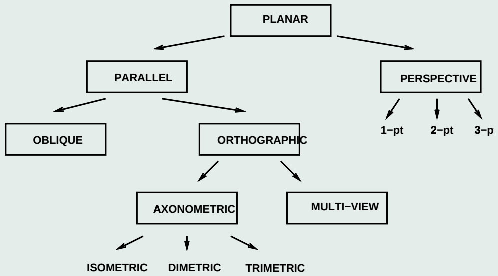
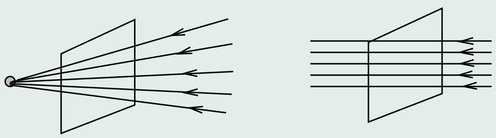
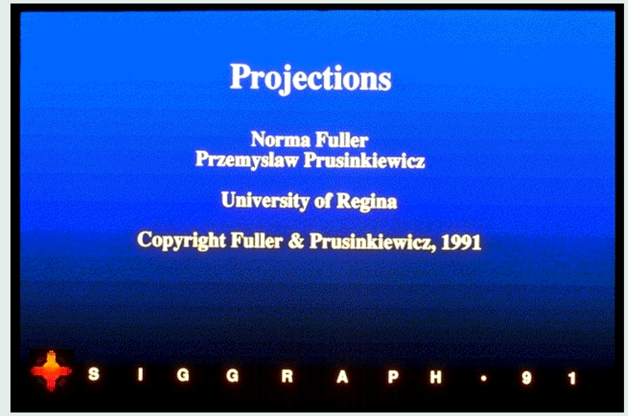
Geometric Projections

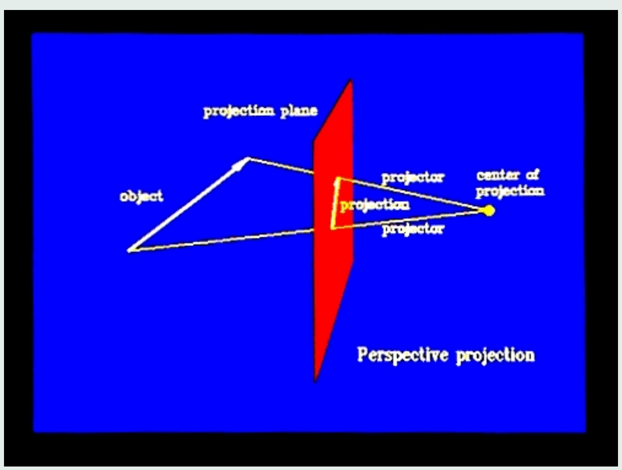


- **Parallel Projection:** projectors are parallel to each other.
- **Perspective Projection:** projectors converge to a point.



