Questions

1. Write the following as a sequence of 4×4 matrices. Scale the scene by a factor 3 in the y direction, while leaving the position of the point $(x, y, z) = (4, -1, 1)$ unchanged.

2. Write the following as a sequence of 4×4 matrices. Rotate the scene θ degrees, such that the axis of rotation is the vector $(1, 1, 0)$ and the origin $(0, 0, 0)$ stays fixed. Define the direction of positive rotation to be *counter*-clockwise when one is looking towards the origin in the direction opposite to the vector, namely one is looking in direction $(-1, -1, 0)$. Assume (as always) a right handed coordinate system.

3. Rewrite the following matrix as a product of a scaling, a rotation, and a translation (not necessarily in that order) :
$$\begin{bmatrix} 0 & -3 & 0 & 10 \\ 2 & 0 & 0 & 11 \\ 0 & 0 & 4 & 12 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

4. Give a transformation that maps the plane $y = f_0$ to the plane $y = 0$ and that maps the plane $y = f_1$ to the plane $y = 1$. Hint: use a translation and a scaling.

5. One can define an ellipsoid surface in $\Re^3$ by scaling, rotating, and translating a unit sphere. Define an ellipsoid whose:

   - center is $(8, 1, 3)$ in world coordinates,
   - x, y, z axes are of length $1, 1, 4$, respectively.
   - z axis is in direction $(1, 1, 0)$ in world coordinates,

   For your answer, it is sufficient that you specify suitable matrices **S, R, T**, and then use them to define a 4×4 quadric matrix **Q**.

6. Let eye $= (0, 0, 0)$ and `lookat` $= (0, 0, 2)$. What is the `GL_MODELVIEW` matrix ?

7. Let the camera be at $(3, 4, 3)$ in world coordinates. Let the z axis of the camera be in direction $(1, -2, 1)$. Let the $\mathbf{V}_{up}$ be $(1, -2, 0)$.

   Give a transformation that maps world coordinates to camera coordinates.

8. When we map object coordinates $\mathbf{x}$ to world coordinates we use $\mathbf{TRx}$ whereas when we map from world to viewer coordinates we typically write $\mathbf{RTx}$. Why do we use the opposite ordering in the two cases?

Solutions

1. Here I'll present two solutions.

   Solution 1: Translate the scene so that the point $(4, -1, 1)$ maps to the origin, then scale the scene in the y direction, then translate the origin back to the point $(4, -1, 1)$.

$$\begin{bmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -4 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

   Verify that the point $(4, -1, 1)$ maps to itself as required. And the scene has been rescaled in the y direction as required. This rescaling is just like changing the "units" on axis, e.g. switching from meters to cm is rescaling by 100. If you are bothered by this solution, think about what's happening in general in 1D if we use this method to rescale the y dimension by s_y by keep $y = y_0$ unchanged:

$$y \to s_y(y - y_0) + y_0 = s_y y - s_y y_0 + y_0$$

   Substitute $y = y_0$ and you'll see that indeed $y_0 \to y_0$.

   Solution 2: First scale the scene in the y direction, and then translate the scene such that the point $(4, -1, 1)$ – which has been mapped to $(4, -3, 1)$ by the rescaling – is mapped back. By inspection, this requires a translation by $(0, 2, 0)$.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

   Either of the above solutions is fine. The second one is simpler for this particular problem, but for other problems you may find the first one to be preferred, especially if rotations are also involved.

2. First rotate to one of the canonical axes (I choose the z-axis), then perform the rotation about the canonical axis, and then rotate the canonical axis back (this can be done by the inverse or transpose of our first rotation matrix).

   First, we look at the rotation matrix that rotates the vector $p = (1, 1, 0)$ to the z-axis. The third row of R is just p itself, and the first and second rows are perpendicular to p and to each other, and also preserve righthandedness.

$$\mathbf{R} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Next, we give the matrix that rotates θ degrees counter-clockwise around the z-axis. For example, rotating by 90 degrees would take the x axis to the y axis.

$$\mathbf{R_z}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Given these matrices, our overall transformation is give by $\mathbf{R}^T\ \mathbf{R_z}(\theta)\ \mathbf{R}$

3. Notice that this transformation takes the origin to the point $(10, 11, 12)$. We can do that the translation last, and concentrate on the upper 3×3 matrix. Doing so, we note that $(1, 0, 0)$ would be mapped to $(0, 2, 0)$ and $(0, 1, 0)$ would be mapped to $(-3, 0, 0)$. Ignoring the scaling aspect, we clearly have a 90 degree counterclockwise (CCW) rotation about the z axis here. We can do the appropriate scaling before or after this rotation.

