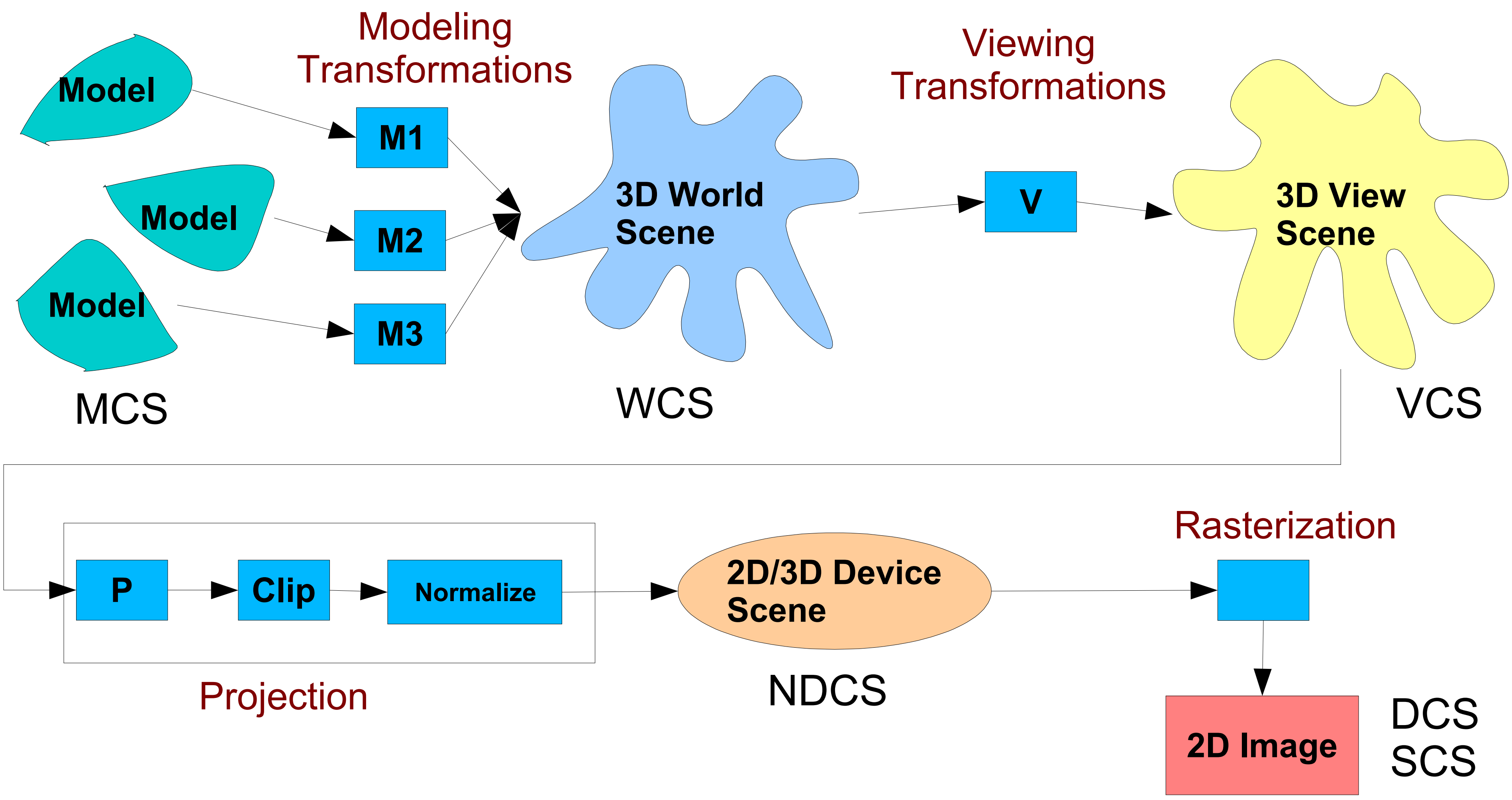
3 DIMENSIONAL VIEWING

Ceng 477 Computer Engineering
Lecture notes
METU

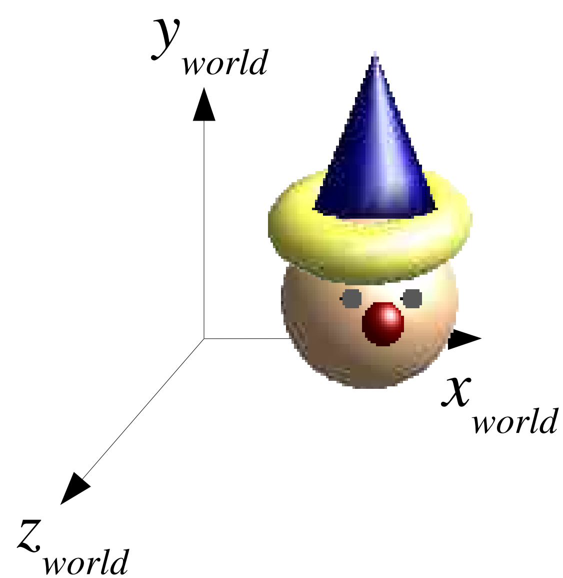
3D Viewing Pipeline, Revisited



- Model coordinates to World coordinates: Modelling transformations

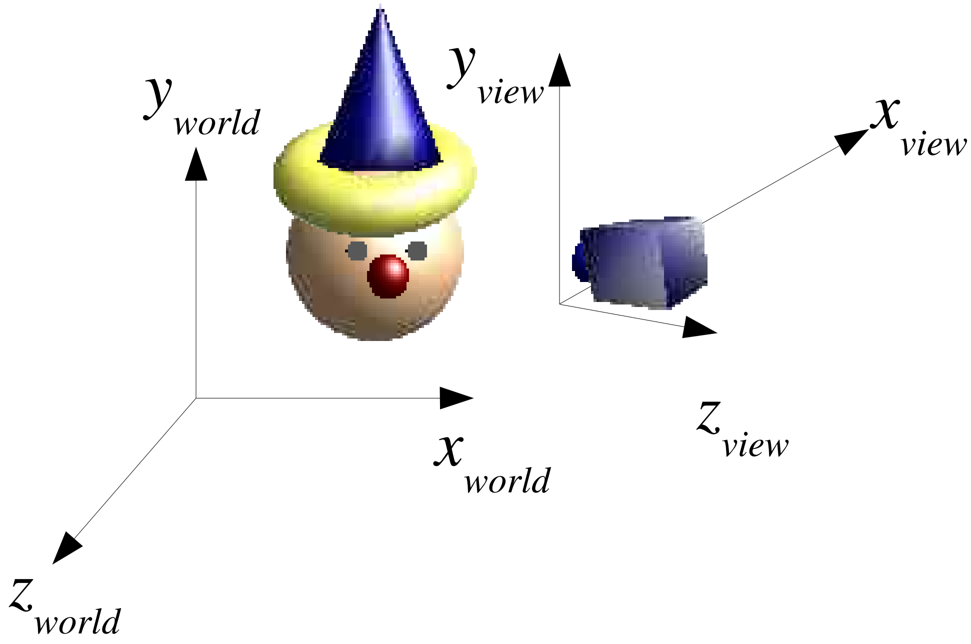
Model coordinates:
1 sphere (head),
2 semispheres (eyes),
1 semisphere (nose),
1 torus (hat),
1 cone (hat).
With their relative coordinates and sizes

World coordinates:
All shapes with their absolute coordinates and sizes.



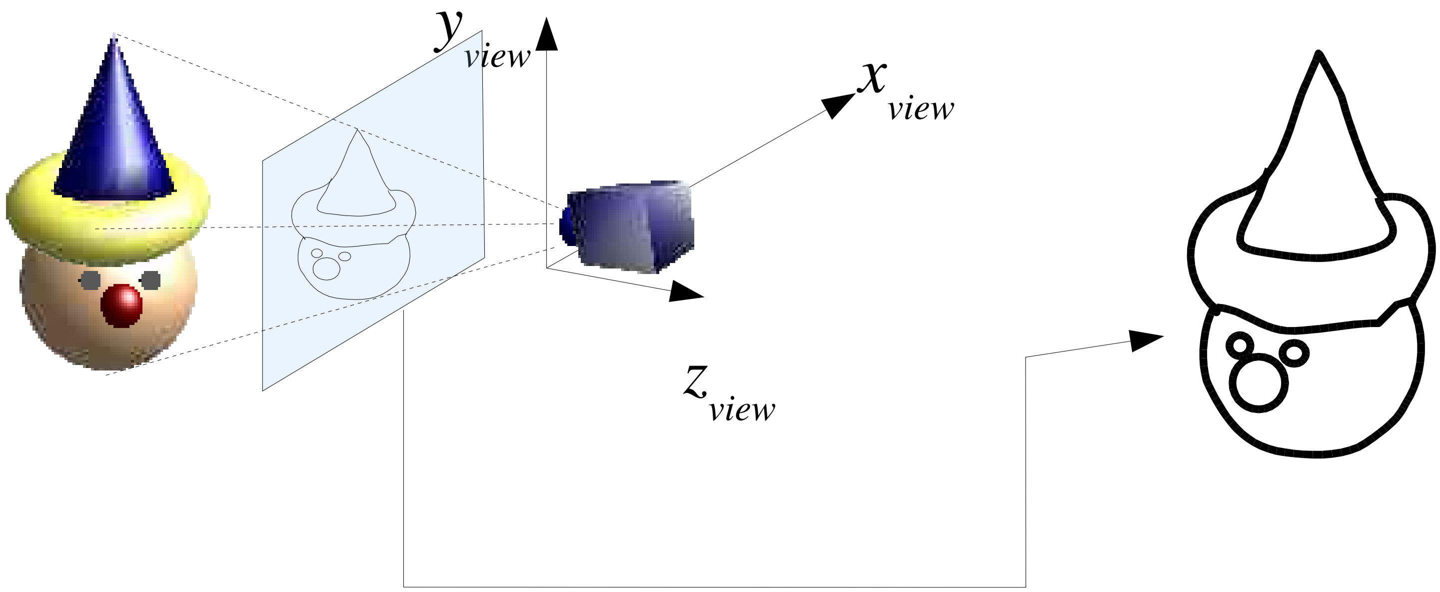
- World coordinates to Viewing coordinates: Viewing transformations