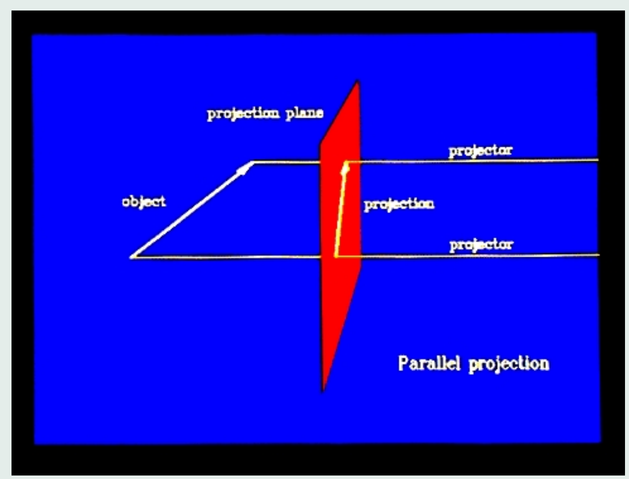


Perspective Projection



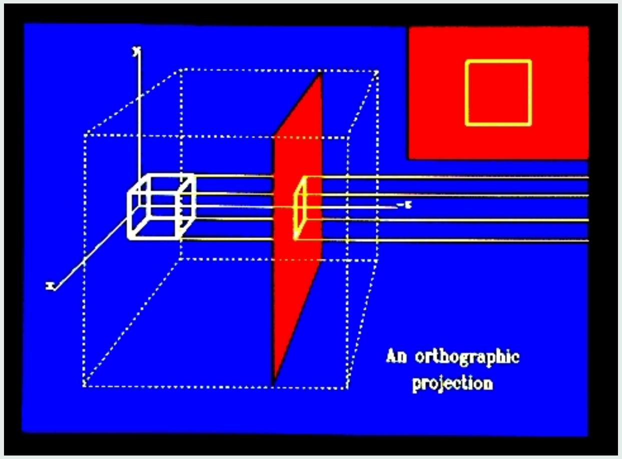
- Projectors converge to a point, the center of projection

Parallel Projection



- Projectors are parallel meeting at infinity

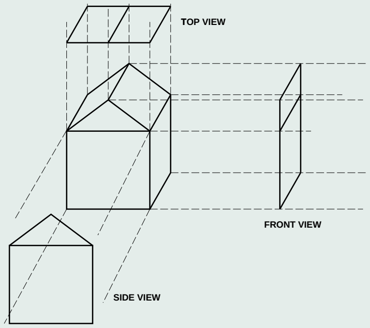
Orthographic Projection



- Projectors are orthogonal to the projection plane

Multi-View Orthographic Projection

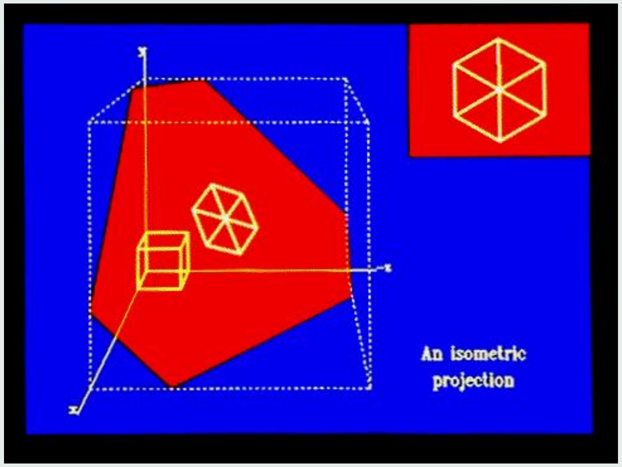
- projectors are also aligned with principal axes, common in architectural drawings (Top, Side and Front Views)



Axonometric Projections

- View plane Normal not aligned with principal axes.
- Useful in revealing 3D nature of objects, but does not preserve shape.
- Classified into Isometric, Dimetric, Trimetric projections
- Isometric projections are the most common.

Isometric Projections



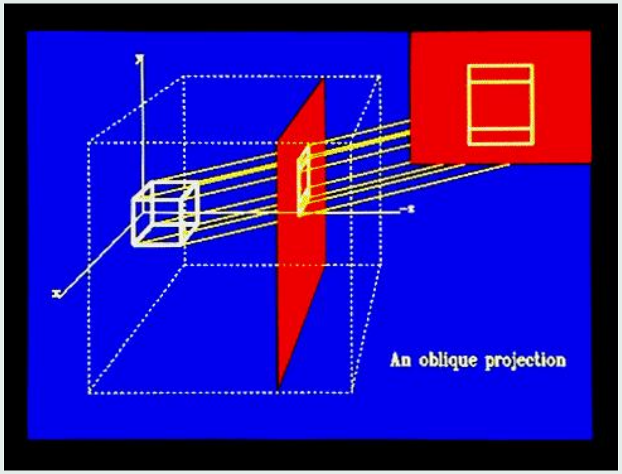
- View Plane Normal makes equal angles with **all** principal axes.

Oblique Projections

Combines the advantages of orthographic and axonometric projections.

