CAMERA SPACE TO CLIP SPACE PERSPECTIVE TRANSFORM

In world space, we assume our camera frame to be at some position in the world and looking at some direction ($\hat{c}$). $\hat{c}$ forms the -Z axis of the camera frame and we move this camera frame to the origin and rotate the frame to align with standard XYZ basis vectors.

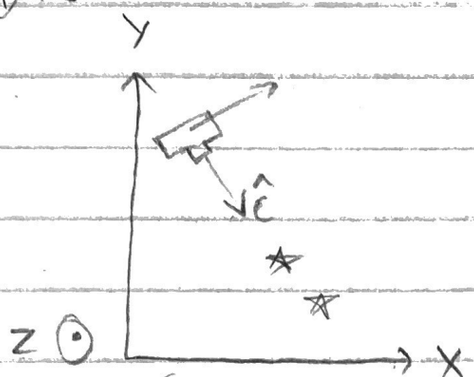


fig- world space.

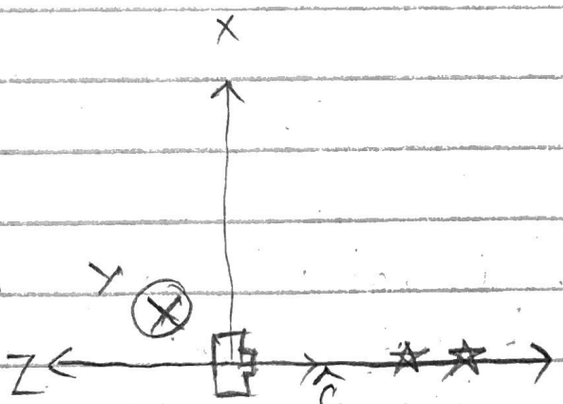


Fig - camera space

Here we assume that we are already in camera space. For perspective transform, we can construct an imaginery line emanating from every point in this camera space, converging at origin. Also, lets place a 2x2 plane in front of the camera at "N" distance or $\langle 0, 0, n \rangle$

Coordinate parallel to XY plane. Those imaginary lines coming out of points in Camera space, would intersect with 2x2 plane and that would result in perspective projection of the points on to the plane.

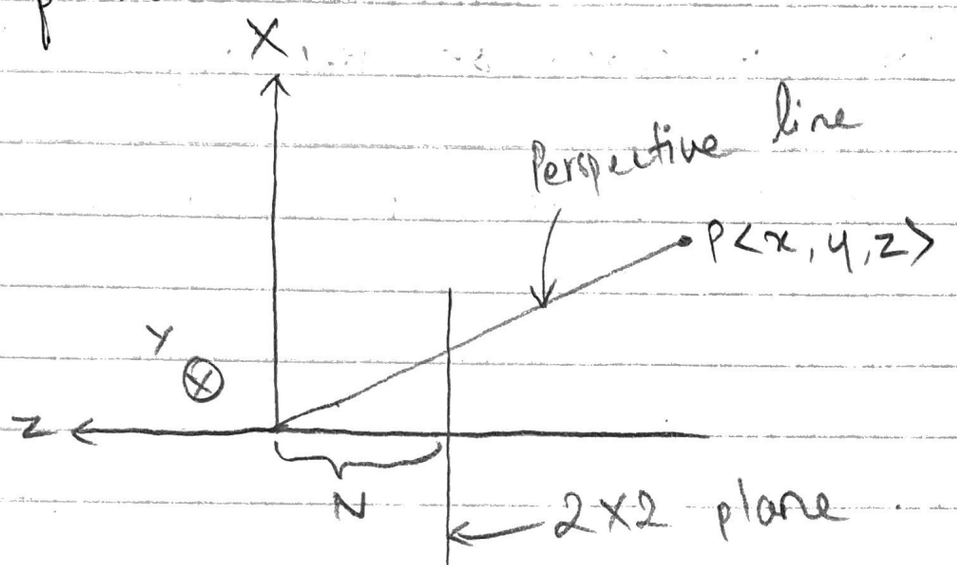


Fig: Perspective projection of point P onto 2x2 plane.

Moreover, we can also imagine cases where perspective lines would not intersect with 2x2 plane. Infact which perspective lines would intersect with plane is given by a pyramid whose tip start at the origin and expands along -Z axis. This is called "viewing pyramid".

