3D Projection

But first...

- Input vs. Output in OpenGL

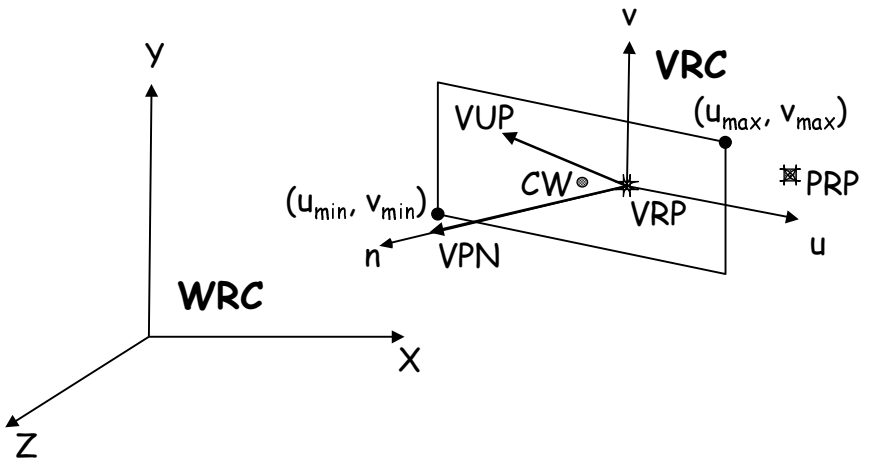
Arbitrary view reference point

- Two coordinate systems
  - World reference coordinate system (WRC)
  - Viewing reference coordinate system (VRC)

Arbitrary view reference point

- Specify viewplane, view coords (WRC)
  - View Reference Point (VRP)
  - View Plane Normal (VPN)
  - View Up Vector (VUP)
- Specify window on the view plane (VRC)
  - Max and min u,v values (window center (CW))
  - Projection Reference Point (PRP)

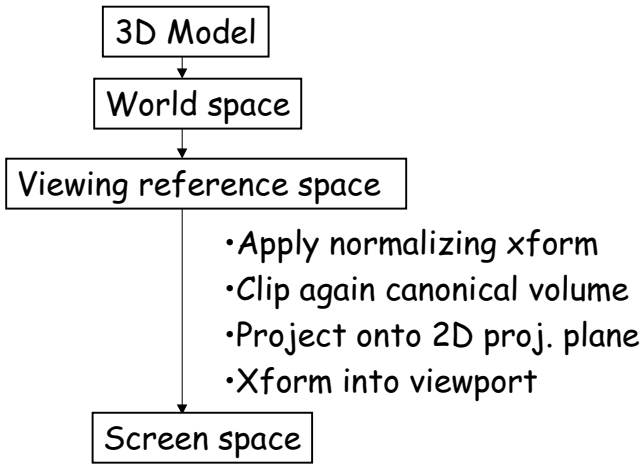
Specifying a view



The view volume

- Parallel
  - Infinite parallelpiped
- Perspective
  - Semi-infinite pyramid
- Limit it
  - Front and back clipping planes

3D viewing process, again



Canonical Volumes

- Make clipping easier
- Parallel
  - $x = -1, x = 1, y = -1, y = 1, z = 0, z = -1$
- Perpective
  - $y = z, y = -z, x = z, x = -z, z = -z_{min}, z = -1$

Normalizing Transformation for Perspective Views

1. Translate VRP to origin
2. Rotate the VRC system so that VPN become z-axis, u become x-axis and v become y-axis
3. Translate so that the CoP given by the PRP is at origin
4. Shear such that the center line of the view volume becomes the z-axis
5. Scale so that the view volume becomes the canonical view volume

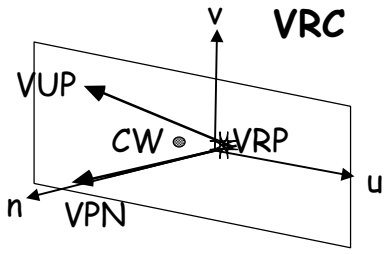
1. Translate VRP to origin

$$\begin{pmatrix} 1 & 0 & 0 & -VRP_x \\ 0 & 1 & 0 & -VRP_y \\ 0 & 0 & 1 & -VRP_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = \mathbf{T(-VRP)}$$

2. Rotate VRC

We want to take u into (1, 0, 0)
v into (0, 1, 0)
n into (0, 0, 1)