World coordinates

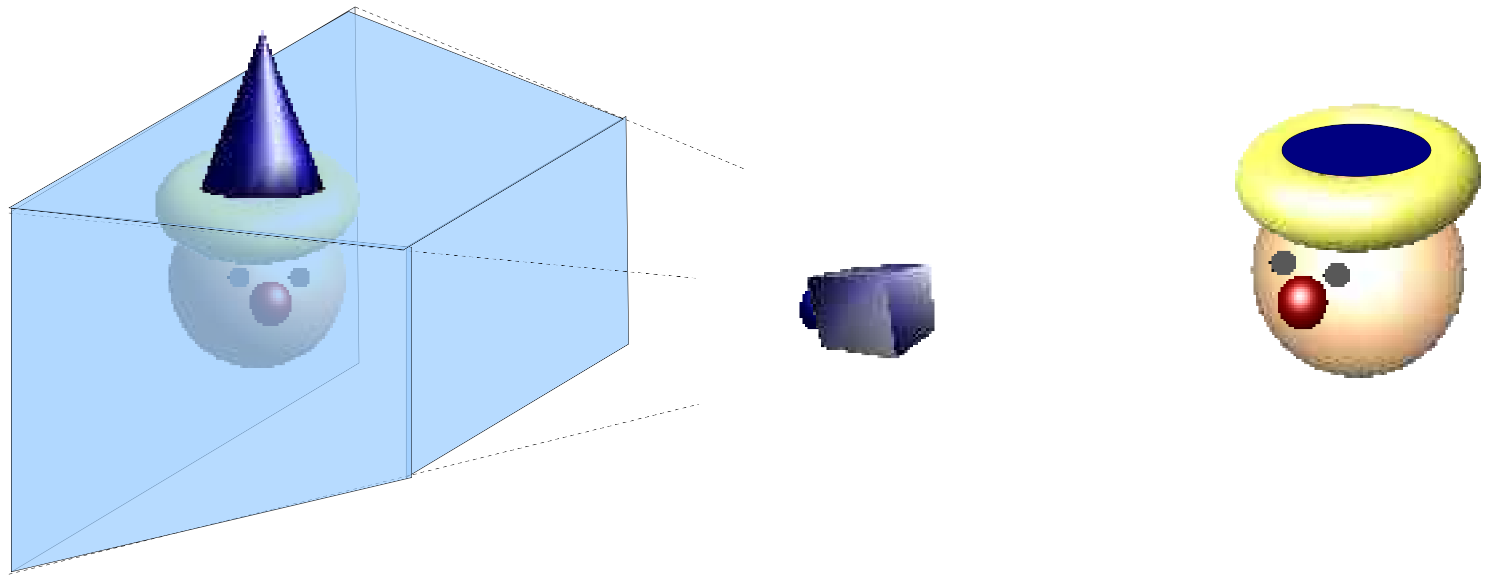


Viewing coordinates:
Viewers (Camera) position and view angles. i.e. rotated/translated

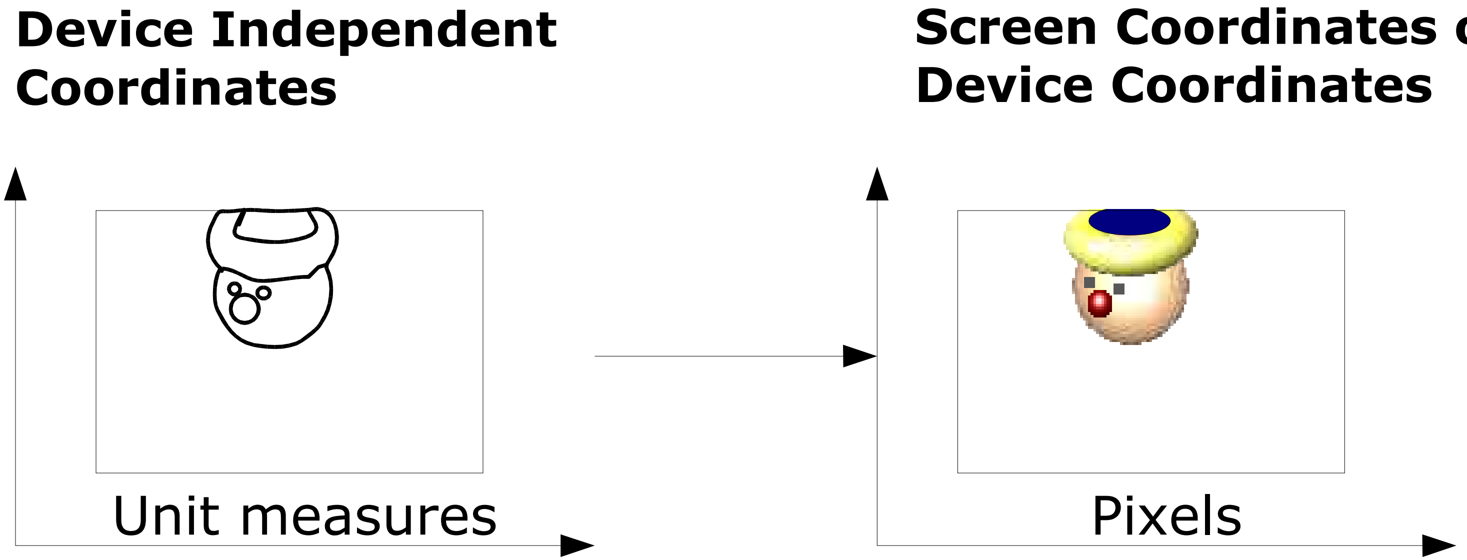
- Projection: 3D to 2D. Perspective or parallel. Depth cue. Visible surface detection.



- Clipping: Find parts of the object in the viewing area. Project, then clip in 2D vs. clip against a view volume, then project.

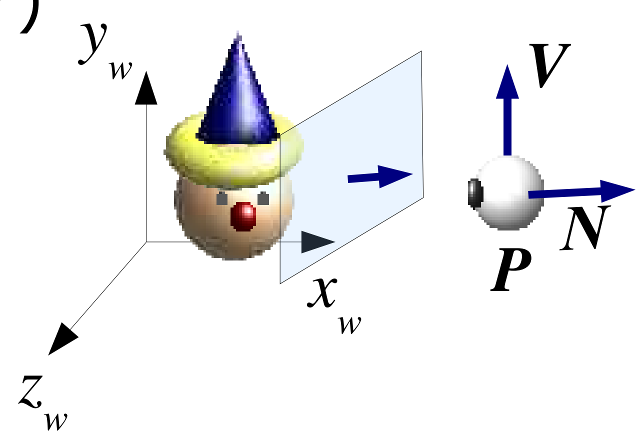


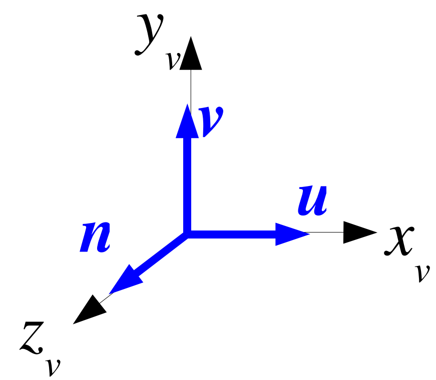
- Device Independent Coordinates to Device Coordinates. Rasterization. Illumination, shading.



Viewing Transformation

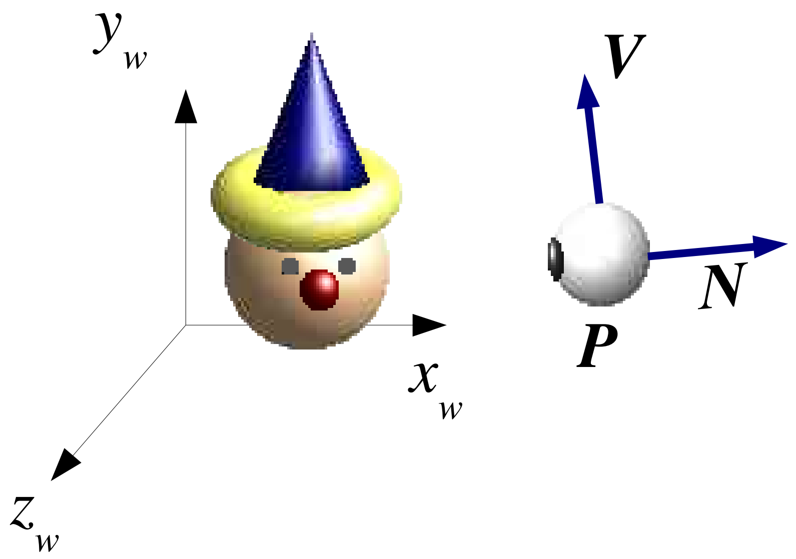
- How to define the viewing coordinate system (or view reference coordinate system):
  - Position of the viewer ***P*** (enough?)
  - Orientation of the viewer:
    - Viewing direction. ***N*** (view plane normal)
    - View up vector. ***V***
    - ***N*** and ***V*** should be orthogonal. (if it is not?)





