Understanding View Frustums and Homogeneous Coordinates

Jaakko Lehtinen, MIT CSAIL

September 15, 2009

Let's look at Figure. 1, a 2D illustration of a canonical projection situation. We have a "camera" situated at the origin $(0, 0)$, looking up along the positive z axis, with a field of view of 90 degrees. It suffices to consider only the 90 degree case, because any other field of view can be constructed from this case by simple scaling of the x axis.

The projection of the point $p = (x, z)$ onto the "image plane" $z = 1$ is given by dividing the coordinates by z, i.e., $p' = (x/z, 1)$. This follows from simple similar triangle arguments. Let's see how this can be written in homogeneous coordinates. First, we add a trailing $w = 1$ to p, yielding $p = (x, z, 1)$. We've now turned a 2D point into a 3D one that lies in the plane $w = 1$. You can think of the w coordinate axis going directly into the page such that $w = 1$ on the page. Now, how do we produce $(x/z, 1, 1)$, the projected point p', from $(x, z, 1)$? Simply by putting z in the w coordinate, because (x, z, z) yields exactly the correct answer when divided by the 3rd w coordinate. This is certainly possible using a matrix:

$$p' = \begin{pmatrix} x/z \\ 1 \\ 1 \end{pmatrix} \propto \begin{pmatrix} x' \\ z' \\ w' \end{pmatrix} = \begin{pmatrix} x \\ z \\ z \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ z \\ 1 \end{pmatrix}. \tag{1}$$

Here the $x'z'w'$ coordinates denote the *post-perspective* 3D coordinates that result from multiplying p by the above matrix, and the symbol $\propto$ denotes "proportional to", i.e., it reflects the fact that we still have to homogenize the result of the matrix-vector multiplication by dividing by w' (which happens to equal z in this case) to get back to the canonical case $w' = 1$. Obviously, once we do that, the result is $(x/z, 1, 1)$, i.e., the correct projected point p'.

Now, looking at the matrix in Eq. (1), it is obvious that it is not invertible. Why? The 2nd and 3rd rows are the same, which leads to a zero determinant, which is equivalent to non-invertibility. What is happening geometrically is that we are flattening the whole 2-dimensional xz space onto the 1D line $z = 1$ – indeed, excluding the line $z = 0$, no matter which point $(x, z, 1)$ you transform by the matrix, the result will always have $z' = 1$ after dividing by w'.

Now, this math is all fine in the sense that it tells us exactly where a given point in view space will project to on the image plane. But what about occlusion? After we've flattened everything onto the image plane, there is no way to tell, using only the information from the projected points, which surfaces lie in front of others! This is not

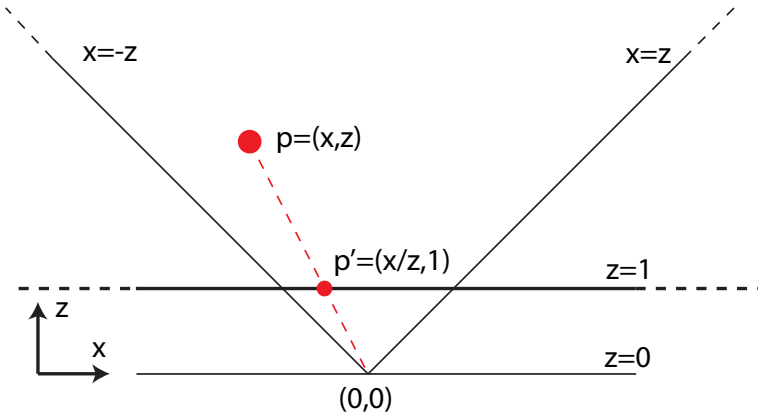

Figure 1: Canonical 2D projection with the camera situated in the origin, looking up. The scene is projected onto the line $z = 1$.

a problem for ray tracing systems, because the rays shot from the camera are (more or less) traced through world space without projecting the object points. However, Graphics processors (GPUs) work in the other direction by projecting triangles on the screen one by one, and they need distance information in order to resolve visibility, i.e., what is in front of what. Fortunately, there is a way around this problem.

