Image Processing and Computer Graphics
Prof. Dr.-Ing. M. Teschner

Exercise 1 - Transformations - Solution

1 Transformation of objects

1. There are different ways that lead to the desired transformation. However, the final transformation obviously is unique except for the scaling coefficient that could be represented in the homogeneous component or in the diagonal entries. We illustrate one possible way.

   First, we transform the point E to the origin:

$$\mathbf{T}_1 := \begin{pmatrix} 1 & 0 & 3 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{pmatrix}.$$

   Then, we rotate the object clockwise by $45°$ which corresponds to a counterclockwise rotation by $315°$:

$$\mathbf{T}_2 := \begin{pmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

   Next, we mirror the object at the x-axis:

$$\mathbf{T}_3 := \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

   After that, the object is scaled by a factor of $1/\sqrt{2}$. Here, we could use the homogeneous coordinate:

$$\mathbf{T}_4 := \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \sqrt{2} \end{pmatrix} \text{ or } \mathbf{T}_4 := \begin{pmatrix} \frac{\sqrt{2}}{2} & 0 & 0 \\ 0 & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

   The last step is to translate the object:

$$\mathbf{T}_5 := \begin{pmatrix} 1 & 0 & 3 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{pmatrix}.$$

As $\mathbf{T}_1$ has to be applied first, it has to be the rightmost transformation, and so on. Hence, the correct order is

$$\mathbf{M} := \mathbf{T}_5\mathbf{T}_4\mathbf{T}_3\mathbf{T}_2\mathbf{T}_1$$

and the final model transform is

$$\mathbf{M} = \begin{pmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{7\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & \frac{13\sqrt{2}}{2} \\ 0 & 0 & \sqrt{2} \end{pmatrix} \text{ or } \mathbf{M} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & \frac{7}{2} \\ \frac{1}{2} & -\frac{1}{2} & \frac{13}{2} \\ 0 & 0 & 1 \end{pmatrix},$$

depending on the representation of $\mathbf{T}_4$.

2. The ModelView transform is given by $\mathbf{V}^{-1}\mathbf{M}$ which results in

$$\begin{pmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{5\sqrt{2}}{2} \\ 0 & 0 & 1 \end{pmatrix} \cdot \mathbf{M} = \begin{pmatrix} \frac{\sqrt{2}}{2} & 0 & \frac{11\sqrt{2}}{2} \\ 0 & -\frac{\sqrt{2}}{2} & \frac{8\sqrt{2}}{2} \\ 0 & 0 & 1 \end{pmatrix} \text{ or } \begin{pmatrix} 1 & 0 & 11 \\ 0 & -1 & 8 \\ 0 & 0 & \sqrt{2} \end{pmatrix}$$

2 Transformation basics in OpenGL

a) *glTranslated*$(2, 0, 0)$ or

GLdouble $trans[16] = \{1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 2, 0, 0, 1\}$; *glMultMatrixd*$(trans)$*;*

b) *glRotated*$(45, 0, 1, 0)$ or

GLdouble $rot[16] = \{\frac{\sqrt{2}}{2}, 0, -\frac{\sqrt{2}}{2}, 0, 0, 1, 0, 0, \frac{\sqrt{2}}{2}, 0, \frac{\sqrt{2}}{2}, 0, 0, 0, 0, 1\}$; *glMultMatrixd*$(rot)$*;*

c)

$$\mathbf{MT} = \begin{pmatrix} \frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & 2 \\ 0 & 1 & 0 & 0 \\ -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

d) First *glTranslated()*, then *glRotated()*. The last mentioned transformation is executed first.

e)

$$\mathbf{MVT} = \mathbf{V}^{-1} \cdot \mathbf{MT} = \begin{pmatrix} \frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & 1 \\ 0 & -1 & 0 & 2 \\ \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} & 3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

3 View Transform

For simplicity, we use the non-homogeneous matrix notation in this solution.

a) The viewing direction is given by $\mathbf{view}' = \mathbf{center} - \mathbf{eye} = (0, -3, 4)^T$. Using the cross product, we obtain $\mathbf{s}' = \mathbf{view}' \times \mathbf{up} = (0, 4, 3)^T$. Normalizing all vectors leads to the desired basis $\mathbf{view} = \frac{1}{5}(0, -3, 4)^T, \mathbf{up} = (1, 0, 0)^T, \mathbf{s} = \frac{1}{5}(0, 4, 3)^T$.

