Projective Transforms

Geoffrey Matthews

Department of Computer Science
Western Washington University

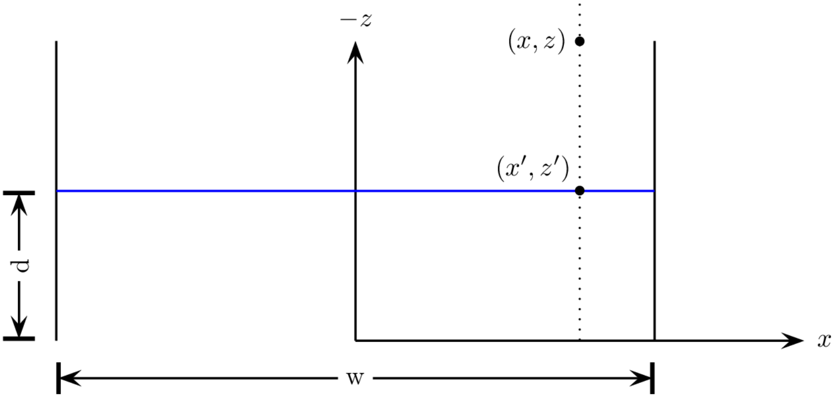
Fall 2011

Online Resources

Readings

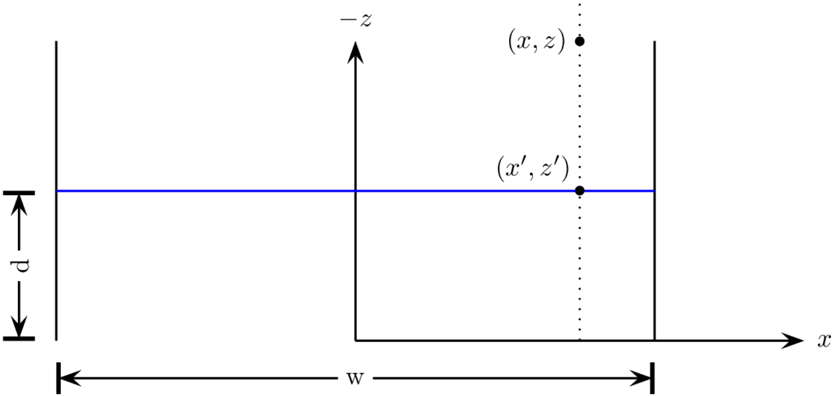
- http://www.songho.ca/opengl/index.html
- http://en.wikipedia.org/wiki/Transformation_matrix
- http://glasnost.itcarlow.ie/~powerk/GeneralGraphicsNotes/projection/projection_viewing.html

