6.837 Tuesday, 3 November 1998

Real life:
- Exercise your right to vote today!

Project 3 **extended**:
- Now due 11:59:59pm **Sunday** night
- Project 4 assigned in class next Tuesday

Today:
- Overview of Rendering Pipeline
- Fill in two more pieces of pipeline:
  - Transformation to Eye Coordinates (H&B S12.2)
  - Perspective Transformation (H&B S12.4)

Thursday:
- Clipping and Screen Space
- End of Pipeline !

Rendering Pipeline

What actually happens between modeling & viewing?
Recall input, output of rendering process:

Input:
- *Geometric model*:
  - Description of all object, surface, and light source geometry and transformations
- *Lighting model:*
  - Computational description of object and light properties, interaction (reflection)
- *Synthetic Viewpoint* (or *Camera*):
  - Eye position and viewing frustum
- *Raster Viewport*:
  - Pixel grid onto which image plane is mapped

Output:
- *Colors/Intensities* suitable for framebuffer display
  - (For example, 24-bit RGB value at each pixel)

Classical Rendering Pipeline

Model Traversal
- Graphics *primitives* issued for display

Modeling Transformations
- Object-space → World-space

Lighting (Shading)
- Vertices *lit* (*shaded*) according to local lighting model

Viewing Transformation
- World-space → Eye-space (Canonical)

Perspective Transformation
- Eye-space → NDC (Clip-space) (Canonical)
- View frustum mapped into Normalized Viewing Volume

Clipping
- Portions of object outside 3D *view volume* are removed

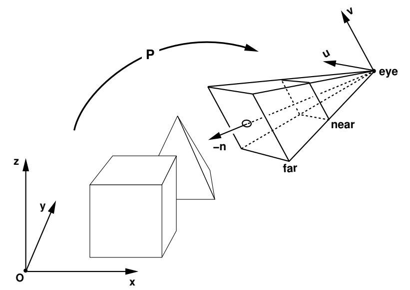
Transformation to Screen Space
- 3D object *projected* onto 2D *image plane*
- Portion of image plane mapped to raster *viewport*

Scan Conversion:
- Rasterization into Spans
  - Object diced into horizontal *spans*, one per raster
  - Depth, color, etc. *interpolated* along edges
- Span Filling
  - Color, etc. interpolated across each span, yielding pixels

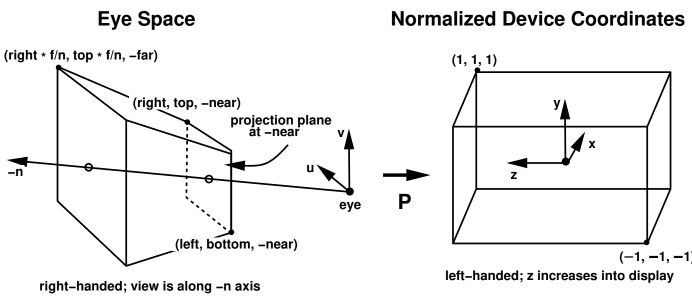
Visibility resolution
- Established *per-pixel*, (typically) in hardware

Today: Eye Space; NDC

Coordinate transformation to Eye Space



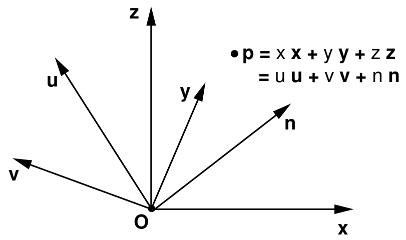
Then, Eye Space to Normalized Device Coordinates



(Sometimes called "clip space")

Change of Orthobasis (xyz to uvn)

"Express **xyz** (i.e., world-space) points in **uvn** basis."



