Lecture 6: Tuesday, 28 September 1999

Administrative:
- C++ review (Kari Anne), W 7-9pm in 34-101
- To view binary `.iv` files, use `ivcat foo.iv`
- `uid_object, *.iv` due this Friday 5pm
  - Exhibition next Tuesday, the 5^{th} in class!

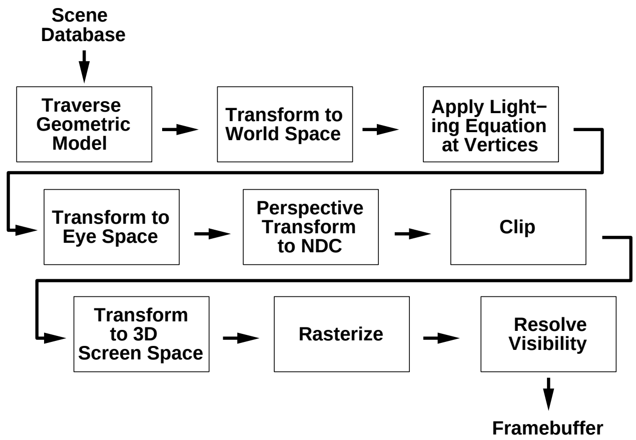
Last time:
- Limit of Sierpinski Tetrahedron (Area A, Volume V)?

Today:
- Fill in more pieces of pipeline:
  - Transformation to Eye Coordinates (H&B S12.2)
    - Transformation of Normals, Planes
  - Perspective Transformation (H&B S12.4)
- Assignment 4: Scene Modeling (Handout)

Thursday, Tuesday:
- Perspective Transformation (Cont.)
- Screen Coordinates
- Backface elimination
- Polygon Scan Conversion

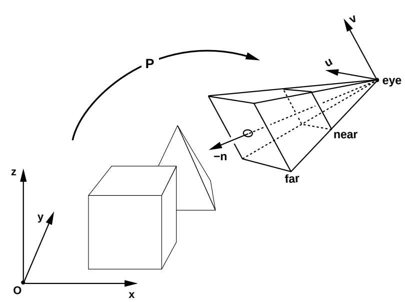
Rendering Pipeline

What actually happens between modeling & viewing?

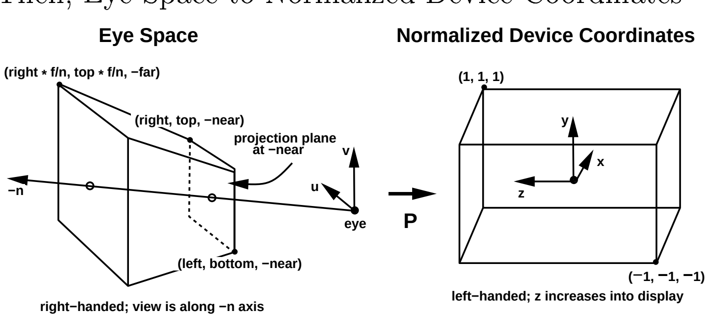


Today: Eye Space; NDC

Coordinate transformation to Eye Space

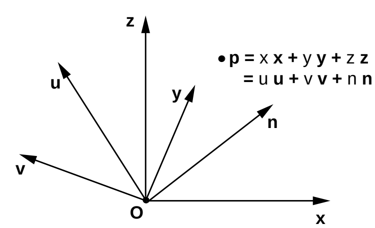


Then, Eye Space to Normalized Device Coordinates



(Sometimes called "clip space")

Orthobasis change: two interpretations



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

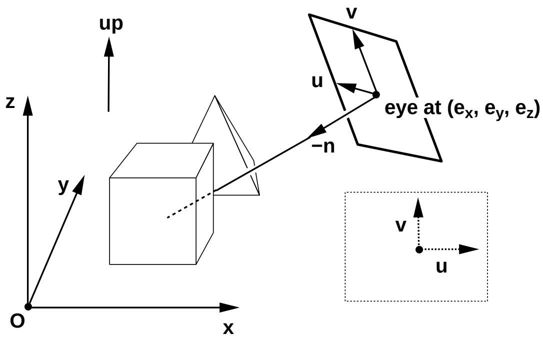
Two interpretations (both work, if applied consistently):
1: **M** "rotates" **uvn** basis into **xyz** basis
2: **M** "reexpresses" every **xyz** point **p** in **uvn** coords!

