# Computer Graphics

3D Viewing

Dr. Mousumi Dutt

CSE, STCET

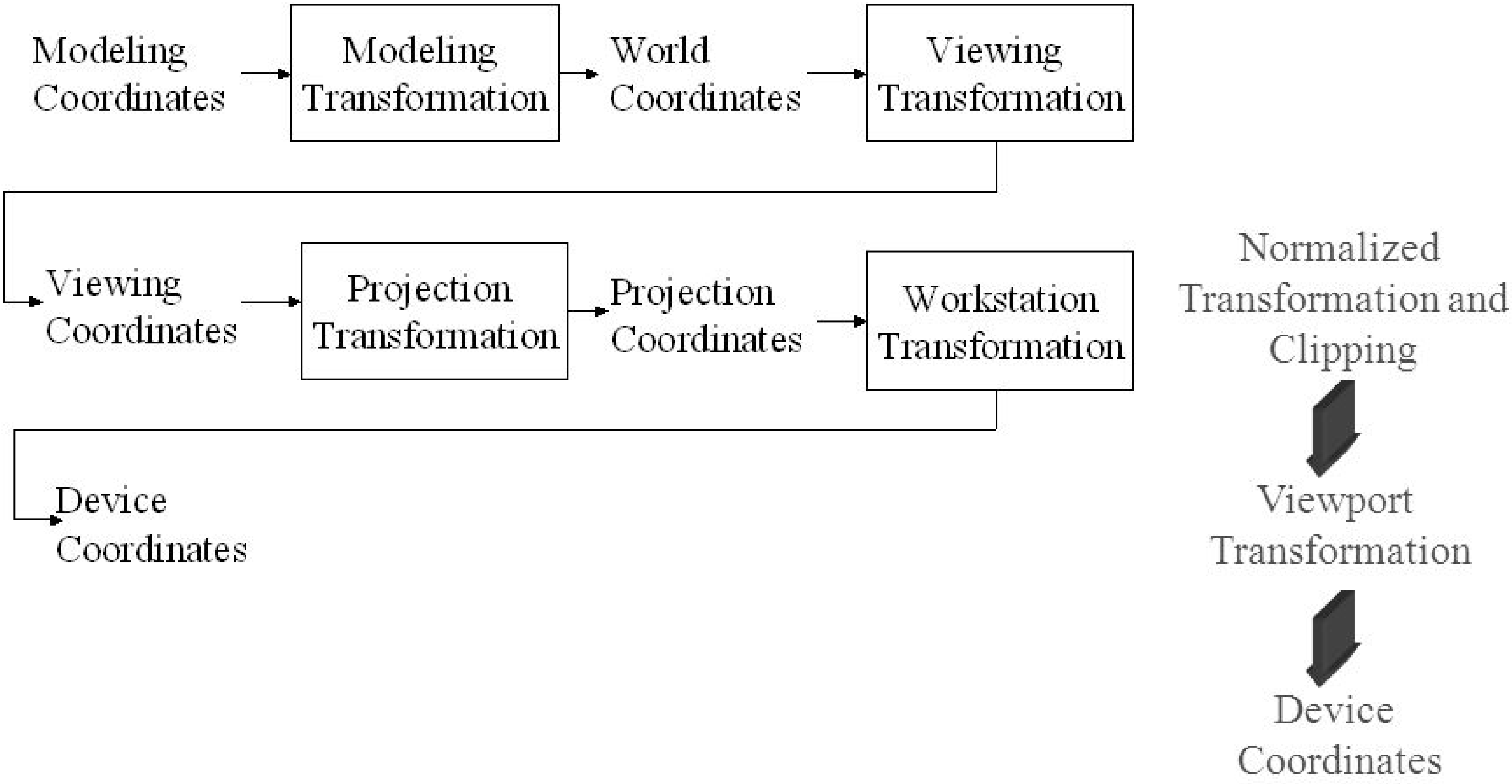
# Introduction

- For three-dimensional graphics applications
  - more choices as to how views are to be generated
- We can view an object from any spatial position:
  - from the front, from above, or from the back
- We could generate a view of what we would see if we were standing in the middle of a group of objects or inside a single object, such as a building
- Three-dimensional descriptions of objects must be projected onto the flat viewing surface of the output device
- The clipping boundaries now enclose a volume of space, whose shape depends on the type of projection we select

# Viewing Pipeline

- The steps for computer generation of a view of a three-dimensional scene are somewhat analogous to the processes involved in taking a photograph
- Which way do we point the camera and how should we rotate it around the line of sight to set the up direction for the picture?
- Finally, when we snap the shutter, the scene is cropped to the size of the "window" (aperture) of the camera, and light from the visible surfaces is projected onto the camera film

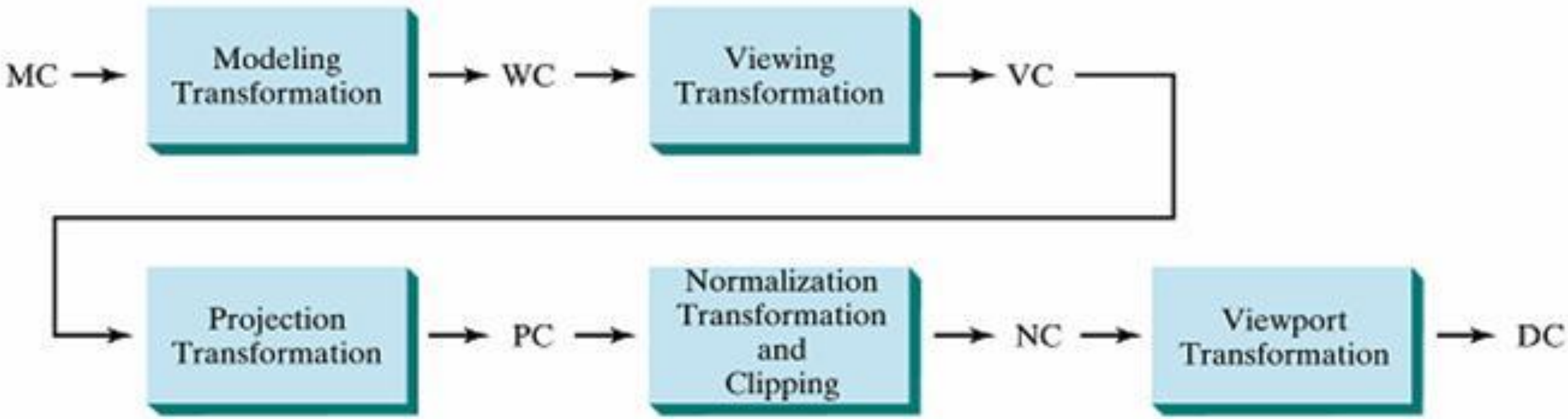
# Viewing Pipeline



# Viewing Pipeline

- Once the scene has been modelled, world-coordinate positions are converted to viewing coordinates
- The viewing-coordinate system is used in graphics packages as a reference for specifying the observer viewing position and the position of the projection plane, which we can think of in analogy with the camera film plane
- Next, projection operations are performed to convert the viewing-coordinate description of the scene to coordinate positions on the projection plane, which will then be mapped to the output device
- Objects outside the specified viewing limits are clipped from further consideration, and the remaining objects are processed through visible-surface identification and surface-rendering procedures to produce the display within the device viewport