We know $\mathbf{p} = x\hat{\mathbf{x}} + y\hat{\mathbf{y}} + z\hat{\mathbf{z}} = u\hat{\mathbf{u}} + v\hat{\mathbf{v}} + n\hat{\mathbf{n}}$
Find transformation $\mathbf{M}$ that, given (x, y, z), produces (u, v, n)

$$\begin{aligned}
\hat{\mathbf{x}} &= (\hat{\mathbf{x}} \cdot \hat{\mathbf{u}})\hat{\mathbf{u}} + (\hat{\mathbf{x}} \cdot \hat{\mathbf{v}})\hat{\mathbf{v}} + (\hat{\mathbf{x}} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}} \\
\hat{\mathbf{y}} &= (\hat{\mathbf{y}} \cdot \hat{\mathbf{u}})\hat{\mathbf{u}} + (\hat{\mathbf{y}} \cdot \hat{\mathbf{v}})\hat{\mathbf{v}} + (\hat{\mathbf{y}} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}} \\
\hat{\mathbf{z}} &= (\hat{\mathbf{z}} \cdot \hat{\mathbf{u}})\hat{\mathbf{u}} + (\hat{\mathbf{z}} \cdot \hat{\mathbf{v}})\hat{\mathbf{v}} + (\hat{\mathbf{z}} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}
\end{aligned}$$

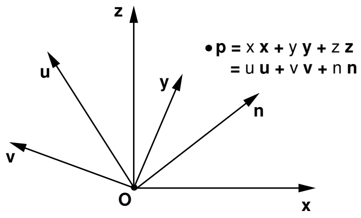
Expand expression for $\mathbf{p}$; collect terms in u, v, n:

$$\begin{aligned}
u &= x(\hat{\mathbf{x}} \cdot \hat{\mathbf{u}}) + y(\hat{\mathbf{y}} \cdot \hat{\mathbf{u}}) + z(\hat{\mathbf{z}} \cdot \hat{\mathbf{u}}) \\
v &= x(\hat{\mathbf{x}} \cdot \hat{\mathbf{v}}) + y(\hat{\mathbf{y}} \cdot \hat{\mathbf{v}}) + z(\hat{\mathbf{z}} \cdot \hat{\mathbf{v}}) \\
n &= x(\hat{\mathbf{x}} \cdot \hat{\mathbf{n}}) + y(\hat{\mathbf{y}} \cdot \hat{\mathbf{n}}) + z(\hat{\mathbf{z}} \cdot \hat{\mathbf{n}})
\end{aligned}$$

In matrix form

$$\begin{pmatrix} u \\ v \\ n \\ 1 \end{pmatrix} = \begin{pmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Two interpretations of orthobasis change



$$\mathbf{M} = \begin{pmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}; \qquad \text{Check:} \qquad \mathbf{M}\begin{pmatrix} u_x \\ u_y \\ u_z \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

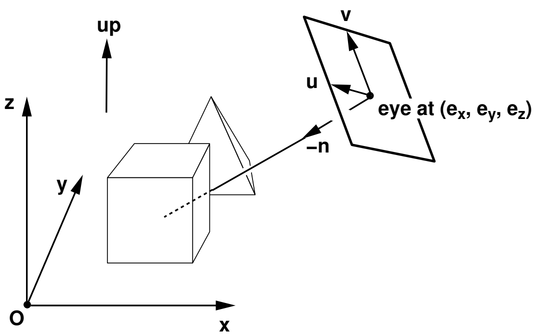
Two interpretations (both work, if applied consistently):
1: $\mathbf{M}$ "rotates" **uvn** basis into **xyz** basis
2: $\mathbf{M}$ "reexpresses" every **xyz** point $\mathbf{p}$ in **uvn** coords!

What is $\mathbf{M}^{-1}$? (What takes **uvn** to **xyz** coords?)

Simply $\mathbf{M}^T$, since $\mathbf{M}^T \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} u_x \\ u_y \\ u_z \\ 1 \end{pmatrix}$, etc.

