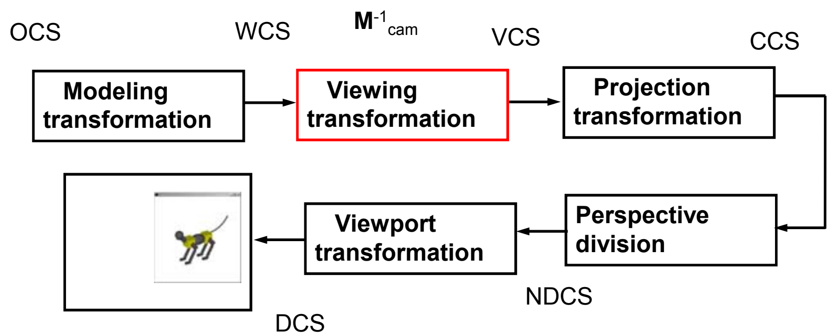
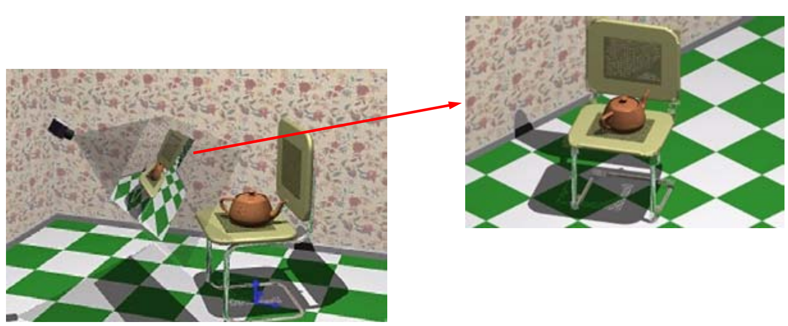
Graphics Pipeline

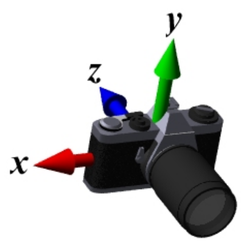


Rendering a 3D Scene From the Point of View of a Virtual Camera



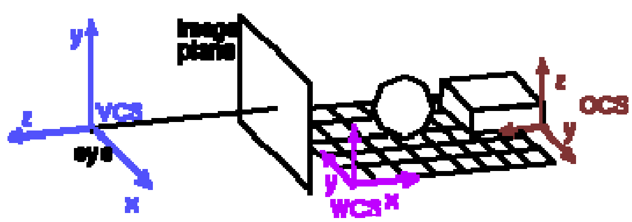
OpenGL Convention

In world coordinates, the camera system is defined as follows:



Camera Transformation

Transforms objects to camera coordinates



$$\left.\begin{array}{l} P_{\text{wcs}} = \mathrm{M}_{\text{cam}} P_{\text{vcs}} \rightarrow P_{\text{vcs}} = \mathrm{M}_{\text{cam}}^{-1} P_{\text{wcs}} \\ P_{\text{wcs}} = \mathrm{M}_{\text{mod}} P_{\text{ocs}} \end{array}\right\} \rightarrow$$

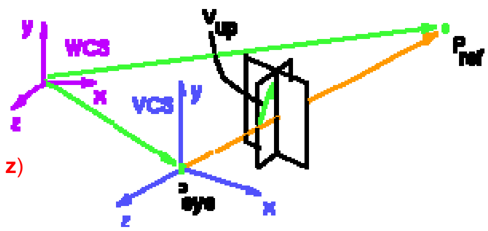
$$P_{\text{vcs}} = \underbrace{\mathrm{M}_{\text{cam}}^{-1} \mathrm{M}_{\text{mod}}}_{\text{Modelview Transformation}} P_{\text{ocs}}$$

Defining $\mathbf{M}_{cam}$

Given:

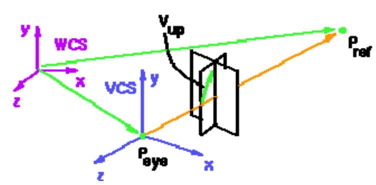
Eye point P_{eye}
Reference point P_{ref}
Up vector $\mathbf{v}_{up}$

($\mathbf{v}_{up}$ is not necessarily orthogonal to $\mathbf{z}$)



To build $\mathbf{M}_{cam}$ we need to define a camera coordinate system [**i** **j** **k** *O*]