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.

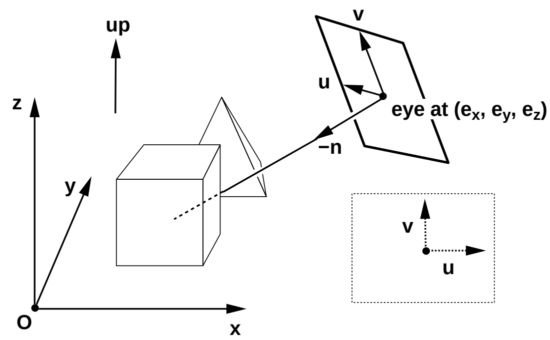
Positioning Synthetic Camera

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)

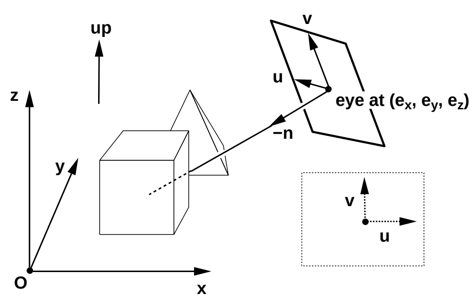
Positioning Synthetic Camera



Given eyepoint **e**, basis $\hat{\mathbf{u}}$, $\hat{\mathbf{v}}$, $\hat{\mathbf{n}}$
Deduce **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:

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

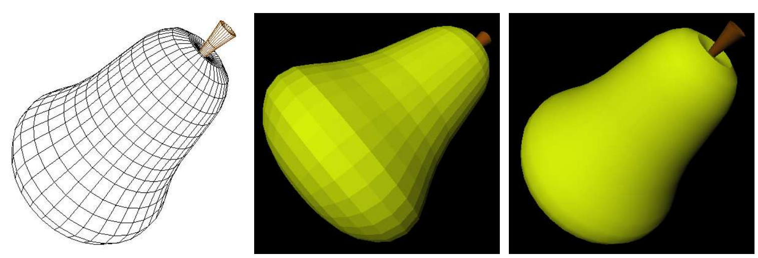
$$\mathbf{M} = \mathbf{RT}$$

"Twist" constraint: Align **v** with world **up** vector (how?)
Given: n, **up**
Compute
Compute
Will this always work ?

Digression: Normals in Inventor

Per-face normals

```
# .../ivexamples/facetedpear.iv
Coordinate3 { ... }
NormalBinding { value PER_FACE }
IndexedFaceSet { coordIndex [ 0, 1, 2, -
```



Per-vertex normals

```
# .../ivexamples/smoothpear.iv
Coordinate3 { ... } # x y z per vertex
Normal { ... } # dx dy dz per vertex
NormalBinding { value PER_VERTEX }
IndexedFaceSet { coordIndex [ 0, 1, 2, -
```

Transforming Normal Vectors

Consider point $\mathbf{p}$, and normal $\mathbf{n}$ such that $\mathbf{n} \perp \mathbf{p}$

$\mathbf{n}^T\mathbf{p} = 0$

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

$\mathbf{n}^T \boxed{} \mathbf{p} = 0$

which can be rewritten as

$$(\mathbf{n}^T\mathbf{M}^{-1})(\mathbf{M}\mathbf{p}) = 0$$

thus the *transpose* of the transformed normal is

$$\mathbf{n}^T\mathbf{M}^{-1}$$

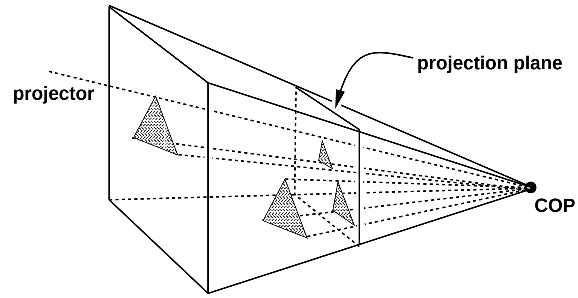
and the transformed normal itself is

$$(\mathbf{n}^T\mathbf{M}^{-1})^T = \mathbf{M}^{-T}\mathbf{n}$$

Normal $\mathbf{n}$, under action of $\mathbf{M}$, becomes $\mathbf{M}^{-T}\mathbf{n}$

