Projective Transform
Practicalities in Projection
Camera Transform

VISIBILITY

Projection

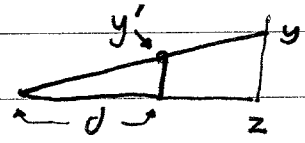
x', y' = where xyz projects to $H(P_{xyz1})$ $p = P_x$

$P_w = z$

$P_{xy} = dx\ dy$ $\quad x' = \frac{dx}{z}$ y'

$z' = ?$

$$\begin{bmatrix} d & & & \\ & d & & \\ ? & ? & ? & ? \\ & & 1 & \end{bmatrix}$$



$z' = z$? $\quad z' = \frac{P_z}{z}$ $\quad P_z = z^2$? no!

instead - preserve ordering

$P_z =$ [illegible] $\quad z' =$ [illegible] $1 - \frac{1}{z}$

$z - 1$

if $z \geq 1$

remember - non-linear "z" $\quad$ near

really, use n and f

$(n+f)z - fn$

$z' = n + f - \frac{fn}{z}$ $\quad$ notice $z' = z$ @ n, f

then transform to NDC

Specifying Projection

FoV vs. Focal Distance

gluPerspective (fovy, aspect, near, far)

glFrustrum (left, right, bottom, top, near, far)

near, far ← distances

↳ only need if assymetric projection

why near / far ?

Coordinate Systems

NDC ← Camera Coords ← World Coords ← Local / Object

projection — Camera transform — modeling positioning

How to specify camera

rotate, translate (rarely scale - projection takes care of that)

lookat — both does, eye, gaze vector, up vector

ℓ_{A-E}

gluLookAt eye xyz ← from

Center xyz at

up xyz (0,1,0)

just a translate / rotate

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

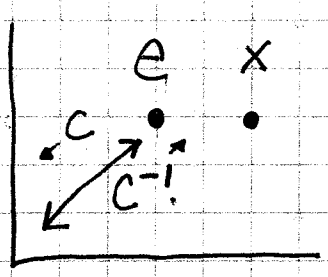
$u = \frac{t \times w}{\|t \times w\|}$

$v = w \times u$

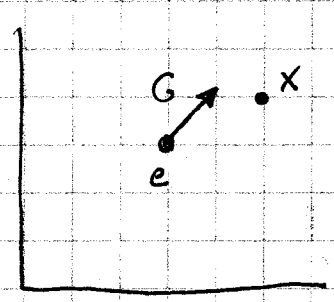
$$\begin{bmatrix} u & v & w & e \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

direction of transform

CAMERA ⇒ WORLD ⇒ CAMERA



$$R\,E^{-1}$$



$$G^{-1}\,E^{-1}$$