Let's define a *view frustum* by truncating the infinite pyramid formed by the $x = -z$ and $x = z$ lines, as seen in Figure 1, by capping it at the front at $z =$ near and at the back at $z =$ far. This is illustrated in Figure 2. The volume that remains inside these lines is called the view frustum. Its boundary is marked in blue. Now, let's take the points on the boundary of the frustum and see what happens to them when we transform them using the projection matrix in Eq. (1). (Remember, the frustum lies, before projection, on the $w = 1$ plane of the 3D xzw space, and it is mapped by the matrix to the post-perspective $x'z'w'$ coordinates, which are still 3D.) The result of this is illustrated in Fig. 3 in the post-perspective $x'z'w'$ space, as seen from the "side" so that x' is coming out of the paper, z' goes up, and w' goes to the right. As we are looking from the side, the frustum is seen as a blue line segment. By looking at the matrix in (1), it's easy to convince oneself of the fact that the entire frustum lies in the $z' = w'$ plane – indeed, the matrix disregards the original w and places the same value (z) in both z' and w'. Now, it's also easy, looking at Fig. 3a), to visually confirm the fact we noted earlier: each point in the frustum, no matter what its original z coordinate was, will land on the $z' = w' = 1$ line after homogenization – just trace it back along the line towards the origin and intersect with the $w' = 1$ line. What this means is that we've lost all information about the distance to the camera in post-perspective space.

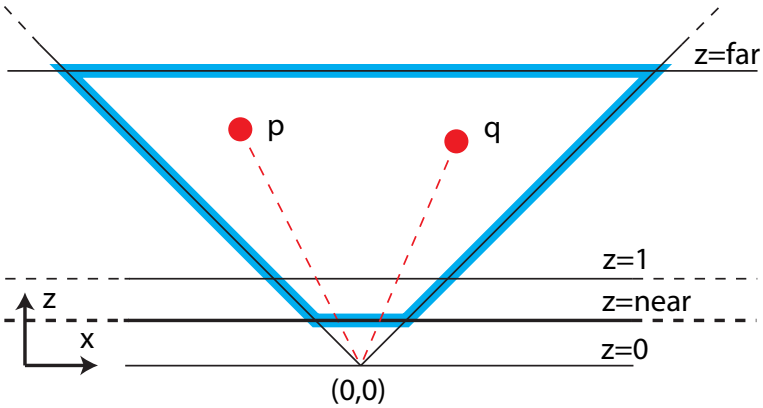

Figure 2: the view frustum.

To retain information about the relative depths of points *after projection*, we use a projection matrix that maps the frustum onto a different plane, while keeping the "image coordinate" x' unchanged. We wish to map points on the "front plane" $z =$ near onto the plane $z' = -1$ and the points on the "back plane" $z =$ far onto the plane $z' = 1$.[1] This is illustrated in Fig. 3b). By looking at the picture it is clear that this indeed happens: At $w' =$ far we have also $z' =$ far, meaning that this point will have $z'/w' = 1$, while at $w' =$ near we have $z' = -$near so that $z'/w' = -1$. The points between the near and far planes are mapped into increasing z'/w' between -1 and 1, meaning that we have retained the information about the points' distance to the camera: Points with smaller z'/w' are closer to the camera. It can also be noted that setting the near plane to 0 reduces to the original case of $z' = w'$ that loses information. The modified projection equation is given by

$$p' \propto \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{\text{far}+\text{near}}{\text{far}-\text{near}} & -\frac{2*\text{far}*\text{near}}{\text{far}-\text{near}} \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ z \\ 1 \end{pmatrix}. \tag{2}$$

The geometric interpretation of this whole process is that we've turned the original viewing frustum (the truncated pyramid) into a rectangle in post-transform (homogenized) space that ranges from -1 to 1 on the x' axis and from -1 to 1 on the z' axis. This is called the *canonical view volume*. The projected screen coordinate x' given by the new mapping is exactly the same as that given by Eq. (1), so you can think of doing the final 2D to 1D projection in the canonical view volume by dropping the z' coordinate. (This is an orthographic projection.) The relationship of the view frustum and the canonical view volume is illustrated in Fig. 4.

[1] DirectX defines the mapping so that the near plane maps to $z' = 0$ instead, while the far plane still maps to $z' = 1$. This amounts to a simple modification of the plane equation.