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 local

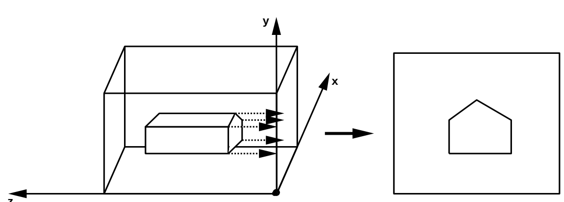
Planar Geometric Projections

"Planar" means projection is onto a plane

"Geometric" means projectors (rays) are straight

Distinction between perspective transformation and perspective projection

Orthographic Projection



No decrease in apparent size with distance from eye

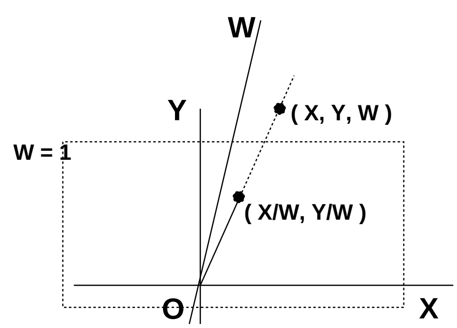
Models eye "at infinity" in world where parallel rays don't appear to converge

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 $W \rightarrow 0$: a $\boxed{}$!

Careful: remember $W < 0$.

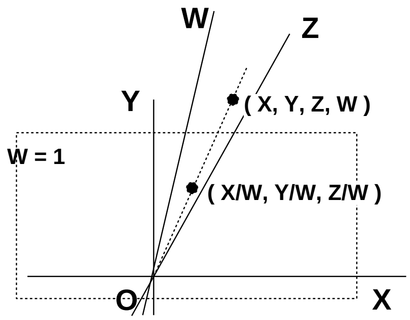
$W < 0$: "Negative finite point"

Operation maps *all of XYW space* **onto** the projection plane $W = 1$!

Note that scalar multiplication has no effect

3D Homogeneous Coordinates

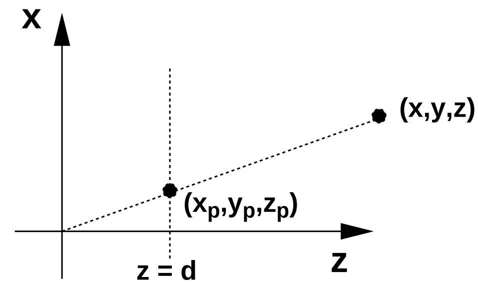
Use representation (X, Y, Z, W) for points
Divide by W, *drop* it to *recover* coordinates $(\frac{X}{W}, \frac{Y}{W}, \frac{Z}{W})$
W "axis": no longer easily visualizable,
but there by analogy



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

Perspective Projection

First: project onto plane $z = d$



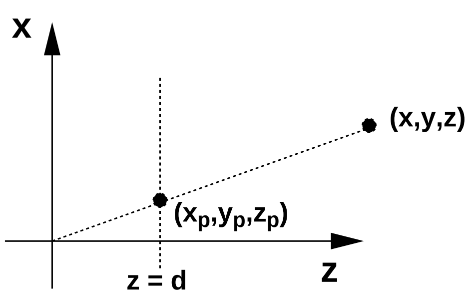
What are coordinates of projected point $\mathbf{P}_p$?
By similar triangles,

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

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

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

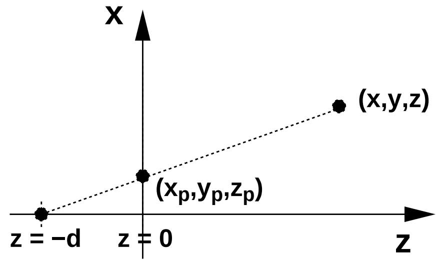
$$\begin{pmatrix} X \\ Y \\ Z \\ W \end{pmatrix} = \mathbf{M} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \\ z/d \end{pmatrix}$$

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

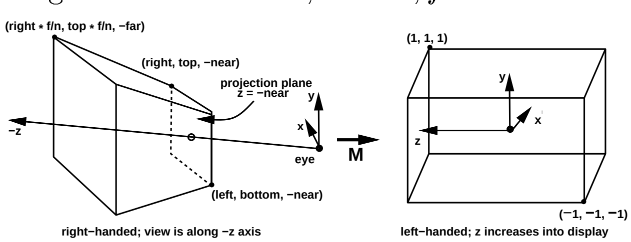
Note that as $d \to \infty$

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

which is simply ____________ !