Transformation Between Coordinate Systems

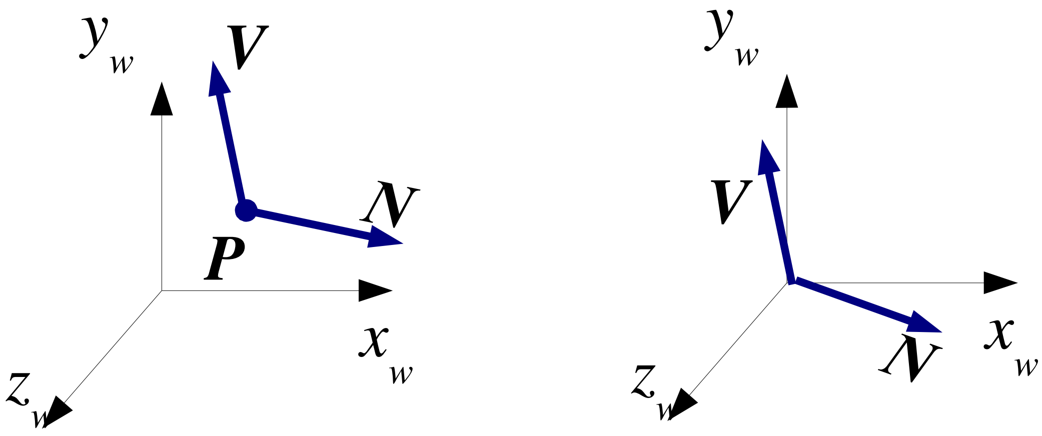
- Given the objects in world coordinates, find the transformation matrix to transform them in view coordinates. ***n, v, u*** : unit vectors defining view coordinates.



- Translation: Move view reference point to origin.

$$\boldsymbol{P} = (x_{0,} y_{0,} z_0)$$

$$\boldsymbol{T} = \begin{bmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



- Rotation: find orthogonal base vectors for the transformation, ***n***, ***v*** and ***u***.

Align $\boldsymbol{n}$ with z_w; $\boldsymbol{v}$ with y_w; $\boldsymbol{u}$ with x_w

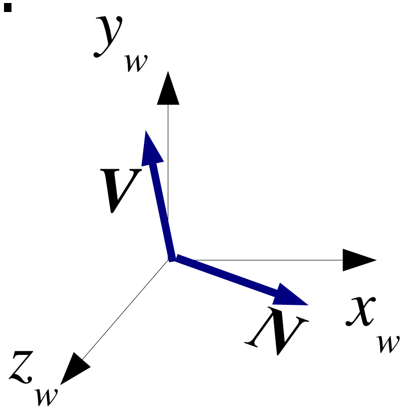
$$\boldsymbol{n} = \frac{\boldsymbol{N}}{|\boldsymbol{N}|}$$

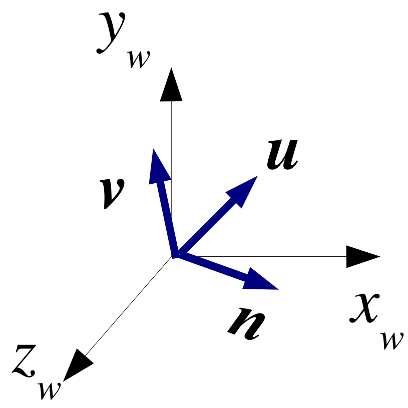
$$\boldsymbol{u} = \frac{\boldsymbol{V} \times \boldsymbol{N}}{|\boldsymbol{V} \times \boldsymbol{N}|}$$

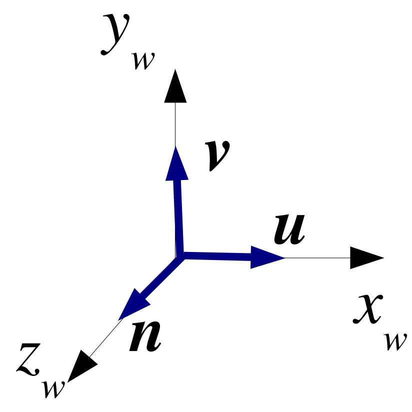
$$\boldsymbol{v} = \boldsymbol{n} \times \boldsymbol{u}$$

$$\boldsymbol{R} = \begin{bmatrix} u_1 & u_2 & u_3 & 0 \\ v_1 & v_2 & v_3 & 0 \\ n_1 & n_2 & n_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{M}_{WC,VC} = \boldsymbol{R} \cdot \boldsymbol{T}$$

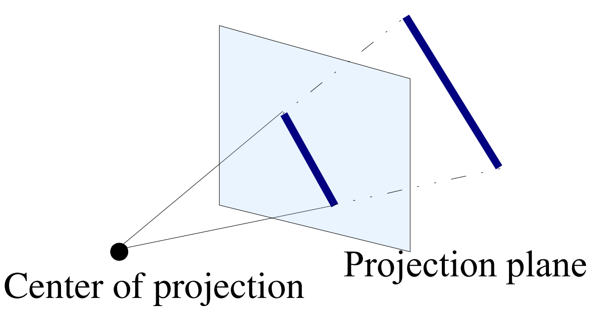




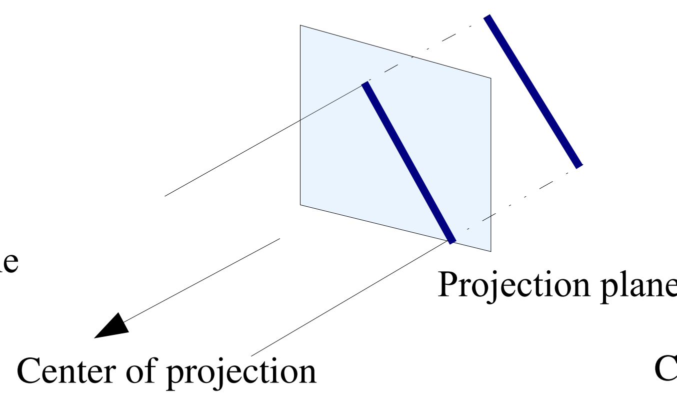


Projections

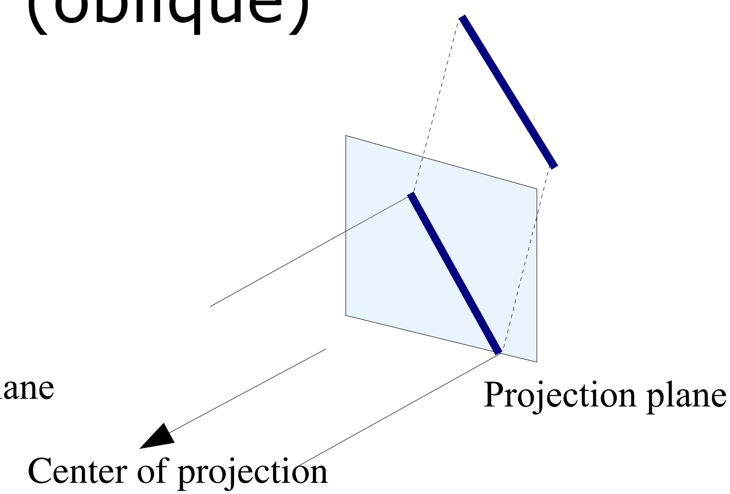
- Classification of projections. Based on:
  - Center of projection: infinity (parallel) or a point (perspective)
  - Projection lines wrt. projection plane: orthogonal (orthographic), another angle (oblique)



