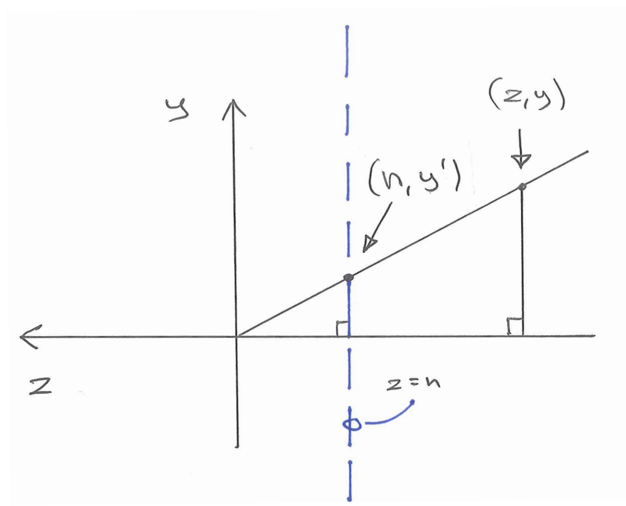
Department of Mathematics and Computer Science
University of Southern Denmark
January 24, 2020
RF

Perspective Projection I

In this note, names denoting vector values are shown in bold (instead of using arrows above the name).

Recall that by perspective projection, we mean moving a point $\boldsymbol{p} = (x, y, z) \in \mathbb{R}^3$ to the point of intersection between the *viewing plane* and the line going through the camera position and $\boldsymbol{p}$. In 3D graphics, one usually takes origo to be the camera position and the viewing plane to be a plane perpendicular to the z-axis. Such a plane is given by the equation $z = n$, where n is some non-zero value chosen by the user. In OpenGL, n is usually negative. In other words, the camera is positioned in origo and is looking down the negative z-axis. Note that such an intersection is defined if and only if $\boldsymbol{p}$ does not lie in the xy-plane. We in the following assume that $\boldsymbol{p} = (x, y, z)$ fulfills $z < 0$, unless otherwise noted.

The following figure shows the situation from the side (y-axis is up).



Using the fact that triangles with same angles are scalings of each other, it follows from the figure that $n = cz$ and $y' = cy$ for some scaling factor c.

From this, we get $n/z = c$ and then $y' = yn/z$. Thus, the projected y-value is $y' = yn/z$. A similar figure can be drawn with the y-axis exchanged by the x-axis (which means looking from below), from which the same argument shows that the projected x-value is $x' = xn/z$. In short, the perspective projection is given by the following mapping $\boldsymbol{f}$:

$$\boldsymbol{f}(x, y, z) = \begin{pmatrix} xn/z \\ yn/z \end{pmatrix}$$

We now prove the following fact. It is interesting by itself, and e.g. implies that triangles are projected to triangles,[1] such that we can just project the three vertices of a triangle, and then find the full projected triangle by drawing lines between the three projected points and filling out the area (rasterization). We furthermore will have good use of the function λ constructed during the proof.

Theorem 1 *For the mapping $\boldsymbol{f}$, line segments not crossing the xy-plane are mapped to line segments.*

Proof: Let $\boldsymbol{l}$ be a given line segment from $\boldsymbol{p}_0 = (x_0, y_0, z_0)$ to $\boldsymbol{p}_1 = (x_1, y_1, z_1)$, where z_0 and z_1 are non-zero and have the same sign. The points on $\boldsymbol{l}$ can be expessed as

$$\boldsymbol{l}(s) = \boldsymbol{p}_0 + s(\boldsymbol{p}_1 - \boldsymbol{p}_0) = \begin{pmatrix} x_0 + s(x_1 - x_0) \\ y_0 + s(y_1 - y_0) \\ z_0 + s(z_1 - z_0) \end{pmatrix}$$

for $s \in [0, 1]$. Let us define

$$\lambda(s) = \frac{z_1 s}{z_0 + (z_1 - z_0)s}.$$

[1] The three line segments constituting the sides of the triangle are projected to three line segments. The interior of the triangle can be seen as the union of all line segments between a corner point and points on its opposing edge. These line segments are mapped to line segments constituting the interior of the mapped triangle.

Looking at $\boldsymbol{f}(\boldsymbol{l}(s))$ we can calculate

$$\boldsymbol{f}(\boldsymbol{l}(s)) = \begin{pmatrix} \dfrac{(x_0 + s(x_1 - x_0))n}{z_0 + s(z_1 - z_0)} \\[2ex] \dfrac{(y_0 + s(y_1 - y_0))n}{z_0 + s(z_1 - z_0)} \end{pmatrix}$$

$$= \begin{pmatrix} \dfrac{x_0 n}{z_0} + \lambda(s)(\dfrac{x_1 n}{z_1} - \dfrac{x_0 n}{z_0}) \\[2ex] \dfrac{y_0 n}{z_0} + \lambda(s)(\dfrac{y_1 n}{z_1} - \dfrac{y_0 n}{z_0}) \end{pmatrix}$$

$$= \boldsymbol{f}(\boldsymbol{p}_0) + \lambda(s)(\boldsymbol{f}(\boldsymbol{p}_1) - \boldsymbol{f}(\boldsymbol{p}_0)),$$

where the calculations behind the second equality are relegated to Lemma 3 below.