Simple Orthogonal Projection



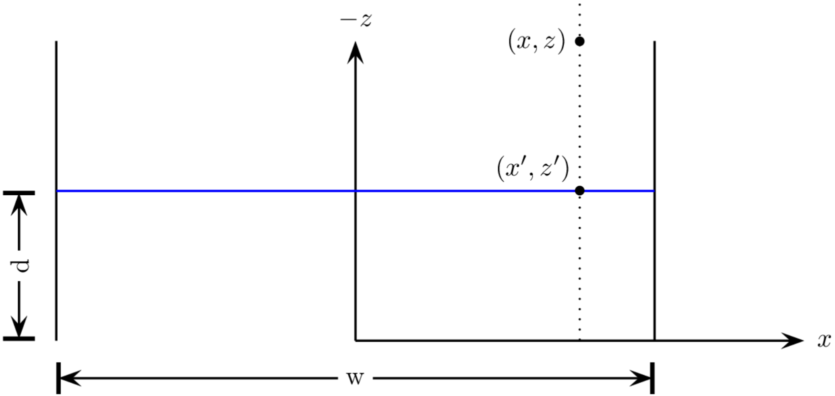
- $z' =$

Simple Orthogonal Projection



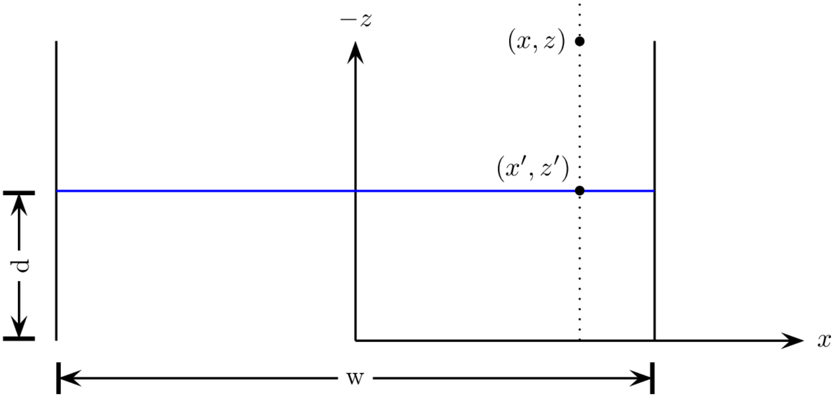
- $z' = -d$
- $x' =$

Simple Orthogonal Projection



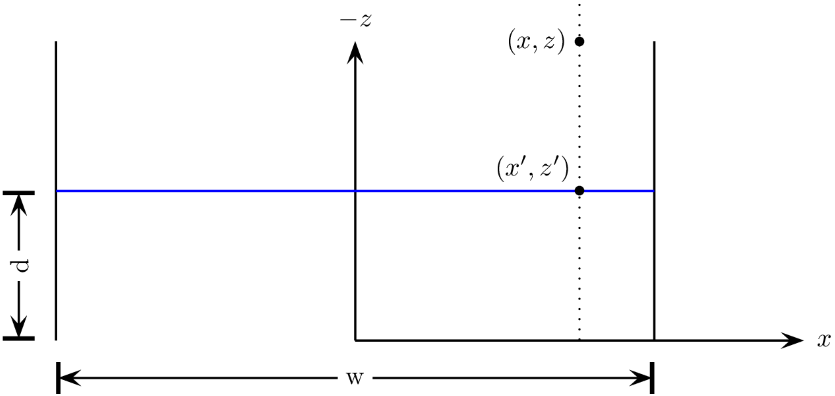
- $z' = -d$
- $x' = x$

Simple Orthogonal Projection



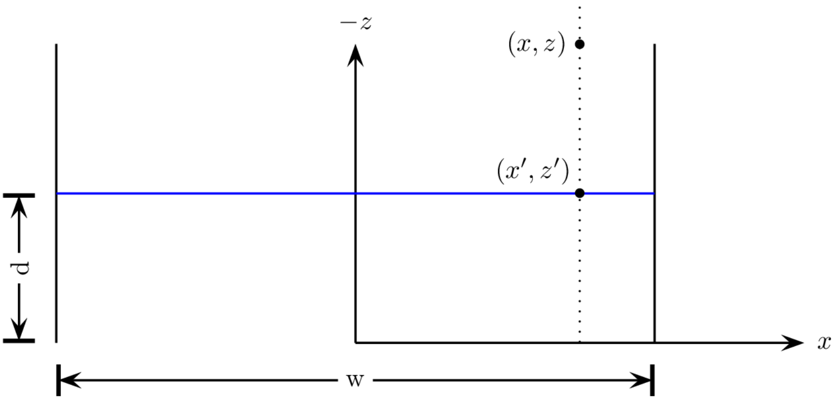
- $z' = -d$
- $x' = x$
- Scale x' to ± 1? $x' =$

Simple Orthogonal Projection



- $z' = -d$
- $x' = x$
- Scale x' to ± 1? $x' = \frac{2x}{w}$

Simple Orthogonal Projection



- $z' = -d$
- $x' = x$
- Scale x' to ± 1? $x' = \frac{2x}{w}$
- What about y?

Simple Orthogonal Projection

$$
\begin{aligned}
x' &= \frac{2}{w}x \\
y' &= \frac{2}{h}y \\
z' &= -d
\end{aligned}
$$

$$
\begin{bmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \\ 0 & 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}
=
\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}
$$

- First two rows are easy.

Simple Orthogonal Projection

$$
\begin{aligned}
x' &= \frac{2}{w}x \\
y' &= \frac{2}{h}y \\
z' &= -d
\end{aligned}
$$

$$
\begin{bmatrix} \frac{2}{w} & 0 & 0 & 0 \\ 0 & \frac{2}{h} & 0 & 0 \\ ? & ? & ? & ? \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}
$$

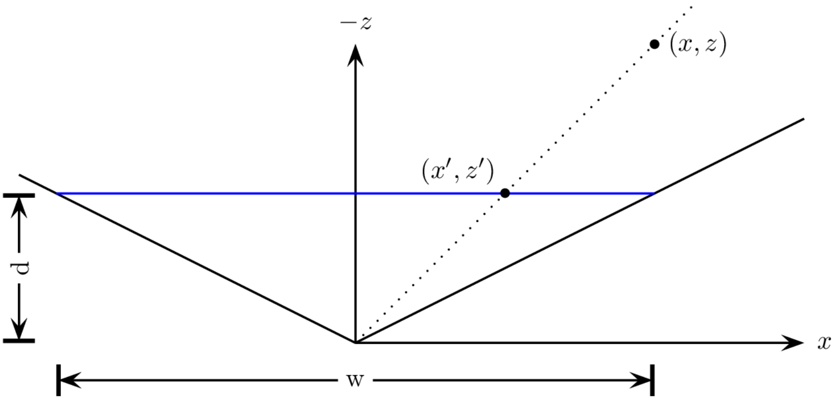
- How can we get z'?

Simple Orthogonal Projection

$$\begin{aligned} x' &= \frac{2}{w}x \\ y' &= \frac{2}{h}y \\ z' &= -d \end{aligned}$$

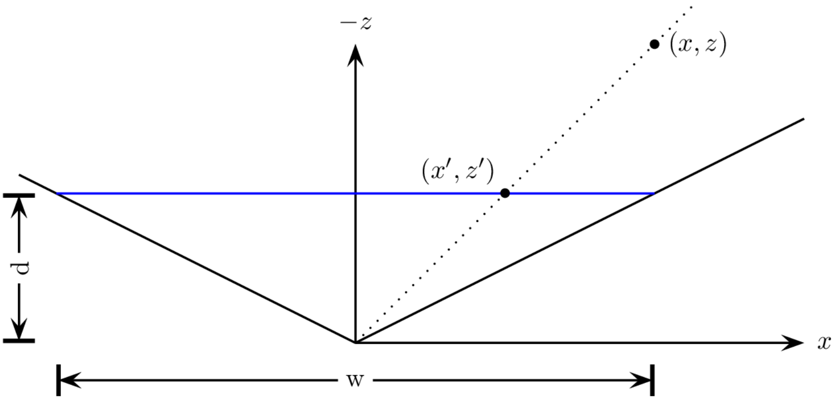
$$\begin{bmatrix} \frac{2}{w} & 0 & 0 & 0 \\ 0 & \frac{2}{h} & 0 & 0 \\ 0 & 0 & 0 & -d \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}$$