   Here are two possible solutions.

$$\mathbf{TRS} = \begin{bmatrix} 1 & 0 & 0 & 10 \\ 0 & 1 & 0 & 11 \\ 0 & 0 & 1 & 12 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

$$\mathbf{TSR} = \begin{bmatrix} 1 & 0 & 0 & 10 \\ 0 & 1 & 0 & 11 \\ 0 & 0 & 1 & 12 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

4. The following solution first does the translation and then does the scaling.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{f_1 - f_0} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -f_0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

5. For a unit sphere $x^2 + y^2 + z^2 = 1$,

$$\begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = 0$$

   and so

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}.$$

The question specifies an ellipsoid that is obtained by transforming a unit sphere. The transformation can be achieved by a scaling in z, followed by some rotation that brings the z axis to direction $(1, 1, 0)$ – i.e. maps (0, 0, 1) to $(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0)$ – followed by a translation to bring the center of the ellipsoid to the desired 3D point:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \mathbf{T}\ \mathbf{R}\ \mathbf{S} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 8 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Next comes the tricky part of the question: to obtain the equation of the ellipse, you need to substitute the second equation into the first. To do this, you need to invert the above equation, i.e.

$$(\mathbf{T}\ \mathbf{R}\ \mathbf{S})^{-1} \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

and then substitute into the equation for a sphere given on the previous page. This gives:

$$\begin{bmatrix} x' & y' & z' & 1 \end{bmatrix} ((\mathbf{T}\ \mathbf{R}\ \mathbf{S})^{-1})^T \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} (\mathbf{T}\ \mathbf{R}\ \mathbf{S})^{-1} \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = 0.$$

[ADDED the following on Jan. 28]

Now recall from linear algebra that $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$. If you forget why this is so, think about it like this: you invert the map $\mathbf{AB}$ by undoing the transformations in the opposite order you applied them (last in, first out, like a stack). So, if $\mathbf{x' = AB\ x}$, then $\mathbf{B^{-1}A^{-1}x'} = \mathbf{x}$. Applying this to the problem at hand gives:

$$\begin{bmatrix} x' & y' & z' & 1 \end{bmatrix} (\mathbf{S}^{-1}\mathbf{R}^{-1}\mathbf{T}^{-1})^T \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \mathbf{S}^{-1}\mathbf{R}^{-1}\mathbf{T}^{-1} \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = 0.$$

Note that the inverse translation (and inverse rotation and scaling) are being applied to **x'**. So we are translating some arbitrary point **x'** by (-8,-1,-3). In particular, $\mathbf{T}^{-1}$ maps the point (8,1,3) to (0,0,0). Similarly, the rotation is the inverse of $\mathbf{R}$ and the scaling is the inverse of $\mathbf{S}$, namely $\mathbf{S}^{-1}$ scales the z axis by $\frac{1}{4}$.

To briefly summarize what we've done: the **x'** variable is a point on the ellipse and the **x** variable is a point on the sphere. The question asks you to make the sphere to an ellipse. You know the equation of the sphere, and you can figure out the mapping from the sphere to the ellipse. So you come up with an equation for the inverse mapping, and then substitute it into the known equation of the sphere. This gives you the equation of the ellipse in variable **x'**.

Please note that there is a somewhat related (**but different**) example in the lecture notes (lecture 3 page 5). I have modified and elaborated that example slightly.

6. The direction of the camera's z axis is

$$\mathbf{p}_c - \mathbf{p}_{lookat} = (0,0,0) - (0,0,2) = (0,0,-2)$$

and so the camera's z axis vector (expressed in a world coordinates basis) is:

$$\mathbf{z}_c = (0,0,-1).$$

As is commonly (but not necessarily) done, we choose $\mathbf{V}_{up} = (0,1,0)$. The direction of the camera's x axis is

$$\mathbf{V}_{up} \times \mathbf{z}_c = (-1,0,0)$$

and this happens to be already normalized. Finally,

$$\mathbf{z}_c \times \mathbf{x}_c = \mathbf{y}_c$$

is the camera's y axis. The modelview matrix for this simple situation is just a rotation matrix with the 3×3 upper left submatrix having *rows* $\mathbf{x}_c, \mathbf{y}_c$, and $\mathbf{z}_c$.

7. To transform from world coordinates to camera coordinates, we translate by $(-3,-4,-3)$ and then rotate onto the vectors $\mathbf{x}_c$, $\mathbf{y}_c$ and $\mathbf{z}_c$.

$$\mathbf{z}_c = (\frac{1}{\sqrt{6}}, -\frac{2}{\sqrt{6}}, \frac{1}{\sqrt{6}}), \quad \mathbf{x}_c = (-\frac{2}{\sqrt{5}}, -\frac{1}{\sqrt{5}}, 0), \quad \mathbf{y}_c = (\frac{1}{\sqrt{30}}, -\frac{2}{\sqrt{30}}, -\frac{5}{\sqrt{30}})$$

The matrix product achieving the desired transformation can be written as:

$$\begin{bmatrix} -\frac{2}{\sqrt{5}} & -\frac{1}{\sqrt{5}} & 0 & 0 \\ \frac{1}{\sqrt{30}} & \frac{-2}{\sqrt{30}} & \frac{-5}{\sqrt{30}} & 0 \\ \frac{1}{\sqrt{6}} & -\frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -3 \\ 0 & 1 & 0 & -4 \\ 0 & 0 & 1 & -3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

8. The camera is itself an object. If we want to position and orient the camera in a scene – just like we would position and orient any object – we would first rotate the camera (object) axes and then apply a translation that brings the camera (object) origin to the desired position. It follows that to map any point in the scene from its representation in camera coordinates *to* its representation in world coordinates, you first rotate and then translate.

Now let's go backwards, and map from world coordinates to camera coordinates. This just undoes the mapping from camera to world, so we first translate (back) and then rotate (back), i.e. the ordering from world to camera is the opposite from camera (or an object) to world.