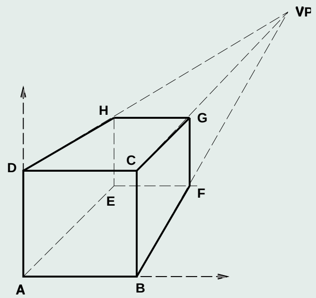
- Viewpoint is off the VPN axis.
- Line joining the viewpoint to the projection point is at an angle α.
- projectors are still orthogonal to the viewplane.
- Special Cases:
  - ⇒ **Cabinet:** $tan\alpha = 1.0$
  - ⇒ **Cavalier** $tan\alpha = 0.5$

Oblique Projections



Perspective Projections

- Classifed as 1-Point, 2-Point and 3-Point Perspective

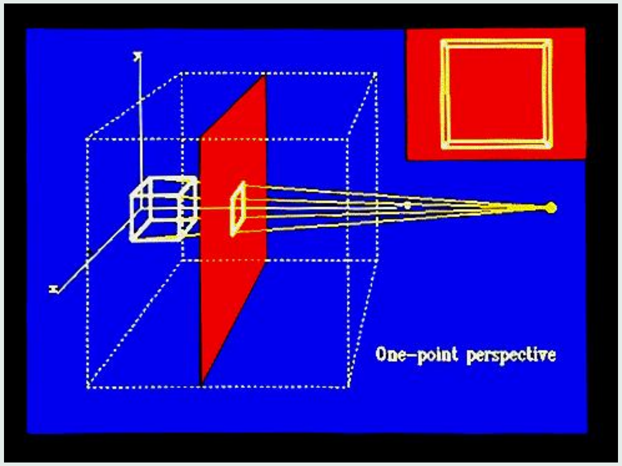


Vanishing Point

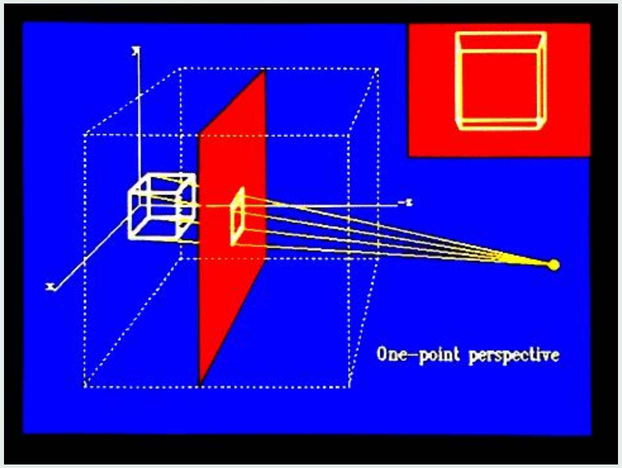
⇒ Lines and faces recede from a view and converge to a point, known as a vanishing point.

⇒ Parallel edges converge at infinity, eg. AD, BC.

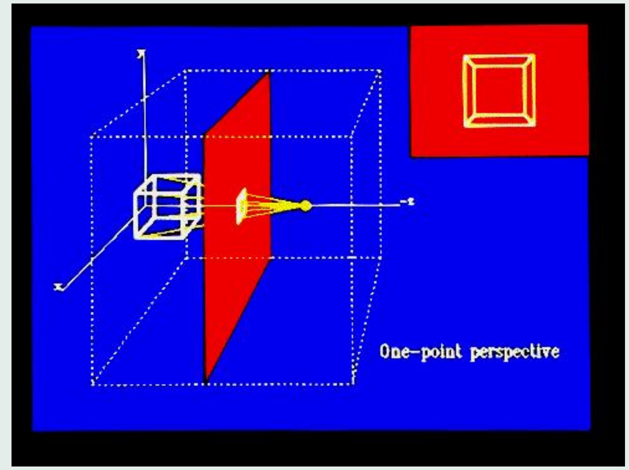
Perspective Projections: 1-Point Perspective