Simple Perspective Projection



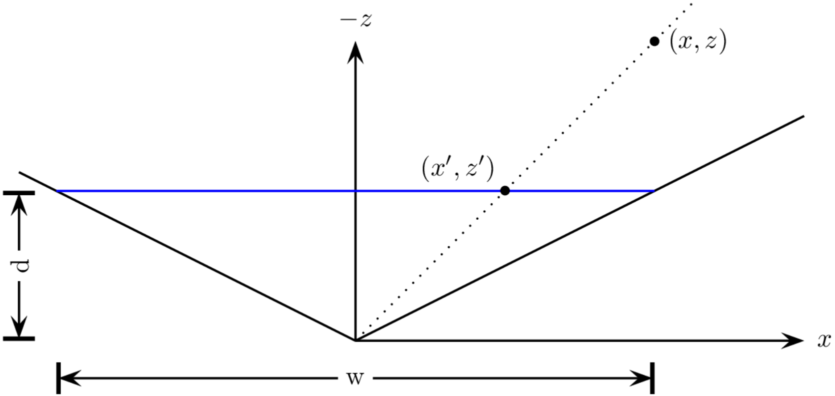
- $z' =$

Simple Perspective Projection



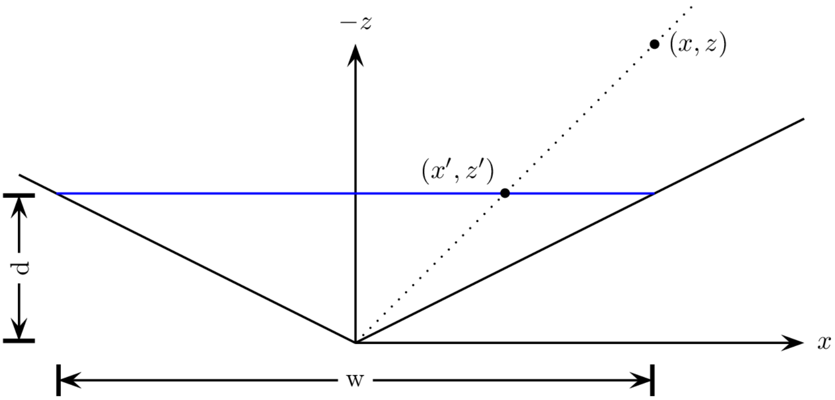
- $z' = -d$
- $x' =$

Simple Perspective Projection



- $z' = -d$
- $x' = \frac{-xd}{z}$ because $\frac{x}{z} = \frac{x'}{z'}$
- Scale x' to ± 1? $x' =$

Simple Perspective Projection



- $z' = -d$
- $x' = \frac{-xd}{z}$ because $\frac{x}{z} = \frac{x'}{z'}$
- Scale x' to ± 1? $x' = \frac{-2xd}{zw}$

Simple Perspective Projection

$$
\begin{aligned}
x' &= \frac{-2d}{zw} x \\
y' &= \frac{-2d}{zh} y \\
z' &= -d
\end{aligned}
$$

$$
\begin{bmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \\ 0 & 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}
=
\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}
$$

- What to do?

Simple Perspective Projection

$$
\begin{aligned}
x' &= \frac{-2d}{zw}x \\
y' &= \frac{-2d}{zh}y \\
z' &= -d
\end{aligned}
$$

$$
\begin{bmatrix} \frac{-2d}{w} & 0 & 0 & 0 \\ 0 & \frac{-2d}{h} & 0 & 0 \\ 0 & 0 & 0 & -d \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{-2xd}{w} \\ \frac{-2yd}{h} \\ -d \\ 1 \end{bmatrix}
$$

- Not quite there, we need to divide x' and y' by z.
- How do we do that?
- And *not* divide z' by z?

Simple Perspective Projection

$$
\begin{aligned}
x' &= \frac{-2d}{zw}x \\
y' &= \frac{-2d}{zh}y \\
z' &= -d
\end{aligned}
$$

$$
\begin{bmatrix} \frac{-2d}{w} & 0 & 0 & 0 \\ 0 & \frac{-2d}{h} & 0 & 0 \\ 0 & 0 & -d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}
=
\begin{bmatrix} \frac{-2xd}{w} \\ \frac{-2yd}{h} \\ -dz \\ 1 \end{bmatrix}
$$

- Now we have to divide x', y' and z' by z
- How can we divide by z?

Simple Perspective Projection

$$
\begin{aligned}
x' &= \frac{-2d}{zw}x \\
y' &= \frac{-2d}{zh}y \\
z' &= -d
\end{aligned}
$$

$$
\begin{bmatrix} \frac{-2d}{w} & 0 & 0 & 0 \\ 0 & \frac{-2d}{h} & 0 & 0 \\ 0 & 0 & -d & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{-2xd}{w} \\ \frac{-2yd}{h} \\ -dz \\ z \end{bmatrix}
$$

- What point does this unhomogenized point represent?