# Viewing Pipeline



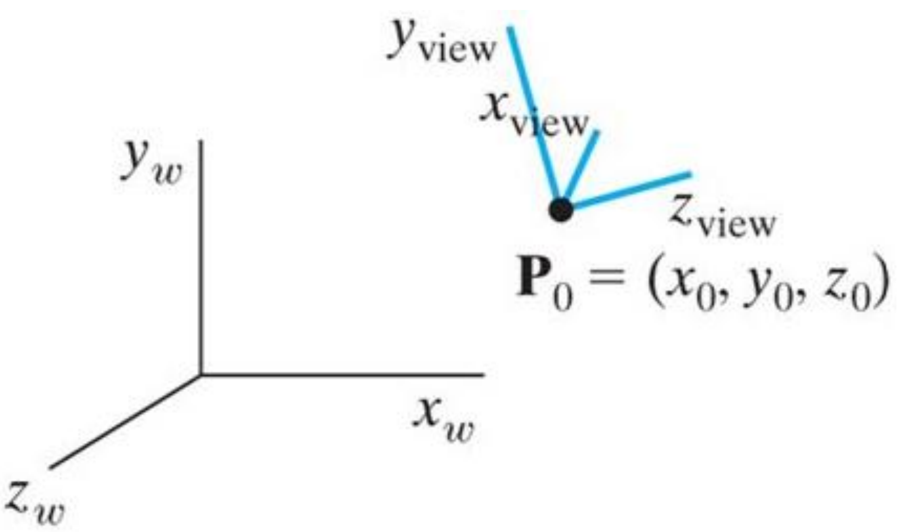

General three-dimensional transformation pipeline, from modeling coordinates to world coordinates to viewing coordinates to projection coordinates to normalized coordinates and, ultimately, to device coordinates.

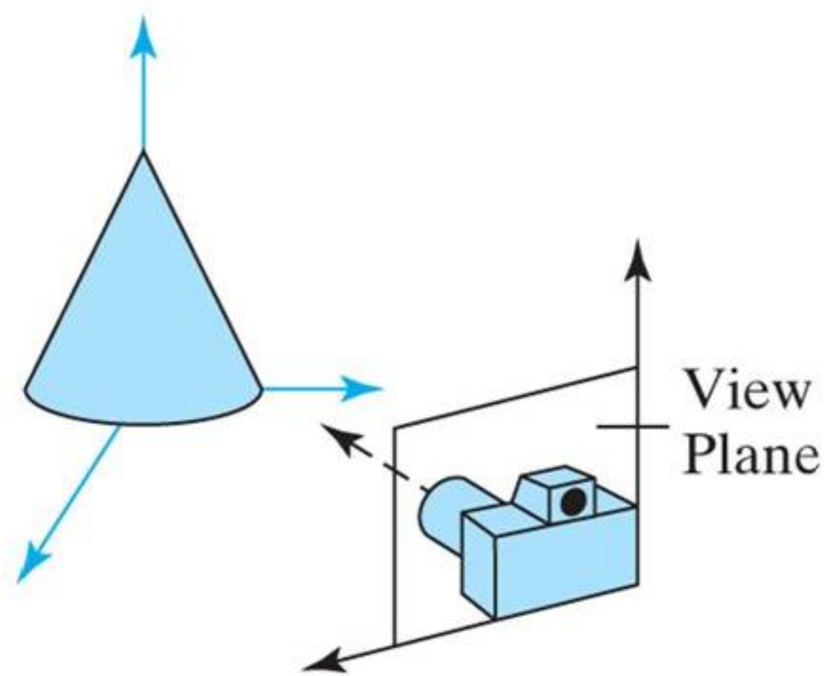
# Viewing Pipeline

- Once the scene has been modelled, world-coordinate positions are converted to viewing coordinates
- The viewing-coordinate system is used in graphics packages as a reference for specifying the observer viewing position and the position of the projection plane, which we can think of in analogy with the camera film plane
- Next, projection operations are performed to convert the viewing-coordinate description of the scene to coordinate positions on the projection plane, which will then be mapped to the output device
- Objects outside the specified viewing limits are clipped from further consideration, and the remaining objects are processed through visible-surface identification and surface-rendering procedures to produce the display within the device viewport

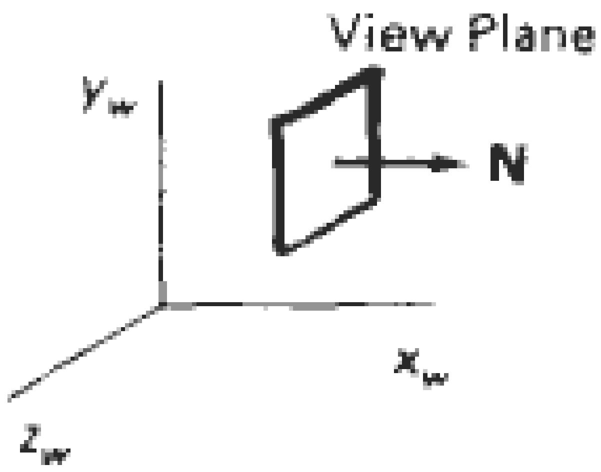
# Viewing Coordinates

- Generating view in 3D is similar to photography
- Establish the viewing coordinates or viewing reference coordinate systems
- View plane or projection plane: set up perpendicular to the viewing $z_v$ axis
- World coordinates ->viewing coordinates ->projected on view plane

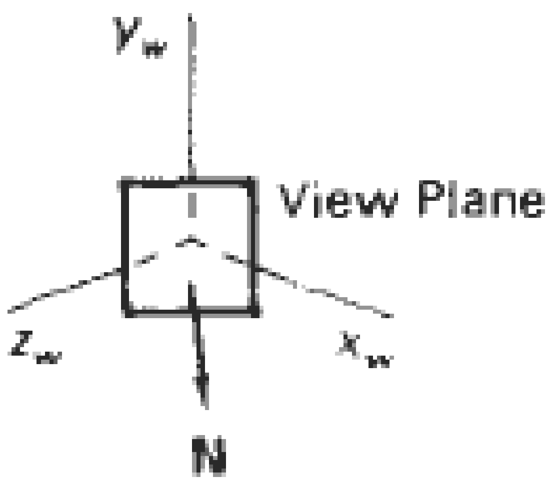


# Viewing Coordinates




(a)

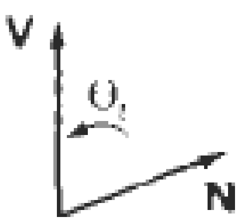



(b)