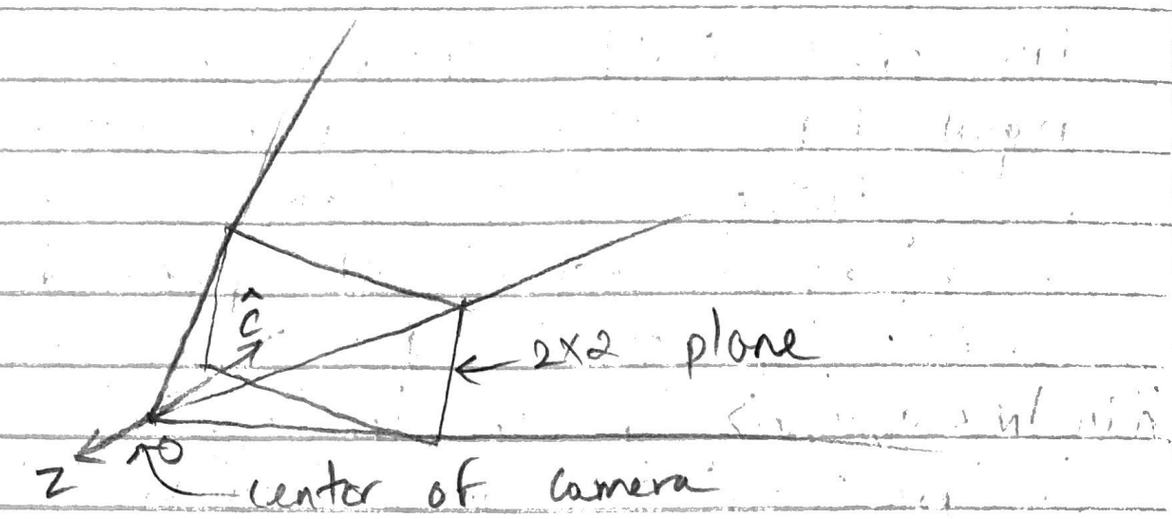
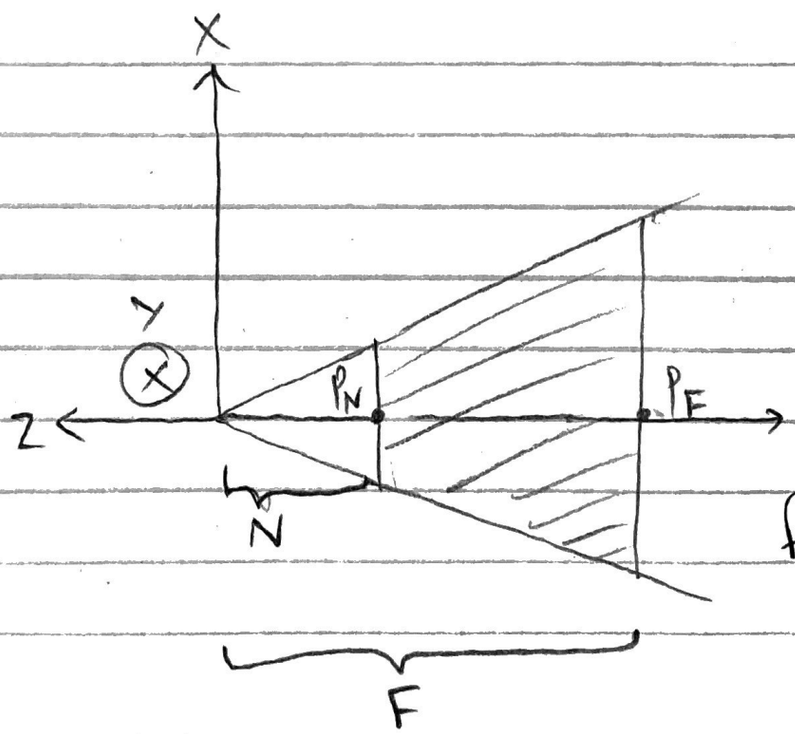


Fig: Viewing pyramid.

Any point inside viewing pyramid in front of 2x2 projection plane would intersect with plane and thus form projection. Also, lets bound this pyramid at the far end, so that anything beyond would be considered out of view and therefore not form final image, although points there may still intersect with projection plane. So, our viewing pyramid would look like so:



$P_N = \langle 0, 0, n \rangle$
$P_F = \langle 0, 0, f \rangle$

fig: Viewing pyramid XZ plane.

Any points inside shaded region in the figure above would appear on our final image. That shaded region which resembles a chopped off pyramid is also called "frustum". Also note in the image "n" in $P_N\langle 0, 0, n\rangle$ and "f" in $P_F\langle 0, 0, f\rangle$ are -ve. However, "N" and "F" are positive.

2X2 projection plane at N distance from origin is also called "Near Plane" and the plane at F distance from origin is also called "Far Plane".

