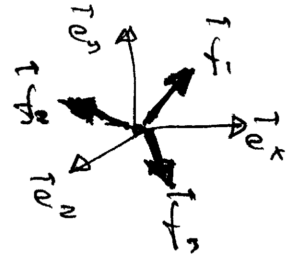
Change of Basis

Two coordinate systems (same origo), given by two ortonormal bases

1) $\vec{e}_x, \vec{e}_y, \vec{e}_z$

2) $\vec{f}_1, \vec{f}_2, \vec{f}_3$



[Orthonormal : vectors have unit length and are pairwise orthogonal, i.e.

$$\vec{e}_a \cdot \vec{e}_b = \begin{cases} 1 & a = b \\ 0 & a \neq b \end{cases}$$
]

We know $\vec{f}_i$'s expressed in $\vec{e}$'s system :

$$\vec{f}_1 = f_{1x} \cdot \vec{e}_x + f_{1y} \cdot \vec{e}_y + f_{1z} \cdot \vec{e}_z$$
$$\vec{f}_2 = f_{2x} \cdot \vec{e}_x + f_{2y} \, \vec{e}_y + f_{2z} \cdot \vec{e}_z$$
$$\vec{f}_3 = f_{3x} \cdot \vec{e}_x + f_{3y} \cdot \vec{e}_y + f_{3z} \cdot \vec{e}_z$$

{The coordinates of the $\vec{f}$'s in system 1) }

Given a vector $\vec{x}$ (or point), we can consider its coordinates in each system.

Let $\begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}$ be the coordinates in system 1)

and let $\begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix}$ —— " —— 2).

We then have:

$$\vec{x} = w_1 \cdot \vec{f}_1 + w_2 \cdot \vec{f}_2 + w_3 \cdot \vec{f}_3$$

$$= w_1 \cdot \underbrace{\left(f_{1x} \cdot \vec{e}_x + f_{1y} \cdot \vec{e}_y + f_{1z} \cdot \vec{e}_z \right)}_{\vec{f}}$$

$$+ w_2 \underbrace{\left(f_{2x} \cdot \vec{e}_x + f_{2y} \cdot \vec{e}_y + f_{2z} \cdot \vec{e}_z \right)}_{\vec{f}}$$

$$+ w_3 \underbrace{\left(f_{3x} \cdot \vec{e}_x + f_{3y} \cdot \vec{e}_y + f_{3z} \cdot \vec{e}_z \right)}_{\vec{f}}$$

$$= \vec{e}_x \cdot (w_1 \cdot f_{1x} + w_2 \cdot f_{2x} + w_3 \cdot f_{3x})$$
$$+ \vec{e}_y \cdot (w_1 \cdot f_{1y} + w_2 \cdot f_{2y} + w_3 \cdot f_{3y})$$
$$+ \vec{e}_z \cdot (w_1 \cdot f_{1z} + w_2 \cdot f_{2z} + w_3 \cdot f_{3z})$$

So for
$$F = \begin{bmatrix} f_{1x} & f_{2x} & f_{3x} \\ f_{1y} & f_{2y} & f_{3y} \\ f_{1z} & f_{2z} & f_{3z} \end{bmatrix}$$
$$\quad \vec{f}_1 \quad \vec{f}_2 \quad \vec{f}_3$$

⇐ Expressed in system 1)

we have

$$\boxed{F \cdot \vec{w} = \vec{v}} \quad *)$$

$\vec{x}$ expressed in system 2) $\qquad$ $\vec{x}$ expressed in system 1)

Note that since the $\vec{f}$'s are orthogonal, we have
$$F^T \cdot F = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = I.$$

Hence,
$$F^T \cdot (F \cdot \vec{w}) = F^T \cdot \vec{v}$$
$$\| $$
$$(F^T \cdot F) \cdot \vec{w}$$
$$\|$$
$$I \cdot \vec{w}$$
$$\|$$
$$\vec{w}$$

$\vec{v} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \vec{x}$ expressed in system 1)

$\vec{w} = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} = \vec{x}$ expressed in system 2)

that is,
$$\boxed{\vec{w} = F^T \cdot \vec{v}} \qquad **)$$

The two equations *) and **) describe how to change between a point (or vector) $\vec{x}$'s coordinates in the two systems.