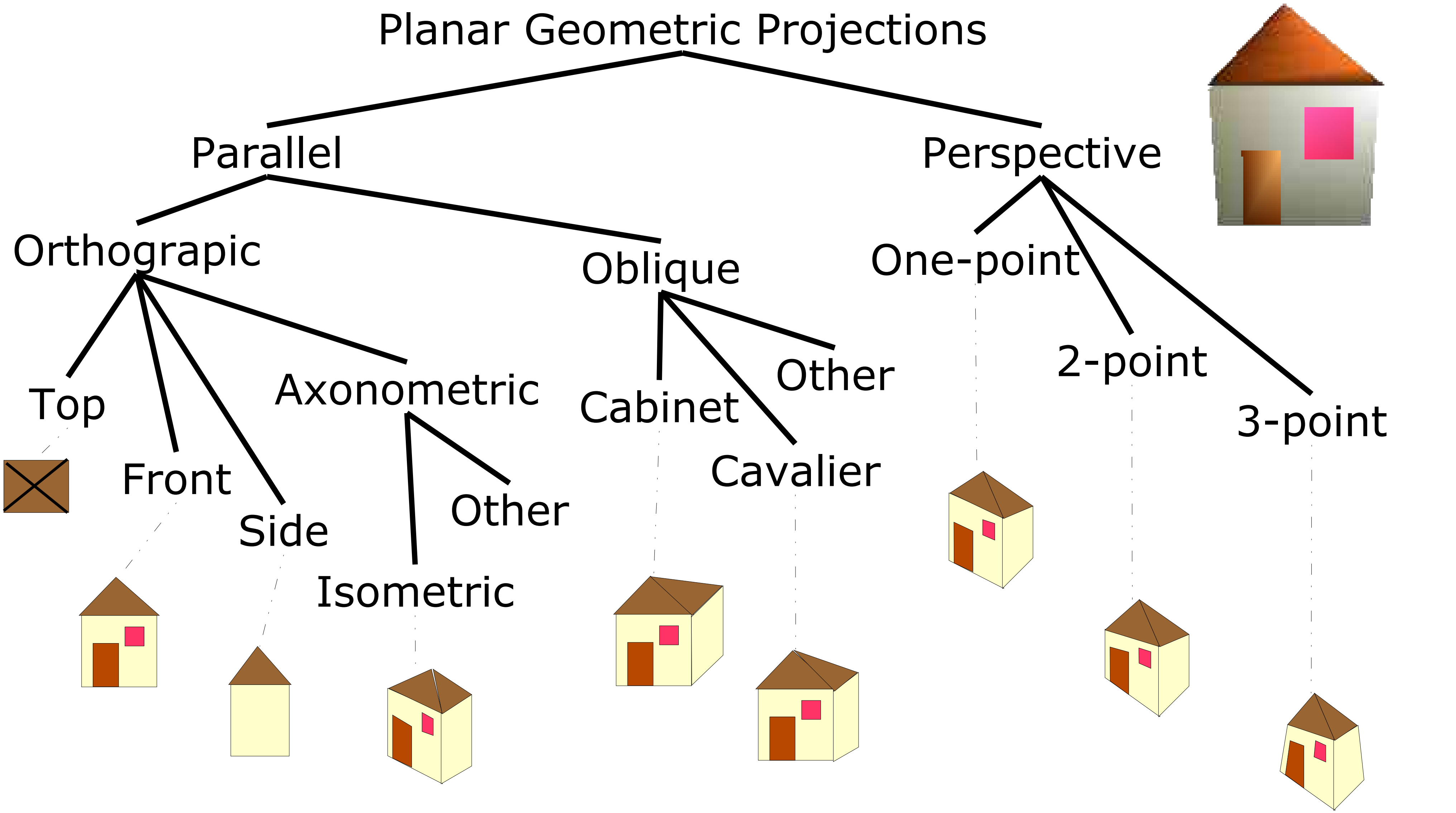
Perspective



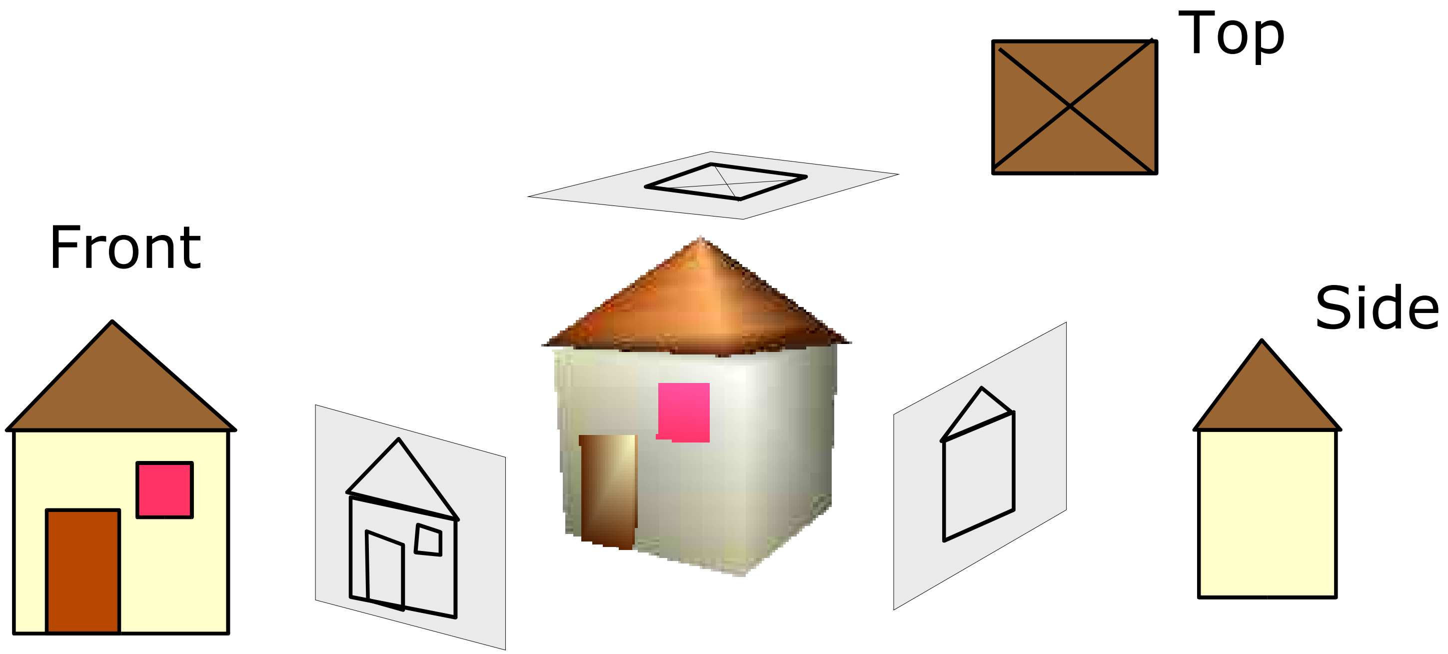
Orthographic



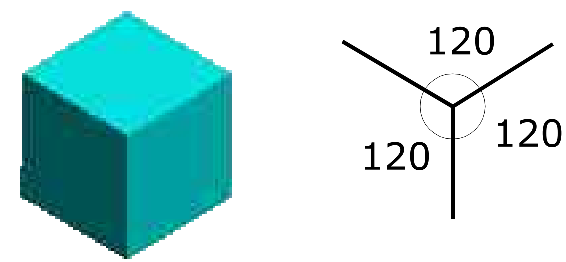
Oblique


Planar Geometric Projections
Parallel
Perspective
Orthograpic
Oblique
One-point
2-point
3-point
Top
Front
Side
Axonometric
Other
Isometric
Cabinet
Cavalier
Other

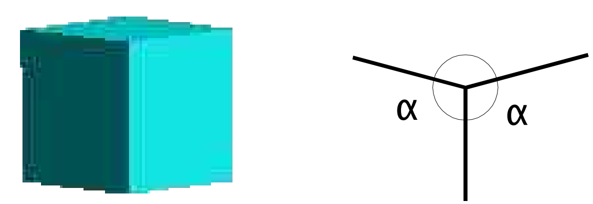
- Multiview Orthographic:
  - Parallel projection to z=0,y=0,x=0 planes
  - Used for engineering drawings, architectural drawing
  - Accurate, scales preserved. Not 3D realistic.



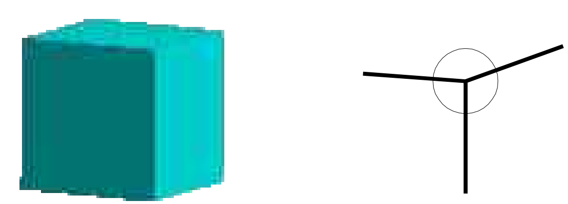
- Axonometric Projections
  - Projection plane is not parallel to coordinate planes
  - Isometric: all angles between principal axes are equal



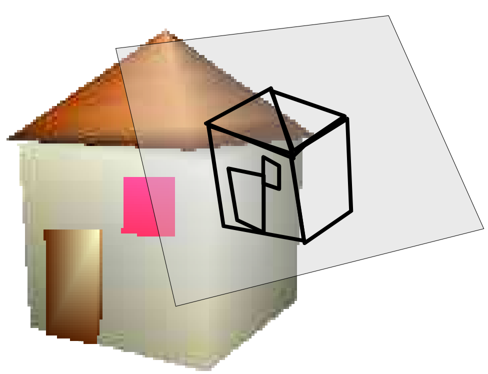
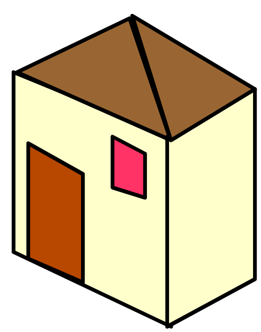
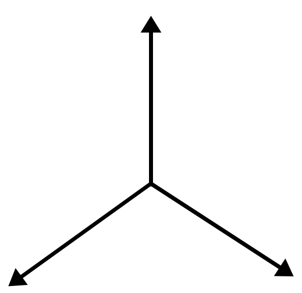
  - Dimetric: angles between two principal axes are equal