Transformation of Finite View Volumes

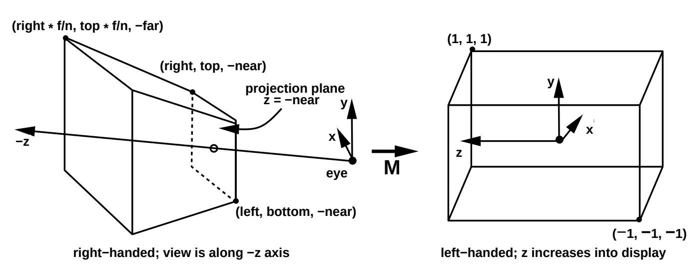
Recall view frustum: left, right, bottom, top, near, far
Looking down $-z$ axis: so, with $n, f > 0$:



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
Important: whereas perspective projection destroys depth information, perspective transformation **preserves depth** (for hidden-surface elimination)

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 $\mathbf{M}$:

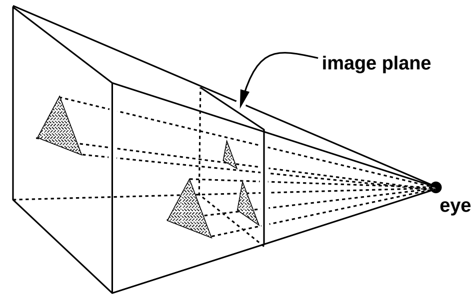
$$\mathbf{M}\begin{pmatrix} l \\ b \\ -n \\ 1 \end{pmatrix} \quad ; \quad \mathbf{M}\begin{pmatrix} r\frac{f}{n} \\ t\frac{f}{n} \\ -f \\ 1 \end{pmatrix} \quad ; \quad \mathbf{M}\begin{pmatrix} (l+r)/2 \\ (b+t)/2 \\ -n \\ 1 \end{pmatrix}$$

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)
Computational elegance:
  Example: 2D homogeneous coordinates (x, y, w)
  Two points make a ________
  Try it: points $(0, 1, 1)$ and $(1, 0, 1)$
  Resulting ________
  What if points are coincident?
Two lines make a ________
  Try it: lines $(1, 1, -1)$ and $(1, -1, 0)$
  Resulting ________ = ________
  What if lines are identical? Parallel?

Qualitative Features of Projection

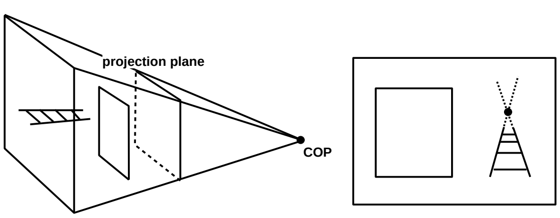
Equal-sized objects project to different sizes!



Scaling with respect to COP has no effect
  (scales object, but increases distance)
Projection does not preserve shape of planar figures
  except ________
Projection does not preserve normals!
  so, don't do lighting after perspective

Qualitative Features of Projection

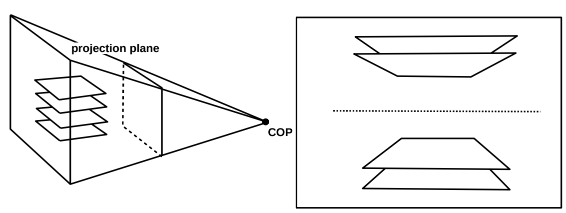
Families of parallel lines have "vanishing points"
  projection of point at infinity in direction of lines



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

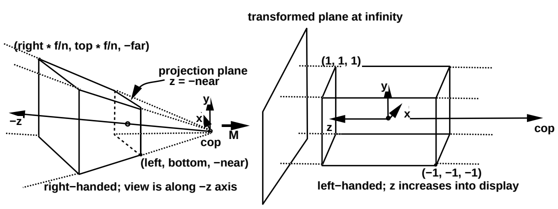
What happens to COP?

