ALIGNING CAMERA AT WORLD SPACE TO THE ORIGIN IN CAMERA SPACE

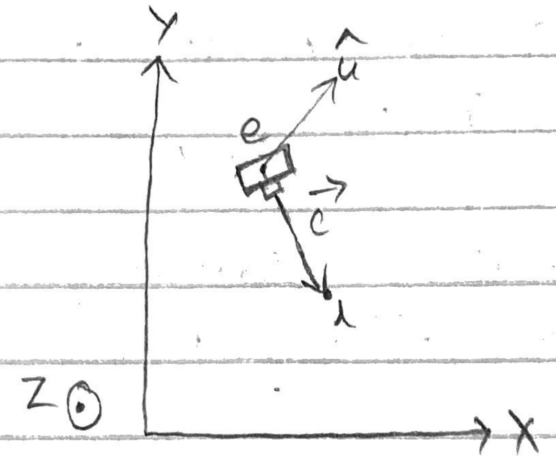

Fig: Camera in world space.

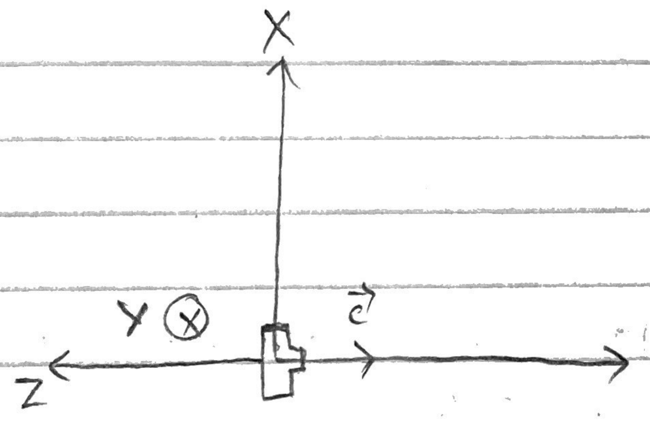

Fig: Camera at origin and looking at -Z axis in camera Space.

In computer graphics, Camera is represented in world space by following parameters:

1> Camera is located at some position (e)
2> Camera is looking at some point (l)
3> Camera has up direction ($\hat{u}$)

Direction at which the camera is looking at is given as:

$$\hat{c} = \frac{l-e}{\| l-e \|}$$

To align this camera at origin in camera space, $\hat{c}$ should be facing $-z$ axis.

Let us construct a camera frame in world space such that its $\hat{z}$ is $-\hat{c}$

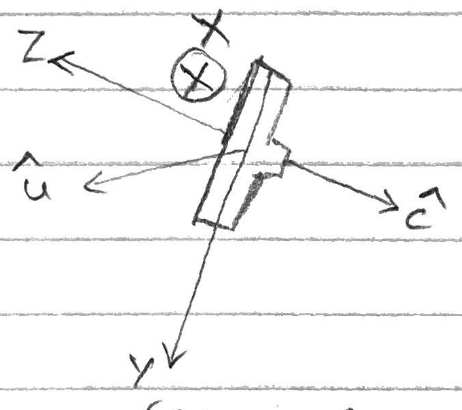

Fig: Camera frame.

We can figure out rest of the axis of its frame by cross products:

$$\hat{ZF} = -\hat{c}$$

$$\hat{XF} = \text{Normalize}(\hat{u} \times \hat{ZF})$$

$$\hat{YF} = \hat{ZF} \times \hat{XF}$$

So, our transformation involves translating camera from e to 0. Then applying rotation such that camera frame aligns with the standard basis vectors.

The translation matrix looks like:

$$T_1 = \begin{bmatrix} 1 & 0 & 0 & -e_x \\ 0 & 1 & 0 & -e_y \\ 0 & 0 & 1 & -e_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation matrix is a bit clever. If we were to write frames of the camera as column vectors of the rotation like:

$$R = [\hat{XF} \quad \hat{YF} \quad \hat{ZF}]$$

Then, that would be rotating standard basis vectors to align with camera frame. What we need to do is apply this rotation in reverse i.e:

$$T_2 = \text{Inverse}(R)$$

$T_2 = R^T$ ($\because$ Inverse of orthonormal vectors is transpose)

$$T_2 = \begin{bmatrix} XF_x & XF_y & XF_z & 0 \\ YF_x & YF_y & YF_z & 0 \\ ZF_x & ZF_y & ZF_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Thus the desired transformation i.e. translation followed by rotation is

$$T = T_2 T_1$$

$$T = T_2 \begin{bmatrix} 1 & 0 & 0 & -e_x \\ 0 & 1 & 0 & -e_y \\ 0 & 0 & 1 & -e_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T = \begin{bmatrix} XF_x & XF_y & XF_z & a \\ YF_x & YF_y & YF_z & b \\ ZF_x & ZF_y & ZF_z & c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where,

$$a = -XF_x \cdot e_x - XF_y \cdot e_y - XF_z \cdot e_z$$

$$b = -YF_x \cdot e_x - YF_y \cdot e_y - YF_z \cdot e_z$$

$$c = -ZF_x \cdot e_x - ZF_y \cdot e_y - ZF_z \cdot e_z$$