CS 430/536
Computer Graphics I

3D Viewing Pipeline

Week 7, Lecture 13

David Breen, William Regli and Maxim Peysakhov
Geometric and Intelligent Computing Laboratory
Department of Computer Science
Drexel University
`http://gicl.cs.drexel.edu`

Overview

- Projection Mathematics
- Canonical View Volume
- Parallel Projection Pipeline
- Perspective Projection Pipeline

Lecture Credits: Most pictures are from Foley/VanDam; Additional and extensive thanks also goes to those credited on individual slides

Projection Mathematics

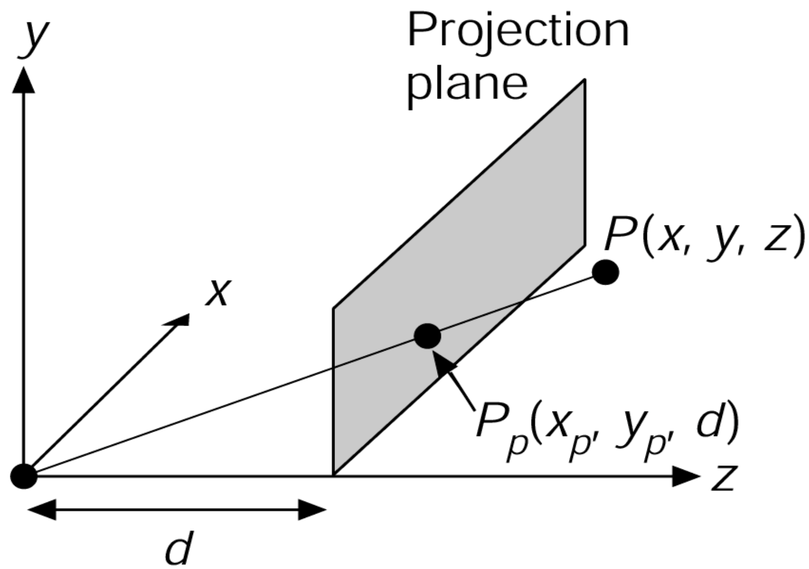
- What is the set of transformations needed to map 3D lines/planes onto a 2D screen positioned in 3D?
- Basic procedure
  - **4D homogeneous** coordinates *to*
  - **3D homogeneous** coordinates *for*
  - **every primitive** *in*
  - the **3D view volume**

Projection Mathematics

We present 4x4 matrices to be used for implementing projections

- Perspective Projections
  - How much to scale objects as a function of distance?
- Parallel Projections
  - (simplified case) Just chop out the z coordinate

The Perspective Projection



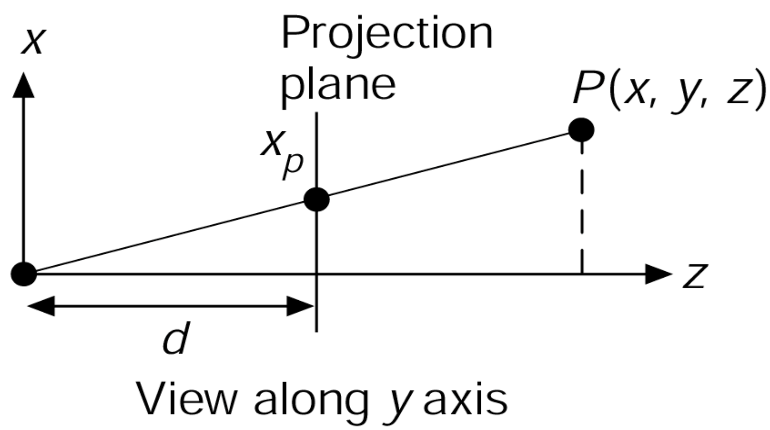
Determining scale

- Consider
  - Point P
  - Projected onto projection plane as point P_p
- Idea: compute ratios via similar triangles