- Pick a position in world coordinate position called view reference point
- It is the origin of viewing-coordinate system
- It is close to or on the surface of the object in a scene
- May be center of the object, or at the center of group of objects, or somewhere out in front of a scene displayed
- The point -> position to place camera
- Select positive direction of $z_v$ axis and orientation of the view plane, specify the view plane normal vector, **N**

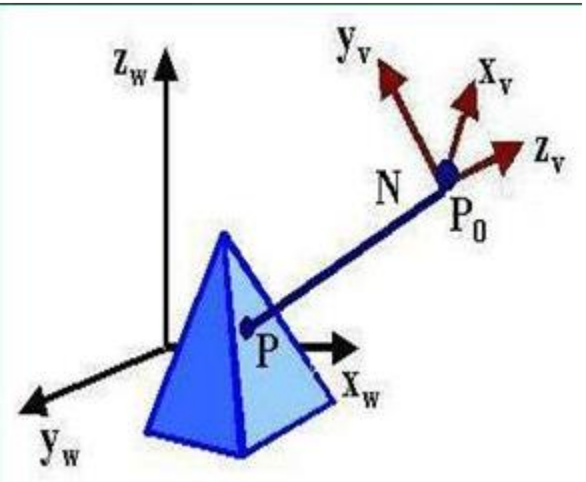
# Viewing Coordinates

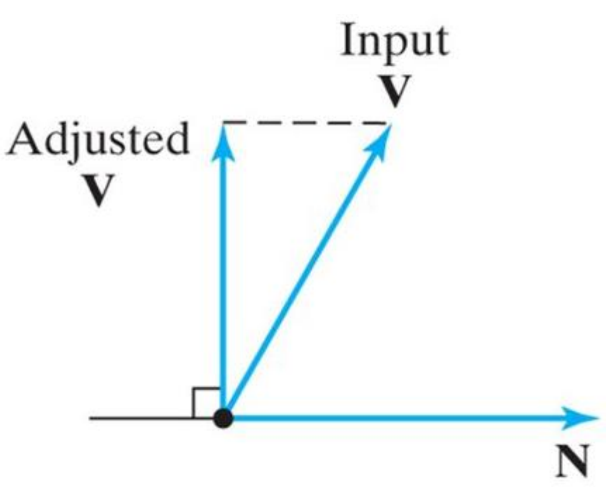
- The view plane normal **N** is the directed line segment from the world origin to the selected coordinate position
- Some graphic package, establish the direction of **N** using the selected the coordinate position as a look-at-point relative to the view reference point
- Consider left handed viewing system, take **N** and positive $z_v$ axis from the viewing origin to the look-at-point
- The magnitude is irrelevant
- **N** will be normalized to unit vector by the viewing calculation
- View up vector: up direction for the view by specifying a vector **V**. It is used to establish the positive direction for the $y_v$ axis



# Viewing Coordinates

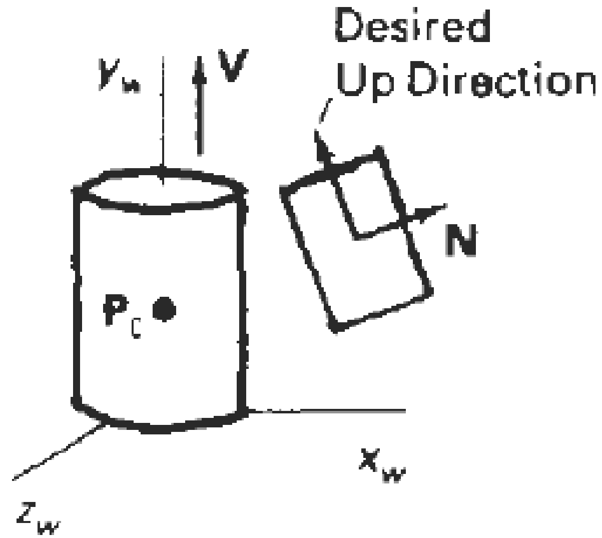
- Vector **V** can also be defined as world coordinate vector
- In some packages it is specified with a twist angle $\Theta_t$ about the $z_v$ axis
- For a general orientation of the normal vector, it can be difficult to determine the direction for **V** that is precisely perpendicular **N**
- Choose the view up vector **V** to be any convenient direction, as long as it is not parallel to **N**



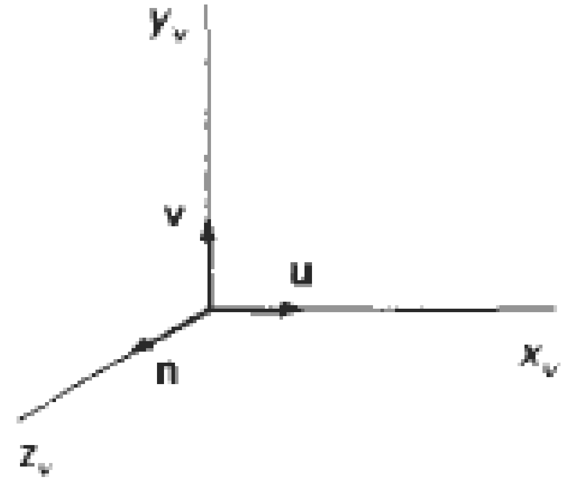
# Viewing Coordinates

- ❑ When view reference point at the center of the object
- ❑ Choose **V** as a world vector (0,1,0) and this vector will be projected into the plane perpendicular to **N** to establish the $y_v$ axis
- ❑ Using **N** and **V**, the third vector U can be computed (perpendicular to both **N** and **V**) to define the direction for $x_v$ axis
- ❑ **V** can be adjusted so that it is perpendicular to both **N** and **U** using the viewing $y_v$ direction



# Viewing Coordinates

- Unit vectors are used to obtain the elements of the world-to-viewing-coordinate transformation matrix




