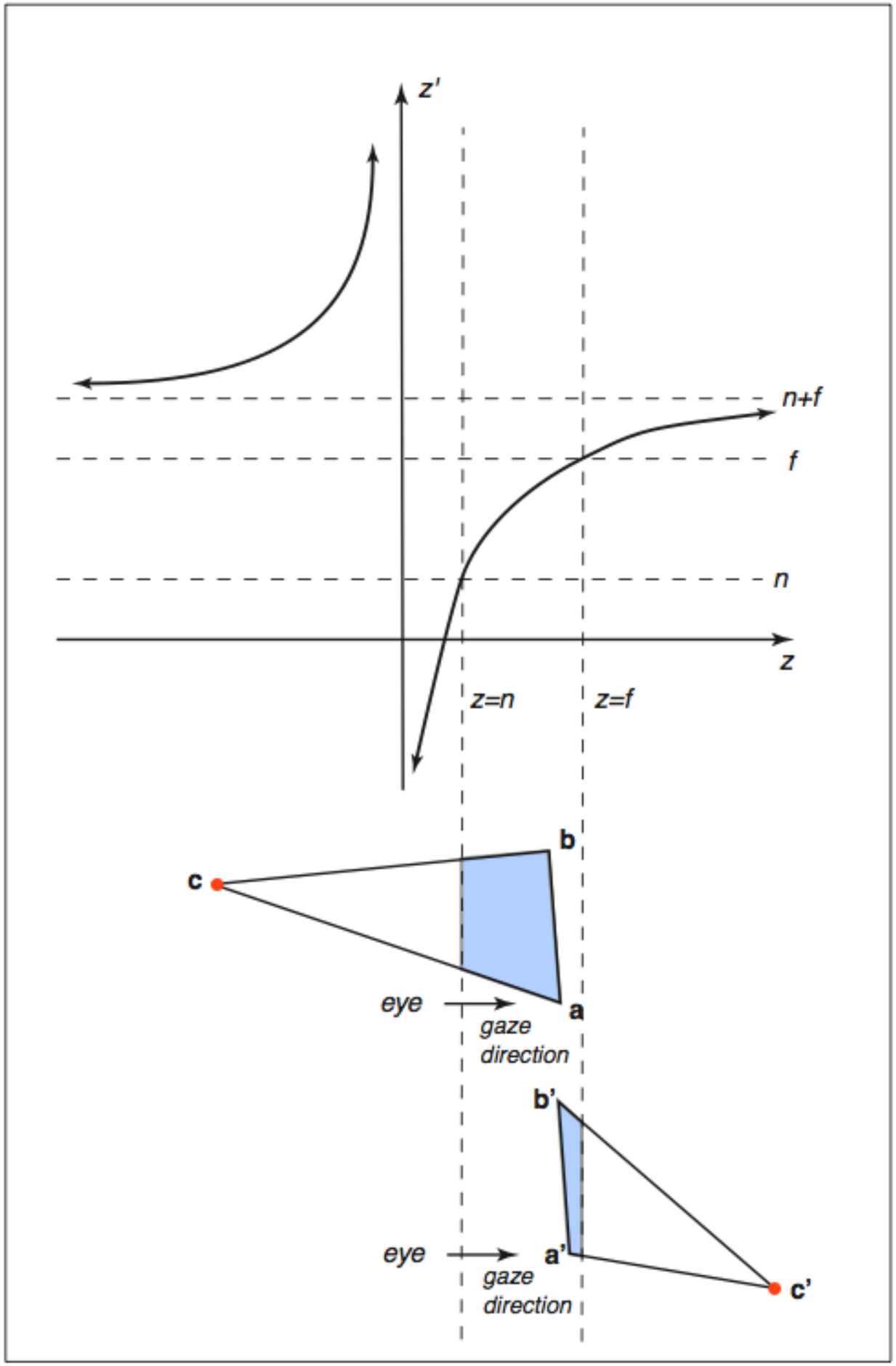
Clipping volume (frustrum) for a perspective projection



