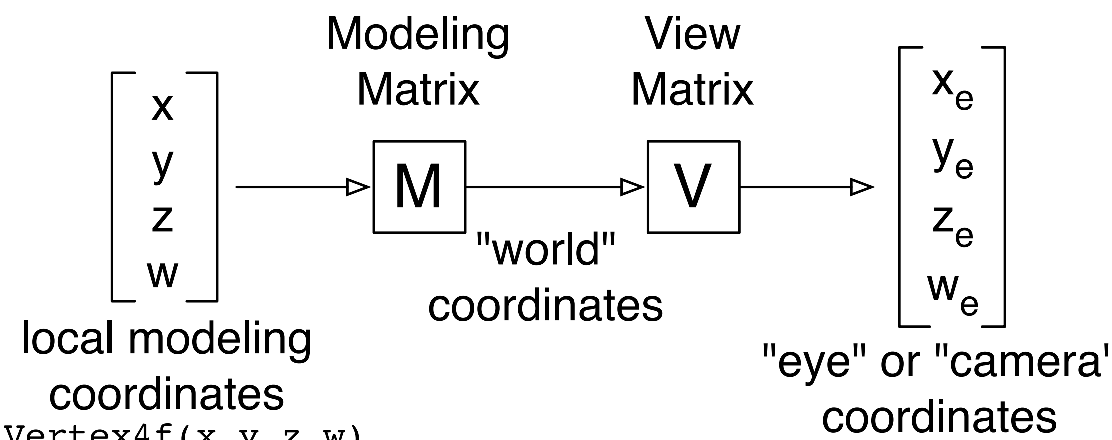
Viewing and Orthographic Projections

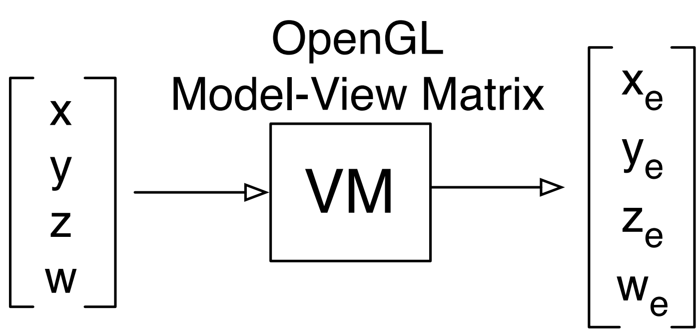
CS 442/452

September 17, 2008

Model-View Transformation Matrix

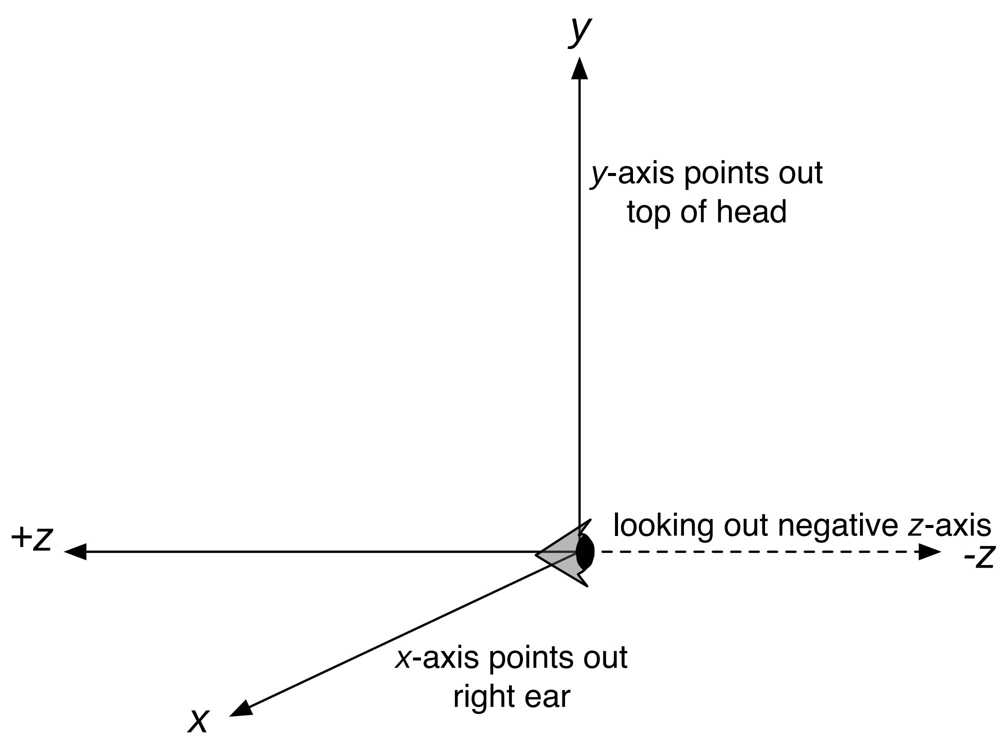


`glVertex4f(x,y,z,w)`

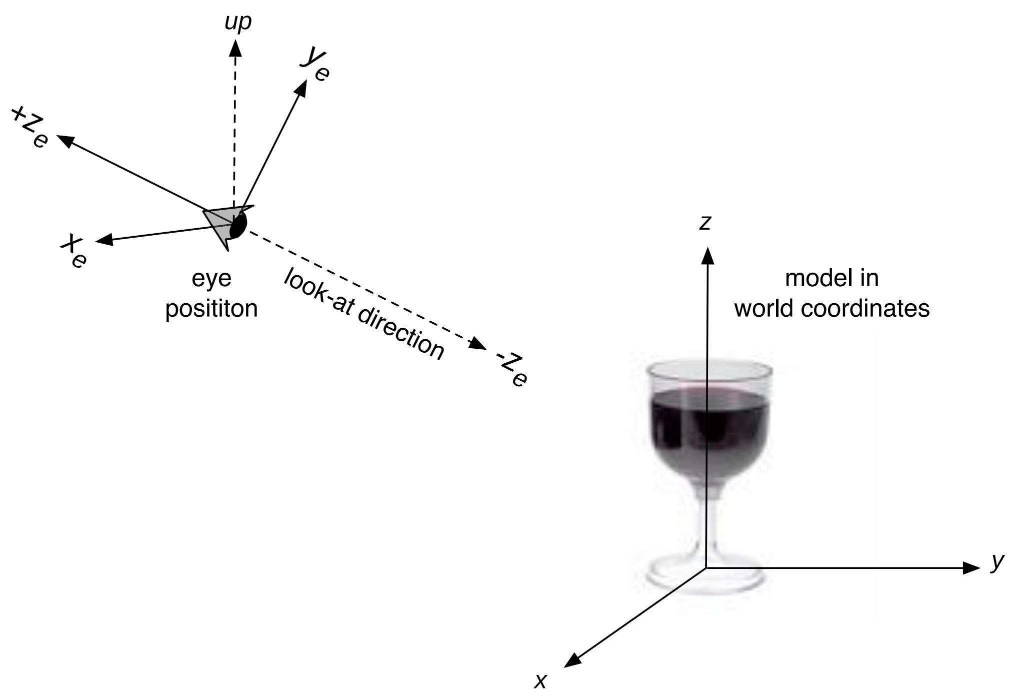


`glMatrixMode(GL_MODELVIEW);`

Eye (or Camera) Coordinate System

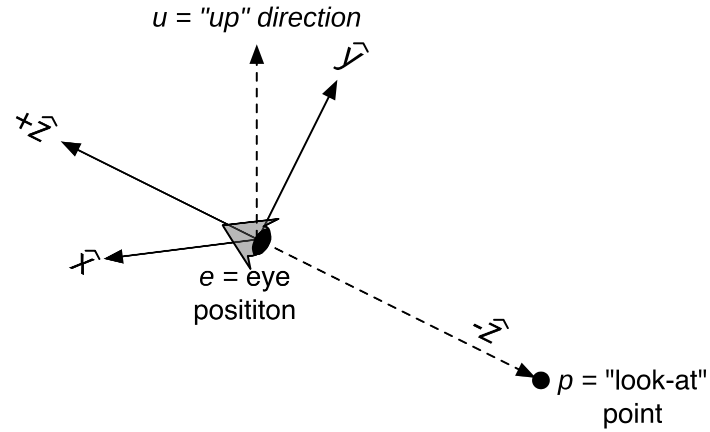


Eye Coordinates vs World Coordinates



The *view matrix* V transforms objects in the scene from world coordinates to eye (camera) coordinates.

