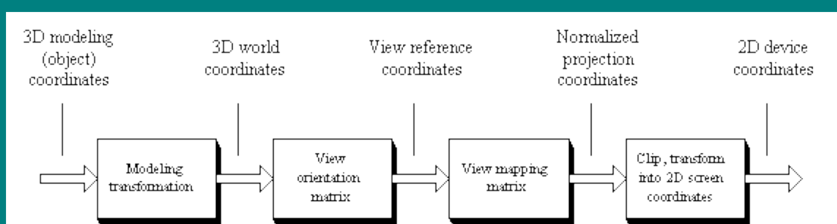
CMSC 435
Introductory Computer Graphics
Viewing

Penny Rheingans
UMBC

Relationship among Coord Systems



The matrix underneath each stage determines the transformation applied at that stage for the perspective and parallel projections

Viewport Transformation

- Window to viewport transform
  - Have projection coordinates (canonical view volume)
    - -1 <= x <= 1, -1 <= y <= 1, -1 <= z <= 1
  - Need device coordinates
    - -0.5<= x <= n_x, -0.5 <= y <= n_y , z unchanged
- Steps

Translate lower left corner to origin:

T(1,1,0)

Scale to correct size:

S(n_x/2, n_y/2,1)

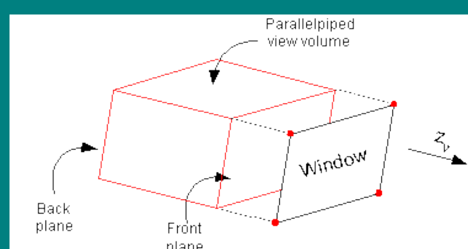
Translate into place:

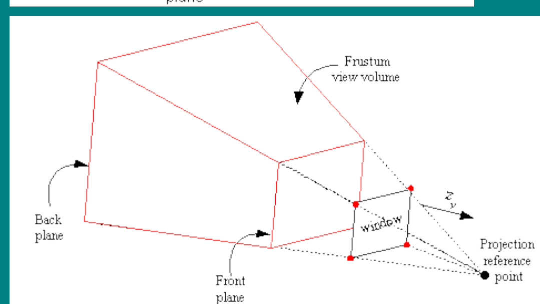
T(-0.5, -0.5 0)

$$M_{vp} = \begin{bmatrix} \frac{n_x}{2} & 0 & 0 & \frac{n_x - 1}{2} \\ 0 & \frac{n_y}{2} & 0 & \frac{n_y - 1}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

View Volumes

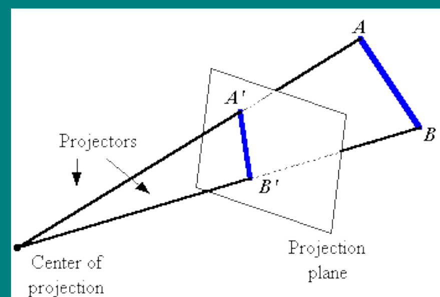
View volume bounded by front, back, top, bottom, and side planes. Front and back planes are parallel to the view plane at positions z_{front} and z_{back} along the z_v axis.



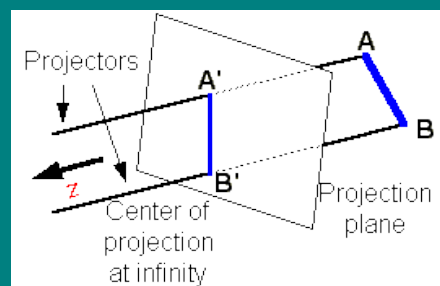


Projection

- Perspective
  - Line AB projects to A'B' (perspective projection)



- Parallel
  - Line AB projects to A'B' (parallel projection)
  - Projectors AA' and BB' are parallel



