Questions

1. Show that a projective transformation maps a quadric surface to a quadric surface.

   Here, by a "projective transformation", I mean any invertible 4×4 matrix that operates on 3D points which are written in 4D homogeneous coordinates.

2. Give the 4×4 projective transformation matrix that is defined by

   `gluPerspective(theta, aspect, near, far)` .

   Rather than just plugging in the values given in the lecture, think about the special constraints on `gluPerspective`, as opposed to `glFrustum`.

3. How would one choose the first four arguments of `glFrustum` to achieve the same view volume as `gluPerspective( 60, 2, 4, 20)` ?

4. Show that a 4×4 projective matrix $\mathbf{M}$ maps lines to lines.

5. In the lecture I showed that when $\mathbf{M}$ maps a plane π to a plane, the normal $\mathbf{n}$ of the mapped plane $\mathbf{M}\pi$ is $(\mathbf{M}^{-1})^T\mathbf{n}$. Suppose you are given a polygon (and say its normal too) and you apply $\mathbf{M}$ to the vertices of a polygon and to the normal. Suppose you want to know the normal of the polygon in the mapped space. Do you need to know $\mathbf{M}^{-1}$ to compute this new normal?

6. [**ADDED Feb. 15**]

   (a) Write the ellipsoid
   $$(x-3)^2 + y^2 + 4z^2 = 20,$$
   in the form $\mathbf{x}^T\mathbf{Q}\mathbf{x} = 0$ where $\mathbf{x}^T = (x, y, z, 1)$.

   (b) What is the equation of this ellipsoid after it has undergone a projective transformation to the view volume between planes $z = -1$ and $z = -8$.

   Hint: It is sufficient for you to write your answer as a product of matrices and vectors. One of these matrices should be of the form:
   $$\mathbf{M} = \begin{bmatrix} f_0 & 0 & 0 & 0 \\ 0 & f_0 & 0 & 0 \\ 0 & 0 & f_0 + f_1 & -f_0 f_1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Answers

1. Consider an invertible 4×4 matrix $\mathbf{M}$. Let $\mathbf{x}$ be the 4D column vector $(x, y, z, 1)^T$. Then,

$$\begin{aligned} 0 &= \mathbf{x}^T \mathbf{Q} \mathbf{x} \\ &= \mathbf{x}^T \mathbf{M}^T (\mathbf{M}^T)^{-1} \mathbf{Q} \mathbf{M}^{-1} \mathbf{M} \mathbf{x} \\ &= (\mathbf{M}\mathbf{x})^T \ (\mathbf{M}^T)^{-1} \mathbf{Q} \mathbf{M}^{-1} \ \mathbf{M}\mathbf{x} \end{aligned}$$

For a quadric surface, we have a symmetric matrix $\mathbf{Q}$, i.e. ($\mathbf{Q} = \mathbf{Q}^T$), and so the matrix $(\mathbf{M}^T)^{-1}\mathbf{Q}\mathbf{M}^{-1}$ is also symmetric, as we show below. Hence, the transformed points $\mathbf{Mx}$ also lie on a quadric surface.

To show that

$$((\mathbf{M}^T)^{-1}\mathbf{Q}\mathbf{M}^{-1})^T = (\mathbf{M}^T)^{-1}\mathbf{Q}\mathbf{M}^{-1}$$

use the property that $(\mathbf{AB})^T = \mathbf{B}^T \mathbf{A}^T$. We get

$$((\mathbf{M}^T)^{-1}\mathbf{Q}\mathbf{M}^{-1})^T = (\mathbf{M}^{-1})^T \ \mathbf{Q}^T ((\mathbf{M}^T)^{-1})^T$$

Now use the property that $(\mathbf{M}^T)^{-1} = (\mathbf{M}^{-1})^T$ for any invertible $\mathbf{M}$, and $\mathbf{Q}$ symmetric, and we are done.

2. OpenGL first applies (lecture 5, bottom p. 3)

$$\mathbf{M}_{projective} = \begin{bmatrix} \text{near} & 0 & 0 & 0 \\ 0 & \text{near} & 0 & 0 \\ 0 & 0 & \text{near} + \text{far} & \text{near} * \text{far} \\ 0 & 0 & -1 & 0 \end{bmatrix}.$$