By Lemma 2 below, for $s \in [0, 1]$ the points in

$$\boldsymbol{f}(\boldsymbol{p}_0) + \lambda(s)(\boldsymbol{f}(\boldsymbol{p}_1) - \boldsymbol{f}(\boldsymbol{p}_0))$$

will be the same as the points

$$\boldsymbol{f}(\boldsymbol{p}_0) + t(\boldsymbol{f}(\boldsymbol{p}_1) - \boldsymbol{f}(\boldsymbol{p}_0))$$

for $t \in [0, 1]$, that is, exactly the points of the line segment from $\boldsymbol{f}(\boldsymbol{p}_0)$ to $\boldsymbol{f}(\boldsymbol{p}_1)$. □

In short, the proof above shows that when traversing the line

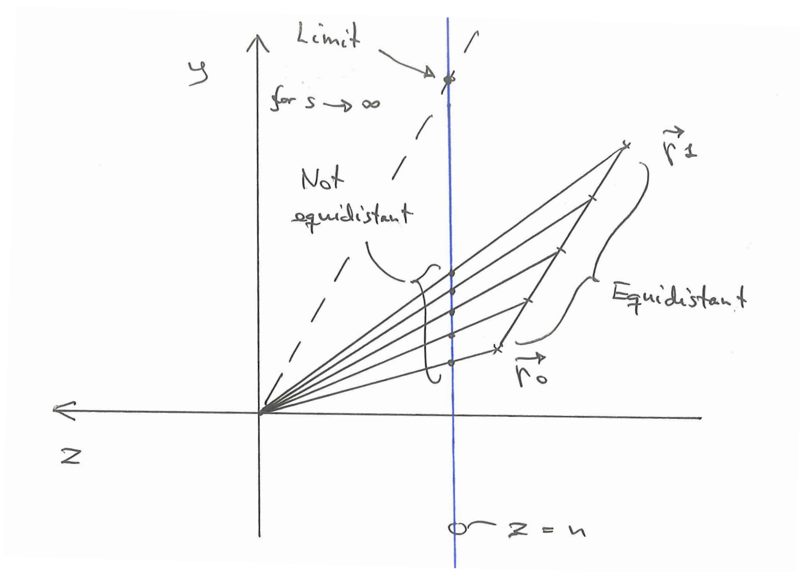
$$\boldsymbol{l}(s) = \boldsymbol{p}_0 + s(\boldsymbol{p}_1 - \boldsymbol{p}_0), \quad s \in [0, 1],$$

between the endpoints $\boldsymbol{p}_0$ and $\boldsymbol{p}_1$, the mapping $\boldsymbol{f}(\boldsymbol{l}(s))$ of the traversed point $\boldsymbol{l}(s)$ will itself traverse the line

$$\boldsymbol{f}(\boldsymbol{p}_0) + t(\boldsymbol{f}(\boldsymbol{p}_1) - \boldsymbol{f}(\boldsymbol{p}_0)), \quad t \in [0, 1]$$

between the mapped endpoints $\boldsymbol{f}(\boldsymbol{p}_0)$ and $\boldsymbol{f}(\boldsymbol{p}_1)$.

However, the lines are *not* traversed at the same speed in the two expressions above. This is to be expected, as the following figure shows.



A very useful aspect of the proof is that we actually know the correlation between the two speeds. That is, we know the relationship between the positions (values of the traversal parameters s and t) for corresponding points $\boldsymbol{l}(s)$ and $\boldsymbol{f}(\boldsymbol{l}(s))$ on the two traversals—it is given by $t = \lambda(s)$.

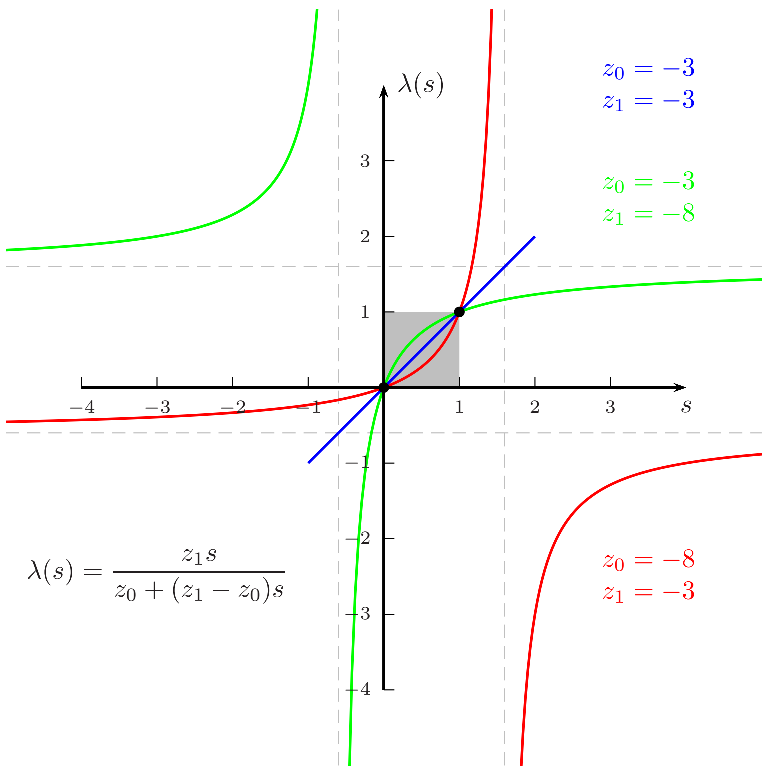
The inverse of λ can be found by solving $t = z_1 s/(z_0 + (z_1 - z_0)s)$ for s. An easy calculation shows this to be