Simple Parallel Tform

View plane is normal to direction of projection

$x_s = x_v,\ y_s = y_v,\ z_s = 0$

Orthographic view volume bounded by

x: l,r = left, right

y: b,t = bottom, top

z: n,f = near, far

$$T_{ort} = \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & \frac{2}{n-f} & -\frac{n+f}{n-f} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Code Fragment

construct M_{vp}

construct M_{orth}

$M = M_{vp}M_{orth}$

for each line segment (a_i, b_i) do

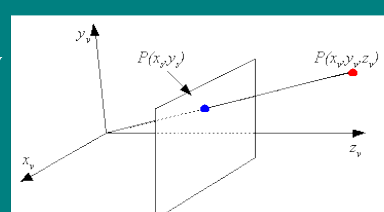
  $p = Ma_i$

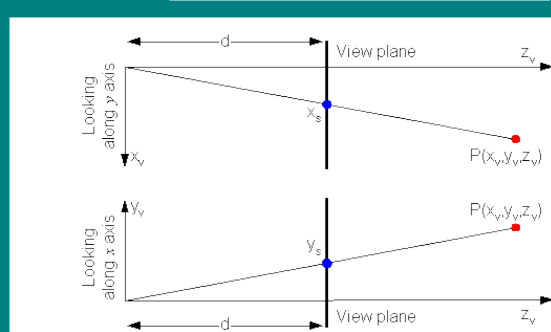
  $q = Mb_i$

  drawline(x_p, y_p, x_q, y_q)

Simple Perspective Tform

- Assume line from center of projection to center of view plane parallel to view plane normal.
- Center of projection is at origin.
- Have $P(x_v, y_v, z_v)$
- Want $P(x_s, y_s)$



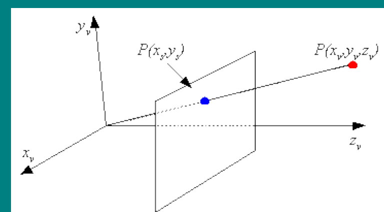


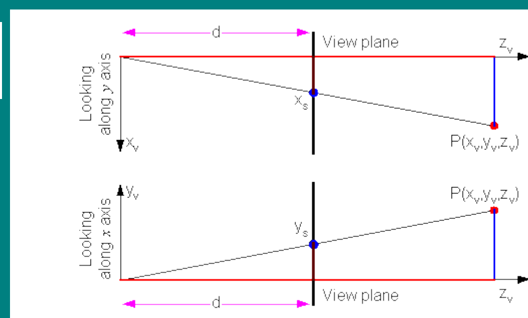
Simple Perspective Tform

- Have $P(x_v, y_v, z_v)$
- Want $P(x_s, y_s)$
- By similar triangles:

$$\frac{x_s}{d} = \frac{x_v}{z_v}, \frac{y_s}{d} = \frac{y_v}{z_v}$$

$$\Rightarrow x_s = \frac{d}{z_v} x_v, y_s = \frac{d}{z_v} y_v$$

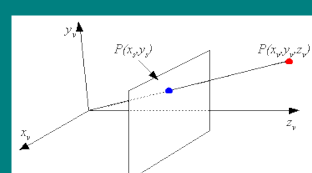




Simple Perspective Tform

- Have $P(x_v, y_v, z_v)$, want $P(x_s, y_s)$
- By similar triangles:

$$\frac{x_s}{n} = \frac{x_v}{z_v}, \frac{y_s}{n} = \frac{y_v}{z_v} \Rightarrow x_s = \frac{x_v}{z_v/n}, y_s = \frac{y_v}{z_v/n}$$



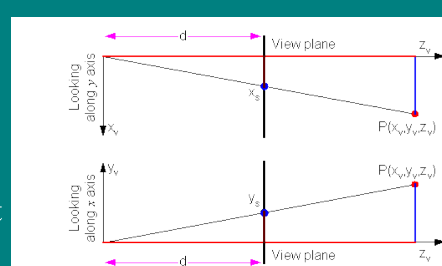
- In homogeneous coords

  $x = x_v$, $y=y_v$, $z = z_v$, $w = z_v/n$

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/n & 0 \end{bmatrix} \begin{bmatrix} x_v \\ y_v \\ z_v \\ 1 \end{bmatrix}$$

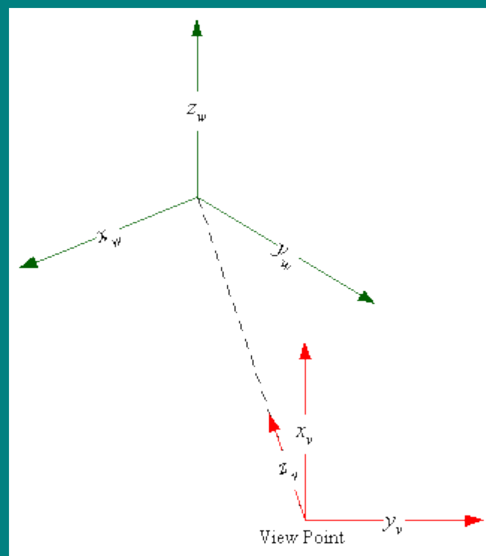
- Do perspective divide to get screen coords