The Perspective Projection

- In the x direction ratio is

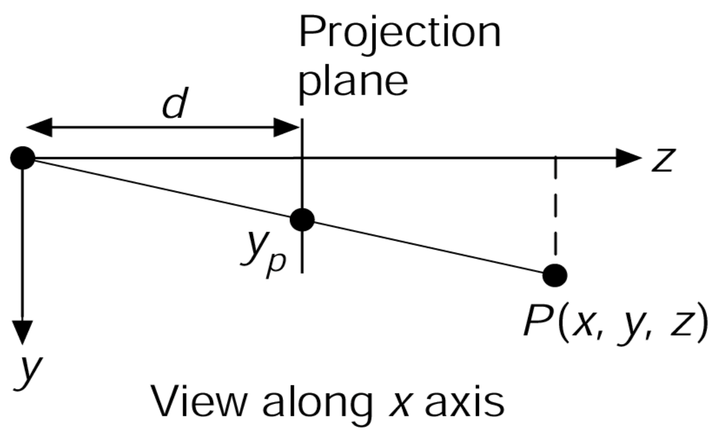
$$\frac{z}{d} = \frac{x}{x_p} \qquad x_p = \frac{x}{z/d}$$



The Perspective Projection

- In the y direction ratio is

$$\frac{z}{d} = \frac{y}{y_p} \qquad y_p = \frac{y}{z/d}$$



The Perspective Projection

- Homogenous perspective projection matrix

$$M_{\text{per}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix}$$

Assumes VPN is z axis.

The Perspective Projection

- Homogenous perspective projection

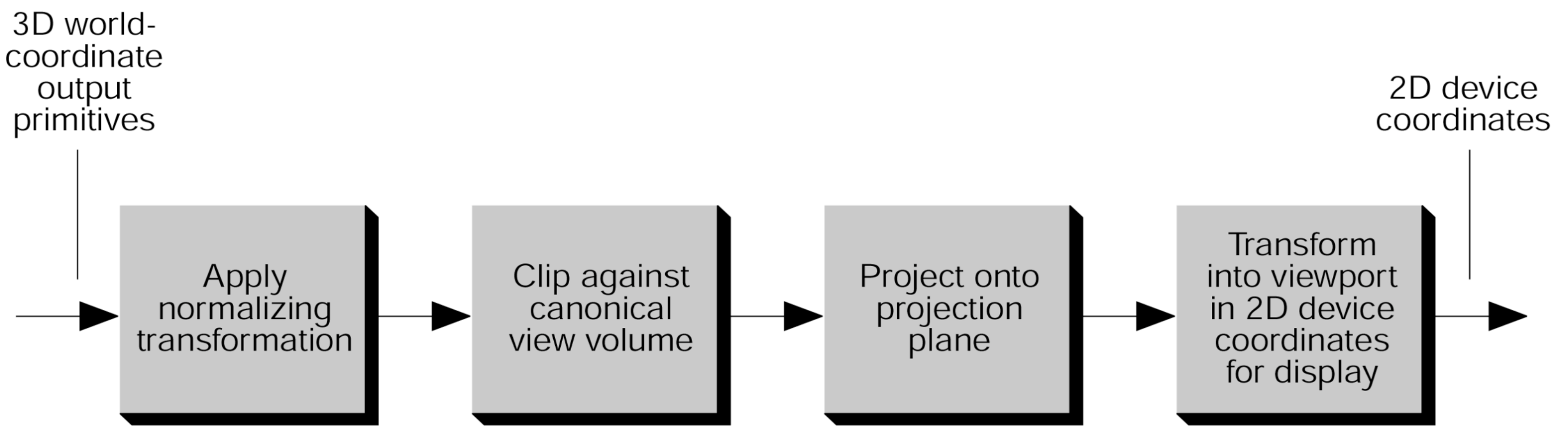
$$\begin{bmatrix} x \\ y \\ z \\ z/d \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix} \bullet \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

The Perspective Projection

- Homogenous perspective projection to 3D

$$\begin{bmatrix} x \\ y \\ z \\ z/d \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix} \bullet \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} \dfrac{x}{z/d} \\ \dfrac{y}{z/d} \\ d \end{bmatrix}$$

Implementing Projections (Foley et al.)