Perspective Projections: 1-Point Perspective



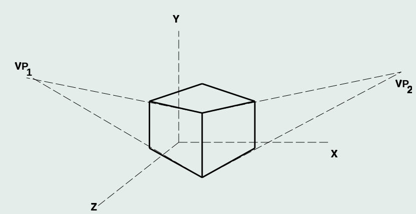
Perspective Projections: 1-Point Perspective

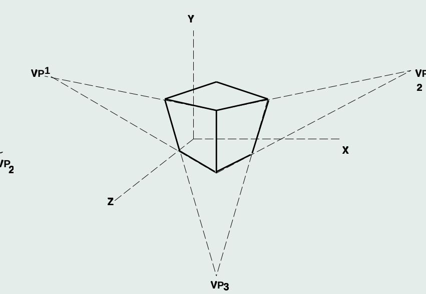


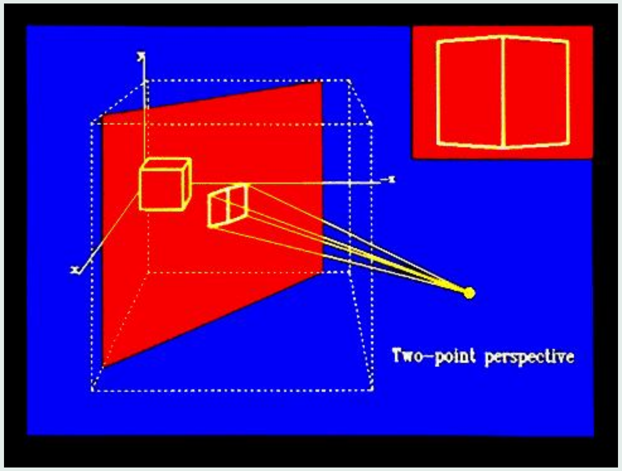
Perspective Projections: 1-Point Perspective

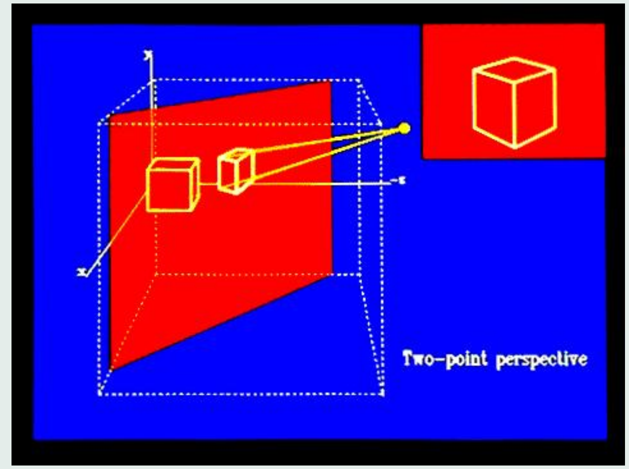
A painting (*The Piazza of St. Mark, Venice*) done by Canaletto in 1735-45 in one-point perspective