Simple Perspective Projection

$$
\begin{aligned}
x' &= \frac{-2d}{zw}x \\
y' &= \frac{-2d}{zh}y \\
z' &= -d
\end{aligned}
$$

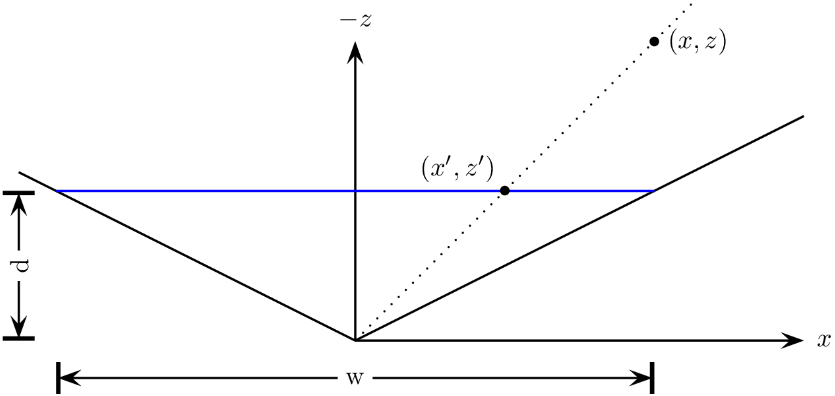
$$
\begin{bmatrix} \frac{-2d}{w} & 0 & 0 & 0 \\ 0 & \frac{-2d}{h} & 0 & 0 \\ 0 & 0 & -d & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}
\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}
= \begin{bmatrix} \frac{-2xd}{w} \\ \frac{-2yd}{h} \\ -dz \\ z \end{bmatrix}
= \begin{bmatrix} \frac{-2xd}{wz} \\ \frac{-2yd}{hz} \\ -d \\ 1 \end{bmatrix}
$$

Simple Perspective Projection

$$
\begin{aligned}
x' &= \frac{-2d}{zw} x \\
y' &= \frac{-2d}{zh} y \\
z' &= -d
\end{aligned}
$$

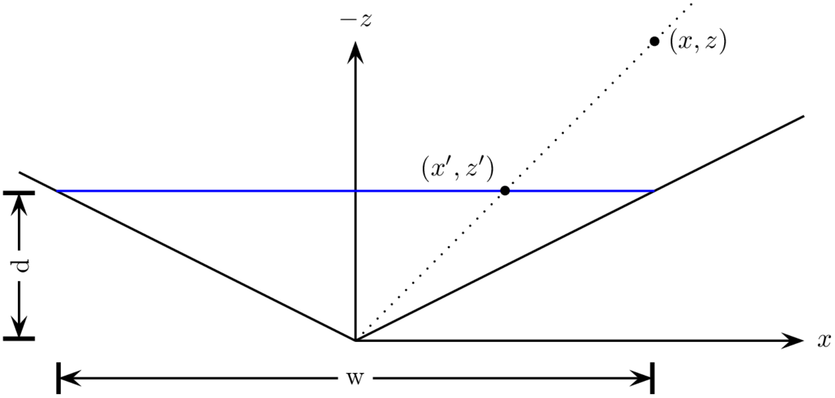
$$
\begin{bmatrix} \frac{2d}{w} & 0 & 0 & 0 \\ 0 & \frac{2d}{h} & 0 & 0 \\ 0 & 0 & d & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix}
\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}
= \begin{bmatrix} \frac{2xd}{w} \\ \frac{2yd}{h} \\ dz \\ -z \end{bmatrix}
= \begin{bmatrix} \frac{-2xd}{wz} \\ \frac{-2yd}{hz} \\ -d \\ 1 \end{bmatrix}
$$

Problems projecting onto $z = -d$ plane.



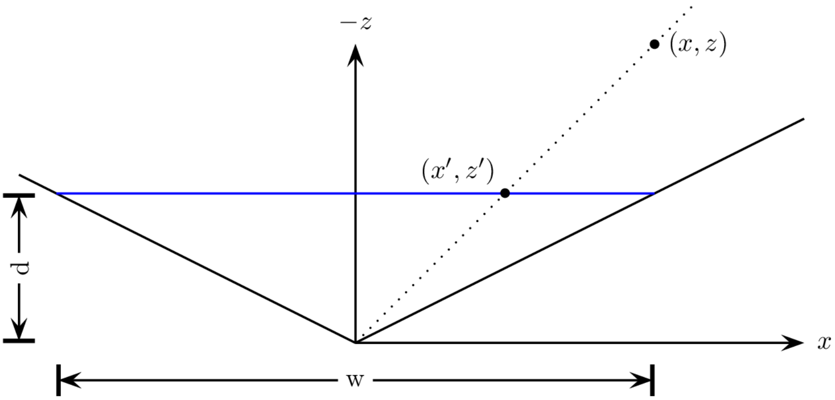
- If two items are on the same projection line, which one colors the image?
- We need to retain depth (z) information to decide.

Problems projecting onto $z = -d$ plane.