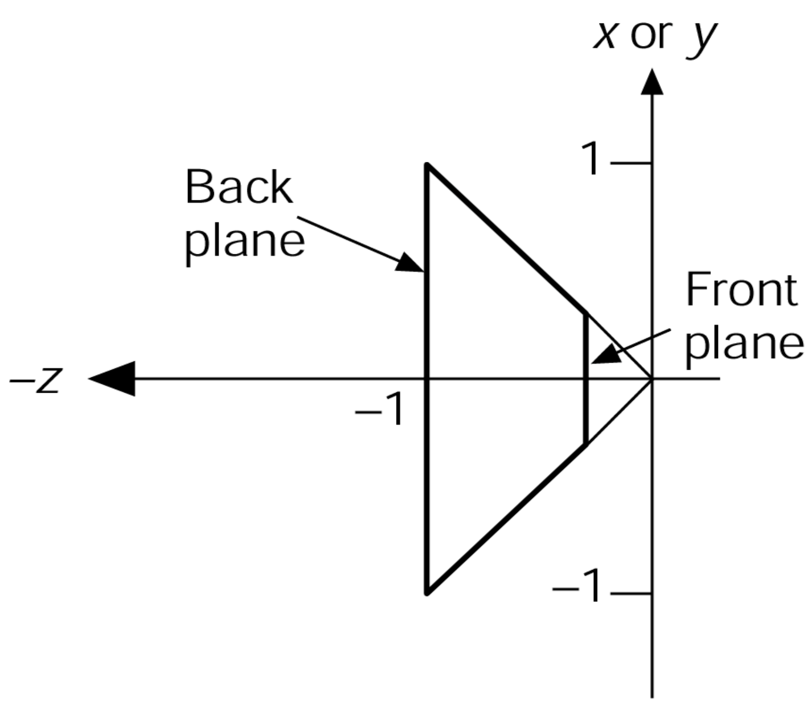


… a sequence of matrix operations and a clipping procedure…

Implementing Projections (Foley et al.)

1. Extend 3D coordinates to homogenous coords
2. Apply normalizing transformation, N_{par} or N_{per}
3. Divide by *W* to map back down to 3D
4. Clip in 3D against canonical view volume
   - parallel or perspective view volume
5. Extend 3D coordinates back to homogenous
6. Perform parallel projection using M_{ort} or
   Perform perspective projection M_{per}
7. Divide by *W* to map from homogenous to 2D coordinates (division effects perspective projection)
8. Translate and scale (in 2D) to device coordinates

Canonical View Volume: Perspective Projection



(b) Perspective

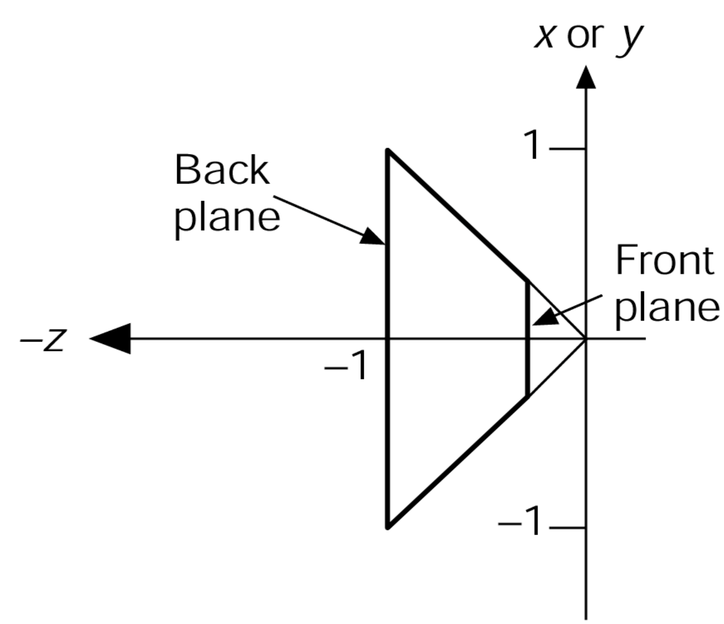
- Defined by 6 planes:
  - $x = z$
  - $x = -z$
  - $y = z$
  - $y = -z$
  - $z = z_{min}$
  - $z = -1$
- Easy to clip against