Perspective Projections: 2, 3 point Perspective



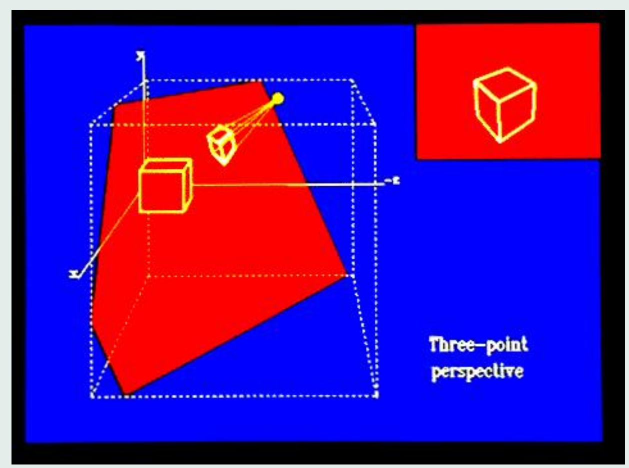


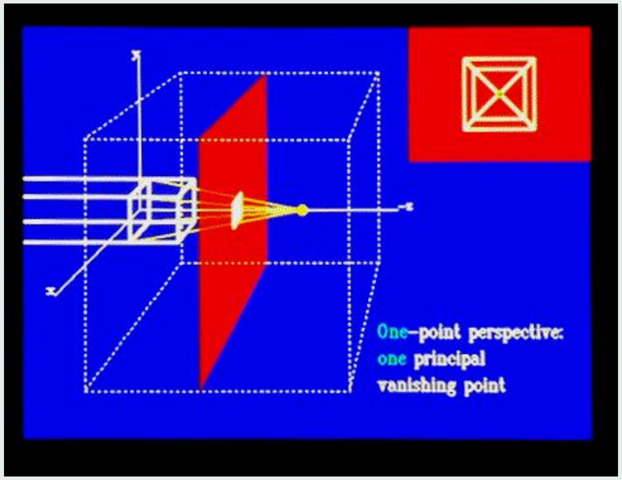


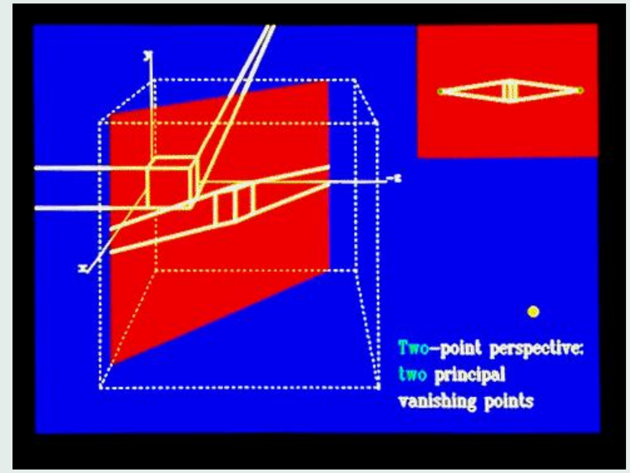


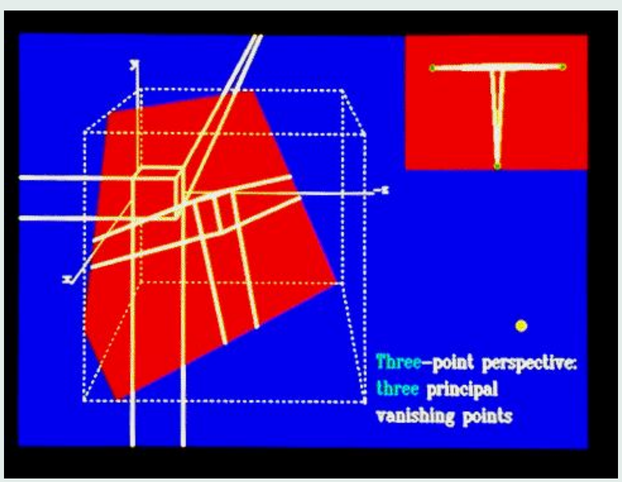
Perspective Projections: 2-Point Perspective

Painting in two point perspective by Edward Hopper *The Mansard Roof* 1923 (240 Kb); Watercolor on paper, 13 3/4 x 19 inches; The Brooklyn Museum, New York









Geometry of Perspective Projection

Problem:

To determine $\vec{P'}$, the projection of the $\vec{P}(p_u, p_v, p_n)$, a point on the object, given the view point $E(e_u, e_v, e_n)$, and $\vec{N}$, the view plane normal (VPN).

Simplification:

$\Rightarrow$ The View Plane contains the origin.