First derive n, u, and v from user input:



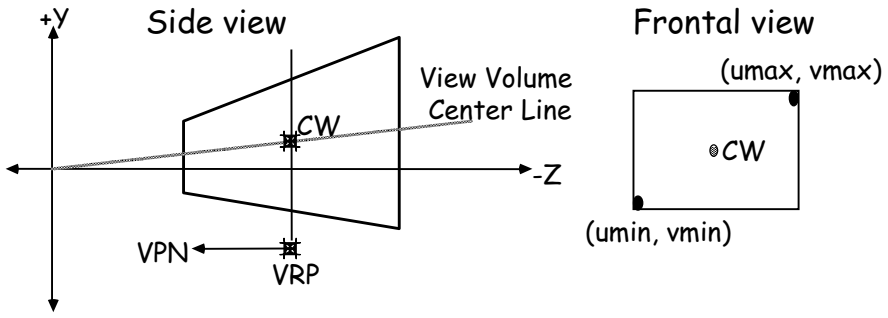
2. Rotate VRC (cont.)

$$\begin{pmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = R_{VRC}$$

3. Translate PRP to the origin

$$\begin{pmatrix} 1 & 0 & 0 & -PRP_u \\ 0 & 1 & 0 & -PRP_v \\ 0 & 0 & 1 & -PRP_n \\ 0 & 0 & 0 & 1 \end{pmatrix} = \mathbf{T(-PRP)}$$

4. Shear such that the center line of the view volume becomes the z-axis



Direction of projection (DoP) = CW - PRP

The center line of the view volume is DoP

Shear (cont.)

Multiply DoP with a matrix to get $(0,0,DoP_z)$

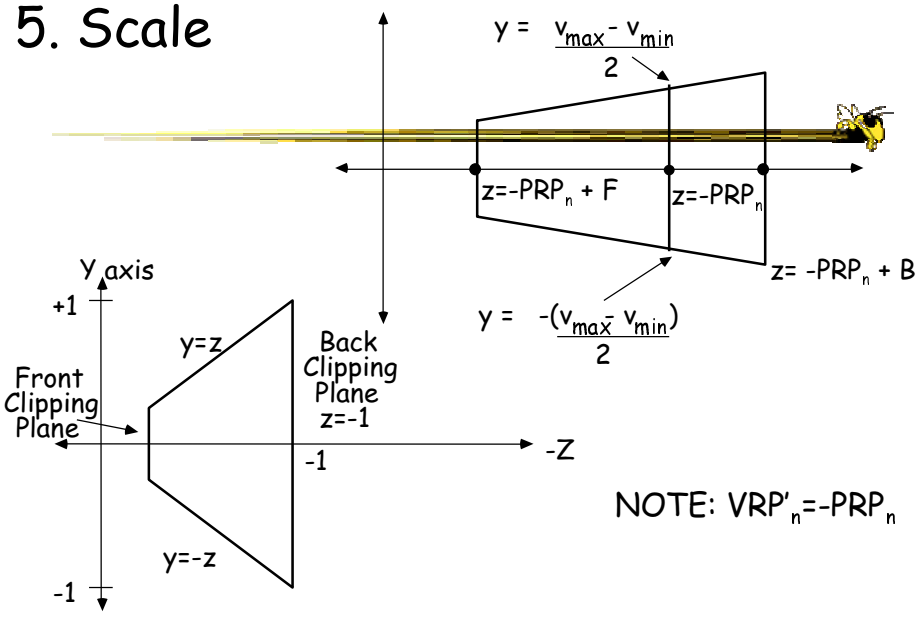
We want SH*DoP = $(0,0,DoP_z)$

$$SH = \begin{pmatrix} 1 & 0 & SHx & 0 \\ 0 & 1 & SHy & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

SHx =

SHy =

5. Scale



NOTE: $VRP'_n = -PRP_n$

5. Scale (cont.)

Scale is done in two steps:

1. First scale in x and y

   xscale = 2 PRP_n/(umax - umin)

   yscale = 2 PRP_n/(vmax - vmin)

2. Scale everything uniformly such that the back clipping plane becomes z =-1

   xscale = 1 / (-PRP_n + B)

   yscale = 1 / (-PRP_n + B)

   zscale = 1 / (-PRP_n + B)

Total Composite Transformation

N_{per} = S_{per} SH_{per} T(-PRP) R T(-VRP)

Use this to transform from the viewing to the world space, then project onto the viewplane.