Arbitrary 3D rotations slide # 8

wish to rotate about an arbitrary vector a

form a new basis uvw with w = a

rotate that basis to the world coordinates x, y, z

rotate by $\phi$ about the z axis

rotate back to the uvw basis

$$\begin{bmatrix} x_u & x_v & x_w \\ y_u & y_v & y_w \\ z_u & z_v & z_w \end{bmatrix} \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} x_u & y_u & z_u \\ x_v & y_v & z_v \\ x_w & y_w & z_w \end{bmatrix}}_{= R_{uvw}}$$

$R^T_{uvw}$

remember:

$$R^T = R^{-1}$$

w is aligned with a

where did u, v come from?

$$w = \frac{a}{\|a\|}$$

arbitrary: need t that is not co-linear with w

t = w + change smallest magnitude component to 1

$w = (1/\sqrt{2}, -1/\sqrt{2}, 0) \Rightarrow t = (1/\sqrt{2}, -1/\sqrt{2}, 1)$

$u = \frac{t \times w}{\|t \times w\|}$ $\quad v = w \times u$

Arbitrary 3D rotations slide # 8

wish to rotate about an arbitrary vector a

form a new basis uvw with w = a

rotate that basis to the world coordinates x, y z

rotate by $\phi$ about the z axis

rotate back to the uvw basis

$$\underbrace{\begin{bmatrix} x_u & x_v & x_w \\ y_u & y_v & y_w \\ z_u & z_v & z_w \end{bmatrix}}_{R^T_{uvw}} \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_u & y_u & z_u \\ x_v & y_v & z_v \\ x_w & y_w & z_w \end{bmatrix} = R_{uvw}$$

remember:

$$R^T = R^{-1}$$

w is aligned with a

where did u, v come from?

$$w = \frac{a}{\|a\|}$$

arbitrary

need t that is not co-linear with w

t = w + change smallest magnitude component to 1

$w = (1/\sqrt{2}, -1/\sqrt{2}, 0) \Rightarrow t = (1/\sqrt{2}, -1/\sqrt{2}, 1)$

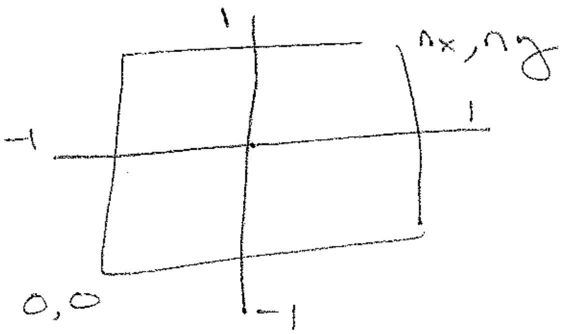
$u = \frac{t \times w}{\|t \times w\|}$ $\quad v = w \times u$

Slide #19
p. 160

# Canonical View Volume

$-1 < x < 1$
$-1 < y < 1$ cube is all x, y z points $\in [-1, 1]^3$
$-1 < z < 1$

screen is $n_x \times n_y$ pixels




$N_x$ might not equal $n_y$ in general

$$\begin{bmatrix} x_{pixel} \\ y_{pixel} \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{n_x}{2} & 0 & \frac{n_x - 1}{2} \\ 0 & \frac{n_y}{2} & \frac{n_y - 1}{2} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{canonical} \\ y_{canonical} \\ 1 \end{bmatrix}$$

$$x_{pixel} = \frac{n_x}{2} x + \frac{n_x - 1}{2}$$

$x = 0 \Rightarrow \frac{n_x - 1}{2}$

$x = -1 \Rightarrow \frac{-n_x + n_x - 1}{2} = -\frac{1}{2}$

$x = 1 \Rightarrow \frac{n_x + n_x - 1}{2} = n_x - \frac{1}{2}$

could carry z along for z buffering

slide #20 p. 162

Orthographic Projection

take 3D line with endpoints a + b
+ use matrix M to take those points
to Ma, Mb in the canonical view volume

Viewer is looking along the minus z axis with
y axis pointing up (right hand coordinate system)

Needs transform to take

$y = b \Rightarrow y = -1$
$y = t \Rightarrow y = +1$
$x = l \Rightarrow x = -1$
$x = r \Rightarrow x = 1$
$z = n \Rightarrow z = 1$
$z = f \Rightarrow z = -1$ | f is more negative than n (confusing - caused by looking along negative z axis)