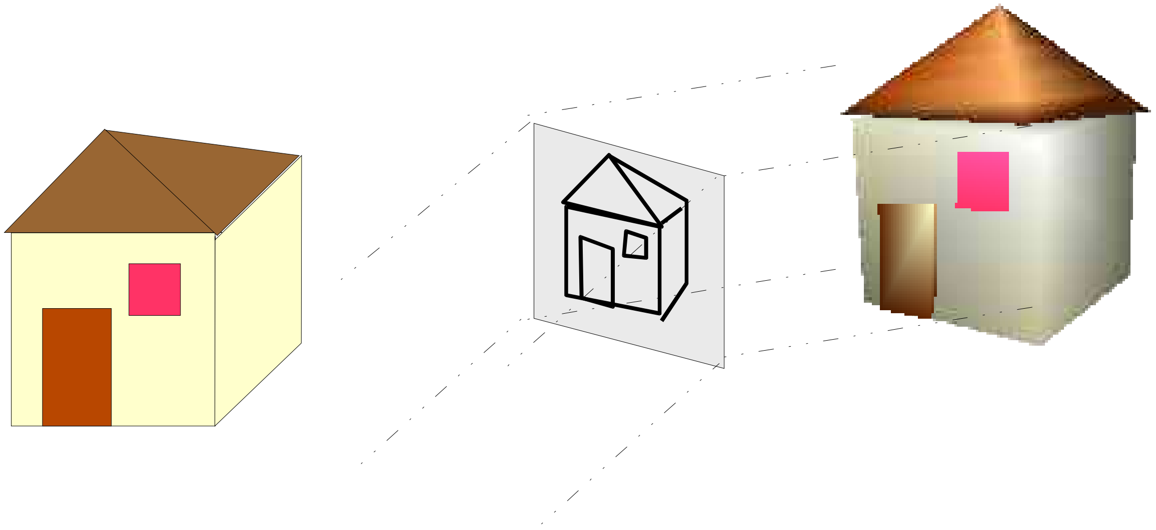
  - Trimetric: all angles different



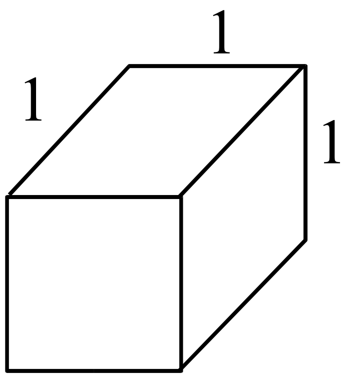
- Isometric Projection (i.e. N=[*c*,*c*,*c*])
  - More realistic. Scales preserved along axes.
  - Used in designs and catalogues. Suitable for rectangular bodies.
  - Exercise: Calculate rotation matrix for isometric projection, up vector on *x*=*z*.

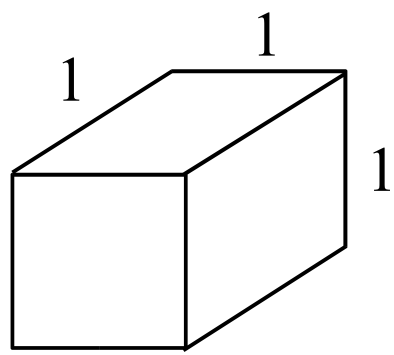




- Oblique projections
  - Projectors have an oblique angle
  - One of the sides have exact dimensions. Others are proportional.
  - Mechanical viewing

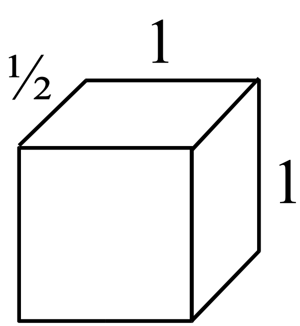


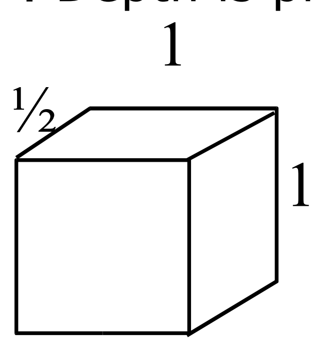
- Types of Oblique projections:
  - *Cavalier*: Angle between projectors and projection plane is 45. Depth is projected full scale



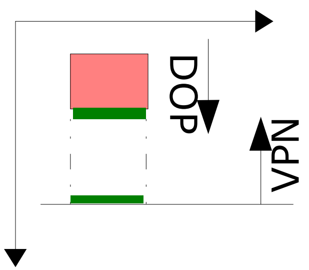


  - *Cabinet*: Angle between projectors and proejction plane is arctan(2)=63.4... . Depth is projected ½ scale.

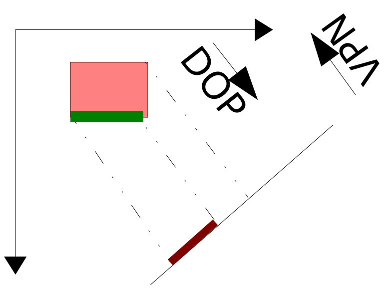




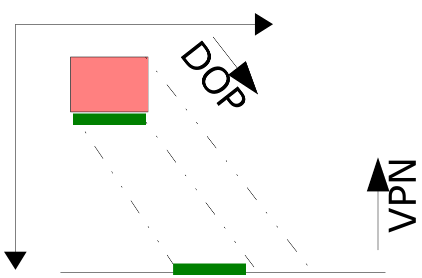
Parallel Projection Summary



- Multiview orthographic
  - VPN || a principal coordinate axis
  - DOP || VPN
  - Single face, exact dimensions



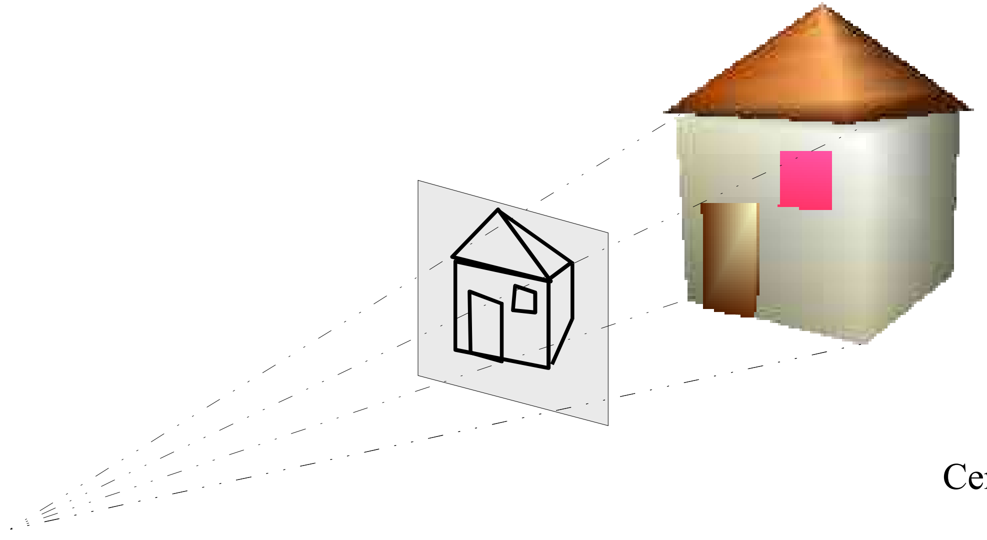
- Axonometric
  - VPN not || a principal coordinate axis
  - DOP || VPN
  - Not exact dimensions

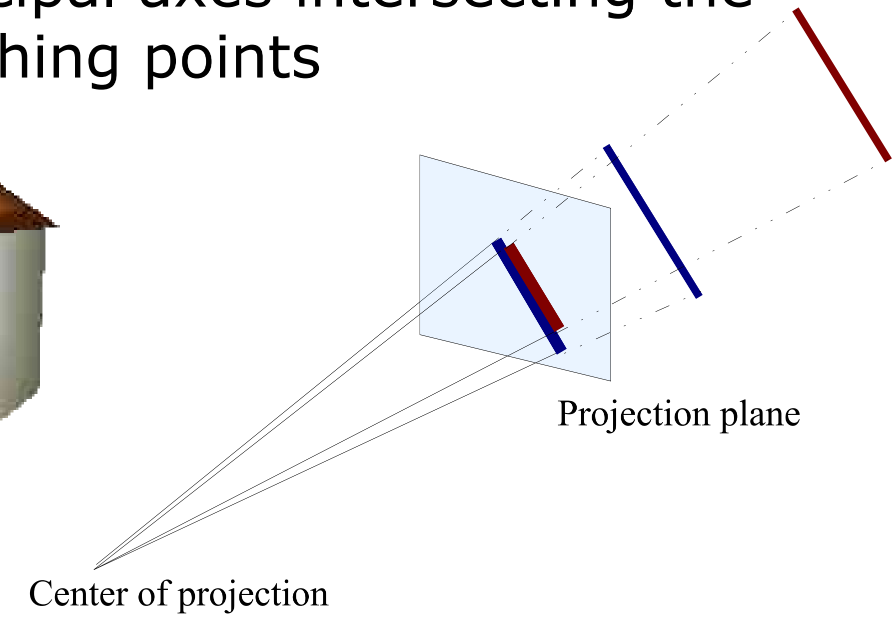


- Oblique
  - VPN || a principal coordinate axis
  - DOP not || VPN
  - One face exact dimension.