Geometry of Perspective Projection

$$p_n(t) = e_n + t(p_n - e_n)$$

$$p_n(t') = e_n + t'(p_n - e_n) = 0$$

Hence $t' = \frac{e_n}{e_n - p_n}$

Projection Point:

$$\begin{aligned} P'_u(t') &= e_u + t'(p_u - e_u) \\ &= e_u + \frac{e_n}{e_n - p_n}(p_u - e_u) = \frac{e_n p_u - e_u p_n}{e_n - p_n} \end{aligned}$$

Similarly,

$$P'_v(t') = \frac{e_n p_v - e_v p_n}{e_n - p_n}$$

Geometry of Perspective Projection

Simplification: $\vec{E}$ is on the $\vec{N}$ axis.

So

$$\vec{E} = (e_u, e_v, e_n) = (0, 0, e_n)$$

$$P' = \left(\frac{e_n p_u}{e_n - p_n}, \frac{e_n p_v}{e_n - p_n}\right)$$

$$= \left(\frac{p_u}{1 - p_n/e_n}, \frac{p_v}{1 - p_n/e_n}\right)$$

Thus P' projects to the 2D point

$$\boxed{(p'_u, p'_v) = \left(\frac{p_u}{1 - p_n/e_n}, \frac{p_v}{1 - p_n/e_n}\right)}$$

Perspective Projection Destroys Depth Information!

All points on the projector project to the same point $\vec{P'}$
Assume $\vec{a}$ to be any point on the projector.

$$P(t) = \vec{e} + t(\vec{a} - \vec{e})$$

$$\begin{aligned} P'_u(t) &= \frac{p_u}{1 - p_n/e_n} \; (e_u = e_v = 0) \\ &= \frac{e_u + t(a_u - e_u)}{1 - (e_n + t(a_n - e_n))/e_n} \\ &= \frac{a_u}{1 - a_n/e_n} \end{aligned}$$

Similarly,

$$P'_v(t) = \frac{a_v}{1 - a_n/e_n}$$

Pseudo Depth

"Need to retain depth information after projection for use in shading and visible surface calculations."

Define ${p_n}' = \frac{p_n}{1 - p_n/e_n}$, as a measure of P's depth.

As p_n increases, ${p_n}'$ increases.

Hence

$$({p_u}', {p_v}', {p_n}') = \left(\frac{p_u}{1 - p_n/e_n}, \frac{p_v}{1 - p_n/e_n}, \frac{p_n}{1 - p_n/e_n}\right)$$

In Homogeneous coordinates,

$$\begin{aligned} P' &= (p_u, p_v, p_n, 1 - p_n/e_n) \\ &= \vec{M}_p \begin{bmatrix} p_u & p_v & p_n & 1 \end{bmatrix}^T \end{aligned}$$

Perspective Projection

$$M_p = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1/e_n & 1 \end{bmatrix}$$

which is the perspective projection transform.

Perspective Projection: General Case

If $\vec{E}$ is off the $\vec{N}$ axis

$$P' = M_p M_s \vec{P}$$

and,

$$M_s = \begin{bmatrix} 1 & 0 & -e_u/e_n & 0 \\ 0 & 1 & -e_v/e_n & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$\Rightarrow$ Moving $\vec{E}$ off the $\vec{N}$ axis introduces a **shear**, given by M_s

View Space Operations

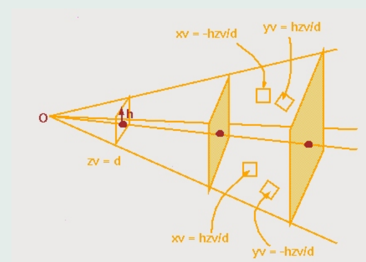
Back Face Elimination

Compare the orientation of the polygon with the view point (or center of projection). Define

$$\begin{aligned} \vec{N}_p &= \text{polygon normal} \\ \vec{N} &= \text{vector to view point} \end{aligned}$$

$$visibility = \vec{N}_p \bullet \vec{N} > 0$$

View Volume



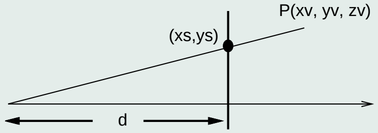
$$x_v = \pm \frac{h z_v}{d}$$
$$y_v = \pm \frac{h z_v}{d}$$

Note

Clipping is more efficiently carried out in screen coordinates.