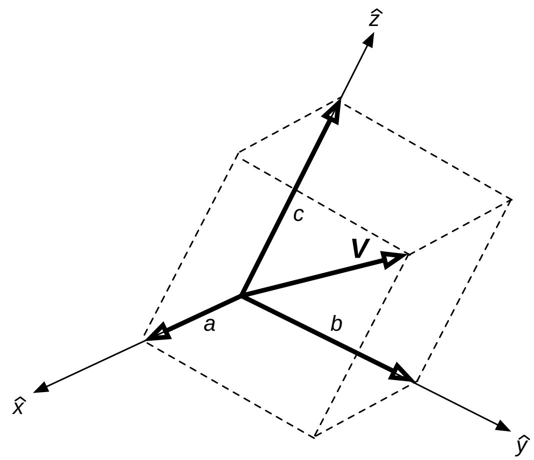
Look-at Transformation Parameters



Orthogonal vectors $(\widehat{x}, \widehat{y}, \widehat{z})$ from **up-direction** u, **eye position** e, and **“look-at” point** p :

$$
\begin{aligned}
z &= e - p \\
\widehat{z} &= \frac{z}{\|z\|} \\
x &= u \times z \\
\widehat{x} &= \frac{x}{\|x\|} \\
\widehat{y} &= \widehat{z} \times \widehat{x}
\end{aligned}
$$

Projections of V onto orthonormal axes



$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \widehat{x} \bullet \mathbf{V} \\ \widehat{y} \bullet \mathbf{V} \\ \widehat{z} \bullet \mathbf{V} \end{bmatrix} = \begin{bmatrix} \widehat{x}_x & \widehat{x}_y & \widehat{x}_z \\ \widehat{y}_x & \widehat{y}_y & \widehat{y}_z \\ \widehat{z}_x & \widehat{z}_y & \widehat{z}_z \end{bmatrix} \begin{bmatrix} \mathbf{V}_x \\ \mathbf{V}_y \\ \mathbf{V}_z \end{bmatrix}$$

$\mathbf{V}$ becomes (a, b, c) in the $(\widehat{x}, \widehat{y}, \widehat{z})$ coordinate system.

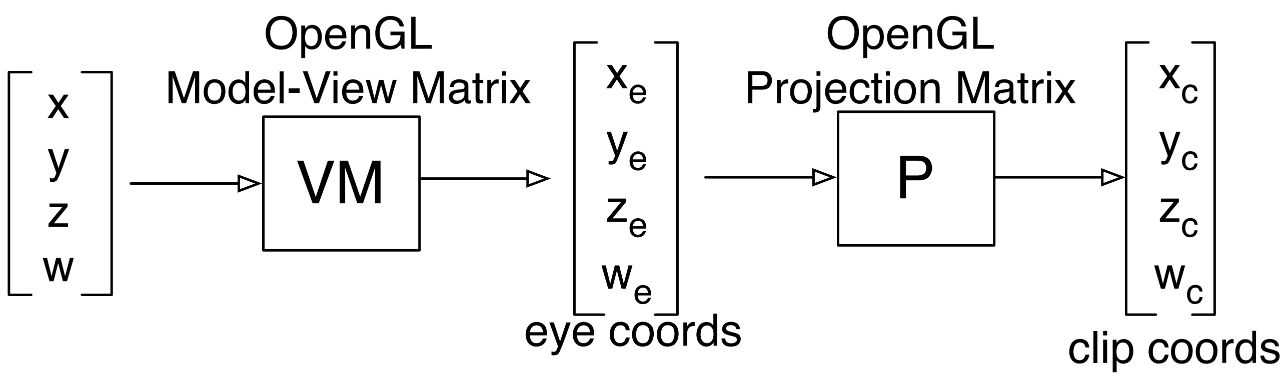
View Matrix from "Look-At" Parameters

$$
\begin{aligned}
e &= \text{eye (camera) position} \\
p &= \text{look-at point} \\
u &= \text{up-direction}
\end{aligned}
$$

$$
\begin{aligned}
z &= e - p \\
\widehat{z} &= z/\|z\| \\
x &= u \times z \\
\widehat{x} &= x/\|x\| \\
\widehat{y} &= \widehat{z} \times \widehat{x}
\end{aligned}
$$

$$
V = \underbrace{\begin{bmatrix} \widehat{x}_x & \widehat{x}_y & \widehat{x}_x & 0 \\ \widehat{y}_x & \widehat{y}_y & \widehat{y}_x & 0 \\ -\widehat{z}_x & -\widehat{z}_y & -\widehat{z}_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{rotate onto } -z} \underbrace{\begin{bmatrix} 1 & 0 & 0 & -e_x \\ 0 & 1 & 0 & -e_y \\ 0 & 0 & 1 & -e_z \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{translate eye}}
$$

Projection Transformation Matrix




```
glMatrixMode(GL_MODELVIEW); glMatrixMode(GL_PROJECTION);
```