  $x_s = x/w$, $y_s = y/w$, $z_s = z/w = n$



World and View Spaces

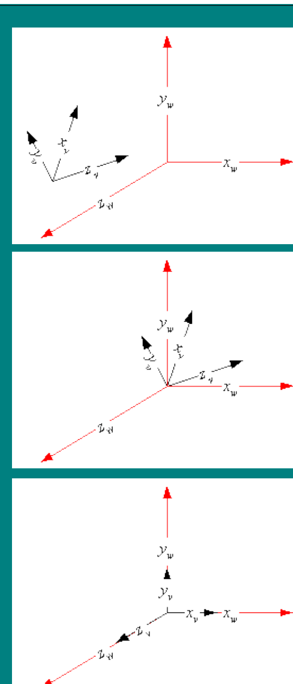
- World space
  - Used for modeling
  - Right-handed
- View space (simple)
  - Camera/viewer at origin
  - View along z_v axis
  - x_v and y_v aligned with display system



$$V = T_{view} \cdot W \Rightarrow \begin{bmatrix} x_v \\ y_v \\ z_v \\ 1 \end{bmatrix} = T_{view} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

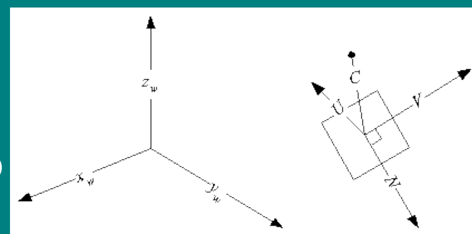
Camera Transform

- Transforms world to wiew coords:
  - Aligning a viewing system with the world coordinate axes using a sequence of translate-rotate tforms.
  - Translate view point to origin of world coordinate space.
  - Rotate to align view coordinate axes (x_v, y_v, z_v) with world coordinate axes (x_w, y_w, z_w)



Basic Viewing System

- Viewing system using
  - camera position C (or e)
  - viewing vector N (or -g)
  - up vector V (or t)
  - view plane distance d (or n)



- The world coordinate system is right-handed, the view coordinate system is left-handed.
- Characteristics
  - View direction controllable
  - Camera up controllable
  - No view volume specified
  - No view plane window specified
  - Perspective projection with viewport as center of projection

Implementing Basic Viewing

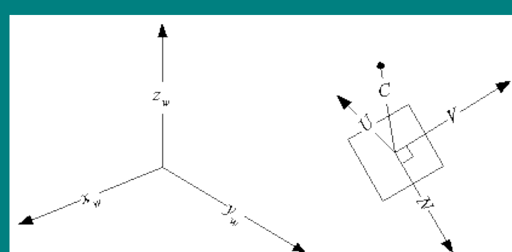
- Translation as before:

  $T(-c_x, -c_y, -c_z)$
- Rotate to align axes:

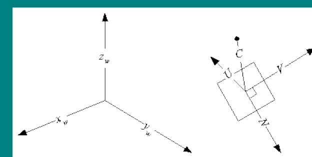
$$R = \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Convert to left-handed coordinates:

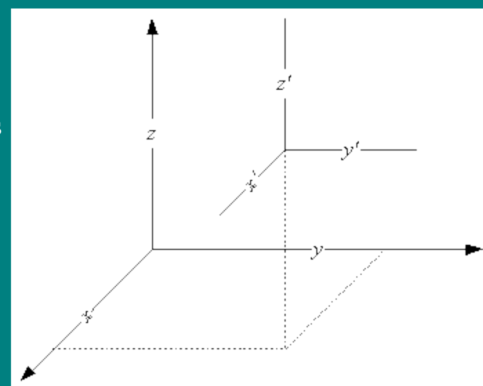
  S(1, 1, -1)



View Transformation

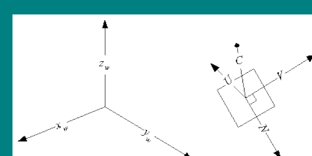


1. Translate origin of world coordinate system to origin of view coordinate system (transformation of coordinate system is inverse of that which moves points)