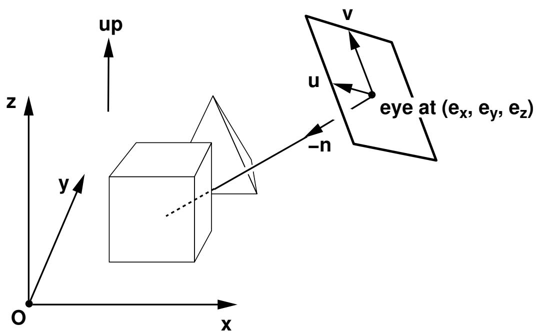
Positioning Synthetic Camera

To achieve effective visual simulation, want
1) eyepoint to be in proximity of modeled scene
2) view directed toward region of interest, and
3) view frustum with a reasonable "twist"



Perspective projection uses *eye coordinates*:
eyepoint at origin
u axis toward "right" of image plane
v axis toward "top" of image plane
view direction along *negative* **n** axis
(will handle "twist" constraint later)

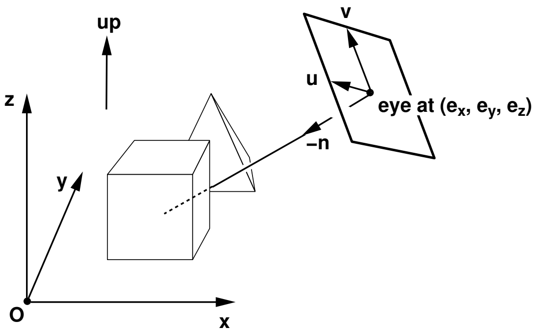
Positioning Synthetic Camera



Given eyepoint $\mathbf{e}$, basis $\hat{\mathbf{u}}$, $\hat{\mathbf{v}}$, $\hat{\mathbf{n}}$
Deduce $\mathbf{M}$ which expresses world in eye coordinates:

$$\mathbf{M}\begin{pmatrix} e_x \\ e_y \\ e_z \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}; \mathbf{M}\begin{pmatrix} e_x - \hat{n}_x \\ e_y - \hat{n}_y \\ e_z - \hat{n}_z \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -1 \\ 1 \end{pmatrix}$$

Positioning Synthetic Camera



Overlay origins, then **change bases**:

$$\mathbf{T} = \begin{pmatrix} 1 & 0 & 0 & -e_x \\ 0 & 1 & 0 & -e_y \\ 0 & 0 & 1 & -e_z \\ 0 & 0 & 0 & 1 \end{pmatrix}; \mathbf{R} = \begin{pmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

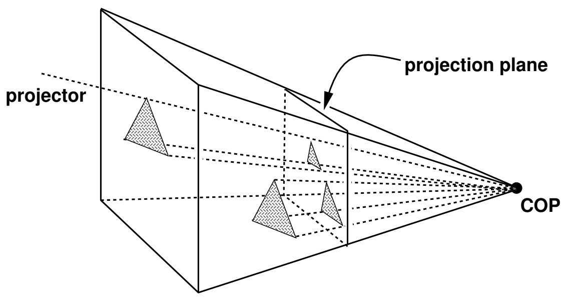
$$\mathbf{M} = \mathbf{RT}$$

Correctness check left as exercise
Still must handle "twist" constraint
Align **v** with world **up** vector (how?)

Recall inputs, outputs of "lookat" procedure

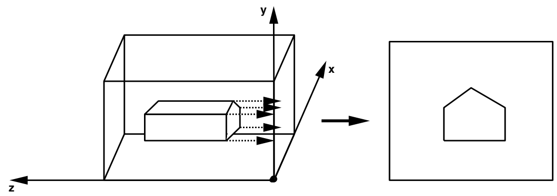
What is Projection?

Converts 3D representation to 2D



Ray family ("projectors") emanates from COP ("center of projection"), through object onto "projection plane"
Parallel (orthographic) projection – COP at infinity
Perspective projection – COP is a local point

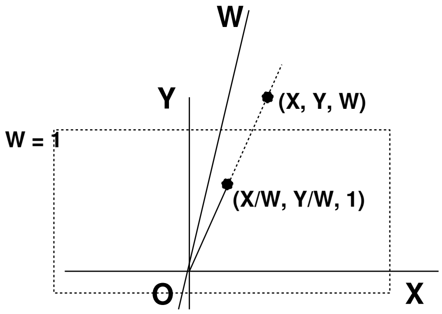
Orthographic Projection



No decrease in apparent size with distance from eye
Models eye "at infinity" in world where parallel rays don't appear to converge
Implementation trivial: simply drop one of (x, y, z)
"Flattened" entity described by only 2 coordinates!