Now imagine a point inside frustum. The projection of X component of this point would look as follows:

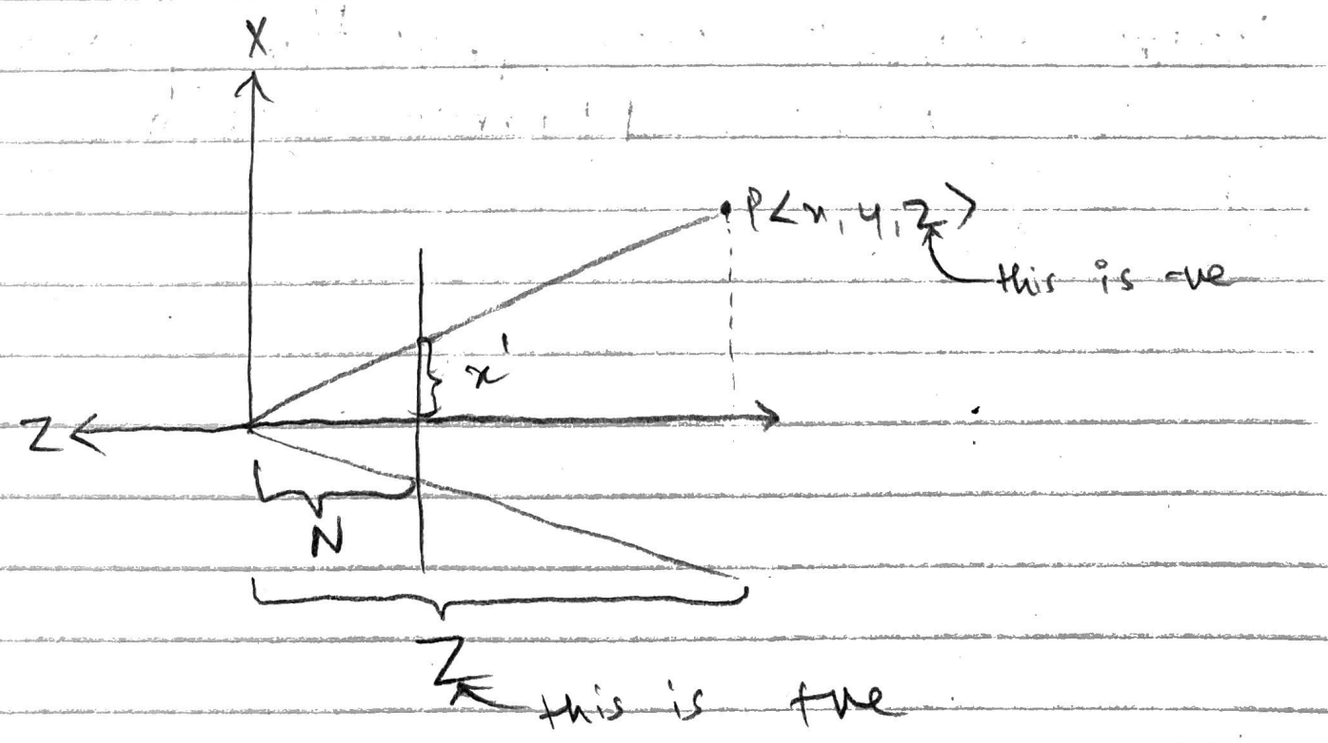


fig: Projection of X component

$$\frac{x'}{N} = \frac{x}{Z} \leftarrow \text{this is +ve}$$

$$x' = x\left(\frac{N}{Z}\right) \quad \text{— (1)}$$

Similarly,

$$y' = y\left(\frac{N}{Z}\right) \quad \text{— (2)}$$

Encoding this division by Z as the 4th coordinate of the output vector, so that our transformation would look as such:

$$T \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ k \\ -z \end{bmatrix} \quad \text{— (3)}$$

Notice how the "z" has -ve in eqn 3, which makes it a +ve number. The "z" in eqn 3 is always -ve for the points lying in the region where camera is facing. "z" in eqn 1 and 2 were absolute distance from the origin so these two different "z"s can be confusing.

Thus in eqn 3, by sticking "-" in front of Z, we achieve division by the Z. Notice that I have placed "k" in the Z coordinate of output vector in eqn 3. Say if we had just put Z there, then by virtue of that -Z as 4th coordinate that would mean every output vector would have -1 as Z coordinate. Turns out we need this Z coordinate information to determine which image sits on top of another.

You may argue why not just leave that as Z and in our program have a special condition that takes into consideration that Z before doing the division. That would certainly simplify things theoretically, but my guess is that having this special condition makes hardware complex and / or makes the graphics pipeline slow.

