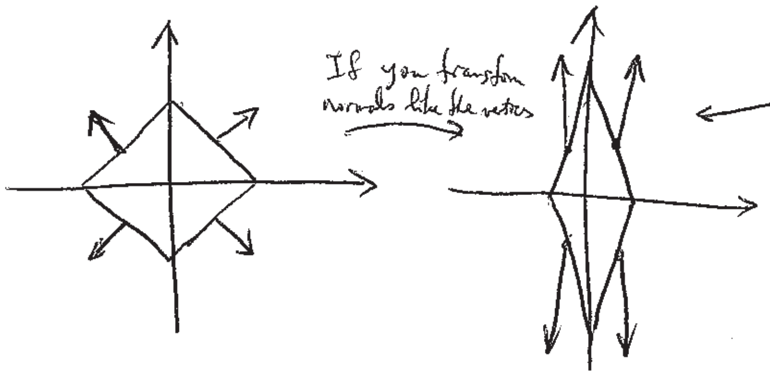
Transforming surface Normals



← the normals are no longer ⊥ to the surface! This is no good!

you want $n^T v = 0$ (n: normal; v: some vector on surface)

$(M_2 n)^T (M_1 v) = 0$ (applying M_1 to v)

we want to apply some M_2 to n s.t. that they are still ⊥!

$$(M_2 n)^T (M_1 v) = 0$$

$$n^T M_2^T M_1 v = 0$$

we want this ($M_2^T M_1$) to be identity I so that $n^T v = 0$

so $$M_2^T M_1 = I$$

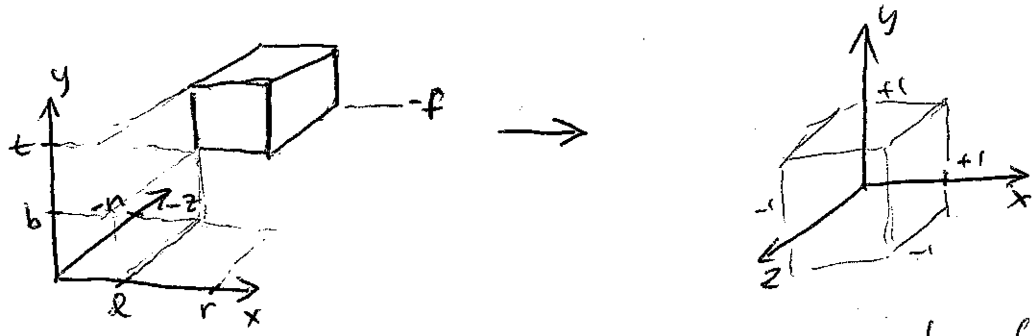
$$M_2^T = M_1^{-1}$$

$$M_2 = (M_1^{-1})^T$$

↑ This is the inverse transpose of the modelview matrix

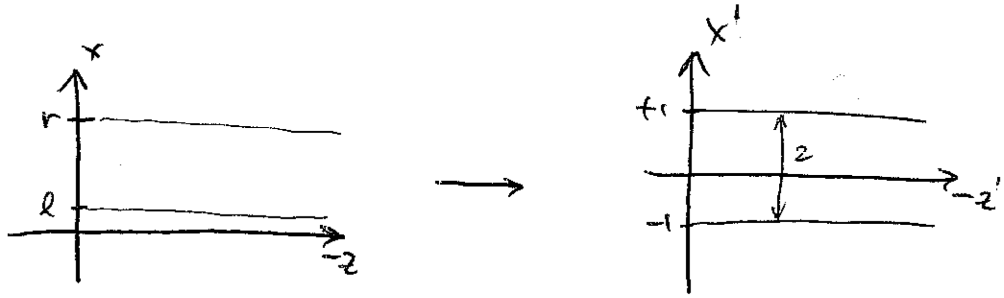
ORTHOGRAPHIC PROJECTION

Task: Take a rectangular volume in eye space defined by (l, r, t, b, n, f) and project it into a cube with sides ± 1.



We need to construct a Matrix M_{ortho} that does this transformation.

First transform the x-coordinates



$$l < x < r \rightarrow -1 < x' < 1$$

distance: $r - l$ $\qquad$ $x' = f(x)$

midpt: $\frac{r+l}{2}$

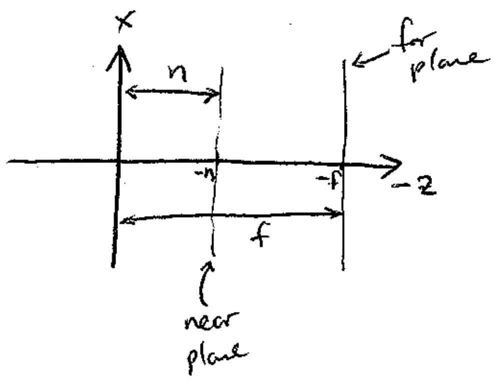
1. first shift x by midpt. This ensures that we are even about 0.
2. Now scale to make length 2

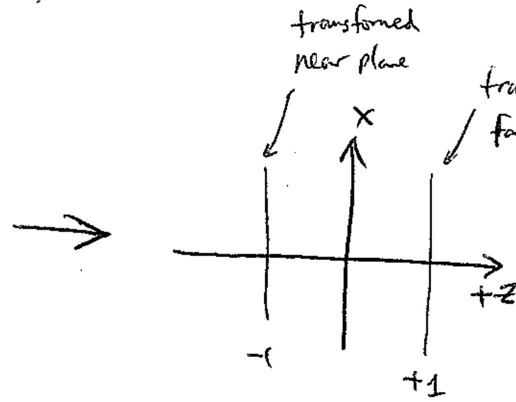
$$x' = \left(\frac{2}{r-l}\right)\left(x - \left(\frac{r+l}{2}\right)\right) = \boxed{\frac{2}{r-l}x - \frac{r+l}{r-l}}$$

Do the same for y:

$$\boxed{y' = \frac{2}{t-b}y - \frac{t+b}{t-b}}$$

Now compute z-coordinate:





note how now the transformed far plane > near plane. So things farther away have bigger z.

distance: $f-n$

midpt: $\frac{-n-f}{2}$

this "−" is to flip the planes so that the far plane is bigger z.

$$z' = \left(\frac{-2}{f-n}\right)\left(z + \frac{n+f}{2}\right)$$

$$\boxed{z' = -\frac{2}{f-n} z - \frac{f+n}{f-n}}$$

Now we can write the orthogonal projection matrix

$$M_{ortho} = \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & -\frac{2}{f-n} & -\frac{f+n}{f-n} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

← same as OpenGL spec