- The viewing system is called uvn system
- To choose the position of the view plane

along $z_v$ axis by specifying the view plane

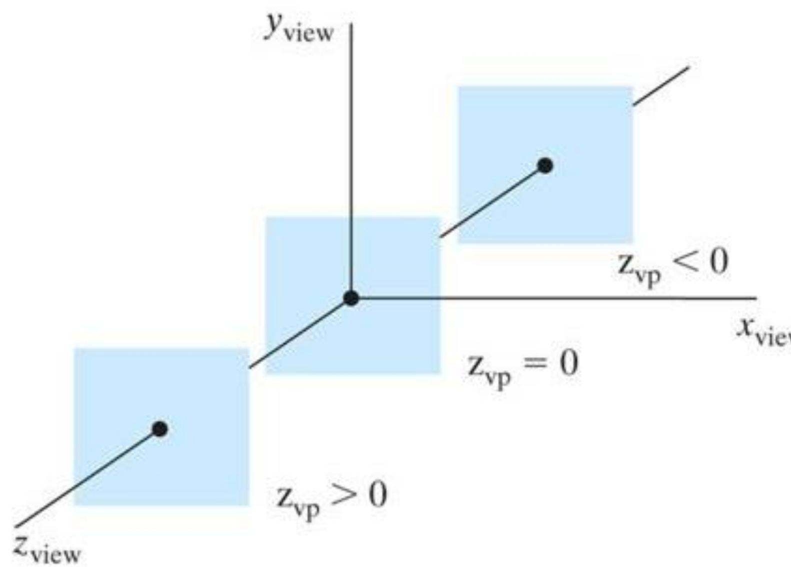
Distance from the viewing origin

- View plane is parallel to $x_v y_v$ axis and the projection of the object to the view plane

correspond to the view of the scene

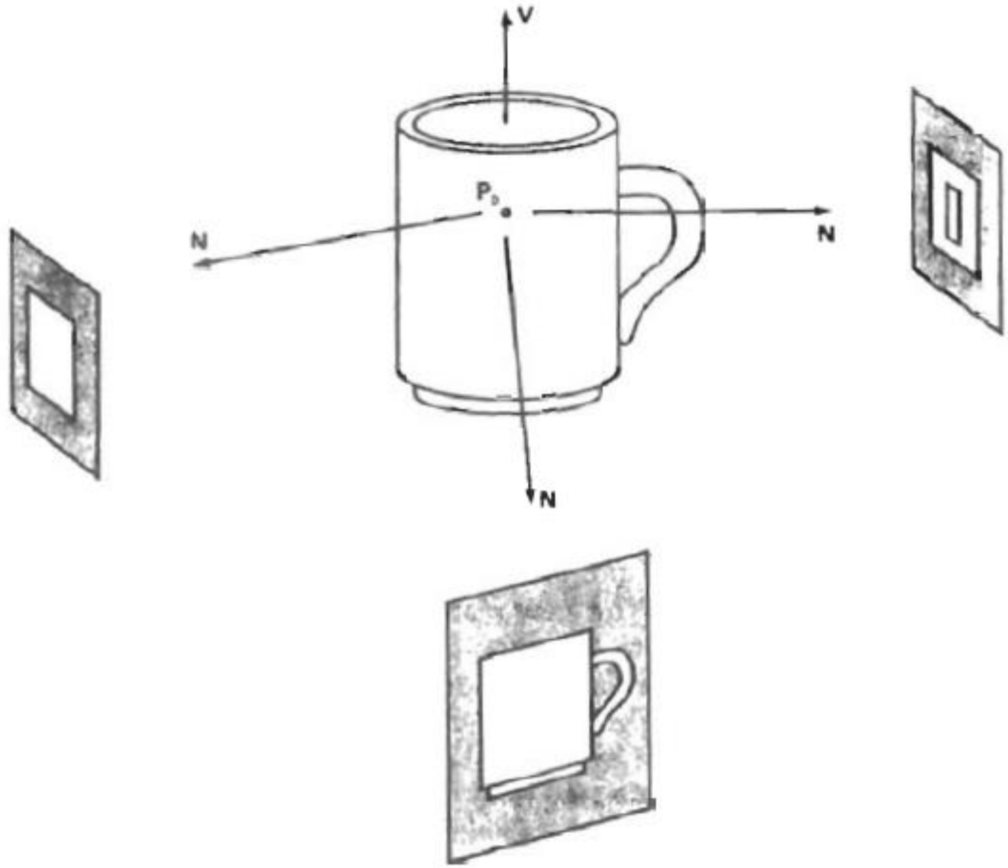
that will be displayed on the output

device



# Viewing Coordinates

- To obtain a series of views of a scene, keep the view reference point fixed and change the direction of **N**
- This corresponds to generating views as we move around the viewing-coordinate origin



# Transformation from World to Viewing Coordinates

1. Translate the view reference point to the origin of the world coordinate system
2. Apply rotations to align the $x_v$, $y_v$, and $z_v$ axes with the world $x_w$, $y_w$, and $z_w$ axes, respectively

If the view reference point is specified at world position $(x_0,y_0,z_0)$, this point is translated to world origin with the matrix transformation

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

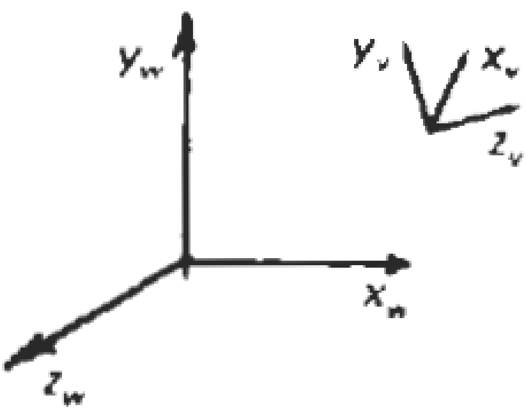
# Transformation from World to Viewing Coordinates

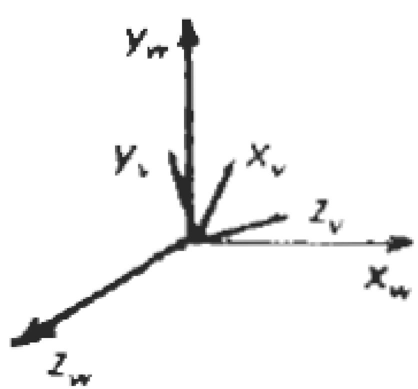
Rotation sequences: $\mathbf{R_z \, R_y \, R_x}$

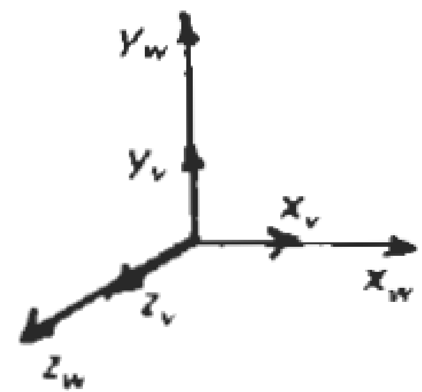
First rotate around $x_w$ axis to bring $z_v$ into $x_w z_w$ plane

Rotate around the $y_w$ axis to align $z_v$ and $z_w$ axis

The final rotation is about $z_w$ axis to align $y_v$ and $y_w$ axis



# Transformation from World to Viewing Coordinates

To calculate uvn vectors and form composite rotation matrix directly

Given vectors **N** and **V**, unit vectors can be calculated as

$$\mathbf{n} = \frac{\mathbf{N}}{|\mathbf{N}|} = (n_1, n_2, n_3)$$

$$\mathbf{u} = \frac{\mathbf{V} \times \mathbf{N}}{|\mathbf{V} \times \mathbf{N}|} = (u_1, u_2, u_3)$$

$$\mathbf{v} = \mathbf{n} \times \mathbf{u} = (v_1, v_2, v_3)$$

Automatically adjusts the direction of **V** (**v** is perpendicular to **n**)