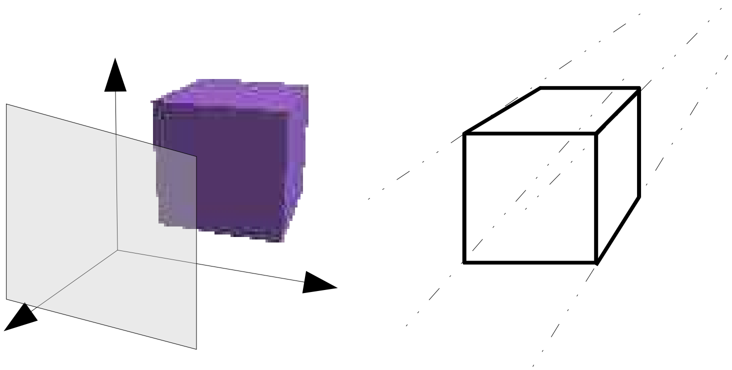
Perspective Projection

- Single point center of projection
- Shapes are projected smaller as their distance increase.
- More realistic (human eye is a perspective projector)
- Depending on number of principal axes intersecting the viewing plane: 1, 2 or 3 vanishing points

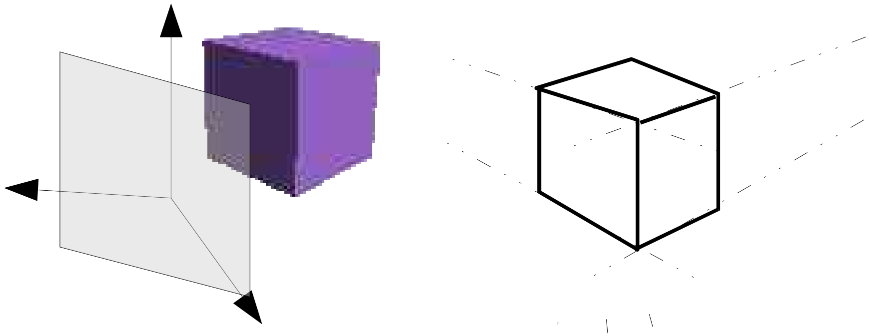




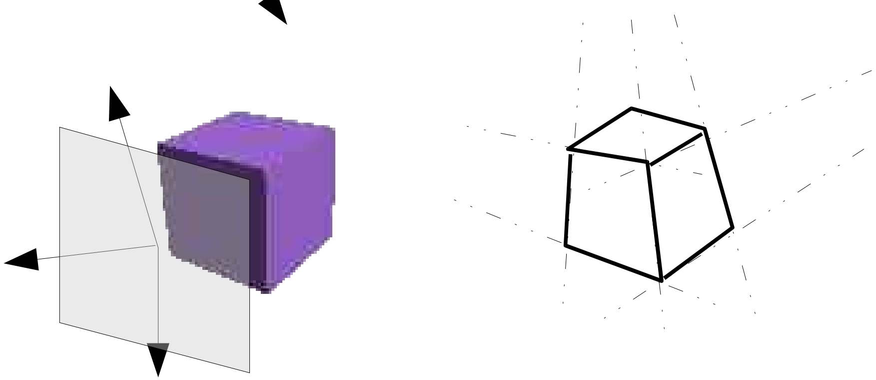
- One point perspective
  only *z* axis intersects
  single vanishing point



- Two-point perspective
  x and *z* axes intersect
  two vanishing points

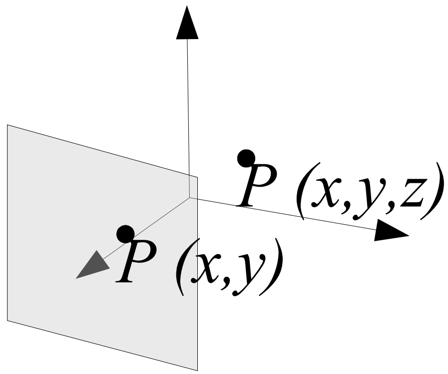


- Three-point perspective
  all axes intersect
  three vanishing points



Axonometric Parallel Projections

- If view plane normal is aligned to one of the axes, simply ignore that axis.



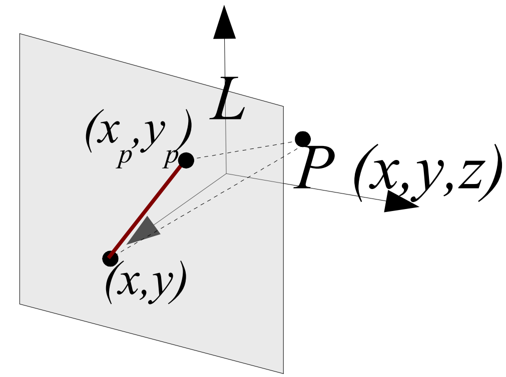
$$x_p = x \qquad y_p = y$$

$$\boldsymbol{M}_{\mathbf{parallel}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Other Axonometric projections:
  apply the required rotation to align view plane normal to *z* axis, and apply the projection above.

Oblique Projections

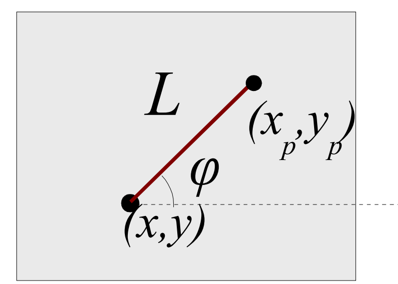
- Two angles selected:
  α: points, projection and view plane
  φ: direction of z axis in projection



L is the distance between: (x, y) and (x_p, y_p)

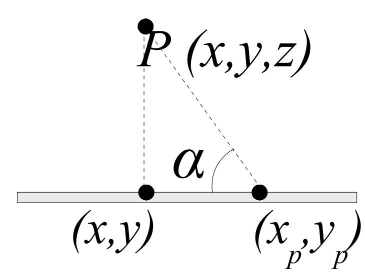
$$L = \frac{z}{\tan\alpha} \qquad \text{If } L_1 = \frac{1}{\tan\alpha}, \text{ also } L \text{ at } z = 1:$$

$$L = z L_1$$

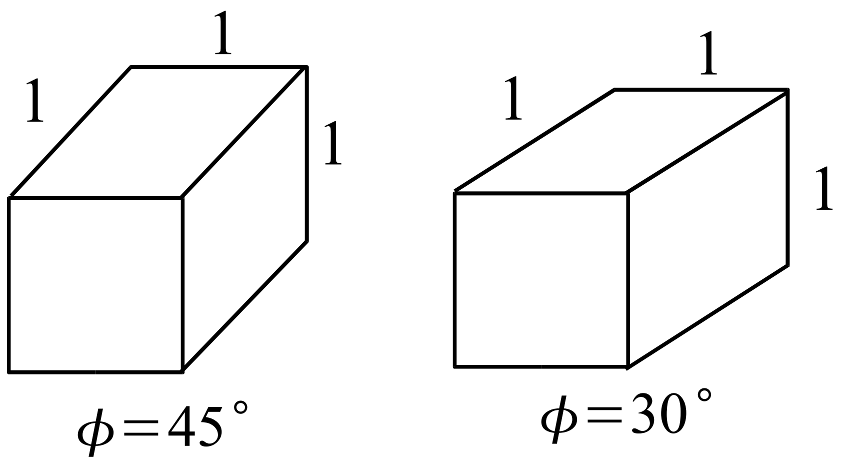


$$x_p = x + z L_1 \cos\phi$$
$$y_p = y + z L_1 \cos\phi$$

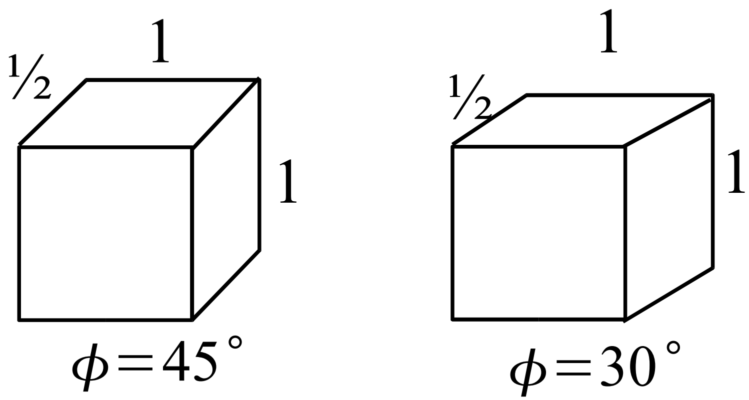
$$\boldsymbol{M}_{\textbf{oblique}} = \begin{bmatrix} 1 & 0 & L_1\cos\phi & 0 \\ 0 & 1 & L_1\sin\phi & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



- Cavalier projection: $\alpha = 45^\circ, L_1 = 1$



- Cabinet projection: $\alpha = 63.4^\circ, L_1 = \frac{1}{2}$



Perspective Projection

- *x* and *y* dimensions depends on values of their values and *z* values.