Therefore there is this wonderful hack, that graphics guys have come up with that preserves the relative order of Z of the points being transformed, although the values may be distorted. Here is that derivation of transform "T":...

$$\begin{bmatrix} N & 0 & 0 & 0 \\ 0 & N & 0 & 0 \\ 0 & 0 & a & b \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} Nx \\ Ny \\ az+b \\ -z \end{bmatrix}$$

When z is "n"

$$\frac{an+b}{-n} = n$$

$$an + b = -n^2$$

$$a = \frac{-n^2 - b}{n} \quad \text{---(4)}$$

When z is "f"

$$\frac{af+b}{-f} = f$$

$$af + b = -f^2$$

$$\left(\frac{-n^2-b}{n}\right)f + b = -f^2 \quad [\text{From (4)}]$$

$$(-n^2 - b)f + bn = -nf^2$$

$$-n^2 f - bf + bn = -nf^2$$

$$b(n-f) = n^2 f - nf^2$$

$$b = \frac{nf(n-f)}{(n-f)}$$

$$b = nf$$

$$b = -N \times -F \quad [\because n = -N \text{ and } f = -F]$$

$$b = NF \quad \text{---(5)}$$

So,

$$a = \frac{-n^2 - NF}{n} \quad [\text{From (4) and (5)}]$$

$$a = \frac{-(n^2 + nf)}{n}$$

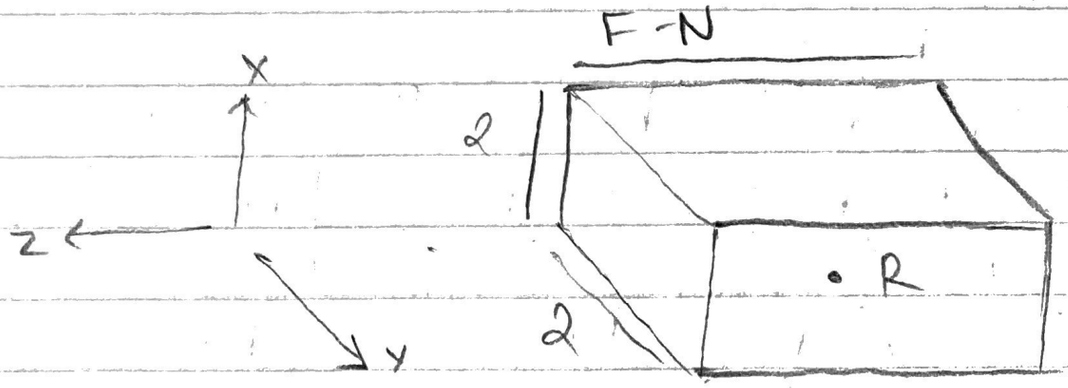
$$a = \frac{N^2 + NF}{N}$$

$$a = N + F$$

Thus, our transformation becomes,

$$\begin{bmatrix} N & 0 & 0 & 0 \\ 0 & N & 0 & 0 \\ 0 & 0 & N+F & NF \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} Nx \\ Ny \\ (N+F)z + NF \\ -z \end{bmatrix}$$

This transformation transforms our frustum to $2 \times 2 \times (F-N)$ rectangular prism.



and the center R is located at $\langle 0, 0, \frac{n+f}{2} \rangle$. So, to transform this box to clips pace which is $2 \times 2 \times 2$ cube, with center at origin, we ought to translate it by $\langle 0, 0, -(n+f)/2 \rangle$ i.e $\langle 0, 0, (N+F)/2 \rangle$ and then scale

the Z by $2/F-N$, which is:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & (2/F-N) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & (N+F)/2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2/F-N & (N+F)/(N-F) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

So, the final camera space to clip space transformation involves, first transforming to rectangular prism then applying the translate / scale operation. This is given as:

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2/(F-N) & (F+N)/(F-N) \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} N & 0 & 0 & 0 \\ 0 & N & 0 & 0 \\ 0 & 0 & N+F & NF \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

$$T = \begin{bmatrix} N & 0 & 0 & 0 \\ 0 & N & 0 & 0 \\ 0 & 0 & c & d \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

where, $c = 2(N+F)/(F-N) - (F+N)/(F-N) = (F+N)/F-N$

and, $d = 2NF/(F-N)$

Thus, $T = \begin{bmatrix} N & 0 & 0 & 0 \\ 0 & N & 0 & 0 \\ 0 & 0 & (F+N)/(F-N) & 2NF/(F-N) \\ 0 & 0 & -1 & 0 \end{bmatrix}$ is the final camera to clip space perspective trans form.