2D Homogeneous Coordinates

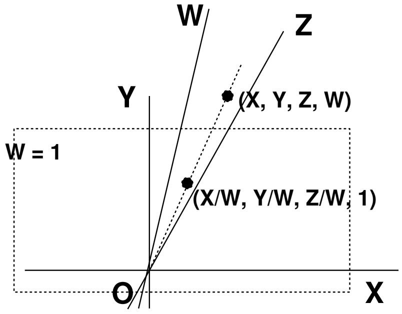
Use representation (X, Y, W) for points



Divide by W, *drop* it to *recover* coordinates $(\frac{X}{W}, \frac{Y}{W})$
$W > 0$: "Positive finite point," what we're used to.
$W = 0$: "Point at infinity." Consider the limit as $W \to 0$: a direction!
Careful: remember $W < 0$.
$W < 0$: "Negative finite point"
Operation maps *all of XYW space* **onto** the projection plane $W = 1$!
Note that scalar multiplication has no effect

3D Homogeneous Coordinates

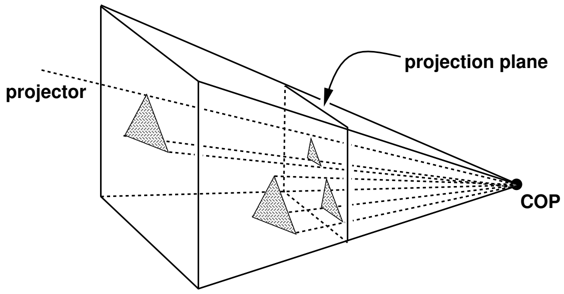
Use representation (X, Y, Z, W) for points
Divide by W, *drop* it to *recover* coordinates $(\frac{X}{W}, \frac{Y}{W}, \frac{Z}{W})$
W "axis": no longer easily visualizable, but there by analogy



Note: when $W = 0$, point at infinity $\equiv$ direction
Again: scalar multiplication doesn't change point!

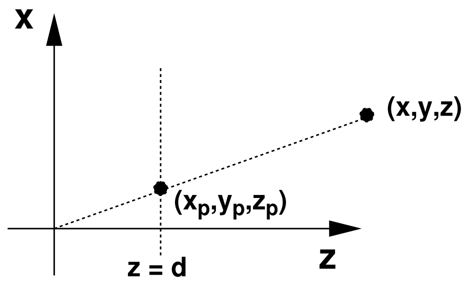
Planar Geometric Projections

"Planar" means projection is onto a plane
"Geometric" means projectors (rays) are straight



Perspective Projection

First: project onto plane $z = d$



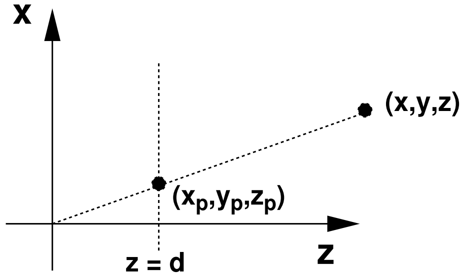
What are coordinates of projected point $\mathbf{P}_p$?
By similar triangles,

$$\frac{x_p}{d} = \frac{x}{z}$$
$$\frac{y_p}{d} = \frac{y}{z}$$

Multiplying through by d yields

$$x_p = \frac{d \cdot x}{z} = \frac{x}{z/d}$$
$$y_p = \frac{d \cdot y}{z} = \frac{y}{z/d}$$
$$z_p = d$$

Perspective Projection



$$x_p = \frac{d \cdot x}{z} = \frac{x}{z/d}$$
$$y_p = \frac{d \cdot y}{z} = \frac{y}{z/d}$$
$$z_p = d$$

Things to note:
- x_p and y_p scale linearly with d
- Division by z shrinks distant objects
- $z = 0$ not allowed (what happens to this plane?)
- Operation ok (i.e., sane) for all other points