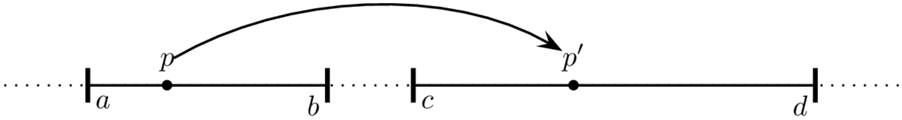
- **Painter's algorithm:** sort all objects and render them back to front. This is an $O(n \log n)$ algorithm.

Problems projecting onto $z = -d$ plane.



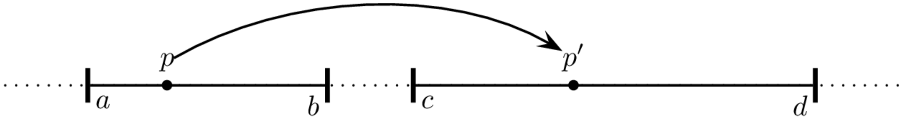
- **Z-buffer algorithm:** maintain a separate buffer where each item writes its depth. Only if an item's depth is smaller than the depth already in the depth buffer does it write to the color buffer. This is an $O(n)$ algorithm.

Linear mapping between frames, one dimension



$$\begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix} = \begin{bmatrix} p' \\ 1 \end{bmatrix}$$

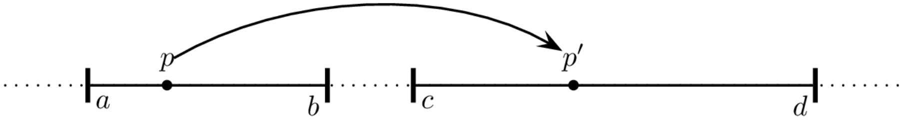
Linear mapping between frames, one dimension



$$\begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix} = \begin{bmatrix} p' \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} ? & ? \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix} = \begin{bmatrix} p' \\ 1 \end{bmatrix}$$

Linear mapping between frames, one dimension

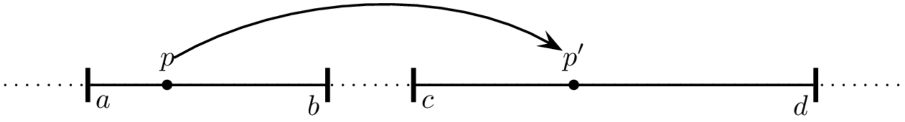


$$\begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix} = \begin{bmatrix} p' \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} ? & ? \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix} = \begin{bmatrix} p' \\ 1 \end{bmatrix}$$

$$c + \left(\frac{p-a}{b-a}\right)(d-c) = p'$$

Linear mapping between frames, one dimension



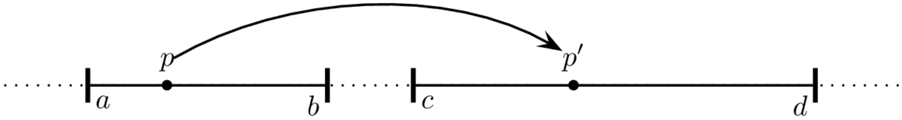
$$\begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix} = \begin{bmatrix} p' \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} ? & ? \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix} = \begin{bmatrix} p' \\ 1 \end{bmatrix}$$

$$c + \left(\frac{p-a}{b-a}\right)(d-c) = p'$$

$$\frac{d-c}{b-a}p + \frac{bc-ad}{b-a} = p'$$

Linear mapping between frames, one dimension



$$\begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix} = \begin{bmatrix} p' \\ 1 \end{bmatrix}$$

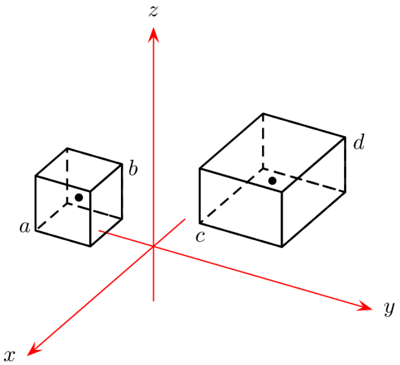
$$\begin{bmatrix} ? & ? \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix} = \begin{bmatrix} p' \\ 1 \end{bmatrix}$$

$$c + \left(\frac{p-a}{b-a}\right)(d-c) = p'$$

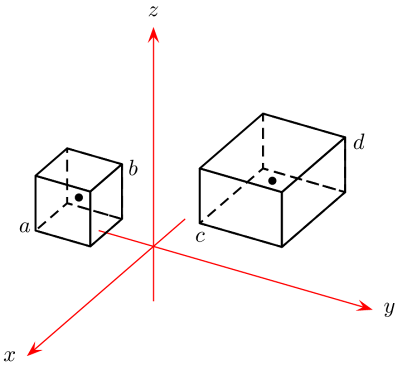
$$\frac{d-c}{b-a} p + \frac{bc-ad}{b-a} = p'$$

$$\begin{bmatrix} \frac{d-c}{b-a} & \frac{bc-ad}{b-a} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix} = \begin{bmatrix} p' \\ 1 \end{bmatrix}$$