Composite rotation matrix:

Transforms **u** onto $\mathbf{x}_w$ axis

**v** onto $\mathbf{y}_w$ axis

**n** onto $\mathbf{z}_w$ axis

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

# Transformation from World to Viewing Coordinates

The complete world to viewing coordinate transformation matrix is obtained as the matrix product

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

This transformation is applied to coordinate description of the objects in the scene to transfer them to the viewing reference frame

# Projection

- The world coordinate description of a object in a scene is converted to viewing coordinate
- The 3D view is projected on a 2D view plane

- Types of projection:

a) Parallel Projection

b) Perspective Projection

- Types of parallel projection:

a) Orthographic projection

b) Oblique projection

# Projection

❑ Types of orthographic projection:

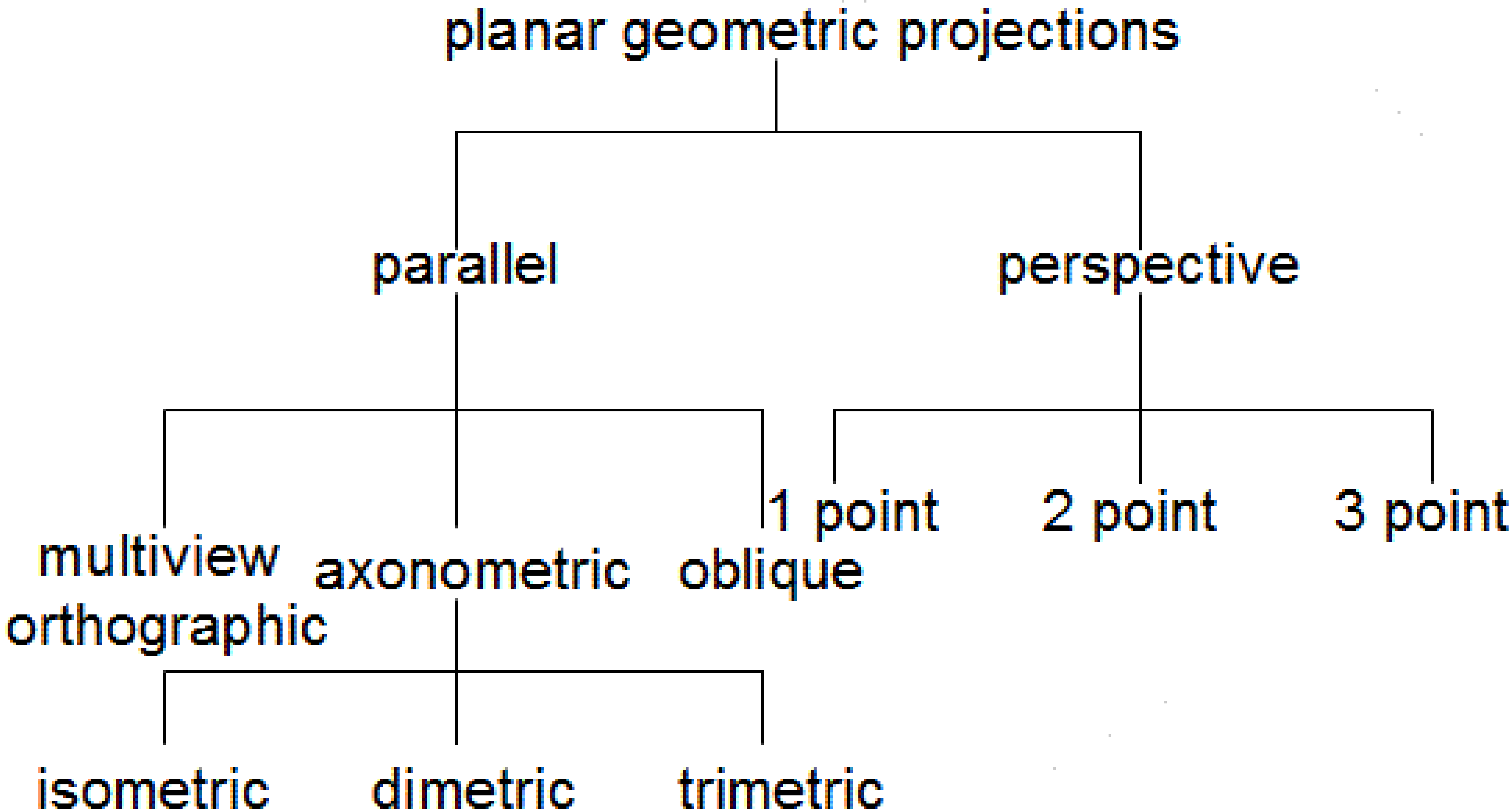
a) Plan view
b) Side elevation view
c) Front elevation view
d) Axonometric projection

❑ Types of axonometric projection:

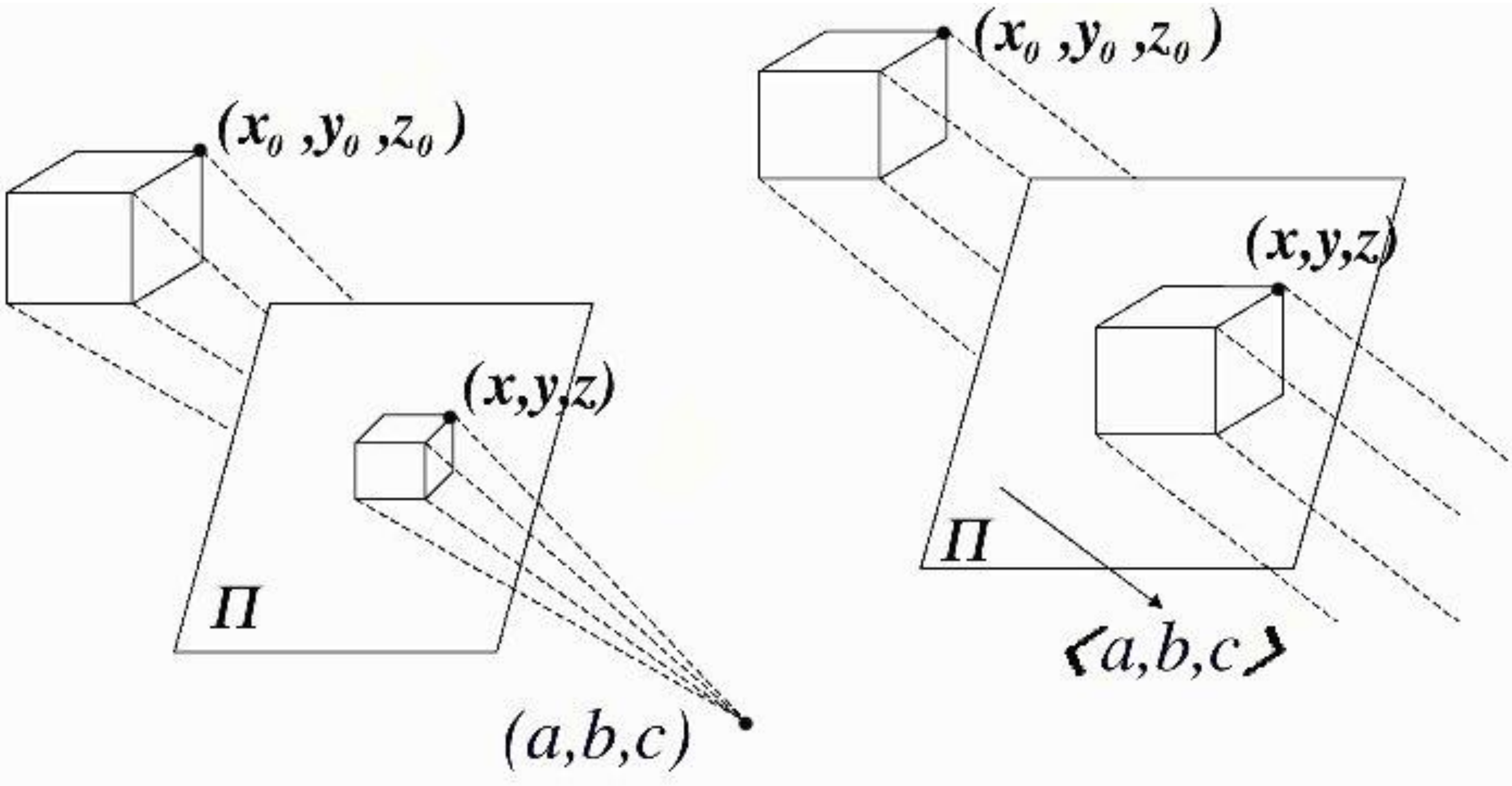
a) Isometric
b) Dimetric
c) Trimetric

❑ Types of oblique projection:

a) Cavalier
b) Cabinet


planar geometric projections
parallel
perspective
multiview orthographic
axonometric
oblique
1 point
2 point
3 point
isometric
dimetric
trimetric

# Projection



# Projection

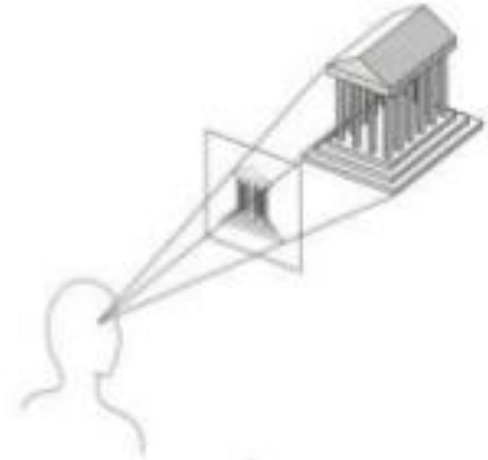
- **Parallel Projection:** coordinate positions are transformed to the view plane along parallel lines
  - Relative positions are maintained

- **Perspective Projection:** object positions are transformed to the view plane along lines that converge to a point called projection reference point or center of projection
  - Realistic view

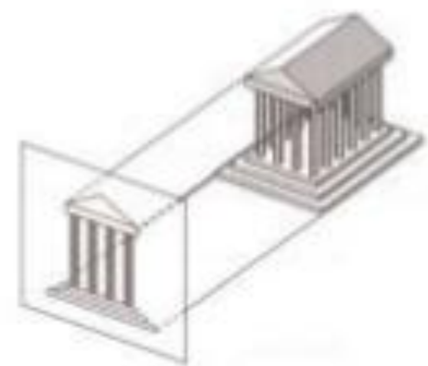
- **The projected view of an object is determined by calculating the intersection of the projection lines with the view plane**

# Projection

- Perspective projection
  + Size varies inversely with distance - looks realistic
  – Distance and angles are not (in general) preserved
  – Parallel lines do not (in general) remain parallel



- Parallel projection
  + Good for exact measurements
  + Parallel lines remain parallel
  – Angles are not (in general) preserved
  – Less realistic looking

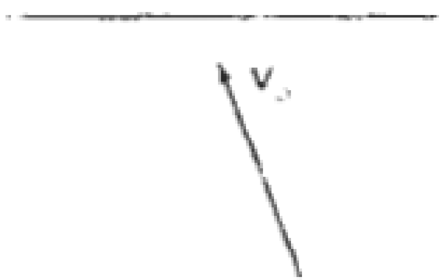


# Parallel Projection

- **Projection vector:** direction of projection lines

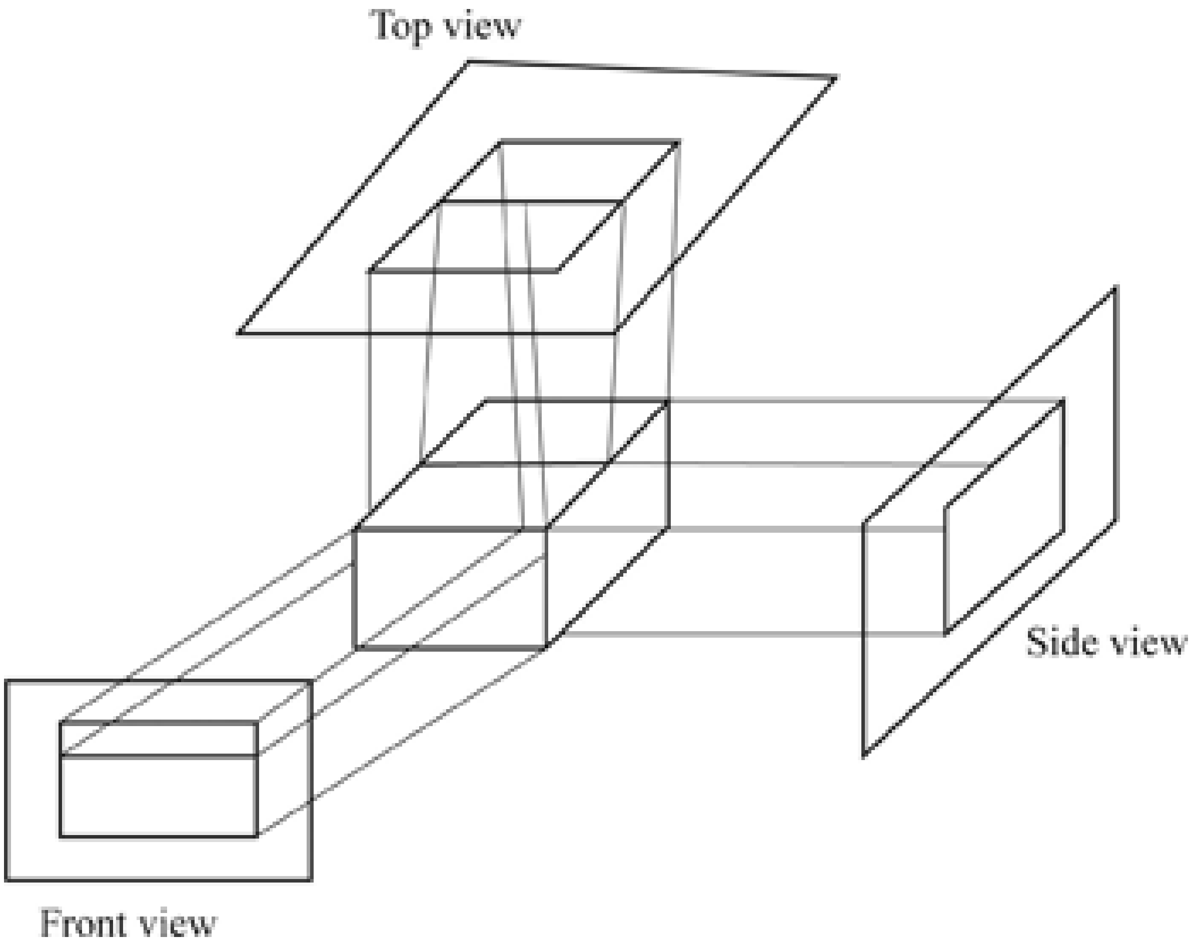
- **Orthographic Projection:** When the projection is perpendicular to the view plane