$$s = \frac{z_0 t}{z_1 - (z_1 - z_0)t},$$

which we may call $s = \lambda^{-1}(t)$. This is useful when doing texture lookup along a line (during shading) based on only knowing the texture coordinates for the endpoints of the line. The interesting points along the line are generated by the pixels which the line covers, i.e., these points are equidistant on the screen (after projection), and have evenly spaced values of t. However, the texture is assumed to be "glued" over the line in world space (before projection), so the texture should be sampled by using the corresponding values of s (which are not equidistant, but can be found via λ^{-1}).

For interpolation across a triangle (rather than a line), a similar correspondence based on barycentric coordinates can be developed (see handout by Eberly, page 83 (not curriculum)). Using these methods during texture lookup is called *interpolation*.

Below, we show the function $\lambda(s)$ for three pairs of values of z_0 and z_1. The gray box illustrates Lemma 2. Asymptotes are also shown. E.g., for $s \to \infty$ we have $\lambda(s) \to z_1/(z_1 - z_0)$, which for the green example is 8/5. That $t = \lambda(s)$ should be bounded for $s \to \infty$ is clearly to be expected from the figure above.



Lemma 2 *If z_0 and z_1 are non-zero and have the same sign, λ is a bijective (i.e., 1-1 and onto) mapping from $[0, 1]$ to $[0, 1]$.*

Proof: We note that $\lambda(0) = 0$ and $\lambda(1) = 1$. Hence, if we can prove λ continous and strictly increasing on $[0, 1]$, we are done.

Addition and multiplication are continous functions. Division is a continous function on intervals where the divisor is not zero. Hence λ is continous if the denominator is not zero. Since $z_0 + (z_1 - z_0)s = 0 \Leftrightarrow s = -z_0/(z_1 - z_0) =$

$z_0/(z_0 - z_1)$, we can see that the denominator in $\lambda(s)$ is not zero for any $s \in [0, 1]$ by a case analysis: i) for $z_1 > z_0 > 0$ or $0 > z_0 \geq z_1$ a denominator of zero means $s < 0$, ii) for $z_0 > z_1 > 0$ or $0 > z_1 \geq z_0$ a denominator of zero means $s > 1$, iii) for $z_0 = z_1$ we simply have $\lambda(s) = s$. In all cases, $\lambda(s)$ is continous on $[0, 1]$.

To show that λ is increasing, we use the quotient rule for differentiation and get $\lambda'(s) = (z_1(z_0 + (z_1 - z_0)s) - z_1 s(z_1 - z_0))/(z_0 + (z_1 - z_0)s)^2 = z_1 z_0/(z_0 + (z_1 - z_0))^2$, hence $\lambda'(s) > 0$ when z_0 and z_1 are non-zero and have the same sign. □

Lemma 3 *We have*

$$\frac{(x_0 + s(x_1 - x_0))n}{z_0 + s(z_1 - z_0)} = \frac{x_0 n}{z_0} + \lambda(s)(\frac{x_1 n}{z_1} - \frac{x_0 n}{z_0}),$$

and similar with y in x's place.

Proof: For x we have

$$\begin{aligned}
\frac{(x_0 + s(x_1 - x_0))n}{z_0 + s(z_1 - z_0)} &= \frac{(x_0(1 - s) + s x_1)n}{z_0 + s(z_1 - z_0)} \\
&= \frac{\frac{x_0 n}{z_0}(z_0 - s z_0) + s x_1 n}{z_0 + s(z_1 - z_0)} \\
&= \frac{\frac{x_0 n}{z_0}(z_0 + s(z_1 - z_0) + s z_1) + s x_1 n}{z_0 + s(z_1 - z_0)} \\
&= \frac{x_0 n}{z_0} + \frac{s x_1 n - s z_1 \frac{x_0 n}{z_0}}{z_0 + s(z_1 - z_0)} \\
&= \frac{x_0 n}{z_0} + \frac{s z_1}{z_0 + s(z_1 - z_0)}(\frac{x_1 n}{z_1} - \frac{x_0 n}{z_0}) \\
&= \frac{x_0 n}{z_0} + \lambda(s)(\frac{x_1 n}{z_1} - \frac{x_0 n}{z_0}).
\end{aligned}$$

The calculations for y are exactly the same.

□