— HARVARD UNIVERSITY —

Computer Science 175

Problem Set 9

Due 25 November 2013, 11:59pm

Michaels Tingley and Traver

{*michaeltingley, mtraver*}*@college.harvard.edu*

Michaels Tingley and Traver completed this problem set in tandem.

Problem 1 (Ex. 10.1)

Image **B** is the only viable image. Here's why.

First, let's consider what it means to change the field of view. As read in section 10.3, a change in the field of view of θ degrees is the same as applying a scale of $-n = \frac{1}{\tan\frac{\theta}{2}}$. Thus, any transformation of the view of the cube must be consistent with a scale. Thus, it is clear that **B** is consistent with this — as the front face of the cube grows in size, the rear face grows slower. However, the growth speed of both is proportional. That is, if the front face grows by 20%, the back also appears to grow by 20%. This is consistent with a change of scale.

Meanwhile, this proportional growth rate is not maintained by answers **A** and **C**. In answer **A**, the rear face does not grow at all. In addition to this not being consistent, it begs the following question: *what is happening on behind the rear face?* That is, if we had a third face that was behind the rear cube face, it's unclear what it would do... answer **A** may even imply that it would *shrink*, which is clearly wrong. Answer **C** also doesn't make sense, because the back face grows faster than the front face. This doesn't make any sense; in addition to this not being consistent with proportionate growth, it begs the following question: *what happens if we decreased the field of view even more?* If it were to continue with this trend, the rear face would grow larger the the front face... which doesn't make geometric sense at all.

Problem 2 (Ex. 11.3)

For problems 2, 3, and 4, we use the simplified projection matrix specified in the assignment.

Let's calculate our NDC's x_n, y_n, and z_n for each case. First, the case with just the projection matrix P:

$$
\begin{aligned}
P\begin{pmatrix} x_e \\ y_e \\ z_e \\ 1 \end{pmatrix} &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} x_e \\ y_e \\ z_e \\ 1 \end{pmatrix} \\
&= \begin{pmatrix} x_e \\ y_e \\ 1 \\ -z_e \end{pmatrix} \\
&\Downarrow
\end{aligned}
$$

$$
x_n = -\frac{x_e}{z_e}, y_n = -\frac{y_e}{z_e}, z_n = -\frac{1}{z_e}
$$

Now we compute the NDC's again, this time replacing P with PS:

$$
\begin{aligned}
PS\begin{pmatrix} x_e \\ y_e \\ z_e \\ 1 \end{pmatrix} &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 3 \end{pmatrix} \begin{pmatrix} x_e \\ y_e \\ z_e \\ 1 \end{pmatrix} \\
&= \begin{pmatrix} 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & -3 & 0 \end{pmatrix} \begin{pmatrix} x_e \\ y_e \\ z_e \\ 1 \end{pmatrix} \\
&= \begin{pmatrix} 3x_e \\ 3y_e \\ 3 \\ -3z_e \end{pmatrix} \\
&\Downarrow
\end{aligned}
$$

$$
x_n = -\frac{3x_e}{3z_e}, y_n = -\frac{3y_e}{3z_e}, z_n = -\frac{3}{3z_e}
$$

We see that the NDC's are the same in each case, and because these are the coordinates that determine what is actually drawn, there is no effect on the scene.

Problem 3 (Ex. 11.2)

In problem 2 we already computed the NDC's for the case with just P, so we only need to compute the case in which we replace P with PQ:

$$\begin{aligned}
PQ\begin{pmatrix} x_e \\ y_e \\ z_e \\ 1 \end{pmatrix} &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{pmatrix}\begin{pmatrix} 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} x_e \\ y_e \\ z_e \\ 1 \end{pmatrix} \\
&= \begin{pmatrix} 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -3 & 0 \end{pmatrix}\begin{pmatrix} x_e \\ y_e \\ z_e \\ 1 \end{pmatrix} \\
&= \begin{pmatrix} 3x_e \\ 3y_e \\ 1 \\ -3z_e \end{pmatrix} \\
&\Downarrow \\
x_n = -\frac{3x_e}{3z_e}&, y_n = -\frac{3y_e}{3z_e}, z_n = -\frac{1}{3z_e}
\end{aligned}$$

We see that x_n and y_n have not changed, so the position of objects on screen will not change. However, z_n has changed; dividing it by 3 has the effect of pushing everything into a smaller range within the (fixed) z-buffer range, so objects may appear to overlap/be drawn in the wrong order if their z_n values are close enough to be "pushed together" by scaling z_n.

Problem 4 (Ex. 11.4)

$$\begin{aligned}
QP\begin{pmatrix} x_e \\ y_e \\ z_e \\ 1 \end{pmatrix} &= \begin{pmatrix} 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{pmatrix}\begin{pmatrix} x_e \\ y_e \\ z_e \\ 1 \end{pmatrix} \\
&= \begin{pmatrix} 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & -1 & 0 \end{pmatrix}\begin{pmatrix} x_e \\ y_e \\ z_e \\ 1 \end{pmatrix} \\
&= \begin{pmatrix} 3x_e \\ 3y_e \\ 3 \\ -z_e \end{pmatrix} \\
&\Downarrow \\
x_n = -3\frac{x_e}{z_e}&, y_n = -3\frac{y_e}{z_e}, z_n = -\frac{1}{z_e}
\end{aligned}$$

We see that x_n and y_n have been multiplied by 3, and z_n has not changed. This has the effect of zooming in toward the center of the scene by a factor of 3.