So, to eg. rotate around any axis $\vec{r}$ (not just the z-axis), create coord. system with its z-axis parallel to $\vec{r}$, move to this system, rotate, and then move back:

$$F \cdot (Rot_z \cdot (F^T \cdot \vec{v})) = \underbrace{(F \cdot Rot_z \cdot F^T)}_{\text{Matrix to rotate around } \vec{r}} \cdot \vec{v}$$

As another example, assume you want the camera to be positioned at the point

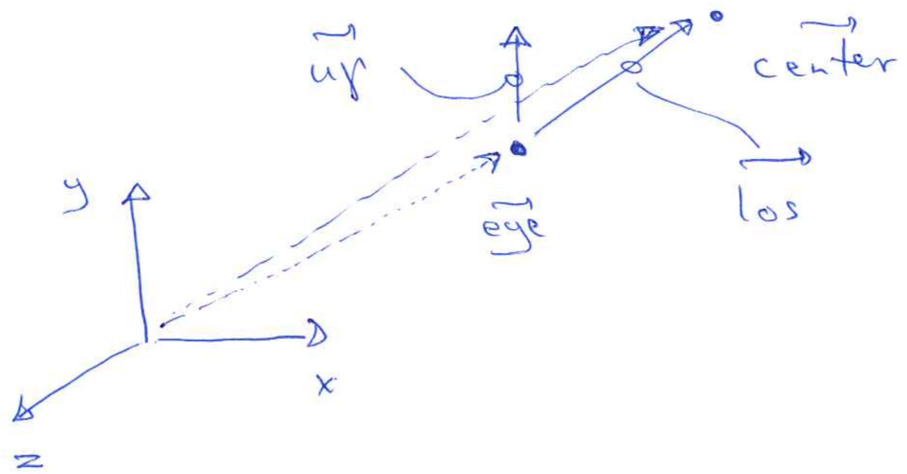
$$\vec{eye} = (e_x, e_y, e_z)$$

looking towards the point

$$\vec{center} = (c_x, c_y, c_z)$$

and with the up-direction of the camera be given by the vector

$$\vec{up} = (u_x, u_y, u_z)$$

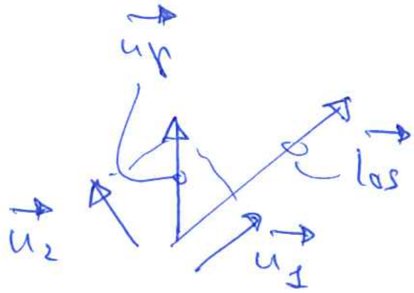


The line of sight is given by

$$\vec{los} = \vec{center} - \vec{eye}$$

and the vector $\vec{up}$ is used as follows:

Let $\vec{u}_1$ and u_2 be the components of $\vec{up}$ parallel, resp. perpendicular, to $\vec{los}$.



$$\vec{u}_1 = \frac{\vec{up} \cdot \vec{los}}{\|\vec{los}\|^2} \cdot \vec{los}$$

$$\vec{u}_2 = \vec{up} - \vec{u}_1$$

[Requirement: $\vec{u}_2 \neq \vec{0}$, i.e., $\vec{up}$ not parallel to $\vec{los}$]

The real camera in OpenGL is always positioned at $\vec{0}$, looking down the negative z-axis, and with the y-axis as the up-direction.

So consider a coordinate system given by

$$\vec{f}_3 = -\vec{los}/\|\vec{los}\|$$

$$\vec{f}_2 = \vec{u}_2/\|\vec{u}_2\|$$

$$\vec{f}_1 = \vec{f}_2 \times \vec{f}_3$$

which is ortonormal and righthanded.

To have the camera positioned in the scene as specified <u>and</u> have it have coordinates as required in OpenGL, we

i) first translate the world by the vector $-\vec{eye}$

ii) then change all point to be expressed in the coordinate system with orthonormal basis $\vec{f_1}, \vec{f_2}, \vec{f_3}$, using the formula $**)$ from page 4.

I.e. we use the transformation matrix

$$F^T \cdot \text{Translate}(-\vec{eye}) .$$

This is analogous to the matrix produced by the call gluLookAt(—).