1994 Foley/VanDam/Finer/Huges/Phillips ICG

Perspective Projection Pipeline

Transforming an arbitrary view volume into the canonical one

1. Translate VRP to the origin
2. Rotate so VPN becomes z, VUP becomes y and u becomes x
3. Translate COP to origin
4. Shear so volume centerline becomes z axis
5. Scale into a canonical view volume for clipping



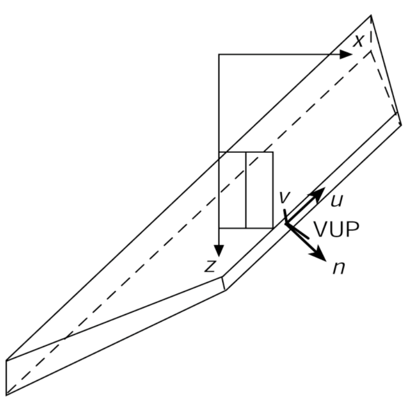
(b) Perspective

1. Translate VRP to the origin

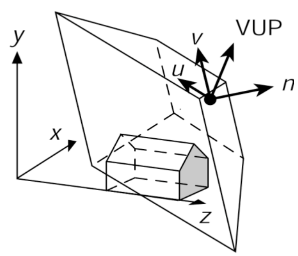
- Simple translation T(-VRP)

$$T = \begin{bmatrix} 1 & 0 & 0 & -vrp_x \\ 0 & 1 & 0 & -vrp_y \\ 0 & 0 & 1 & -vrp_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

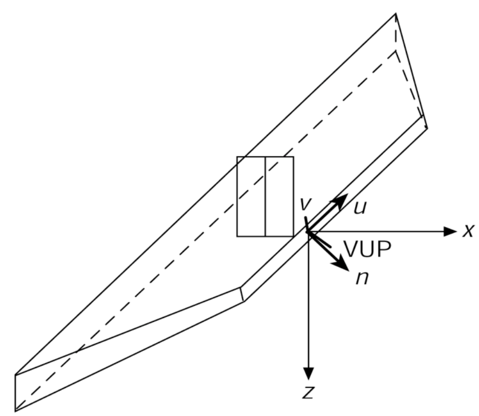
VRP, VPN & VUP in
World Coordinates (x,y,z)



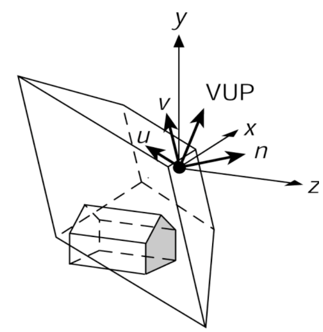
Top Projection



Off-Axis Projection



Top Projection

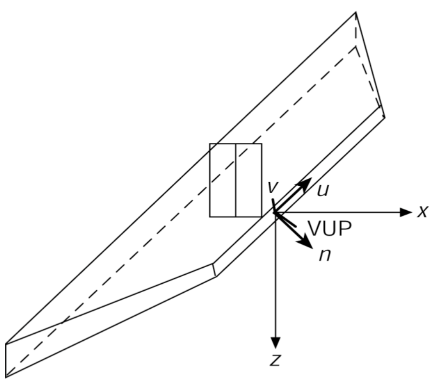


Off-Axis Projection

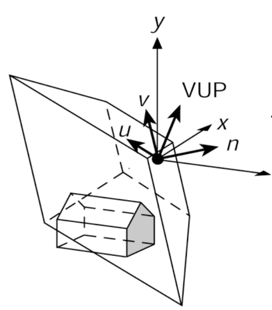
2. Rotate

- $R_z = \frac{VPN}{|VPN|}$ VPN rotated to z
- $R_x = \frac{VUP \times R_z}{|VUP \times R_z|}$ VUP rotated to y
- $R_y = R_z \times R_x$
- $R_x = [r_{1x}, r_{2x}, r_{3x}]$...