Using Field of View

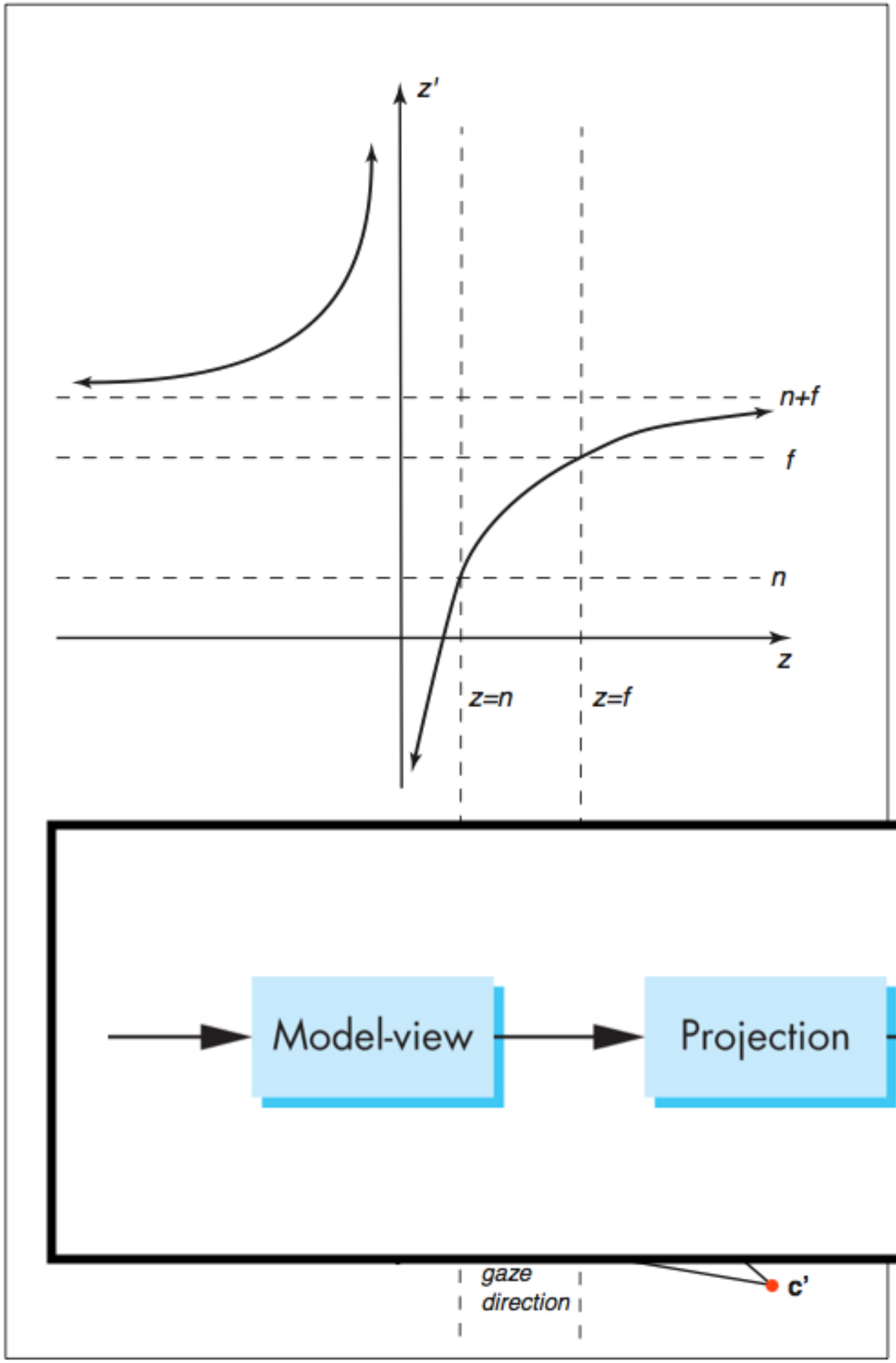
With **`glFrustum`** it is often difficult to get the desired view **`gluPerpective(fovy, aspect, near, far)`** often provides a better interface





[Shirley, Marschner]

Clipping after the perspective transformation can cause problems



OpenGL clips **after** projection and **before** perspective division

$$-w \leq x \leq w$$

$$-w \leq y \leq w$$

$$-w \leq z \leq w$$

[Shirley, Marschner]