Perspective Projection

Matrix formulation

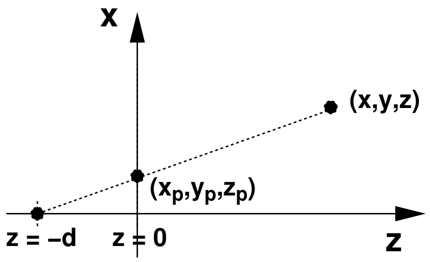
$$\mathbf{M} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{pmatrix}$$
$$\begin{pmatrix} X \\ Y \\ Z \\ W \end{pmatrix} = \mathbf{M} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \\ z/d \end{pmatrix}$$

Finally, recover projected point

$$\begin{pmatrix} X/W \\ Y/W \\ Z/W \end{pmatrix} = \begin{pmatrix} x_p \\ y_p \\ z_p \end{pmatrix} = \begin{pmatrix} \frac{x}{z/d} \\ \frac{y}{z/d} \\ d \end{pmatrix}$$

Another Perspective Projection

Projection *plane* at $z = 0$
Center of projection at $z = -d$



$$x_p = \frac{d \cdot x}{z + d} = \frac{x}{(z/d) + 1}, y_p = \frac{d \cdot y}{z + d} = \frac{y}{(z/d) + 1}$$

In matrix form,

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \mathbf{0} & 0 \\ 0 & 0 & 1/d & 1 \end{pmatrix}$$

Matrix Formulation: Limit

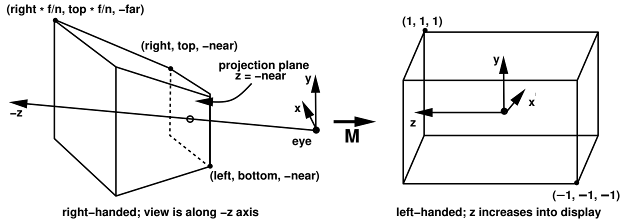
Note that as $d \to \infty$

$$\mathbf{M} \to \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{0} & 1 \end{pmatrix}$$

which is simply orthographic projection!

Projection of Finite View Volumes

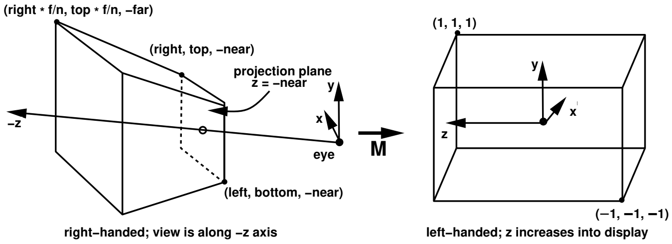
Recall view frustum: left, right, bottom, top, near, far
Looking down $-z$ axis: so, with $n, f > 0$:



Wish to project this into "canonical" view volume:
parallelepiped bounded by $x = \pm 1$, $y = \pm 1$, $z = \pm 1$
Called NDC, or sometimes Clip Coordinates

Matrix Formulation

$$\mathbf{M} = \begin{pmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & -(\frac{f+n}{f-n}) & -(\frac{2fn}{f-n}) \\ 0 & 0 & -1 & 0 \end{pmatrix}$$



Check action of $\mathbf{M}$:

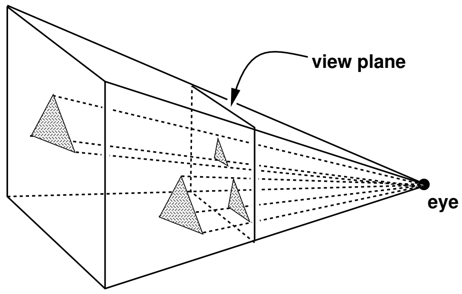
$$\mathbf{M}\begin{pmatrix} l \\ b \\ -n \\ 1 \end{pmatrix} \quad ; \quad \mathbf{M}\begin{pmatrix} r\frac{f}{n} \\ t\frac{f}{n} \\ -f \\ 1 \end{pmatrix} \quad ; \quad \mathbf{M}\begin{pmatrix} (l+r)/2 \\ (b+t)/2 \\ -n \\ 1 \end{pmatrix}$$

What Good are Homogeneous Coordinates?