3D Screen Space

Perspective Projection



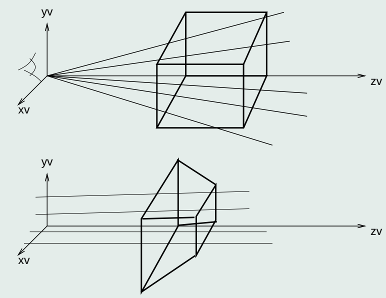
$$x_s = dx_v/z_v, y_s = dy_v/z_v$$

In homogeneous coordinates,

$$X = x_v, Y = y_v, Z = z_v, w = z_v/d$$

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

Perspective Transformation



$$x_s = \frac{dx_v}{hz_v}$$

$$y_s = \frac{dy_v}{hz_v}$$

$$z_s = \frac{f(1 - d/z_v)}{f - d}$$

Perspective Transformation

In homogeneous coordinates,

$$X = (d/h)x_v, Y = (d/h)y_v, Z = \frac{fz_v}{f-d} - \frac{df}{f-d}, w = z_v$$

and

$$M_{persp} = \begin{bmatrix} d/h & 0 & 0 & 0 \\ 0 & d/h & 0 & 0 \\ 0 & 0 & f/(f-d) & -df/(f-d) \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Perspective Transformation

$$M_{persp} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & f/(f-d) & -df/(f-d) \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} d/h & 0 & 0 & 0 \\ 0 & d/h & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

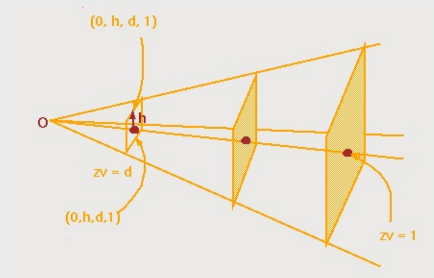
$$= M_{persp2} M_{persp1}$$

Note:

⇒ M_{persp1} converts a truncated pyramid to a regular pyramid (unit slopes for the side planes)

⇒ M_{persp2} maps the regular pyramid into a box.

Perspective Transformation



Consider the following points:

⇒ $(0, h, d, 1) \longrightarrow (0, d, d, 1) \longrightarrow (0, d, 0, d)$

⇒ $(0, -h, d, 1) \longrightarrow$??

⇒ $(0, fh/d, f, 1)$??

⇒ $(0, -fh/d, f, 1) \longrightarrow$??

PHIGS Viewing System

- A View Reference Coordinate (VRC) system.
- View Reference Point (VRP) - origin of VRC
- Projection Reference Point (PRP) - center of projection
- Allows oblique projections (center of proj. to center of view plane is not parallel to View Plane Normal (VPN)
- Near, Far clipping planes and projection plane defined.
- Projection window of any aspect ratio and located anywhere in the view plane.

View Orientation

- VRP - point in World coords.
- VPN - vector in World coords.
- View Up (VUV) - vector in world coords.

$$\begin{aligned}\vec{U} &= V\vec{U}V \times V\vec{P}N \\ \vec{V} &= V\vec{P}N \times \vec{U}\end{aligned}$$

View Mapping

- Projection type (parallel or perspective)
- PRP - point in VRC space
- View plane distance - in VRC space
- Front and Back plane distances
- View plane window size (4 limits)
- Projection viewport limits in normalized projection space (4 limits)

Normalizing Transformation

- Translate PRP to the origin.