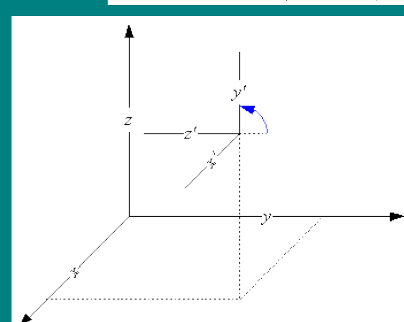


$$T_1 = \begin{bmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 1 & -c_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

View Transformation

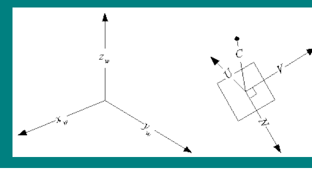


2. Rotate coordinate system 90° about x' axis. Use θ = -90.

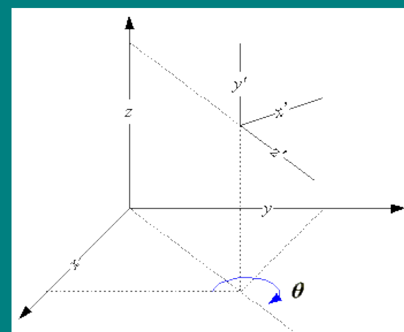


$$T_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

View Transformation

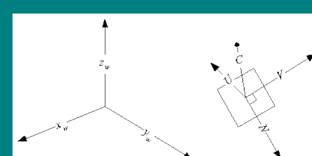


3. Rotate about y' by θ so that $(0,0,c_z)$ lies on z' axis.

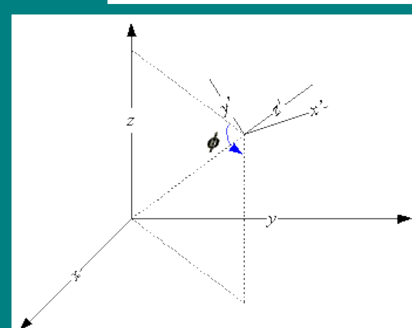


$$T_3 = \begin{bmatrix} \cos(-\theta) & 0 & \sin(-\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(-\theta) & 0 & \cos(-\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

View Transformation

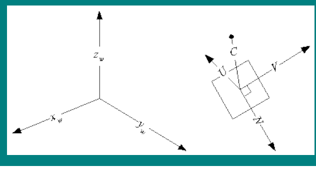


4. Rotate about x' by ϕ so that the origin of the original coordinate system lies on z' axis.



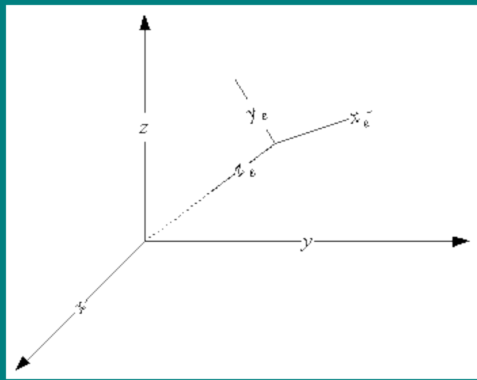
$$T4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(-\phi) & -\sin(-\phi) & 0 \\ 0 & \sin(-\phi) & \cos(-\phi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\phi & \sin\phi & 0 \\ 0 & -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

View Transformation

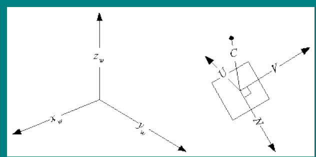


5. Reflect z’ axis to create left-handed coordinate system.

$$T_5 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

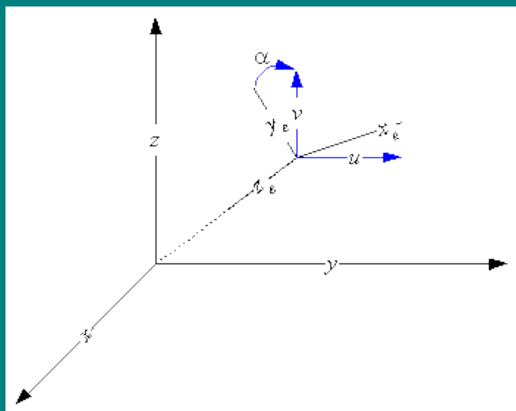


View Transformation



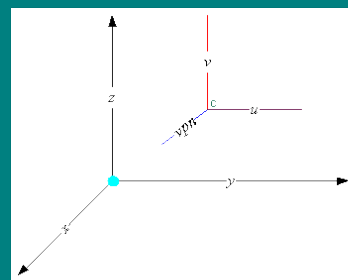
6. Twist about z’ so that y’ aligns with V.