Linear mapping between frames, three dimensions



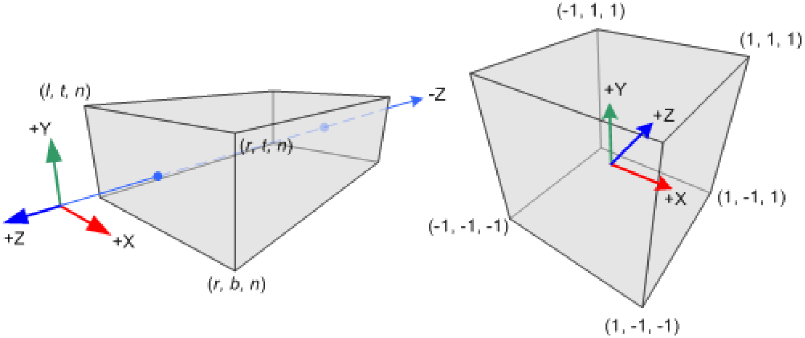
Linear mapping between frames, three dimensions



$$c + \left(\frac{p-a}{b-a}\right)(d-c) = p'$$

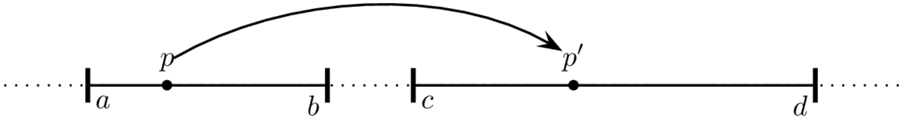
$$\frac{d-c}{b-a}p + \frac{bc-ad}{b-a} = p'$$

Orthographic Projection in OpenGL



- Map an arbitrary axially aligned bounding box (AABB) to normalized device coordinates (NDC), the ± 1 cube.
- x: $r \Rightarrow 1 \qquad l \Rightarrow -1$
- y: $t \Rightarrow 1 \qquad b \Rightarrow -1$
- z: $f \Rightarrow 1 \qquad n \Rightarrow -1$

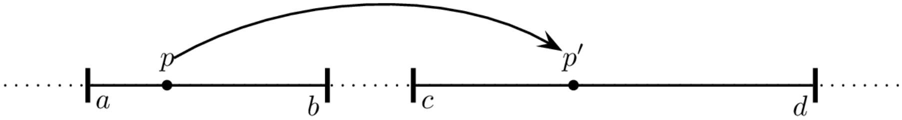
Orthographic Projection in OpenGL



- Map x between l and r to between -1 and $+1$

$$x' = \frac{d-c}{b-a}x + \frac{bc-ad}{b-a}$$

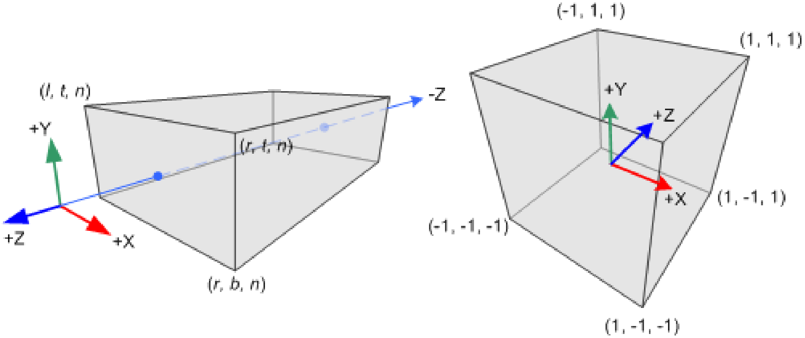
Orthographic Projection in OpenGL



- Map x between l and r to between -1 and $+1$

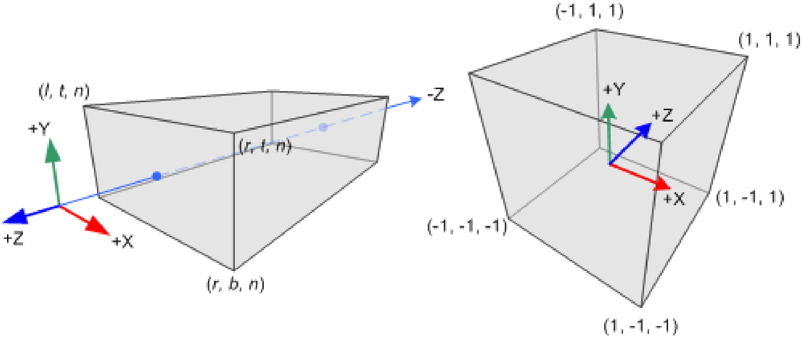
$$
\begin{aligned}
x' &= \frac{d-c}{b-a}x + \frac{bc-ad}{b-a} \\
&= \frac{1-(-1)}{r-l}x + \frac{-r-l}{r-l} \\
&= \frac{2}{r-1}x - \frac{r+l}{r-l} \\
y' &= \frac{2}{t-b}y - \frac{t+b}{t-b} \\
z' &= \frac{-2}{f-n}z - \frac{f+n}{f-n}
\end{aligned}
$$

Orthographic Projection in OpenGL



$$\begin{bmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & \frac{-2}{f-n} & -\frac{f+n}{f-n} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}$$