Problem 5 (Ex. 12.3)

$$
\begin{aligned}
QV &= \begin{pmatrix} 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{W}{2} & 0 & 0 & \frac{W-1}{2} \\ 0 & \frac{H}{2} & 0 & \frac{H-1}{2} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} \frac{3W}{2} & 0 & 0 & \frac{3(W-1)}{2} \\ 0 & \frac{3H}{2} & 0 & \frac{3(H-1)}{2} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix}
\end{aligned}
$$

Now let's consider the difference between $V \begin{pmatrix} x_n \\ y_n \\ z_n \\ 1 \end{pmatrix}$ (this is the normal transformation used with the viewport) and $QV \begin{pmatrix} x_n \\ y_n \\ z_n \\ 1 \end{pmatrix}$.

$$
\begin{aligned}
V \begin{pmatrix} x_n \\ y_n \\ z_n \\ 1 \end{pmatrix} &= \begin{pmatrix} \frac{W}{2} & 0 & 0 & \frac{W-1}{2} \\ 0 & \frac{H}{2} & 0 & \frac{H-1}{2} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_n \\ y_n \\ z_n \\ 1 \end{pmatrix} \\
&= \begin{pmatrix} \frac{1}{2}\left(x_n W + W - 1\right) \\ \frac{1}{2}\left(y_n H + H - 1\right) \\ \frac{1}{2}\left(z_n + 1\right) \\ 1 \end{pmatrix}
\end{aligned}
$$

$$
\begin{aligned}
QV \begin{pmatrix} x_n \\ y_n \\ z_n \\ 1 \end{pmatrix} &= \begin{pmatrix} \frac{3W}{2} & 0 & 0 & \frac{3(W-1)}{2} \\ 0 & \frac{3H}{2} & 0 & \frac{3(H-1)}{2} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_n \\ y_n \\ z_n \\ 1 \end{pmatrix} \\
&= \begin{pmatrix} \frac{3}{2}\left(x_n W + W - 1\right) \\ \frac{3}{2}\left(y_n H + H - 1\right) \\ \frac{1}{2}\left(z_n + 1\right) \\ 1 \end{pmatrix}
\end{aligned}
$$

Now let's examine $QV \begin{pmatrix} x_n \\ y_n \\ z_n \\ 1 \end{pmatrix} \Big/ V \begin{pmatrix} x_n \\ y_n \\ z_n \\ 1 \end{pmatrix}$ (that is, element-wise divide). We find that this gives $\begin{pmatrix} 3 \\ 3 \\ 1 \\ 1 \end{pmatrix}$. This is interesting, because it means that the change implied by this transformation is simply to scale up the x_w and y_w coordinates by a factor of 3, effectively zooming in. *However*, it is important to realize that this scaling is done *after* applying the viewport matrix. Therefore, while in our previous results (i.e., the result from 11.4) the scaling was done 'intuitively' (exploding the image from the center), here the scaling is done from the bottom left corner, which is the origin of the viewport coordinate system.

Problem 6

First we use a texture viewport matrix to convert our query point $[0.45, 0.63]^t$ into texture "window" coordinates within our 512×512 texture:

$$\begin{pmatrix} 512 & 0 & 0 & -\frac{1}{2} \\ 0 & 512 & 0 & -\frac{1}{2} \\ - & - & - & - \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0.45 \\ 0.63 \\ - \\ 1 \end{pmatrix} = \begin{pmatrix} 229.9 \\ 322.06 \\ - \\ 1 \end{pmatrix}$$

These coordinates are closest to the pixel center at $[230, 322]^t$. Now we convert these integer-valued coordinates back to the domain of the canonical unit square by applying the inverse of the texture viewport matrix used above. But since it's a pretty simple matrix, it's faster to just do some quick math to achieve the same effect without having to do matrix inversion and multiplication:

$$\begin{aligned} \frac{230 + \frac{1}{2}}{512} &= 0.450195 \\ \frac{322 + \frac{1}{2}}{512} &= 0.629883 \end{aligned}$$

So the pixel center closest to the query point, in the domain of the canonical unit square, is at $[0.450195, 0.629883]^t$.

Problem 7

Given the following,

$$v = \begin{pmatrix} a & b & c \end{pmatrix} \begin{pmatrix} x_n \\ y_n \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} w_n x_n \\ w_n y_n \\ w_n \end{pmatrix} = w_n \begin{pmatrix} x_n \\ y_n \\ 1 \end{pmatrix} = P' \begin{pmatrix} x_e \\ y_e \\ z_e \\ 1 \end{pmatrix}$$

$$1 = \begin{pmatrix} d & e & f \end{pmatrix} \begin{pmatrix} x_n \\ y_n \\ 1 \end{pmatrix}$$

we wish to interpolate v, which is affine w.r.t. NDC's but not eye coordinates, with respect to eye coordinates. Using the constant 1 function, we see that

$$w_n \begin{pmatrix} v \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \end{pmatrix} \begin{pmatrix} x_n \\ y_n \\ 1 \end{pmatrix} w_n$$

Substituting in from the relationship between NDC's and eye coordinates,

$$w_n \begin{pmatrix} v \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \end{pmatrix} P' \begin{pmatrix} x_e \\ y_e \\ z_e \\ 1 \end{pmatrix}$$

and dividing through by w_n,

$$\begin{pmatrix} v \\ 1 \end{pmatrix} = \frac{1}{w_n} \begin{pmatrix} a & b & c \\ d & e & f \end{pmatrix} P' \begin{pmatrix} x_e \\ y_e \\ z_e \\ 1 \end{pmatrix}$$

We can now compute the value of v using eye coordinates, so we can therefore interpolate it w.r.t. eye coordinates.