$$\frac{z_{prp}-z}{z_{prp}-z_{vp}}=\frac{y}{y_p} \qquad d_p=z_{prp}-z_{vp}$$

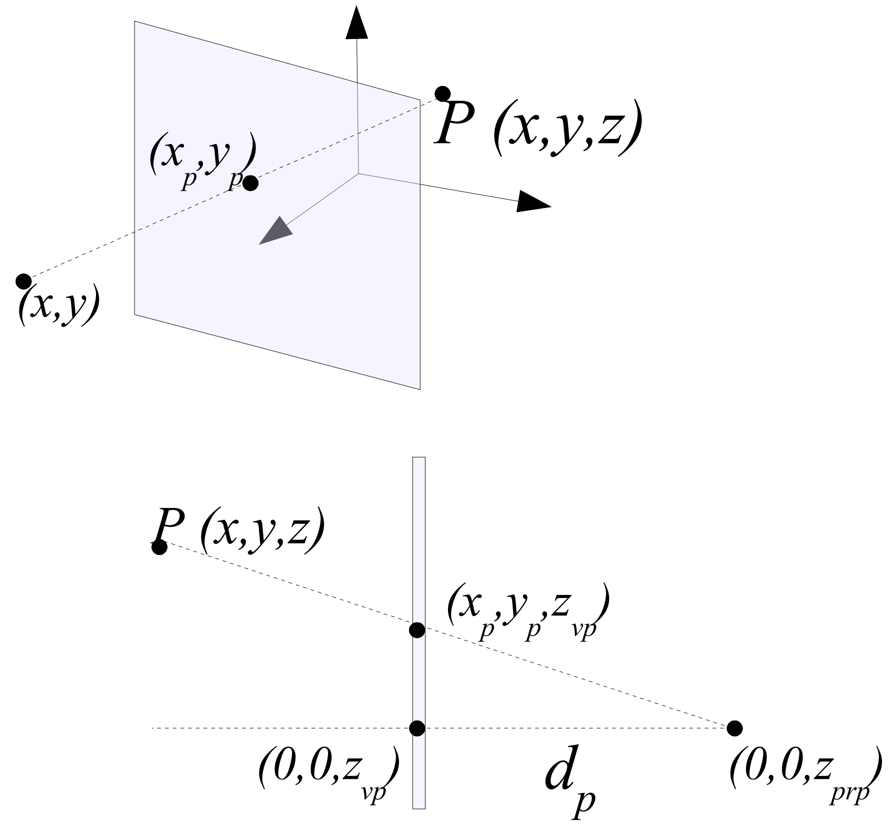
$$y_p=y\frac{d_p}{z_{prp}-z} \quad \text{similarly:} \quad x_p=x\frac{d_p}{z_{prp}-z}$$

Special case: $z_{vp}=0$

$$x_p=x\frac{1}{1-z/z_{prp}} \qquad y_p=y\frac{1}{1-z/z_{prp}}$$

Special case: $z_{prp}=0$

$$x_p=x\frac{z_{vp}}{z} \qquad y_p=y\frac{z_{vp}}{z}$$



- Perspective projection is not linear. Solution: homogenous coordinates, forth axis:

$$\begin{bmatrix} x_h \\ y_h \\ z_h \\ h \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -z_{vp}/d_p & z_{vp}(z_{prp}/d_p) \\ 0 & 0 & -1/d_p & z_{prp}/d_p \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$h = \frac{z_{prp} - z}{d_p}$$

$$x_p = \frac{x_h}{h}$$

$$y_p = \frac{y_h}{h}$$

- Where does z_h comes from?

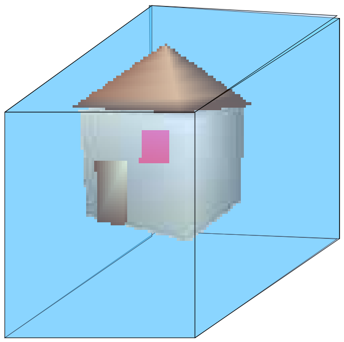
- Special cases

$$z_{vp} = 0 \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1/z_{prp} & 1 \end{bmatrix}$$

$$z_{prp} = 0 \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/z_{vp} & 0 \end{bmatrix}$$

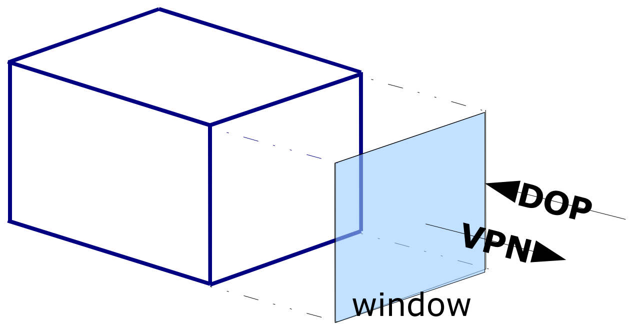
View Volumes

- Clipping planes:
  - left, right, top, bottom
  - front, back

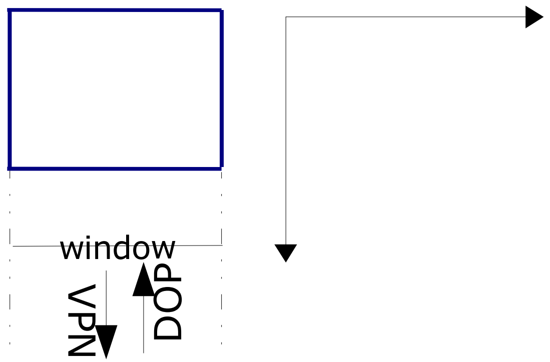


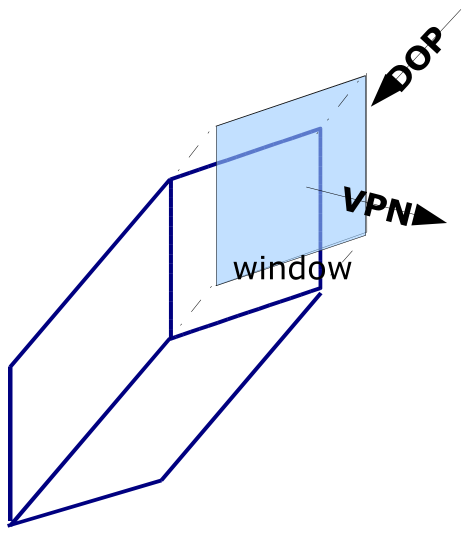
- Front: objects closer not focusable, hide most of the scene, perspective projection give distorted shapes.
- Back: very small, not visible but expensive.
- Clip the objects out of the view volume

- View volumes (parallel)

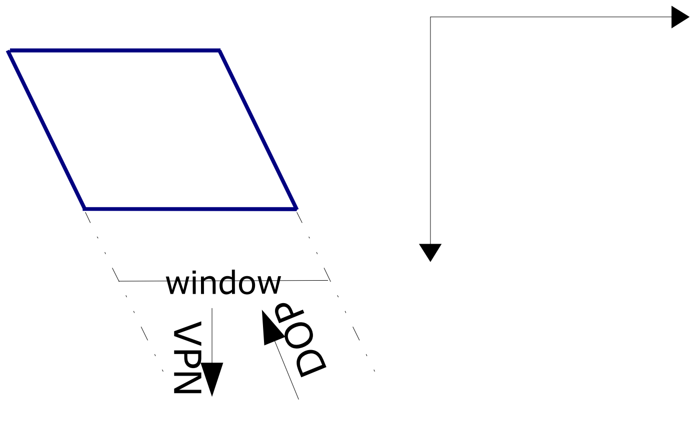


orthogonal

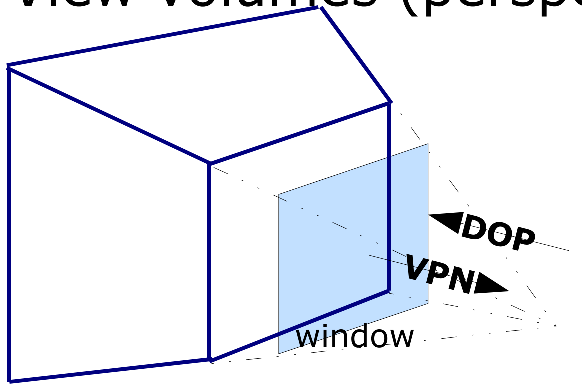




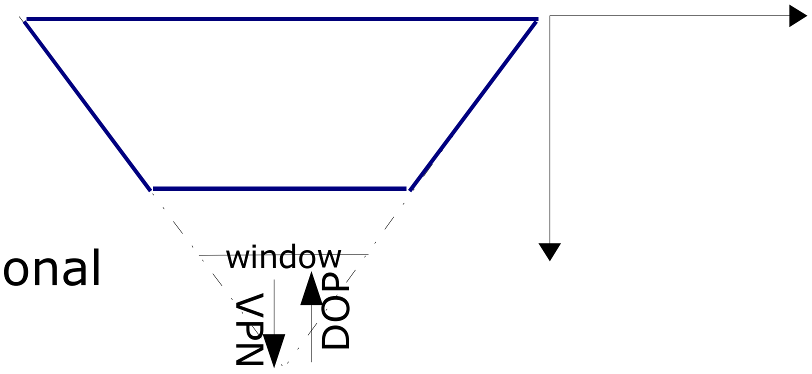
oblique

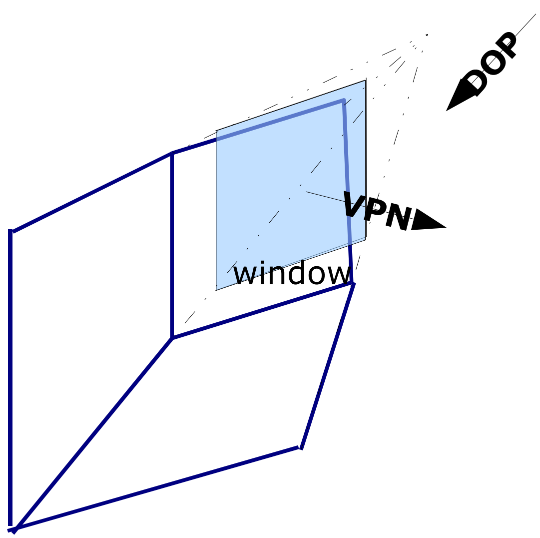


- View volumes (perspective)

