Department of Mathematics and Computer Science
University of Southern Denmark
January 3, 2018
RF

Perspective Projection II

As in Note I, names denoting vector values are shown in bold (instead of using arrows above the name).

In note I, we considered projection as a mapping from $\mathbb{R}^3$ to $\mathbb{R}^2$, and derived the mathematical expresssion for it. However, in the rendering pipeline, it is desirable to make projection a two-step process, where the view frustrum given by values l, r, b, t, n, f (for left, right, bottom, top, near, far) is first mapped to a standard box with dimensions

$$[-1, 1] \times [-1, 1] \times [-1, 1]$$

(called *clip space*) in such a way that:

1. Lines are mapped to lines.
2. Points on a line are inside the view frustum before the mapping if and only if they are inside the box after the mapping.
3. The expressions for x and y-coordinate after the mapping are those from note I (possibly with translations and scalings applied to it).

The main advantage is that by points 1 and 2, clipping can be done against this single, standardized box for all frustrums, hence the clipping algorithm can be hardcoded in the rendering pipeline. Also, the perspective projection is retained, by point 3.

The cores step is creating a mapping

$$\boldsymbol{P}(x, y, z) = \begin{pmatrix} P_1(x, y, z) \\ P_2(x, y, z) \\ P_3(x, y, z) \end{pmatrix}$$

which maps the view frustrum to an axis-aligned box with dimensions

$$[l, r] \times [b, t] \times [-1, 1]$$

and which fulfills

1. Lines are mapped to lines.
2. Points on a line are inside the view frustum before the mapping if and only if they are inside the box after the mapping.
3. The expressions for x and y-coordinate after the mapping are exactly those of $\boldsymbol{f}$ from note I:

$$\begin{aligned} P_1(x, y, z) &= xn/z \\ P_2(x, y, z) &= yn/z \end{aligned}$$

We now consider mapping a line $\boldsymbol{l}(s)$ not parallel to the near plane [given by $z = n$, which is the same as not being parallel to the xy-plane]. It is easy to verify in the end that the final mapping also works for lines parallel to this plane. We let $\boldsymbol{q}_0$ be the point on the line lying in the near plane $z = n$, and $\boldsymbol{q}_1$ be the point on the line lying in the far plane $z = f$. All points on the line is given by

$$\boldsymbol{l}(s) = \boldsymbol{q}_0 + s(\boldsymbol{q}_1 - \boldsymbol{q}_0)$$

for $s \in \mathbb{R}$.

Let $(x'_0, y'_0) = \boldsymbol{f}(\boldsymbol{q}_0)$ and $(x'_1, y'_1) = \boldsymbol{f}(\boldsymbol{q}_1)$ and consider the parametric line expression

$$\boldsymbol{\alpha}(t) = \begin{pmatrix} x'_0 \\ y'_0 \\ -1 \end{pmatrix} + t \begin{pmatrix} x'_1 - x'_0 \\ y'_1 - y'_0 \\ 1 - (-1) \end{pmatrix}$$

We now define the mapping $\boldsymbol{P}$ for all points $\boldsymbol{l}(s)$ on the line through $\boldsymbol{q}_0$ and $\boldsymbol{q}_1$ by mapping $\boldsymbol{l}(s)$ to $\boldsymbol{\alpha}(t)$ for $t = \lambda(s)$, where λ is the function from *Perspective Projection I.*

For the points $\boldsymbol{l}(s)$, we have:

- Point 3 above is fulfilled, by the definition of $\boldsymbol{f}$ and the proof of Theorem 1 in *Perspective Projection I.*

- Point 1 above is fulfilled by construction ($\boldsymbol{\alpha}$ is a line by construction).
- Point 2 above is fulfilled by

$$
\begin{aligned}
& \boldsymbol{l}(s) \text{ lies in frustrum} \\
\Updownarrow \quad & \\
& s \in [0,1] \text{ and } \boldsymbol{f}(\boldsymbol{l}(s)) \in [l,r] \times [b,t] \\
\Updownarrow \quad & \\
& \lambda(s) \in [0,1] \text{ and} \\
& (x'_0 + \lambda(s)(x'_1 - x'_0), y'_0 + \lambda(s)(y'_1 - y'_0)) \in [l,r] \times [b,t] \\
\Updownarrow \quad & \\
& \boldsymbol{P}(\boldsymbol{l}(s)) = \alpha(\lambda(s)) \in [l,r] \times [b,t] \times [-1,1]
\end{aligned}
$$

  where the second step follows from Lemma 2 and from the proof of Theorem 1 from *Perspective Projection I.*

The next step is to express the mapping $\boldsymbol{P}$ (so far only defined on the line $\boldsymbol{l}(s)$) in terms of the (x,y,z) coordinates of the point $\boldsymbol{l}(s)$ instead of in terms of the line parameters s and $t = \lambda(s)$ as above.

We already know the result for the first two coordinate expressions P_1 and P_2, by point 3: $P_1(x,y,z) = xn/z$ and $P_2(x,y,z) = yn/z$).

The third coordinate function P_3 by construction of $\boldsymbol{\alpha}$ has the value $-1 + \lambda(s)(1-(-1)) = 2\lambda(s) - 1$. From the definition of $\boldsymbol{l}(s)$, we know that $z = n + s(f-n)$, from which we see $s = (z-n)/(f-n)$. Inserting this expression for s and the definition of $\lambda(s)$ into the value above for P_3 leads to[1]

$$P_3(x,y,z) = \frac{f + n - 2fn/z}{f-n}.$$

Since we now have $\boldsymbol{P}$ expressed entirely in terms of (x,y,z)

$$\boldsymbol{P}(x,y,z) = \begin{pmatrix} xn/z \\ yn/z \\ \dfrac{f+n-2fn/z}{f-n} \end{pmatrix},$$

we see that the mapping can be expressed independently of the actual line $\boldsymbol{l}$, and in fact will fulfill points 1–3 for all lines, and also is defined for the entire

[1] Calculations can be found in the handwritten notes *Perspective Projection, Derivation*, pages 10-11 (where λ is called $\tilde{f}$)

space (more precisely, on the two parts of $\mathbb{R}^3$ where $z < 0$ and where $z > 0$, since this is required in Theorem 1 from *Perspective Projection I*).

Massaging the mapping $\boldsymbol{P}$ such that it maps the frustum into $[-1,1] \times [-1,1] \times [-1,1]$ rather than $[l,r] \times [b,t] \times [-1,1]$ is a simple matter of applying translation and scaling on P_1 and P_2. See the handwritten notes *Perspective Projection, Derivation*, pages 12-14.