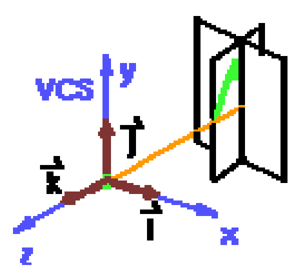
Camera Coordinate System



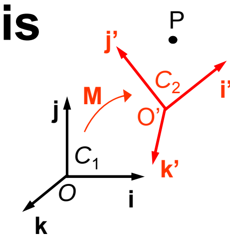
$$\mathbf{k} = \frac{P_{\text{eye}} - P_{\text{ref}}}{|P_{\text{eye}} - P_{\text{ref}}|}$$

$$\mathbf{i} = \frac{\mathbf{v}_{\text{up}} \times \mathbf{k}}{|\mathbf{v}_{\text{up}} \times \mathbf{k}|}$$

$$\mathbf{j} = \mathbf{k} \times \mathbf{i}$$



Reminder: Change of Basis



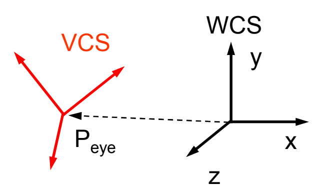
$$P_{C_1} = \mathrm{M} P_{C_2}$$

$$P_{C_1} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} i'_x & j'_x & k'_x & O'_x \\ i'_y & j'_y & k'_y & O'_y \\ i'_z & j'_z & k'_z & O'_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \mathrm{M} P_{C_2}$$

Building $\mathrm{M}_{\mathrm{cam}}$

Change of basis

Our reference system is WCS,
we know the camera parameters with respect to the world



Align WCS with VCS

$$\mathrm{M}_{\text{cam}} = \underset{\text{Translation}}{\begin{bmatrix} 1 & 0 & 0 & P_{\text{eye}_x} \\ 0 & 1 & 0 & P_{\text{eye}_y} \\ 0 & 0 & 1 & P_{\text{eye}_z} \\ 0 & 0 & 0 & 1 \end{bmatrix}} \underset{\text{Rotation}}{\begin{bmatrix} i_x & j_x & k_x & 0 \\ i_y & j_y & k_y & 0 \\ i_z & j_z & k_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}$$

$$P_{\text{WCS}} = \mathrm{M}_{\text{cam}} P_{\text{VCS}}$$

Building M_{cam} Inverse

Invert the smart way

$$\mathbf{M}_{\text{cam}}^{-1} = \left(\begin{bmatrix} 1 & 0 & 0 & P\text{eye}_x \\ 0 & 1 & 0 & P\text{eye}_y \\ 0 & 0 & 1 & P\text{eye}_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i_x & j_x & k_x & 0 \\ i_y & j_y & k_y & 0 \\ i_z & j_z & k_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \right)^{-1}$$

$$= \begin{bmatrix} i_x & j_x & k_x & 0 \\ i_y & j_y & k_y & 0 \\ i_z & j_z & k_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 & 0 & P\text{eye}_x \\ 0 & 1 & 0 & P\text{eye}_y \\ 0 & 0 & 1 & P\text{eye}_z \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1}$$

Building M_{cam} Inverse

Invert the smart way

$$\mathbf{M}_{\text{cam}}^{-1} = \begin{bmatrix} i_x & j_x & k_x & 0 \\ i_y & j_y & k_y & 0 \\ i_z & j_z & k_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 & 0 & P\text{eye}_x \\ 0 & 1 & 0 & P\text{eye}_y \\ 0 & 0 & 1 & P\text{eye}_z \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1}$$

$$= \begin{bmatrix} \overbrace{\boxed{\begin{matrix} i_x & i_y & i_z \\ j_x & j_y & j_z \\ k_x & k_y & k_z \end{matrix}}}^{\text{Transpose}} & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \\ \begin{matrix} 0 & 0 & 0 \end{matrix} & 1 \end{bmatrix} \begin{bmatrix} \begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix} & \overbrace{\boxed{\begin{matrix} -P\text{eye}_x \\ -P\text{eye}_y \\ -P\text{eye}_z \end{matrix}}}^{\text{Negate}} \\ \begin{matrix} 0 & 0 & 0 \end{matrix} & 1 \end{bmatrix}$$

$$P_{\text{vcs}} = \mathbf{M}_{\text{cam}}^{-1} P_{\text{wcs}}$$

Summary of the Modelview Transformation

1. ***An affine transformation composed of elementary affine transformations***
2. ***The camera transformation is a change of basis***
3. ***The modelview transformation preserves:***

- lines and planes
- parallelism of lines and planes
- affine combinations of points and relative ratios