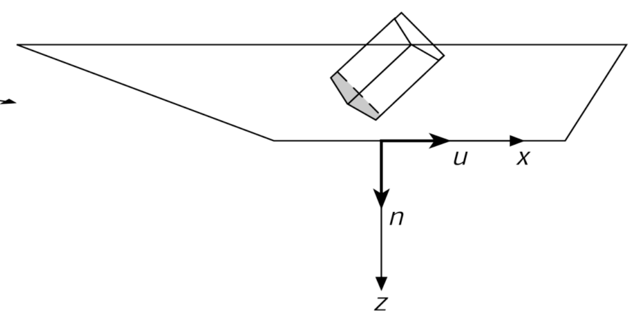
$$R = \begin{bmatrix} r_{1x} & r_{2x} & r_{3x} & 0 \\ r_{1y} & r_{2y} & r_{3y} & 0 \\ r_{1z} & r_{2z} & r_{3z} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



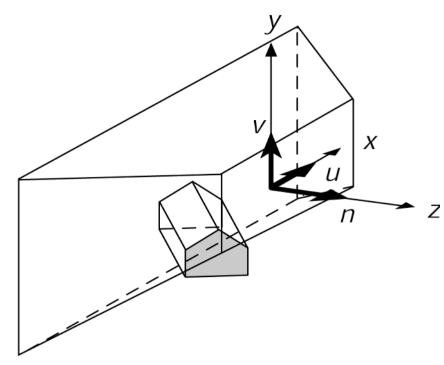
Top Projection



Off-Axis Projection



Top Projection



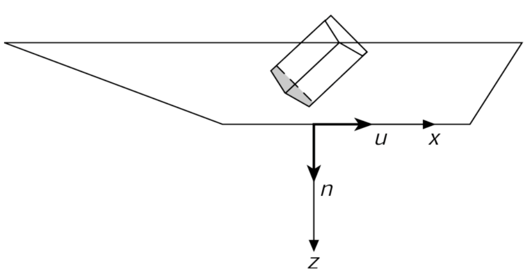
Off-Axis Projection

3. Translate

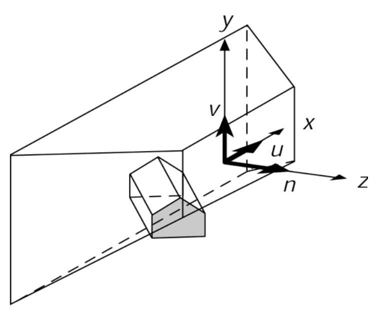
- Simple translation T(-PRP)

$$T = \begin{bmatrix} 1 & 0 & 0 & -prp_u \\ 0 & 1 & 0 & -prp_v \\ 0 & 0 & 1 & -prp_n \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

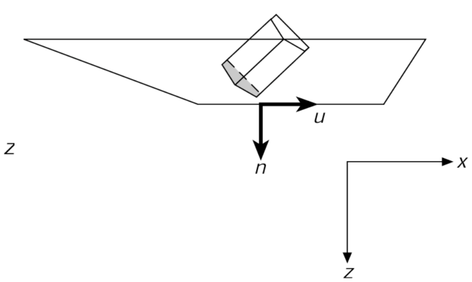
PRP in VRC
Coordinates (u,v,n)



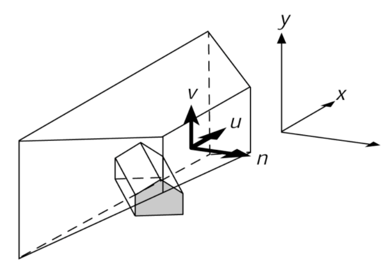
Top Projection



Off-Axis Projection



Top Projection



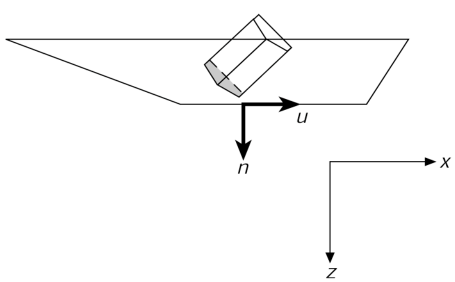
Off-Axis Projection

1994 Foley/VanDam/Finer/Huges/Phillips ICG

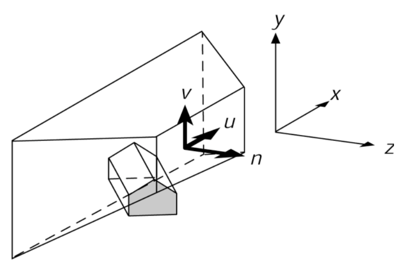
4. Shear

- Goal is to transform the center line to the z axis
- Same as parallel projection
- Shear matrix is the same!