After translation,

- View plane distance = d
- Near plane distance = n
- Far plane distance = f
- Window parameters,
  - $\Rightarrow$ u_{max}, u_{min} become x_{max}, x_{min}
  - $\Rightarrow$ v_{max}, v_{min} become y_{max}, y_{min}

Shear Projection Window Center

Shear the center of the projection window such that the center line becomes the z_v axis.

$$M_{sh} = \begin{bmatrix} 1 & 0 & \frac{x_{max}+x_{min}}{2d} & 0 \\ 0 & 1 & \frac{y_{max}+y_{min}}{2d} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Scale into symmetric (unit slope) view volume

$$M_s = \begin{bmatrix} \frac{2d}{x_{max}-x_{min}} & 0 & 0 & 0 \\ 0 & \frac{2d}{y_{max}-y_{min}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

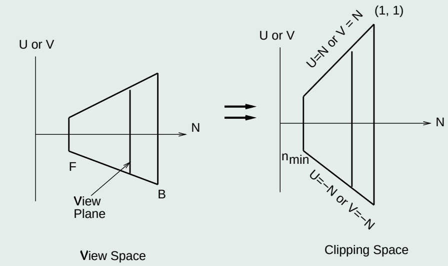
Perspective Transformation

$$M_persp = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & f/(f-n) & -fn/(f-n) \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

Normalizing Transformation - Properties

- Planes are preserved.
- Planes parallel to the view plane are shifted.
- The 4 sidewalls of the view volume become 4 faces of a parallelepiped, view point is at infinity.
- Also known as **pre-warping**.
- Useful for visible surface calculations.

Clipping in 3D



Clip boundaries are as follows:

$$-g_n \leq g_u \leq g_n$$
$$-g_n \leq g_v \leq g_n$$
$$n_{min} \leq g_n \leq 1$$

Use the Cohen-Sutherland Algorithm with these boundaries defining the clip volume.

Cohen-Sutherland Clipping Algorithm

- Extension of the 2D algorithm to 3D.
- Note that the scaling to unit slope planes facilitates efficient clipping in 3D.

⇒ $g_u > -g_n$ - Point to right of volume

⇒ $g_u < g_n$ - Point to left of volume

⇒ $g_v > -g_n$ - Point above volume

⇒ $g_v < g_n$ - Point below volume

⇒ $g_n < n_{min}$ - Point in front of volume

⇒ $g_n > 1$ - Point behind volume

3-D Clipping in Homogeneous Coordinates

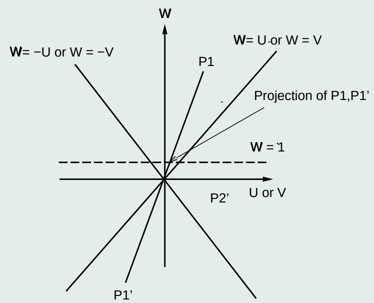
$$-1 \leq \frac{g_u}{g_w} \leq 1, -1 \leq \frac{g_v}{g_w} \leq 1, 0 \leq \frac{g_n}{g_w} \leq 1$$

OR

$$-g_w \leq g_u \leq g_w, -g_w \leq g_v \leq g_w, 0 \leq g_n \leq g_w$$

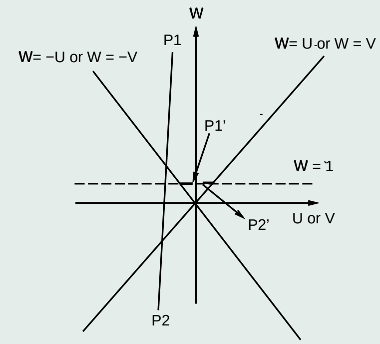
Equalities reverse if $g_w \leq 0.0$

Clipping Points and Lines



- If g_w component is **negative**, multiply through by -1, then clip.
- Same procedure when both endpoints of a line have negative g_w
- When endpoints of a line have **opposite** g_w, what do we do??

Clipping Points and Lines



- $P1$ and $P2$ are on opposite side of $W = 0$ plane.
- The projection of P_1P_2 has 2 parts; one segment extends to $-\infty$, the other to $+\infty$.
- Solution: Clip twice, once with each region.
- Better, Clip with top region, negate end points, and clip again with top region - Need to have only 1 clip region.