Perspective matrix:

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

Transform Center of Projection by $\mathbf{M}$:

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



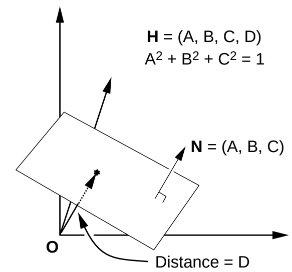
Plane Equations

Consider four-tuple $\mathbf{H} = (A, B, C, D)$ such that

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

and

$$A^2 + B^2 + C^2 = 1$$



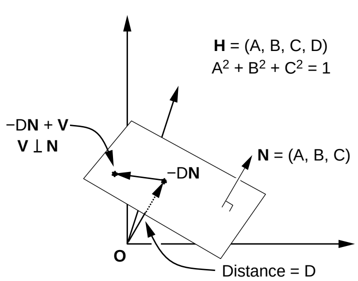
that is, that describes all points $\mathbf{p}$ such that

$$\mathbf{Hp} = 0$$

$\mathbf{H}$: has normal $\mathbf{N} =$ ☐ (Why? ☐)
and lies -D units from origin

Check that point $\mathbf{O}$ - D$\mathbf{N}$ lies on plane

$$\begin{pmatrix} A & B & C & D \end{pmatrix}\begin{pmatrix} -DA \\ -DB \\ -DC \\ 1 \end{pmatrix} = -D(A^2+B^2+C^2)+D = 0$$



Check: point $\mathbf{O}$ - D$\mathbf{N}$ + $\mathbf{v}$, for any $\mathbf{v} \perp N$

$$\begin{aligned} \mathbf{H}(\mathbf{O} - D\mathbf{N} + \mathbf{v}) &= \begin{pmatrix} A & B & C & D \end{pmatrix}\begin{pmatrix} -DA + V_x \\ -DB + V_y \\ -DC + V_z \\ 1 \end{pmatrix} \\ &= -D(A^2 + B^2 + C^2) + \mathbf{N} \cdot \mathbf{V} + D \\ &= 0 \end{aligned}$$

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?

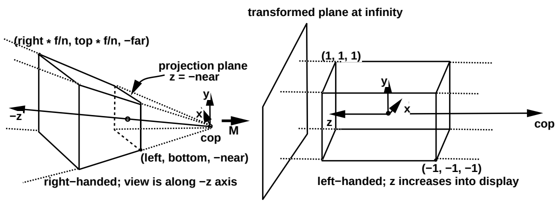
Plane postmultiplied by matrix *inverse*:

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

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

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

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



What happens to projection plane?

Again, postmultiply by matrix inverse:

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

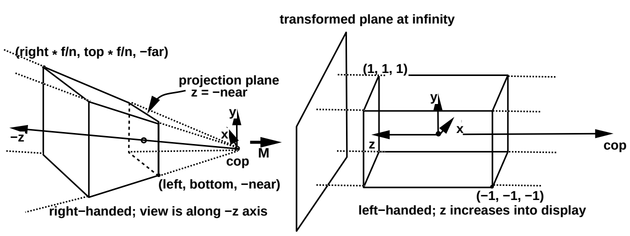
$$\mathbf{M}^{-1} = \begin{pmatrix} \frac{r-l}{2n} & 0 & 0 & \frac{r+l}{2n} \\ 0 & \frac{t-b}{2n} & 0 & \frac{t+b}{2n} \\ 0 & 0 & 0 & -1 \\ 0 & 0 & \frac{1}{2f} - \frac{1}{2n} & \frac{1}{2f} + \frac{1}{2n} \end{pmatrix}$$

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

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

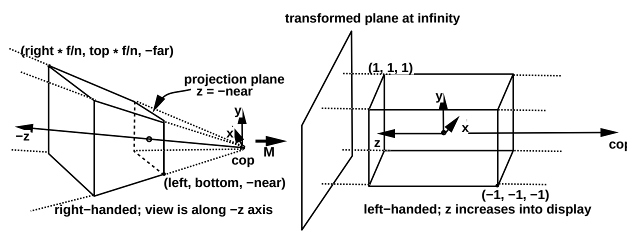
Transformation of projection plane (cont.)

But this is the plane !



What happens to plane at infinity?

Its equation is $\mathbf{H} =$



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

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

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

This is the plane $z = \frac{-(n+f)}{n-f} = 3$!