A viewport transformation, or 'windowing transformation' maps a unity rectangle to the desired window/viewport coordinates.

Viewport Transformation from the Canonical Volume

The viewport transformation assumes that a geometry is within the canonical view volume.

If we are transforming into a viewport with n_x horizontal pixels and n_y vertical pixels, then we can squeeze the view volume into the correct planar dimensions with the following transformation:

$$\begin{pmatrix} x_{screen} \\ y_{screen} \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{n_x}{2} & & & \frac{n_x-1}{2} \\ & \frac{n_y}{2} & & \frac{n_y-1}{2} \\ & & 1 & \\ & & & 1 \end{pmatrix} \begin{pmatrix} x_{cannonical} \\ y_{cannonical} \\ 1 \end{pmatrix}$$

Thus we define M_vp as:

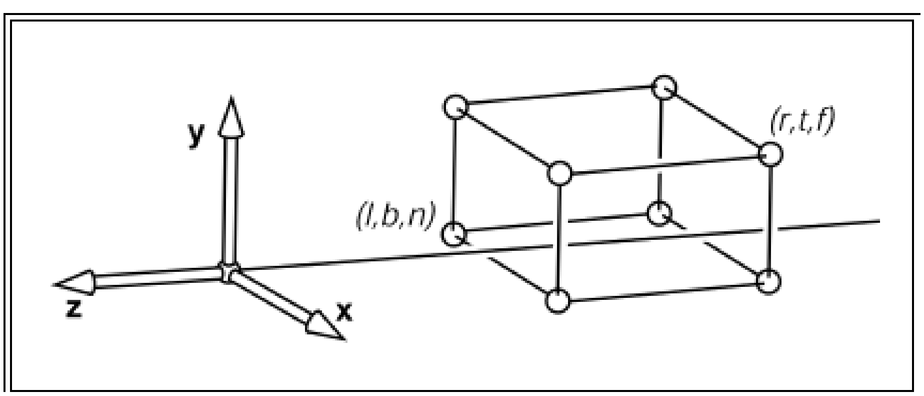
$$M_{vp} = \begin{pmatrix} \frac{n_x}{2} & & & \frac{n_x-1}{2} \\ & \frac{n_y}{2} & & \frac{n_y-1}{2} \\ & & 1 & \\ & & & 1 \end{pmatrix}$$

This matrix uses $\frac{n_y}{2}$ because the viewcube goes from -1 to 1-- to get a n_y length viewing port the top and bottom half must be n_y length each.

This matrix also translates each point into the positive quadrant. *why?*

Viewport Transformation from any cube

In order to generalize this process to any arbitrary view-volume, we build a matrix to transform from arbitrary cube to the canonical volume



$$\begin{pmatrix} \frac{2}{r-l} & & & -\frac{r+l}{r-l} \\ & \frac{2}{t-b} & & -\frac{t+b}{t-b} \\ & & \frac{2}{n-f} & -\frac{n+f}{n-f} \\ & & & 1 \end{pmatrix}$$

This matrix performs two functions:

1. It scales all points s.t. they fit within the canonical view volume
2. it translates all points by $-\frac{2dist_{arb}+len_{arb}}{len_{arb}}$ Where $dist_{arb}$ is the distance of the arbitrary plane from the origin *why*?