Formulate *translation* as 4×4 matrix operation
(instead of as separate addition, as in L3)
Formulate *projection* (non-linear) as matrix op
Map interior of view frustum to canonical parallelepiped
Allow specification of "points at infinity": directions
Infinitely distant point light sources
Normals (specified by planes to which they're $\perp$)
(Correct clipping of certain complex entities
requires clipping in homogeneous coordinates)

Qualitative Features of Projection

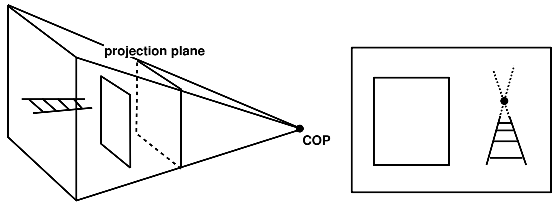
Equal-sized objects project to different sizes!



Scaling with respect to COP has no effect
(scales object, but increases distance)
Projection maps lines to lines, but ...
Projection does not preserve shape of planar figures
except those on planes parallel to projection plane
Projection does not preserve normals!
so, don't do lighting after perspective

Qualitative Features of Projection

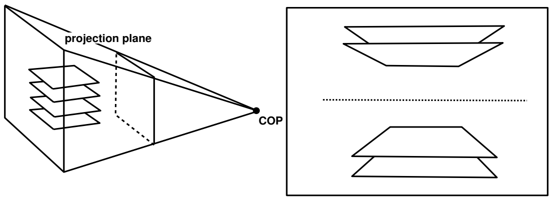
Families of parallel lines have "vanishing points"
projection of point at infinity in direction of lines



Except: line families *parallel* to projection
plane have no (local) vanishing points

Qualitative Features of Projection

Families of parallel *planes* have vanishing *lines*:
projection of line at infinity which lies in all planes



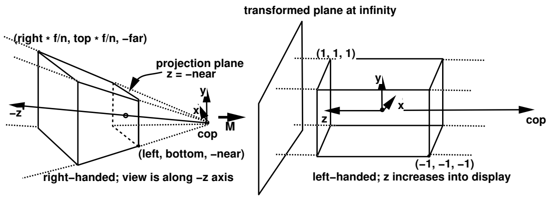
Except, of course plane families *parallel* to
projection plane

What happens to COP?

Perspective matrix:

$$\mathbf{M} = \begin{pmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

$$\begin{aligned} \mathbf{COP'} &= \mathbf{M}\ \mathbf{COP} = \mathbf{M}\begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 & -\frac{2fn}{f-n} & 0 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 & -1 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix} \end{aligned}$$



Transforming Plane Equations

Recall that $Ax + By + Cz + D = 0$ can be written:

$$\begin{pmatrix} A & B & C & D \end{pmatrix}\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = 0$$

Suppose space is transformed by transformation $\mathbf{M}$
What "happens" to plane equation $\mathbf{H}$ under $\mathbf{M}$?

$$\mathbf{Hp} = 0$$

For any (non-singular) transformation matrix $\mathbf{M}$

$$\mathbf{HM}^{-1}\mathbf{Mp} = 0$$

Which can be rewritten (reparenthesized) as:

$$(\mathbf{HM}^{-1})(\mathbf{Mp}) = 0$$

Thus the transformed plane equation **H'** is

$$\mathbf{HM}^{-1}$$

Plane $\mathbf{H}$, under action of $\mathbf{M}$, becomes $\mathbf{HM}^{-1}$

What happens to plane through COP?