OpenGL then normalizes the view volume so it maps to $[-1,1] \times [-1,1] \times [-1,1]$. It does os in a few steps. First, the x and y range of the viewing window is already centered on $(0,0)$ so we don't need to translate it. We do need to translate z, however, to bring $z = -\text{near}$ to $z = 0$. After that, we scale each of the x, y, z dimensions, remembering to flip the z axis. The scale factors can be derived from the definitions of θ and `aspect` and use basic trigonometry. Finally, we translate so that the volume is centered at the origin. Assuming I have not made an error, the normalization is:

$$\mathbf{M}_{normalize} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{2}{\text{near} * \texttt{aspect} * 2 * \tan(\frac{\theta}{2})} & 0 & 0 & 0 \\ 0 & \frac{2}{\text{near} * 2 * \tan(\frac{\theta}{2})} & 0 & 0 \\ 0 & 0 & \frac{2}{\text{near} - \text{far}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \text{near} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3. Since the `aspect` parameter is 2, we have

$$\texttt{left} = -\texttt{right} = 2 * \texttt{bottom} = -2 * \texttt{top}.$$

Since the `fov` is 60 degrees, the angle from the center of the image to the edge of the image (along either the x or y axis) is 30 degrees. The `near` plane is at a distance 4. Using $\tan(30) = \frac{1}{\sqrt{3}}$ we get that

$$(\texttt{left}, \texttt{right}, \texttt{bottom}, \texttt{top}) = (-\frac{8}{\sqrt{3}},\ \frac{8}{\sqrt{3}},\ \ -\frac{4}{\sqrt{3}},\ \frac{4}{\sqrt{3}})$$

4. I showed in the lecture that a projective transformation maps $\mathbf{M}$ planes to planes. It follows that $\mathbf{M}$ maps lines to lines. The reason is that any line can be represented as the intersection of two planes. But since two planes π_1 and π_2 map to planes, $\mathbf{M}\ \pi_1$ and $\mathbf{M}\ \pi_2$, the line which is the intersection of π_1 and π_2 must map to the intersection of planes $\mathbf{M}\ \pi_1$ and $\mathbf{M}\ \pi_2$ which itself must be a line (since the intersection of two planes is a line).

   Note that it follows immediately that triangles map to triangles. I made that claim during the lecture but didn't prove it. So now its proved.

5. No, we can compute the normal instead by taking the cross product

$$\mathbf{n} \equiv \mathbf{M}\mathbf{e_1} \times \mathbf{M}\mathbf{e_2}$$

   for two consecutive edges $\mathbf{e_1}$ and $\mathbf{e_2}$ in the polygon. That is, $\mathbf{M}\mathbf{e_1}$ and $\mathbf{M}\mathbf{e_2}$ are direction vectors, representing the directions of transformed edges of the polygon.

6. (a) We can get the matrix $\mathbf{Q}$ in two ways. One is to use the formula from class (not so interesting and requires less understanding). Expanding the equation, we get

$$x^2 - 6x - 11 + y^2 + 4z^2 = 0,$$

   which can be written in the form $\mathbf{x}^T\mathbf{Q}\mathbf{x} = 0$ with:

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & -3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ -3 & 0 & 0 & -11 \end{bmatrix}.$$

   A second way to get $\mathbf{Q}$ is to notice that x is translated by 3 and that otherwise we can write the matrix down by inspection.

$$\mathbf{x^T}\mathbf{Q}\mathbf{x} \ = \ \mathbf{x^T} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & -20 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}.$$

(b) The projective transformation for near and far clipping planes at $z = -1$ and $z = -8$, is:

$$\mathbf{M} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -9 & -8 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

There are two ways we can write out the equation for the transformed ellipsoid.

The first is easier, and closer to the description I gave in Exercises 3 Question 5 (see also slides of the review lecture 13). If $\mathbf{x}$ is a point on the ellipsoid, then $\mathbf{x}' = \mathbf{M}\mathbf{x}$ is a point on the transformed ellipsoid. The points on the original ellipsoid satisfy $\mathbf{x}^T\mathbf{Q}\mathbf{x} = 0$. We write $\mathbf{M^{-1}x'} = \mathbf{x}$ and substitute, which gives

$$(\mathbf{M^{-1}x'})^T \ \mathbf{Q} \ \mathbf{M^{-1}x'} = 0$$

or

$$\mathbf{x}'^{\ T} \ \mathbf{M^{-T}Q} \ \mathbf{M^{-1}x'} = 0.$$

The second (which is the one I originally posted) involves a trick. Write $\mathbf{x}^T\mathbf{Q}\mathbf{x} = 0$ by inserting two products of matrices with their inverses:

$$(\mathbf{x}^T\mathbf{M}^T)(\mathbf{M}^{-T}\mathbf{Q}\mathbf{M}^{-1})(\mathbf{M}\mathbf{x}) = 0.$$

Thus, the coefficient matrix of the quadric in the projected view volume is $\mathbf{M}^{-T}\mathbf{Q}\mathbf{M}^{-1}$, i.e. the equation for the transformed ellipsoid is

$$\mathbf{x}'\mathbf{M}^{-T}\mathbf{Q}\mathbf{M}^{-1}\mathbf{x}' = 0$$

where $\mathbf{x}' = \mathbf{M}\mathbf{x}$ is the transformed $\mathbf{x}$. The notation $\mathbf{M}^{-T}$ stands for $(\mathbf{M}^{-1})^T$.