- Otherwise **Oblique Projection**

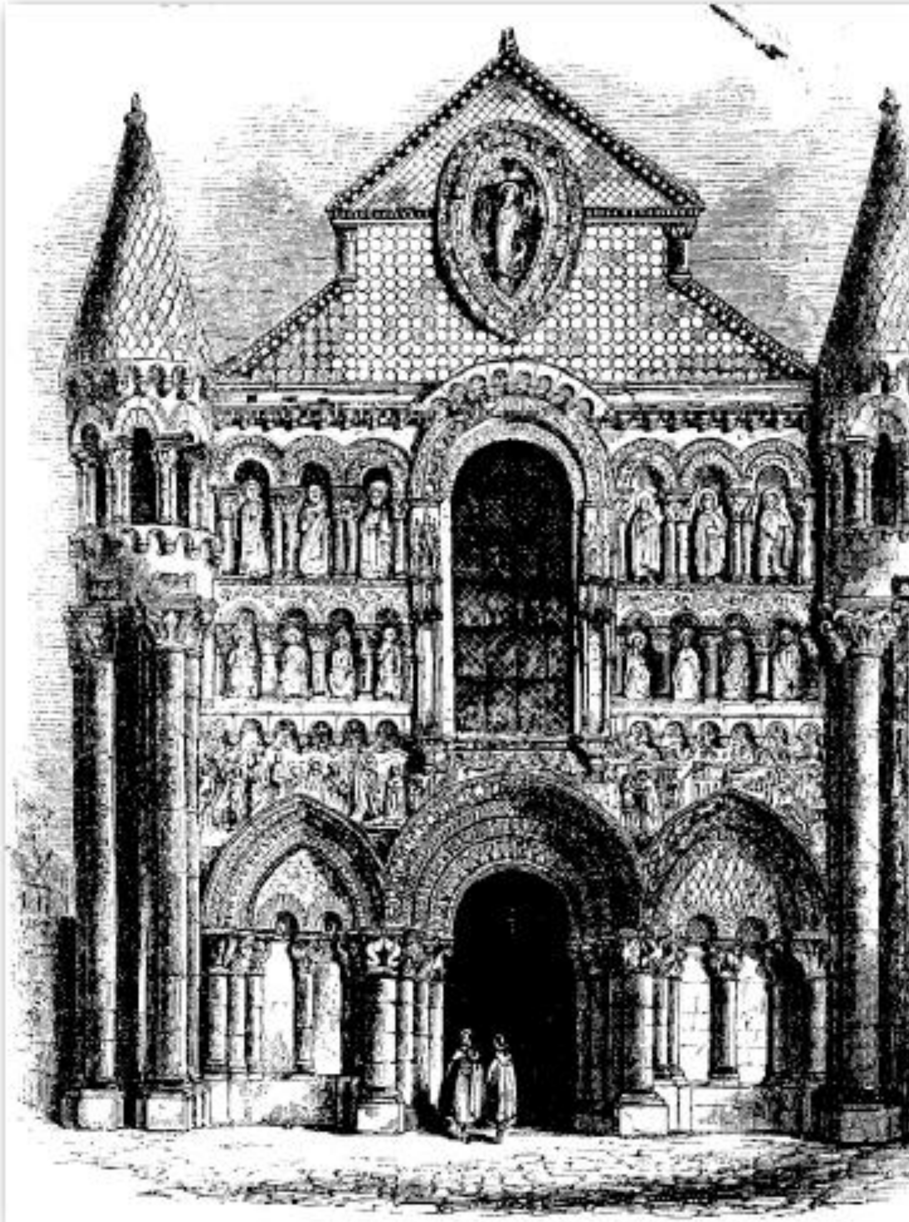



# Orthographic Projection

- Elevation: front, side, rear orthographic projection of an object
- Plan View: top orthographic projection of an object
- Application: engineering and architectural drawings