b) We look for a linear mapping $\mathbf{A}$ that maps $\mathbf{s}_0$ to $\mathbf{s}$, $\mathbf{view}_0$ to $\mathbf{view}$ and $\mathbf{up}_0$ to $\mathbf{up}$. Thus, one could solve the equation $\mathbf{AB}_0 = \mathbf{B}$, or one could use some basic Linear Algebra knowledge: If $\mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3)$ with column vectors $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$, then $\mathbf{A} \cdot (1, 0, 0)^T = \mathbf{a}_1, \mathbf{A} \cdot (0, 1, 0)^T = \mathbf{a}_2, \mathbf{A} \cdot (0, 0, 1)^T = \mathbf{a}_3$. Thus, the desired basis transformation is given by

$$\mathbf{A} = (\mathbf{s}, \mathbf{up}, -\mathbf{view}) = \begin{pmatrix} 0 & 1 & 0 \\ \frac{4}{5} & 0 & \frac{3}{5} \\ \frac{3}{5} & 0 & -\frac{4}{5} \end{pmatrix} \tag{1}$$

c) If we transform the camera from the origin to its new position and orientation using $\mathbf{V}$, the rotation has to be executed first, then the translation. Otherwise, the position would be changed by the rotation, as the origin is always the center of the rotation. For the ModelView transform, we need the inverse view transform. Thus, in this case, the translation has to be executed first, and the rotation afterwards.

d) We need the inverse view transform on the ModelView stack. Therefore, the array has to be $(0, 1, 0, 0, \frac{4}{5}, 0, \frac{3}{5}, 0, \frac{3}{5}, 0, -\frac{4}{5}, 0, 0, 0, -5, 1)$.

   For the implementation, it is easier to invert the rotational and translational parts separately and put them onto the ModelView stack.

e) Using the dot product, it is straightforward to show that the vectors are orthogonal:

$$\begin{aligned} \mathbf{up} \cdot \mathbf{view} &= (\mathbf{up}' - \mathbf{view}(\mathbf{up}' \cdot \mathbf{view})) \cdot \mathbf{view} \\ &= \mathbf{up}' \cdot \mathbf{view} - (\mathbf{view} \cdot \mathbf{view})(\mathbf{up}' \cdot \mathbf{view}) \\ &= \mathbf{up}' \cdot \mathbf{view} - \mathbf{up}' \cdot \mathbf{view} = 0. \end{aligned}$$

   Note that this formula holds because $\mathbf{view}$ is normalized.

   Most of the solutions did not show the general case, but used the vectors given in the exercise. Of course, this was also ok.

   For the second alternative, there was a mistake in the exercise: It should be $\mathbf{up}^\times = \mathbf{s} \times \mathbf{view} = -\mathbf{view} \times \mathbf{s}$, so the sign of $\mathbf{up}^\times$ was wrong, but those who calculated the alternative way recognized the mistake. Again, calculating the cross products with the special vectors given in the exercise was a possible way.

   However, it can also be shown in the general case:

$$\mathbf{up}^\times = \mathbf{s} \times \mathbf{view} = (\mathbf{view} \times \mathbf{up}') \times \mathbf{view} = -\mathbf{view} \times (\mathbf{view} \times \mathbf{up}')$$

   This term is of the form $\mathbf{a} \times (\mathbf{b} \times \mathbf{c})$, which is equal to $\mathbf{b}(\mathbf{a} \cdot \mathbf{c}) - \mathbf{c}(\mathbf{a} \cdot \mathbf{b})$. This rule can be shown by simply calculating the components of the two cross products, or it can be found on Wikipedia (see “Kreuzprodukt” or “Cross product”). Finally, we get

$$\begin{aligned} \mathbf{up}^\times &= -(\mathbf{view}(\mathbf{view} \cdot \mathbf{up}') - \mathbf{up}'(\mathbf{view} \cdot \mathbf{view})) \\ &= \mathbf{up}' - \mathbf{view}(\mathbf{view} \cdot \mathbf{up}') = \mathbf{up} \end{aligned}$$