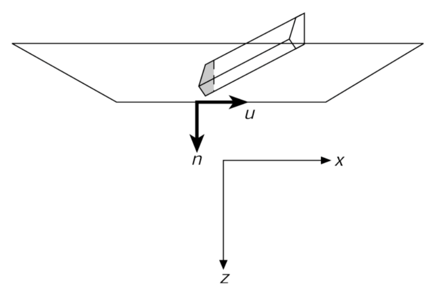
$$SH_{par} = \begin{bmatrix} 1 & 0 & \dfrac{1/2(u_{\max} + u_{\min}) - prp_u}{prp_n} & 0 \\ 0 & 1 & \dfrac{1/2(v_{\max} + v_{\min}) - prp_v}{prp_n} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



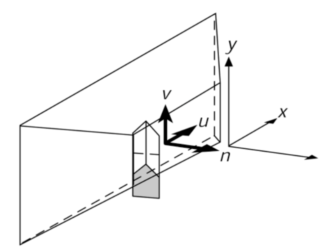
Top Projection



Off-Axis Projection



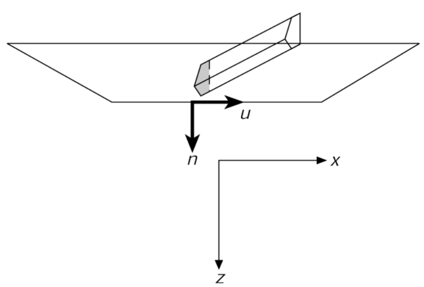
Top Projection



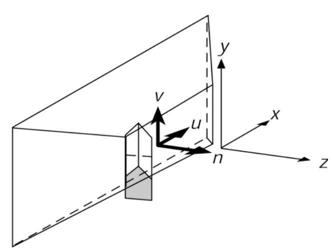
Off-Axis Projection

5. Scaling

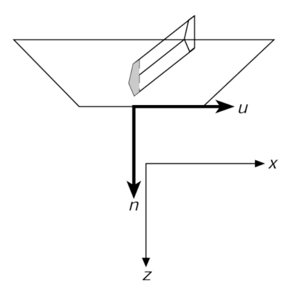
- We can define VRP after transformation as
  - $VRP' = SH_{par} \bullet T(-PRP) \bullet [0,0,0,1]^T$
- $vrp'_z = -prp_n$ since shear does not affect z coordinates
- First, we scale differentially in x and y to set plane slopes to 1 and -1 $\left[\dfrac{-2vrp_z'}{u_{\max} - u_{\min}},\ \dfrac{-2vrp_z'}{v_{\max} - v_{\min}},\ 1\right]$
- Second, we scale uniformly by $\dfrac{-1}{vrp_z' + B}$
  - Back clipping plane is $z = -1$, front clipping plane is $z = -\dfrac{vrp_z' + F}{vrp_z' + B}$
  - Projection plane is $d = z_{proj} = \dfrac{-vrp_z'}{vrp_z' + B}$