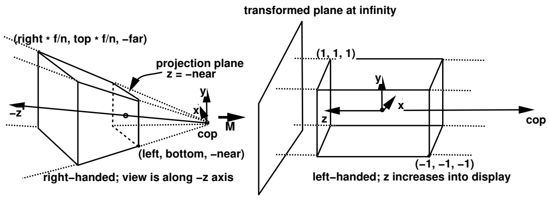
Plane postmultiplied by matrix *inverse*:

$$\mathbf{M}^{-1} = \begin{pmatrix} \frac{r-l}{2n} & 0 & 0 & \frac{r+l}{2n} \\ 0 & \frac{t-b}{2n} & 0 & \frac{t+b}{2n} \\ 0 & 0 & 0 & -1 \\ 0 & 0 & \frac{1}{2f} - \frac{1}{2n} & \frac{1}{2f} + \frac{1}{2n} \end{pmatrix}$$

consider plane $z = 0$, $\mathbf{H} = \begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix}$

$$\begin{aligned} \mathbf{H}' &= \mathbf{HM}^{-1} = \begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix} \mathbf{M}^{-1} \\ &= \begin{pmatrix} 0 & 0 & 0 & -1 \end{pmatrix} \end{aligned}$$

What points lie on $\mathbf{H}'$?



What happens to projection plane?

Again, postmultiply by matrix inverse:

$$\mathbf{M} = \begin{pmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

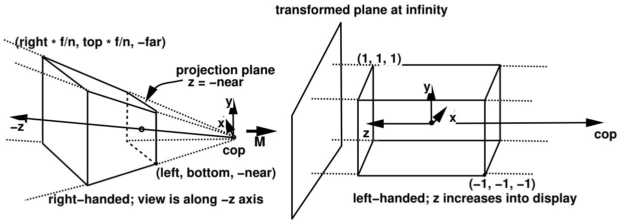
$$\mathbf{M}^{-1} = \begin{pmatrix} \frac{r-l}{2n} & 0 & 0 & \frac{r+l}{2n} \\ 0 & \frac{t-b}{2n} & 0 & \frac{t+b}{2n} \\ 0 & 0 & 0 & -1 \\ 0 & 0 & \frac{1}{2f} - \frac{1}{2n} & \frac{1}{2f} + \frac{1}{2n} \end{pmatrix}$$

consider plane $z = -n$, $\mathbf{H} = \begin{pmatrix} 0 & 0 & 1 & n \end{pmatrix}$

$$\begin{aligned} \mathbf{H}' &= \mathbf{HM}^{-1} = \begin{pmatrix} 0 & 0 & 1 & n \end{pmatrix} \mathbf{M}^{-1} \\ &= \begin{pmatrix} 0 & 0 & \frac{n-f}{2f} & \frac{n-f}{2f} \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 & 1 & 1 \end{pmatrix} \end{aligned}$$

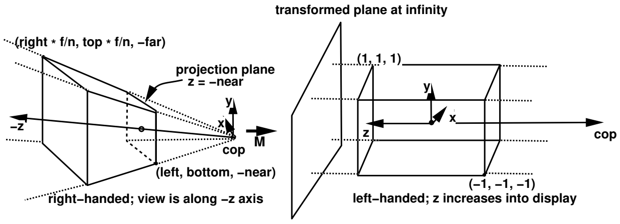
Transformation of projection plane (cont.)

But this is the plane $z + 1 = 0$, or $z = -1$!



What happens to plane at infinity?

Its equation is $\mathbf{H} = \begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix}$



$$\begin{aligned} \mathbf{H}' &= \mathbf{HM}^{-1} = \begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{M}^{-1} \\ &= \begin{pmatrix} 0 & 0 & n-f & n+f \end{pmatrix} \end{aligned}$$

Suppose $n = 1$, $f = 2$. Then

$$\mathbf{H}' = \begin{pmatrix} 0 & 0 & -1 & 3 \end{pmatrix}$$

This is the plane $z = 3$!

Aggregate Coordinate Transformations

Concatenate to existing matrix:

$$\begin{pmatrix} x_S \\ y_S \\ z_S \end{pmatrix} = \mathbf{S}\ \mathbf{P}\ \mathbf{E}\ \mathbf{M} \begin{pmatrix} x_O \\ y_O \\ z_O \\ 1 \end{pmatrix}$$

M Modeling transformations (Object to World)
E Translate, Rotate World into Eye coordinates
P Perspective transformation into NDC (device-ind.)
S Transformation into screen coordinates (device-dep.)
(Note: projection from 4- to 3-vector occurs after **P**!)