$$T_6 = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



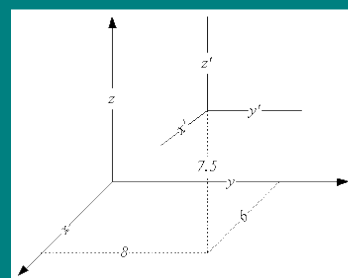
Viewing Example

Camera at (6,8,7.5)
View towards (0,0,0)
VPN (-6,-8, -7.5)
View up (-3.6, -4.8, 8.8)

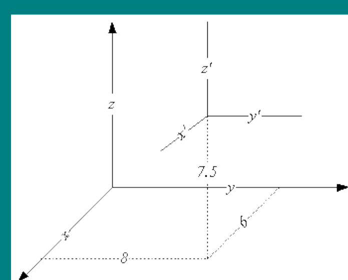


1. Translate world origin to view origin

$$T_1 = \begin{bmatrix} 1 & 0 & 0 & -6 \\ 0 & 1 & 0 & -8 \\ 0 & 0 & 0 & -7.5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

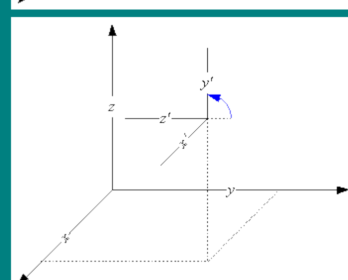


Viewing Example



2. Rotate 90° about x'.

$$T_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



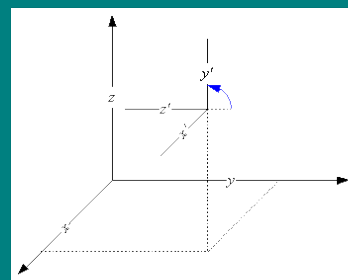
Viewing Example

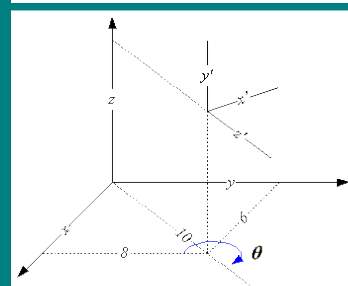
3. Rotate about y’ by θ so that $(0,0,c_z)$ lies on the z’ axis.

$\cos\theta = -8/10$

$\sin\theta = -6/10$

$$T_3 = \begin{bmatrix} -.8 & 0 & -.6 & 0 \\ 0 & 1 & 0 & 0 \\ .6 & 0 & -.8 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$





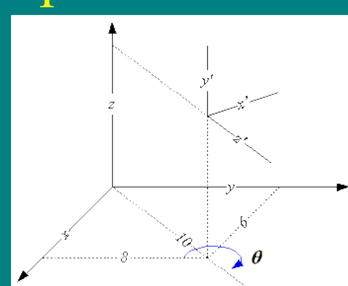
Viewing Example

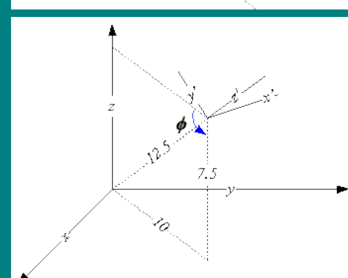
4. Rotate about x’ by ϕ so that the origin of the original coordinate system lies on the z’ axis.