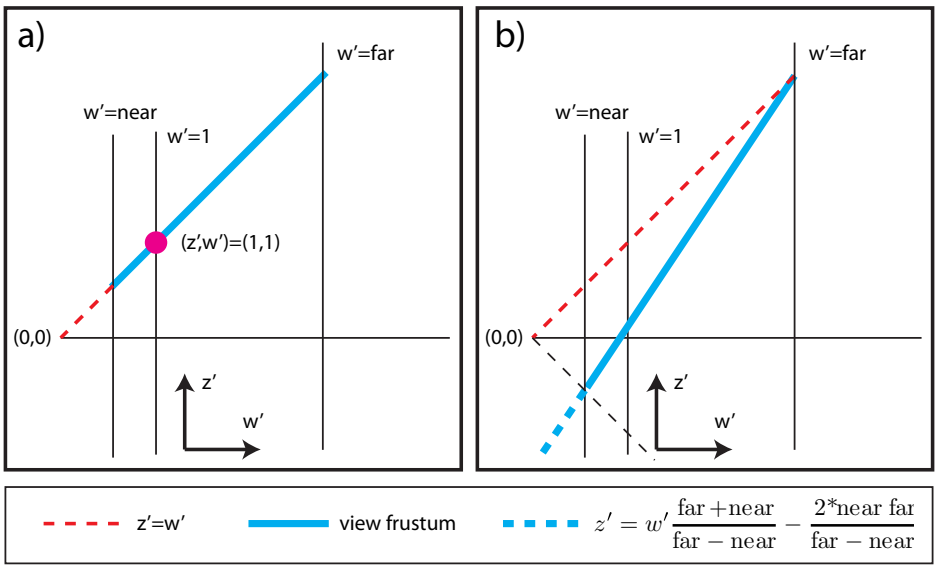

Figure 3: **a)** The effect of the projection matrix of Eq. (1) on the view frustum as seen in post-perspective $x'z'w'$ coordinates from the "side view" $z'w'$. The transformed frustum lies entirely on the $z' = w'$ plane, meaning that all points within it will project to $z' = 1$ when homogenized (divided by w'). **b)** The invertible projection matrix. The plane on which the view frustum is mapped is chosen so that the points on $z = \text{far}$ land on the plane $z' = 1$ when homogenized, and the points on $z = \text{near}$ land on $z' = -1$. The points in between are mapped to intermediate values in a monotonous but non-uniform manner. Note that choosing near $= 0$ reduces to a).

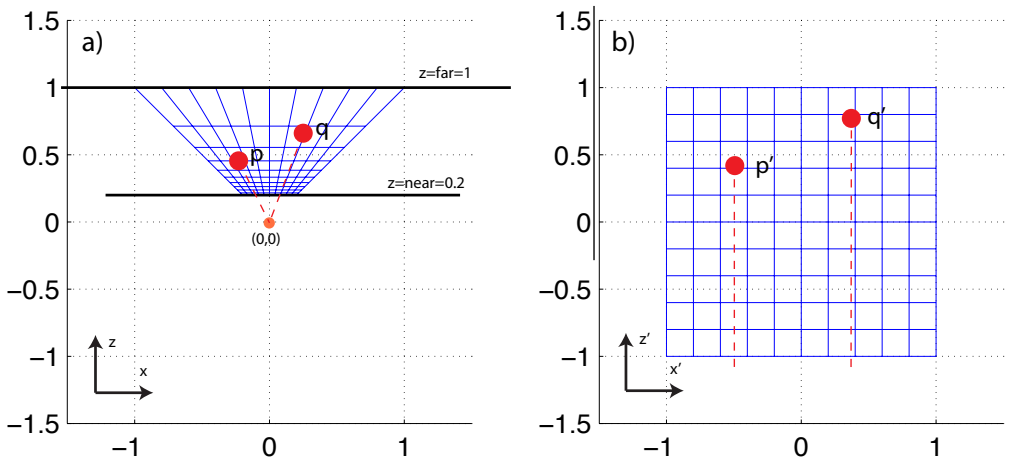

Figure 4: **a)** The view frustum before projection and **b)** after projection and homogenization to $w' = 1$. Screen coordinates for the points p and q are obtained from transformed and homogenized points p' and q' by simply dropping the z' coordinate. The near and far planes have been chosen to maintain equal scales on both sides of the figure, but they can, of course, be arbitrary.

One neat property of the new projection is that *it is invertible*. This means that you can take any point (x', z', w') in the canonical view volume, multiply it by the inverse of the projection matrix from Eq. (2), homogenize the result, and (ha!) you've got the original "un-projected" point $(x, z, 1)$ in the view-volume. Thanks to projective equivalence, this gives the same result as multiplying the point $(x'/w', z'/w', 1)$ by the inverse and then homogenizing again.

The above discussion generalizes trivially to the 3D xyz case, although visualization becomes much harder for obvious reasons because the homogeneous space is 4D. One merely adds the y (resp. y') coordinate and treats it similarly to x (resp. x') above. The post-transform $x'y'z'w'$ space, where the view frustum is a cube (after homogenization to $w' = 1$), is often called *clip space*.