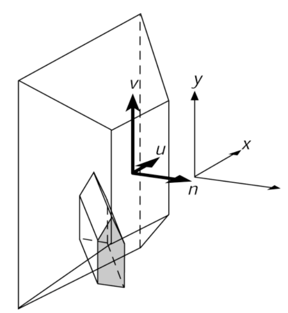
Top Projection



Off-Axis Projection



Top Projection

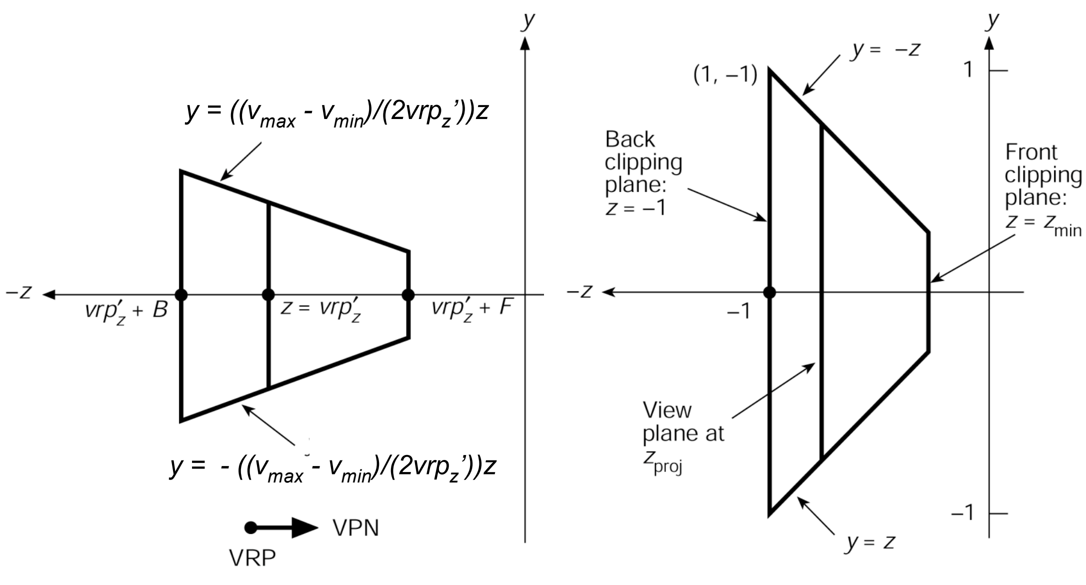


Off-Axis Projection

5. Scaling (*Cont*)

- We can combine the transformations

$$S_{per} = \begin{bmatrix} \frac{2vrp_z'}{(u_{\max}-u_{\min})(vrp_z'+B)} & 0 & 0 & 0 \\ 0 & \frac{2vrp_z'}{(v_{\max}-v_{\min})(vrp_z'+B)} & 0 & 0 \\ 0 & 0 & \frac{-1}{vrp_z'+B} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



(a) Before scaling

(b) After scaling

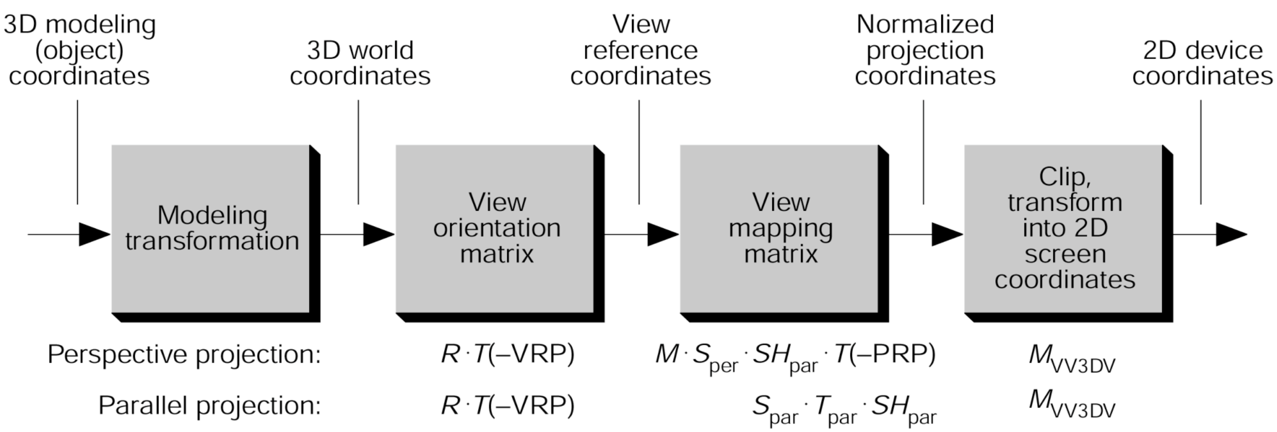
Figure is wrong in Foley et al.!

Perspective Projection Pipeline

$$N_{per} = (S_{per} \bullet (SH_{par} \bullet (T(-PRP) \bullet (R \bullet T(-VRP)))))$$

- Apply to all model vertices
- $P' = N_{per} P$

Summary of 3D Transforms



- We know how to take any projection and convert it into a canonical View Volume
- 3D edges can be clipped against it and projected onto screen