$\cos\phi = 10/12.5$

$\sin\phi = 7.5/12.5$

$$T_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & .8 & -.6 & 0 \\ 0 & .6 & .8 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

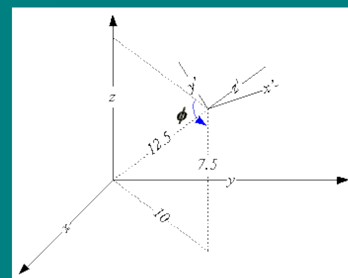


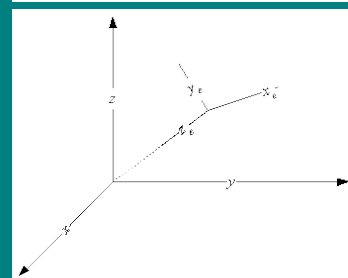


Viewing Example

5. Reflect z' axis to create left-handed coordinate system.

$$T_5 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$





Viewing Example

6. Twist about z' axis so that y' aligns with V.

$$\alpha = \cos^{-1}\left(\frac{V \cdot y_e}{|V| \cdot |y_e|}\right)$$

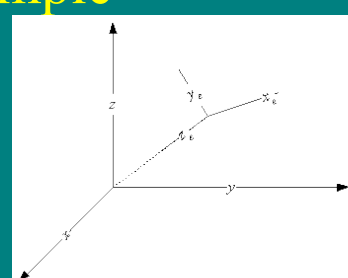
where $y_e = y_s{}^T y_w = T_5 T$

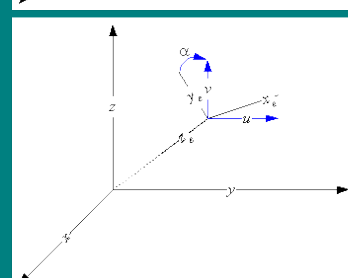
$V=(-3.6,-4.8, 8)$

$y_e = (-3.6, -4.8, 8)$

$\alpha = 0$

$$T_6 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

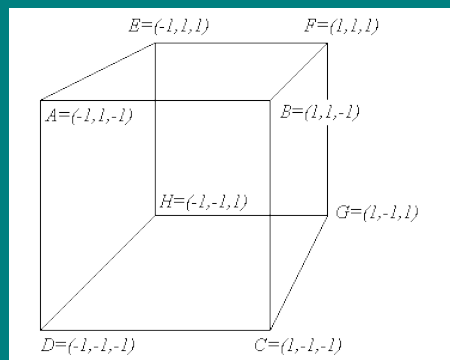




Viewing Example

- Multiply it all together
- Cube at origin

$$V = T_6T_5T_4T_3T_2T_1 = \begin{bmatrix} -.8 & .6 & 0 & 0 \\ -.36 & -.48 & .8 & 0 \\ -.48 & -.64 & -.6 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} T_1$$



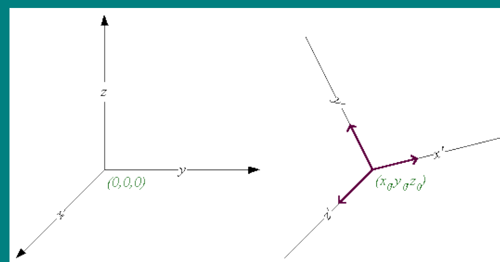
Compositions of Translations and Rotations

- Resulting matrix has form

$$M = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Basis Rotation Shortcut

- Where u'_x, u'_y, u'_z are unit basis vectors
- Assume we've already performed translation, so $x'_0 = y'_0 = z'_0 = 0$
- Can rotate to align basis vectors using



$$R = \begin{bmatrix} u'_{x1} & u'_{x2} & u'_{x3} & 0 \\ u'_{y1} & u'_{y2} & u'_{y3} & 0 \\ u'_{z1} & u'_{z2} & u'_{z3} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{where}\ \begin{aligned} u'_x &= \begin{bmatrix} u'_{x1} & u'_{x2} & u'_{x3} \end{bmatrix} \\ u'_y &= \begin{bmatrix} u'_{y1} & u'_{y2} & u'_{y3} \end{bmatrix} \\ u'_z &= \begin{bmatrix} u'_{z1} & u'_{z2} & u'_{z3} \end{bmatrix} \end{aligned}$$

- Expressed in coordinates of S

Applying the Shortcut

- Given view direction vector N

$$n = \frac{N}{|N|} = (n_1, n_2, n_3)$$

- Given view up vector V

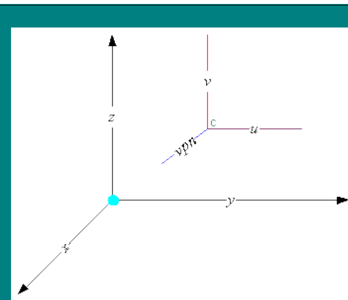
$$u = \frac{N \times V}{|N \times V|} = (u_1, u_2, u_3)$$

$$v = u \times n = (v_1, v_2, v_3)$$

$$R = \begin{bmatrix} u_1 & u_2 & u_3 & 0 \\ v_1 & v_2 & v_3 & 0 \\ n_1 & n_2 & n_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Shortcut Example

