3D Rendering Geometry

CS 355: Interactive Graphics and Image Processing

Rendering Geometry

- Transform from object to world coordinates
- Transform from world to camera coordinates
- Clipping: near plane, far plane, field of view (we're going to skip this for the moment)
- Perspective projection
- View transformation

Object to World

- Like what you’ve done in 2D, only in 3D:
  - Scale (while still at origin in object space)
  - Rotate (while still at origin in object space)
  - Translate to position the object

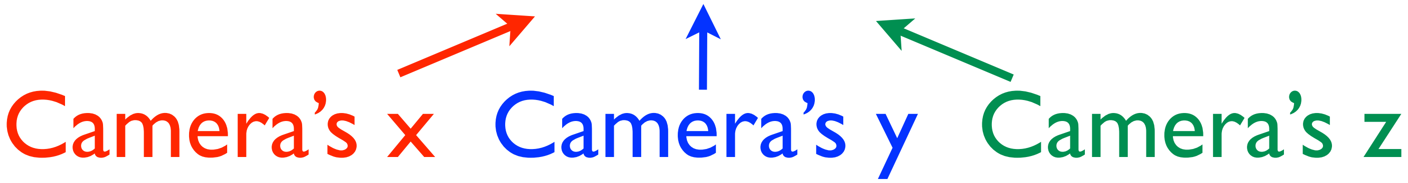
World to Camera

- Suppose that you know
  - Position of camera in world coordinates

$$\mathbf{c} = (c_x, c_y, c_z)$$

  - Orientation of camera as given by *a set of basic vectors* in world coordinates

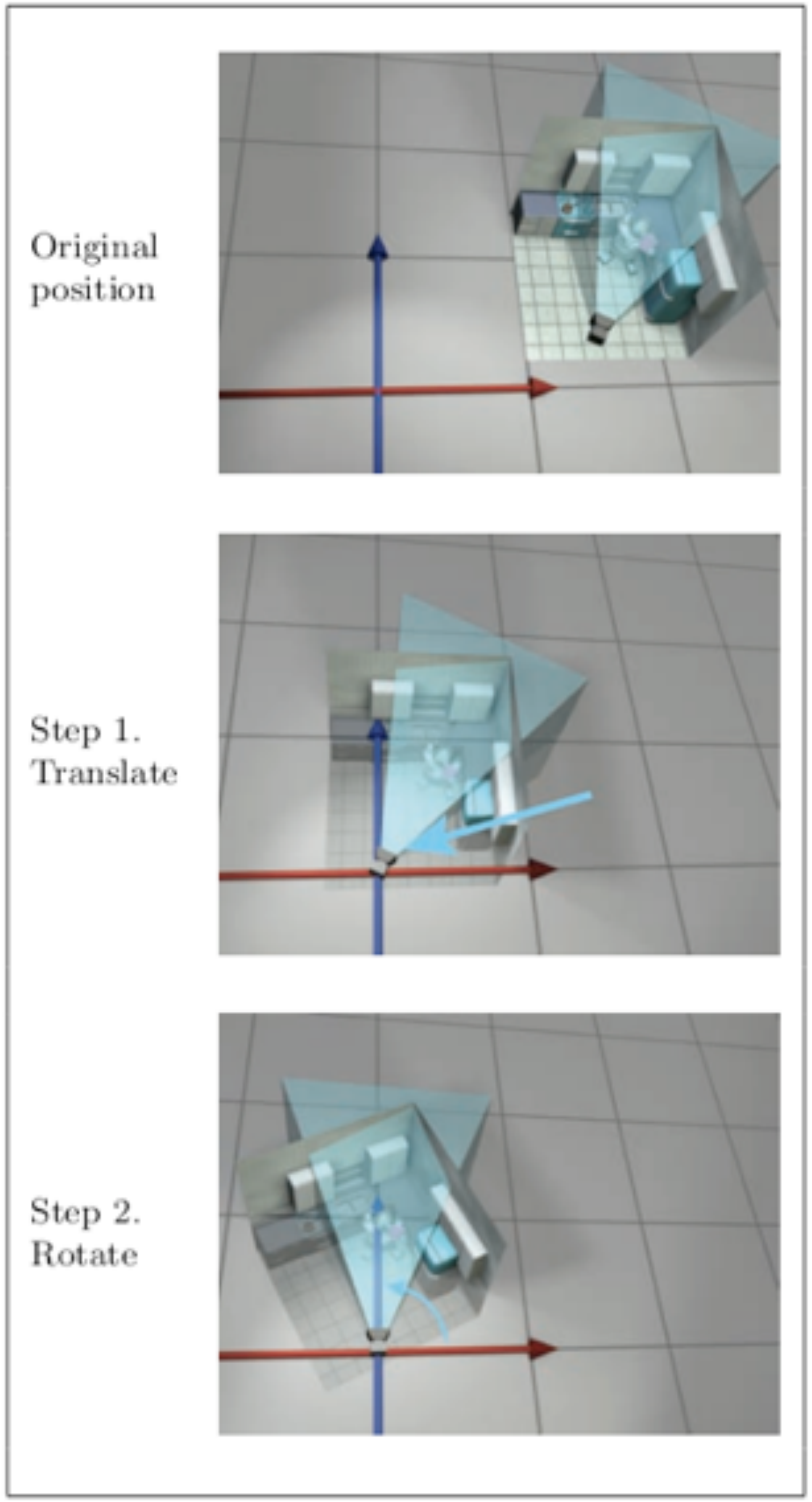
$$\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$$



Camera's x Camera's y Camera's z

World to Camera

- Two steps:
  - **Translate**
    everything to be relative to the camera position
  - **Rotate**
    into the camera's viewing orientation



World to Camera

- Two steps:
  - **Translate**
    everything to be relative to the camera position
  - **Rotate**
    into the camera's viewing orientation

$$\begin{bmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 1 & -c_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} e_{11} & e_{12} & e_{13} & 0 \\ e_{21} & e_{22} & e_{23} & 0 \\ e_{31} & e_{32} & e_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Putting It Together

$$\begin{bmatrix}\boxed{\begin{matrix}x\\y\end{matrix}}\\ f\\ 1\end{bmatrix} \sim \begin{bmatrix}X\\Y\\Z\\Z/f\end{bmatrix} = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 1/f & 0\end{bmatrix} \begin{bmatrix}e_{11} & e_{12} & e_{13} & 0\\ e_{21} & e_{22} & e_{23} & 0\\ e_{31} & e_{32} & e_{33} & 0\\ 0 & 0 & 0 & 1\end{bmatrix} \begin{bmatrix}1 & 0 & 0 & -c_x\\ 0 & 1 & 0 & -c_y\\ 0 & 0 & 1 & -c_z\\ 0 & 0 & 0 & 1\end{bmatrix} \begin{bmatrix}\boxed{\begin{matrix}X\\Y\\Z\end{matrix}}\\ 1\end{bmatrix}$$

Normalize Project Rotate Translate

Rendering Geometry

- ✓ Transform from object to world coordinates
- ✓ Transform from world to camera coordinates
- Clipping: near plane, far plane, field of view
- ✓ Perspective projection
- View transformation

Coming up...

- Specifying camera pose and orthogonalizing the rotation
- Clipping space
- Screen transformation