What operations will do this for us?
move + scale ⇐ more intuitive
or
scale + move

$$\begin{bmatrix} x_{can} \\ y_{can} \\ z_{can} \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & 0 \\ 0 & \frac{2}{t-b} & 0 & 0 \\ 0 & 0 & \frac{2}{n-f} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -\frac{l+r}{2} \\ 0 & 1 & 0 & -\frac{b+t}{2} \\ 0 & 0 & 1 & -\frac{n+f}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Now have points in canonical view volume

add in generalization of previous equation ⇒ canonical view volume

$$\begin{bmatrix} n_x/2 & 0 & 0 & \frac{n_x-1}{2} \\ 0 & n_y/2 & 0 & \frac{n_y-1}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

do nothing to z, just carry it along

Combine 3 matrices to get

$$\begin{bmatrix} x_{pixel} \\ y_{pixel} \\ z_{canonical} \\ 1 \end{bmatrix} = M_o \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

z is now ∓ $[-1, 1]$

ignored for projection

used for z buffer.

compute $M_o$

for each line segment $(a_i, b_i)$

$p = M_o a_i$

$q = M_o b_i$

drawline$(x_p, y_p, x_q, y_q)$

Arbitrary View Positions

eye position $e$ (center of eye/lens)

gaze direction $g$

view-up vector $t$ (bisects camera/head + points up)

$$w = -\frac{g}{\|g\|}$$

$$u = \frac{t \times w}{\|t \times w\|}$$

$$v = w \times u$$

remember: cross product is 3D vector $\perp$ to ~~two~~ orig. vectors

$\|a \times b\| = \|a\|\|b\| \sin\phi$

what do we need to add to our pipeline?

a conversion from the coordinate system

$x, y, z$ @ $o$ $\Rightarrow$ $(u, v, w)$ @ $e$

book calls this canonical
we'll call it world

move $e \to o$ + align $uvw$ to $xyz$

$$M_v = \begin{bmatrix} x_u & y_u & z_u & 0 \\ x_v & y_v & z_v & 0 \\ x_w & y_w & z_w & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -x_e \\ 0 & 1 & 0 & -y_e \\ 0 & 0 & 1 & -z_e \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

compute $M_v$

compute $M_o$

$M = M_o M_v$

for each line segment $(a_i, b_i)$

$p = M a_i$

$q = M b_i$

drawline $(x_p, y_p, x_q, y_q)$

# Perspective Projection

Slide # 24 p. 166

would like to add another matrix to our chain

but how to handle the divide by z?

$$y_s = \frac{d}{z}\, y \; ?$$

trick: use that extra coordinate from last class ($w$ or $h$)

& let it take on values other than 1

$$\begin{bmatrix} hx \\ hy \\ hz \\ h \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad \text{by dividing by } h$$

perspective matrix:

$$M_p = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{n+f}{n} & -f \\ 0 & 0 & \frac{1}{n} & 0 \end{bmatrix}$$

points in $z = n$ plane are unchanged:

$\cancel{n\left(\frac{n+f}{n}\right) - nf = n^2}$

$\cancel{z' = z\,\frac{n+f}{n} - f \xrightarrow{\text{homogenize}} n+f -}$

$$M_p \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z\frac{n+f}{n} - f \\ \frac{z}{n} \end{bmatrix} \xrightarrow{\text{homogenize}} \begin{bmatrix} \frac{nx}{z} \\ \frac{ny}{z} \\ n+f-\frac{fn}{z} \\ 1 \end{bmatrix}$$

good properties:

pts on $z=n$ plane are unchanged

$n/z = 1$ $\quad$ $x, y, z$ unchanged

pts on $z=f$ plane

$z$ unchanged, $x, y$ squished appropriately

both $n$ + $z$ (inside the view volume) are negative so no flips in $x, y$

preserves relative $z$ values (can be used for z buffer)

map lines → lines

map planes → planes

$p \approx hp$ so we can multiply $M$ by an arbitrary constant (it will get divided out later)

$$M_p = \begin{bmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & n+f & -fn \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

more attractive version of $M_p$

$M_p^{-1}$ is also important → selection on screen. What is 2D mouse pointing to in 3D space

$$M_p^{-1} = \begin{bmatrix} \frac{1}{n} & 0 & 0 & 0 \\ 0 & \frac{1}{n} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{1}{fn} & \frac{n+f}{fn} \end{bmatrix}$$

or equivalently

$$M_p^{-1} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 0 & fn \\ 0 & 0 & -1 & n+f \end{bmatrix}$$

$$M = M_o M_p M_v$$

to canonical viewvolume

squish for perspective

take eye point to origin + align $uvw$ to $xyz$

compute $M_o$

compute $M_v$

compute $M_p$

$M = M_o M_p M_v$

for each line segment $(a_i, b_i)$ do

$p = M a_i$

$q = M b_i$

$\text{drawline}\left(\frac{x_p}{h_p}, \frac{y_p}{h_p}, \frac{x_q}{h_q}, \frac{y_q}{h_q}\right)$