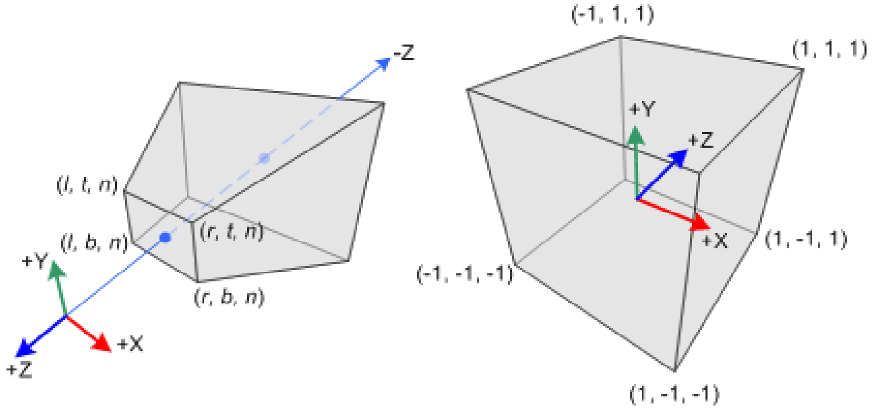
Orthographic Projection in OpenGL



- If symmetric around z-axis, $r = -l$ and $t = -b$, then

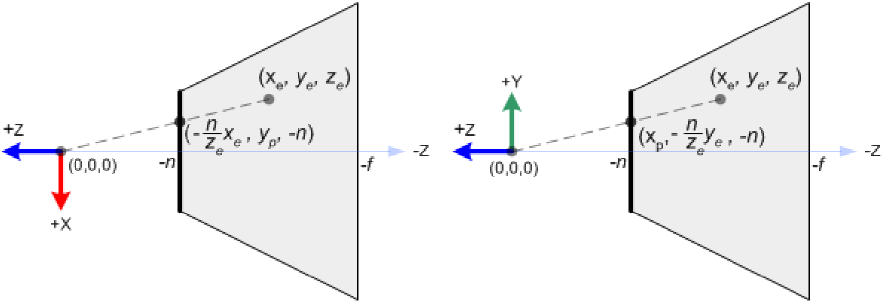
$$\begin{bmatrix} \frac{1}{r} & 0 & 0 & 0 \\ 0 & \frac{1}{t} & 0 & 0 \\ 0 & 0 & \frac{-2}{f-n} & -\frac{f+n}{f-n} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}$$

Projective Transforms in OpenGL



- Map an arbitrary view frustrum to normalized device coordinates (NDC), the ± 1 cube.

Projective Transforms in OpenGL



- Find the x and y coordinates using similar triangles.

$$
\begin{aligned}
x' &= -\frac{n}{z}x \\
y' &= -\frac{n}{z}y
\end{aligned}
$$

Projective Transforms in OpenGL

- It looks like dividing by $-z$ is going to be a good idea again, so let's fill in the bottom row of our matrix.

$$\begin{aligned} x' &= -\frac{n}{z}x \\ y' &= -\frac{n}{z}y \end{aligned}$$

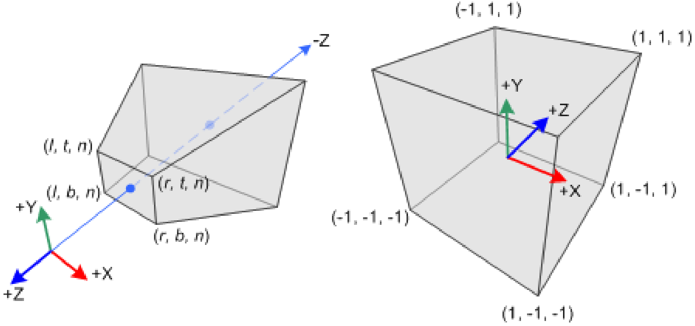
Projective Transforms in OpenGL

- It looks like dividing by $-z$ is going to be a good idea again, so let's fill in the bottom row of our matrix.

$$
\begin{aligned}
x' &= -\frac{n}{z}x \\
y' &= -\frac{n}{z}y
\end{aligned}
$$

$$
\begin{bmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \\ 0 & 0 & -1 & 0 \end{bmatrix}
\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}
=
\begin{bmatrix} x_u \\ y_u \\ z_u \\ -z \end{bmatrix}
=
\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}
$$

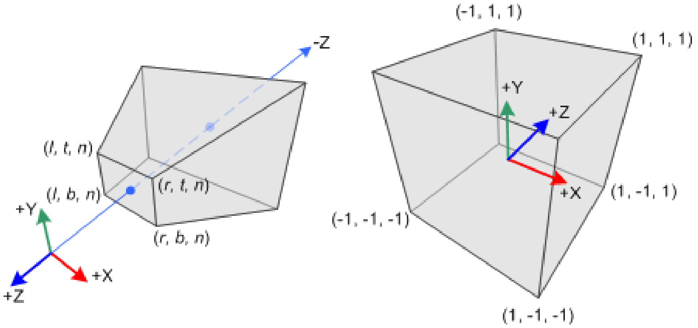
Projective Transforms in OpenGL



- Now we map x and y to normalized device coordinates, as before:

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

Projective Transforms in OpenGL



- Now we map x and y to normalized device coordinates, as before:

$$\begin{aligned} x' &= \left(\frac{2}{r-l}\right)\left(-\frac{n}{z}x\right) - \frac{r+l}{r-l} \\ &= \left[\left(\frac{2n}{r-l}\right)x + \left(\frac{r+l}{r-l}\right)z\right]/(-z) \end{aligned}$$

- Now we can get more rows of our matrix.

$$x' = \left[\left(\frac{2n}{r-l}\right)x + \left(\frac{r+l}{r-l}\right)z\right]/(-z)$$

$$\begin{bmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x_u \\ y_u \\ z_u \\ -z \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}$$