Camera at (6,8,7.5)
View towards (0,0,0)
VPN (-6,-8, -7.5)
View up (-3.6, -4.8, 8.8)



$$n = \frac{N}{|N|} = (n_1, n_2, n_3) = (-6/12.5, -8/12.5, -7/12.5) = (-.48, -.64, -.6)$$

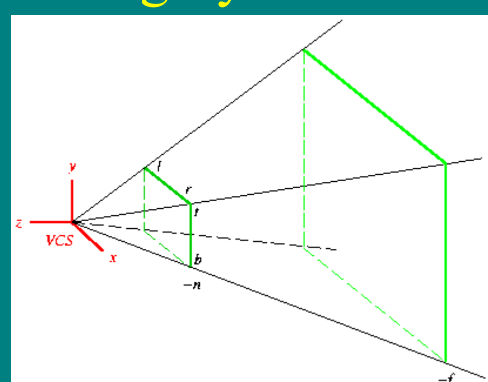
$$u = \frac{N \times V}{|N \times V|} = (u_1, u_2, u_3) = \frac{(-8 \cdot 8 - -7.5 \cdot -4.8, -7.5 \cdot -3.6 - -6 \cdot 8, -6 \cdot 4.8 - -8 \cdot -3.6)}{|N \times V|} = \frac{(-100, 75, 0)}{|N \times V|} = (-.8, .6, 0)$$

$$v = u \times n = (v_1, v_2, v_3) = (-.36, -.48, .8)$$

$$R = \begin{bmatrix} -.8 & .6 & 0 & 0 \\ -.36 & -.48 & .8 & 0 \\ -.48 & -.64 & -.6 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} T_1$$

Advanced Viewing System

- Frustum from six planes
- Left-handed system
- Using
  - Camera position (C)
  - View direction (N, $-Z_v$)
  - View up (Y_v)
  - Distance to near (n) and far (f) plane



- Characteristics
  - View position and direction controllable
  - Camera up controllable
  - View volume specified, but view plane constrained to be coincident with near plane
  - Perspective with center of projection at view point

Advanced Viewing System

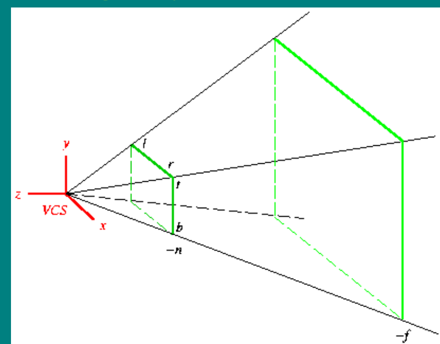
View volume specified by

$x_v = [r,l]z_v/n$ (sides)

$y_v = [t,b]z_v/n$ (top/bottom)

$z_v = n,f$ (near/far)

View plane has dimensions (r-l)×(t-b)



- Want 3D screen space for
  - 3D clipping
  - Visibility calculation
- Choose z_s such that
  - Z_s normalized for maximum precision
  - x,y positions unchanged on near plane

Projection for Advanced View

- Full perspective transform
  - $x = (2n/(r-l))x_v/z_v + ((l+r)/(l-r))$
  - $y = (2n/(t-b))y_v/z_v + ((t+b)/(b-t))$
  - $z = ((f+n)/(n-f))z_v + 2fn/(f-n)$
- Using homogeneous coordinates
  - $x = (2n/(r-l))x_v + ((l+r)/(l-r))z_v$
  - $y = (2n/(t-b))y_v + ((t+b)/(b-t))z_v$
  - $z = ((f+n)/(n-f)) + 2fn/(f-n)/z_v$
  - $w = z_v$
- So

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{l+r}{l-r} & 0 \\ 0 & \frac{2n}{t-b} & \frac{b+t}{b-t} & 0 \\ 0 & 0 & \frac{f+n}{n-f} & \frac{2fn}{f-n} \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_v \\ y_v \\ z_v \\ 1 \end{bmatrix}$$