Computer Graphics - Exercise 07

Maximilian Richter

Winter Semester 2021

1 Active and Passive Rotation (Theory)

a) Generic transformation matrix

$$R(\phi) = \begin{pmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{pmatrix} \quad \text{with } \psi = \frac{2\pi\phi}{360°}$$

Applying this transformation for $\phi = 30°$ to $\mathbf{p} = (0.5, 2)^T$

$$\mathbf{p}' = R(\phi)\mathbf{p} = \begin{pmatrix} \cos(\frac{2\pi 30°}{360°}) & -\sin(\frac{2\pi 30°}{360°}) \\ \sin(\frac{2\pi 30°}{360°}) & \cos(\frac{2\pi 30°}{360°}) \end{pmatrix} \begin{pmatrix} 0.5 \\ 2 \end{pmatrix} = \begin{pmatrix} \sqrt{3}/4 - 1 \\ 1/2 + \sqrt{3} \end{pmatrix} \approx \begin{pmatrix} -0.57 \\ 1.98 \end{pmatrix}$$

b) The basis in this frame of reference is

$$\mathcal{B} = \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$$

The basis of the transformed coordinate system is

$$\mathcal{B} = \left\{ \begin{pmatrix} \sqrt{3}/2 \\ -1/2 \end{pmatrix}, \begin{pmatrix} 1/2 \\ \sqrt{3}/2 \end{pmatrix} \right\}$$

c) The basis transformation matrix $R_{\mathcal{B},\mathcal{B}'}(\phi)$ can be computed from the passive rotation basis, by applying the inverse of the rotation matrix $R(\phi)$

$$R_{\mathcal{B},\mathcal{B}'}(\phi) = R(\phi)^{-1} = R(\phi)^T = \begin{pmatrix} \cos(\psi) & \sin(\psi) \\ -\sin(\psi) & \cos(\psi) \end{pmatrix}$$

This can be verified by applying the transformation matrix to the basis vectors

$$\mathbf{b}_1' = R_{\mathcal{B},\mathcal{B}'}(\phi)\mathbf{b}_1 = \begin{pmatrix} \sqrt{3}/2 & 1/2 \\ -1/2 & \sqrt{3}/2 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \sqrt{3}/2 \\ -1/2 \end{pmatrix}$$

$$\mathbf{b}_2' = R_{\mathcal{B},\mathcal{B}'}(\phi)\mathbf{b}_2 = \begin{pmatrix} \sqrt{3}/2 & 1/2 \\ -1/2 & \sqrt{3}/2 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1/2 \\ \sqrt{3}/2 \end{pmatrix}$$

d) The basis transformation can be used to transform the point coordinates from Figure 1c from the Basis $\mathcal{B}$ to $\mathcal{B}'$ by using the inverse of the basis transformation

$$\mathbf{p}_{\mathcal{B}'}' = R_{\mathcal{B},\mathcal{B}'}^{-1}(\phi)\mathbf{p}_{\mathcal{B}} = \begin{pmatrix} \sqrt{3}/2 & -1/2 \\ 1/2 & \sqrt{3}/2 \end{pmatrix} \begin{pmatrix} 0.5 \\ 2 \end{pmatrix} \approx \begin{pmatrix} -0.57 \\ 1.98 \end{pmatrix}$$

e) One can observe, that the choice of the basis is not important. Active and passive rotation both yield the same result in their respective basis. A visual difference would only be notable if the camera would not be fixed to the basis.

2 Transformations and Homogeneous Coordinates

a) The y-axis has "opposite" signs because rotation in mathematics is defined counter clock-wise. Switching the signs would also yield a valid rotation, however this rotation would be clock-wise and thus an backwards rotation in a right-system.

b) Let $\mathbf{M}$ be a linear transformation. Normal vectors are then transformed by multiplication with the inverse transpose of the transformation

$$\mathbf{n}' = (\mathbf{M}^{-1})^T \mathbf{n}$$

c)

Coordinates	Pros	Cons
World	Parameter t is already distance	Might be difficult to compute
Object	No complicated shear transformations	Matrix inversion, Distance t is not true distance

d) In perspective projection, a 3D point in a truncated pyramid frustum (eye coordinates) is mapped to a cube, the range of x-coordinate from $[l, r]$ to $[-1, 1]$, the y-coordinate from $[b, t]$ to $[-1, 1]$ and the z-coordinate from $[-n, -f]$ to $[-1, 1]$. The homogeneous projection matrix can be expressed as

$$M = \begin{pmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & \frac{-f+n}{f-n} & \frac{-2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

The projected and normalized coordinates are finally calculated by dividing the w-component of the clip coordinates

$$\begin{pmatrix} x_c \\ y_c \\ z_c \\ w_c \end{pmatrix} = M \cdot \begin{pmatrix} x_e \\ y_e \\ z_e \\ w_e \end{pmatrix} \quad \Rightarrow \quad \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} x_c/w_c \\ y_c/w_c \\ z_c/w_c \end{pmatrix}$$