Projective Transforms in OpenGL

- Now we can get more rows of our matrix.

$$x' = \left[\left(\frac{2n}{r-l}\right)x + \left(\frac{r+l}{r-l}\right)z\right] / (-z)$$

$$\begin{bmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x_u \\ y_u \\ z_u \\ -z \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ ? & ? & ? & ? \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x_u \\ y_u \\ z_u \\ -z \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}$$

Projective Transforms in OpenGL

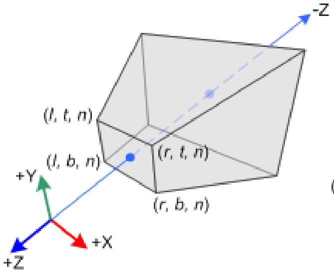
- Finding the third row of our matrix.
- We know z' does not depend on x or y, so let's try this:

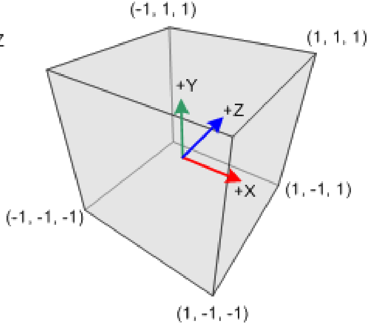
$$\begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & A & B \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x_u \\ y_u \\ z_u \\ -z \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}$$

$$z' = \frac{Az + B}{-z}$$

- Now solve for A and B

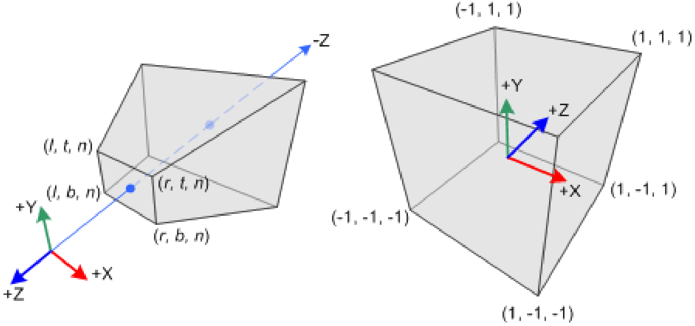
Projective Transforms in OpenGL





$$z' = \frac{Az + B}{-z}$$

Projective Transforms in OpenGL



$$z' = \frac{Az + B}{-z}$$

- When $z = -n$, $z' = -1$
- When $z = -f$, $z' = +1$

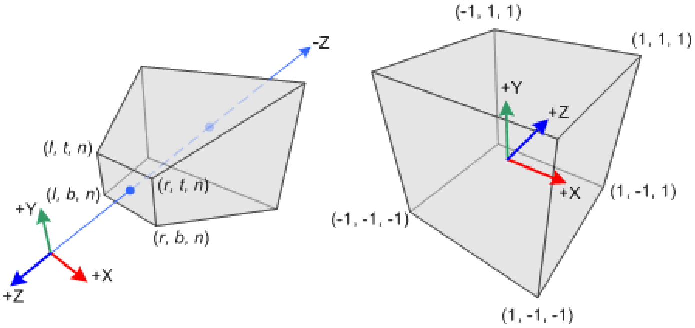
Projective Transforms in OpenGL

$$z' = \frac{Az + B}{-z}$$

- When $z = -n$, $z' = -1$
- When $z = -f$, $z' = +1$

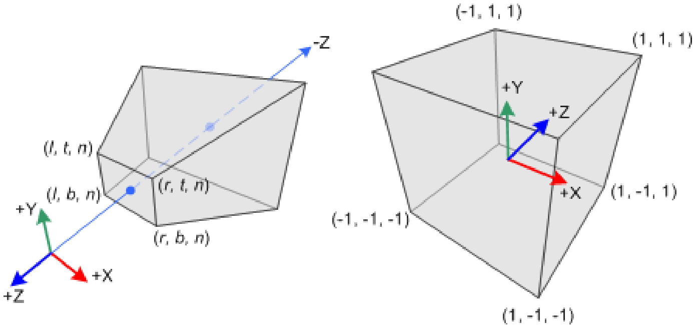
$$\begin{aligned}
-1 &= \frac{-nA + B}{n} \\
1 &= \frac{-fA + B}{f} \\
A &= -\frac{f + n}{f - n} \\
B &= -\frac{2fn}{f - n}
\end{aligned}$$

Projective Transforms in OpenGL



$$\begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x_u \\ y_u \\ z_u \\ -z \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}$$

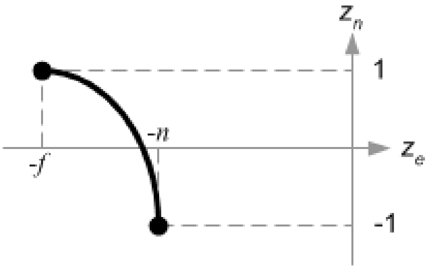
Projective Transforms in OpenGL



- When symmetric, $r = -l$ and $t = -b$, it simplifies:

$$\begin{bmatrix} \frac{n}{r} & 0 & 0 & 0 \\ 0 & \frac{n}{t} & 0 & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x_u \\ y_u \\ z_u \\ -z \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}$$

Z fighting in OpenGL



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