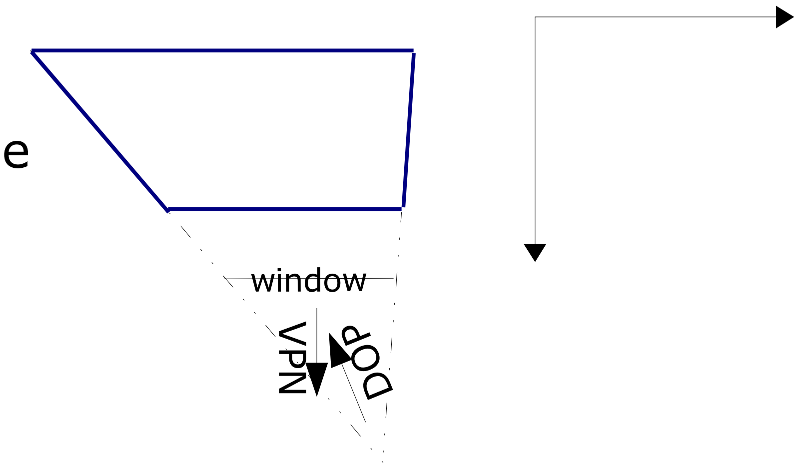


DOP ortogonal



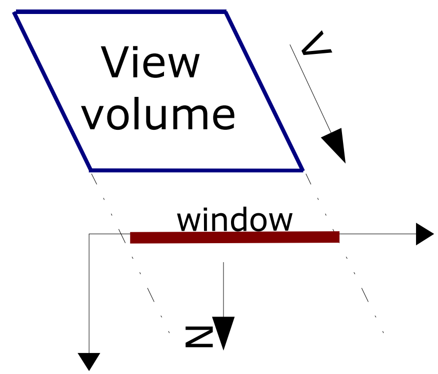


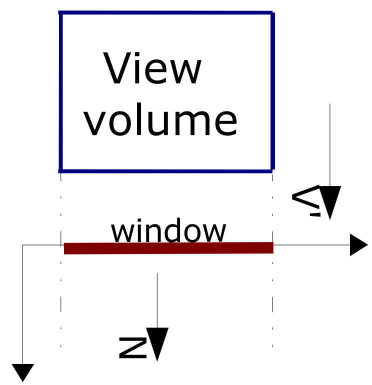
DOP oblique



General Parallel Projections

- Given the projection reference point and viewing vector, find the projection transformation





if $\boldsymbol{V}=(p_x, p_y, p_z)$, align $\boldsymbol{V}$ on $\boldsymbol{N}$:

$$\boldsymbol{V}'=\boldsymbol{M}\cdot\boldsymbol{V}=\begin{bmatrix}0\\0\\p_z\\0\end{bmatrix}$$

$\boldsymbol{M}$ is a shearing transformation mapping $\boldsymbol{V}$ to $\boldsymbol{N}$, solve:

$$\boldsymbol{M}=\begin{bmatrix}1&0&a&0\\0&1&b&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$$

$$p_x+a\,p_z=0 \qquad a=\frac{-p_x}{p_z}$$

$$p_y+b\,p_z=0 \qquad b=\frac{-p_y}{p_z}$$

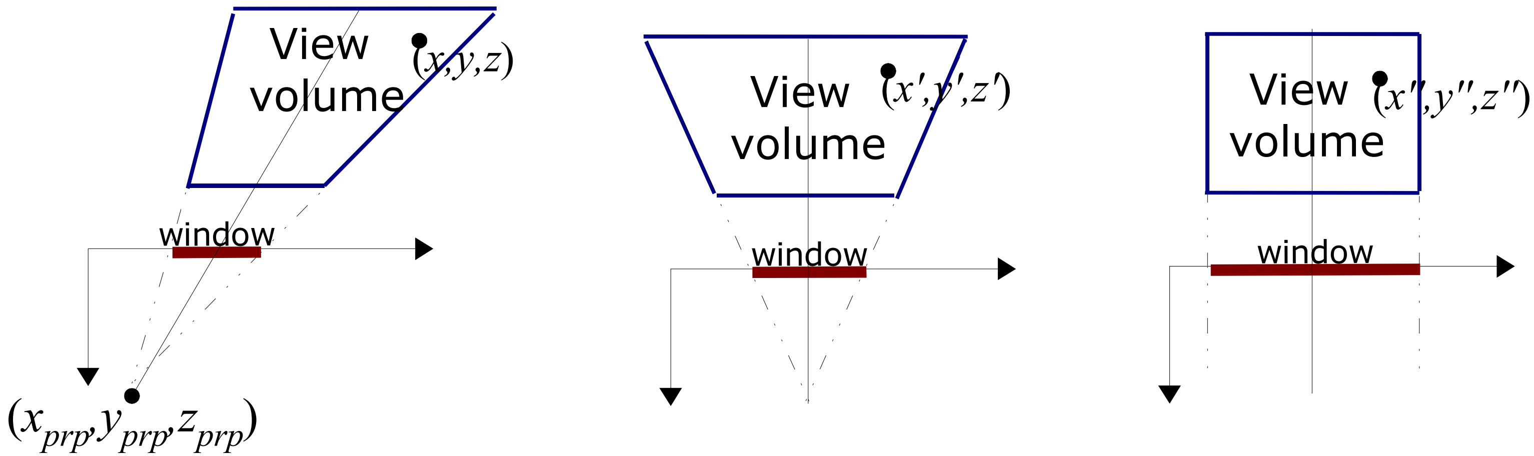
- Parallel projection transformation:

$$\boldsymbol{M}_{\textbf{parallel}} = \begin{bmatrix} 1 & 0 & -p_x/p_z & 0 \\ 0 & 1 & -p_y/p_z & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Note the similarity in oblique projection calculations.

General Perspective Projection

- Given a perspective point and window, find the projection transformation giving a parallel view volume.



1. Translate perspective point to $(0, 0, z_{prp})$.
2. Shear so that line from perspective point to window center be orthogonal to viewing frustum
3. Map frustum to a parallelpipe

Translate and shear to align $(x_{prp}, y_{prp}, z_{prp})$ vector to view plane normal

$$\boldsymbol{M}_{shear} = \begin{bmatrix} 1 & 0 & a & -a z_{prp} \\ 0 & 1 & b & -b z_{prp} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad a = -\frac{x_{prp} - (xw_{min} + xw_{max})/2}{z_{prp}} \quad b = -\frac{y_{prp} - (yw_{min} + yw_{max})/2}{z_{prp}}$$

$$x' = x + a(z - z_{prp})$$
$$y' = y + b(z - z_{prp})$$
$$z' = z$$

Map values in frustum to a parallelpipe

$$x'' = x'\left(\frac{z_{prp} - z_{vp}}{z_{prp} - z}\right) + x_{prp}\left(\frac{z_{vp} - z}{z_{prp} - z}\right)$$
$$y'' = y'\left(\frac{z_{prp} - z_{vp}}{z_{prp} - z}\right) + y_{prp}\left(\frac{z_{vp} - z}{z_{prp} - z}\right)$$

$$\boldsymbol{M}_{scale} = \begin{bmatrix} 1 & 0 & \frac{-x_{prp}}{d} & \frac{x_{prp} z_{vp}}{d} \\ 0 & 1 & \frac{-y_{prp}}{d} & \frac{y_{prp} z_{vp}}{d} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1/d & z_{prp}/d \end{bmatrix} \quad \boldsymbol{M}_{pers} = \boldsymbol{M}_{scale} \cdot \boldsymbol{M}_{shear}$$

Normalized View Volume

- Mapping the projected view volume into a unit cube:
  - Device independent standart view volume
  - Clipping is simplified, unit clipping planes
  - Simpler depth cueing
- Scale and translate into normalized view volume:

$$x=0, \quad x=1, \quad y=0, \quad y=1, \quad z=0, \quad z=1$$

$$\begin{bmatrix} D_x & 0 & 0 & K_x \\ 0 & D_y & 0 & K_y \\ 0 & 0 & D_z & K_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$D_x = \frac{xv_{max} - xv_{min}}{xw_{max} - xw_{min}}$$

$$D_y = \frac{yv_{max} - yv_{min}}{yw_{max} - yw_{min}}$$

$$D_z = \frac{zv_{max} - zv_{min}}{z_{back} - z_{front}}$$

$$K_x = xv_{min} - xw_{min} D_x$$
$$K_y = yv_{min} - yw_{min} D_y$$
$$K_z = zv_{min} - z_{front} D_z$$

Clipping

- 2D clipping algorithms can be extended to handle 3D clipping. Instead of intersection with border lines, intersection with planes are considered

- Cohen-Sutherland bits:

$$
\begin{aligned}
&bit1 = 1, && \text{if } x < xv_{min} \text{(left)} \\
&bit2 = 1, && \text{if } x > xv_{max} \text{(right)} \\
&bit3 = 1, && \text{if } y < yv_{min} \text{(below)} \\
&bit4 = 1, && \text{if } y > yv_{max} \text{(above)} \\
&bit5 = 1, && \text{if } z < zv_{min} \text{(front)} \\
&bit6 = 1, && \text{if } z > zv_{max} \text{(back)}
\end{aligned}
$$

- Parametric line equations

$$
\begin{aligned}
&x = x_1 + (x_2 - x_1)u, \qquad 0 \leqslant u \leqslant 1 \\
&y = y_1 + (y_2 - y_1)u \\
&z = z_1 + (z_2 - z_1)u
\end{aligned}
$$