Cameras & Projection

The textbook discussion is different from mine. To simplify presentation, mostly look at 2D.

\[ \bar{e} = \begin{pmatrix} \bar{y} \\ \bar{z} \end{pmatrix} \]

\[ \bar{p} = \begin{pmatrix} y_p \\ z_p \end{pmatrix} = ? \]

\[ y_p = -\frac{y}{z} \] by design, because \[ \frac{y}{z} = -\frac{1}{2} \]

This is a non-linear transform! (similar triangles)

Good news: homogeneous coordinates can handle this.

\[ \tilde{e} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \]

8 Orthographic Projection

- simpler than the perspective/central projection
- we'll convert central projection into this.
- useful in applications in Architecture and Engineering
What does the projection matrix look like?

\[
\begin{pmatrix}
\frac{1}{2} \\
\frac{1}{2} \\
0
\end{pmatrix}
\rightarrow
\begin{pmatrix}
\frac{1}{2} \\
0 \\
0
\end{pmatrix}
\]

\[ N = \begin{pmatrix}
0 \\
0 \\
1
\end{pmatrix} \]

To view the "View frustum" (box in this case), we want to transform it eventually to fit on the screen. So we transform the box into "Normalized Device Coordinates".

Step 1: Translate the "View Box" to origin

\[
\begin{pmatrix}
t + \frac{b}{2} \\
\frac{f}{2} + \frac{h}{2} \\
0
\end{pmatrix}
\]

Move h origin using

\[
T = \begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
-\frac{b}{2} & 0 & 1
\end{pmatrix}
\]
Step 1: Translate the "View Box" to origin

\[
\begin{pmatrix}
\frac{t+b}{2} \\
\frac{m+f}{2}
\end{pmatrix}
\]

Move base origin using

\[
T = \begin{bmatrix}
1 & 0 & -\frac{t+b}{2} \\
0 & 1 & -\frac{m+f}{2} \\
0 & 0 & 1
\end{bmatrix}
\]

Step 2: Scale to make the half-size = 1

E.g., along y box is \( t-b \) → 1

\[
S = \begin{bmatrix}
\frac{2}{t-b} & 0 & 0 \\
0 & \frac{2}{m-f} & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

Projection matrix

\[
P_0 = S^T = \begin{bmatrix}
\frac{2}{t-b} & \frac{t+b}{t-b} & 0 \\
\frac{2}{m-f} & \frac{m+f}{m-f} & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

Exercise: do this for the x-coordinate
Perspective

Projective Transformation
("unhinging" transform)

Depth information is retained, even though distorted