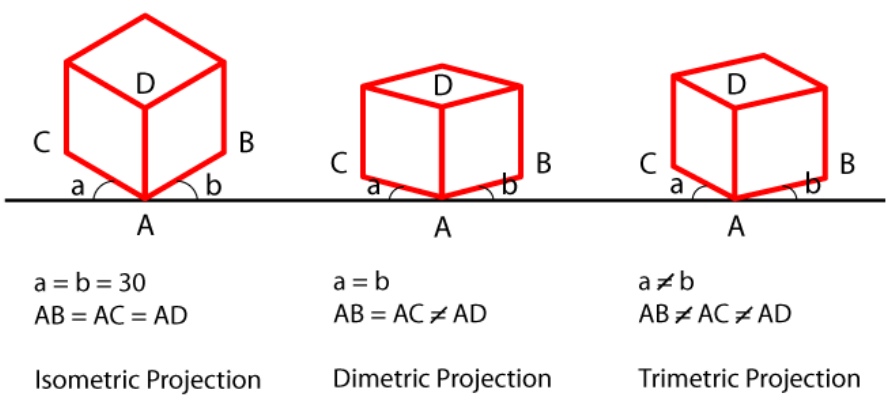


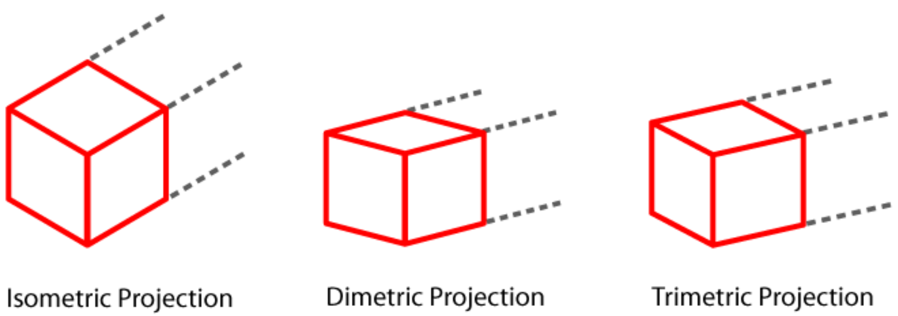
# Orthographic Projection



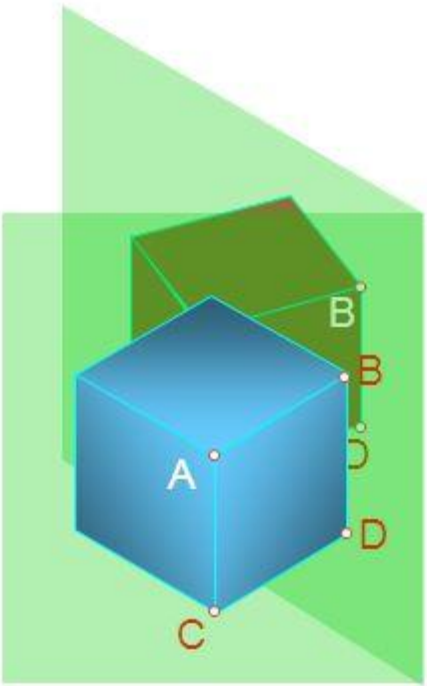
# Axonometric Orthographic Projection

- Display more than one face at a time
- **Isometric projection:** by aligning projection plane such that it intersects each coordinate axis in which the object is defined (principal axes) at the same distance from the origin



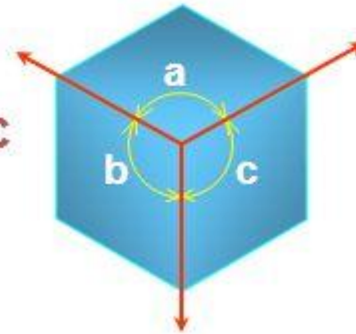



# Axonometric Projection




## Type of axonometric drawing

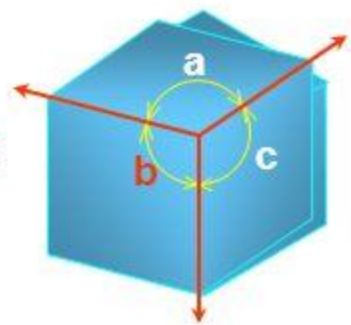
1. Isometric




**Axonometric axis**

All angles are equal.

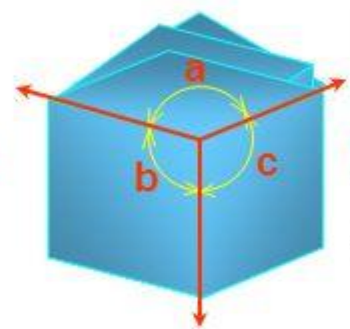
2. Dimetric




**Axonometric axis**

Two angles are equal.

3. Trimetric

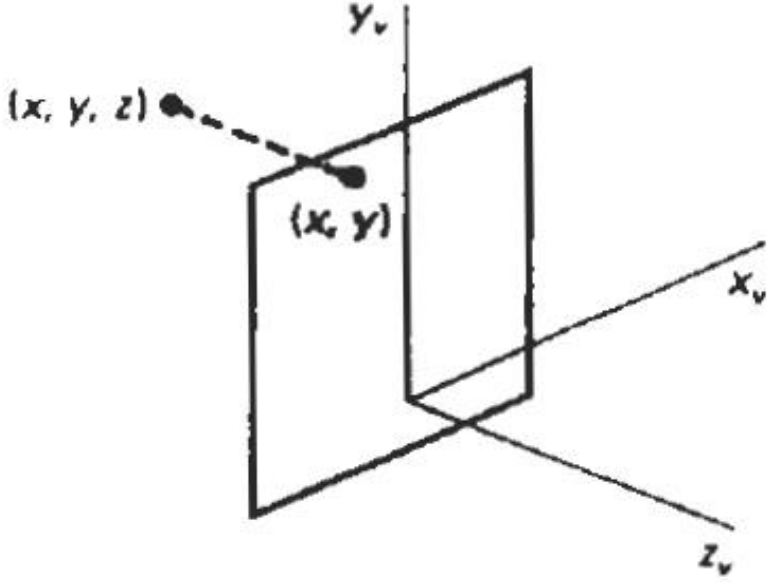



**Axonometric axis**

None of angles are equal.

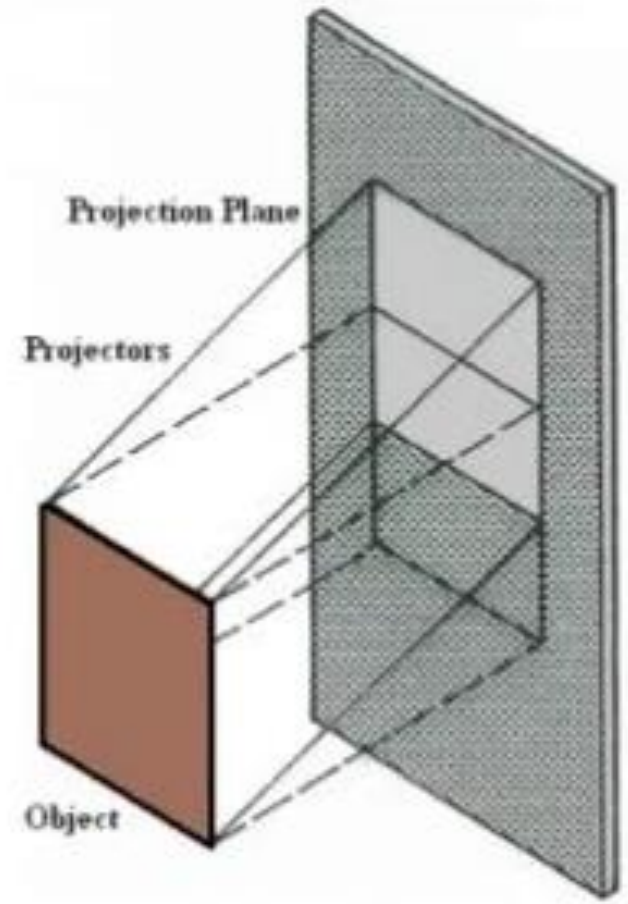
# Transformation Equation of Orthographic Parallel Projection

- View plane is placed at position $z_{vp}$ along the $z_v$ axis
- Any point (x,y,z) in viewing coordinate is transformed to projection coordinate as $x_p=x$; $y_p=y$
- Original z coordinate value is preserved for depth information needed in depth cueing and visible surface determination procedures



# Oblique Projections

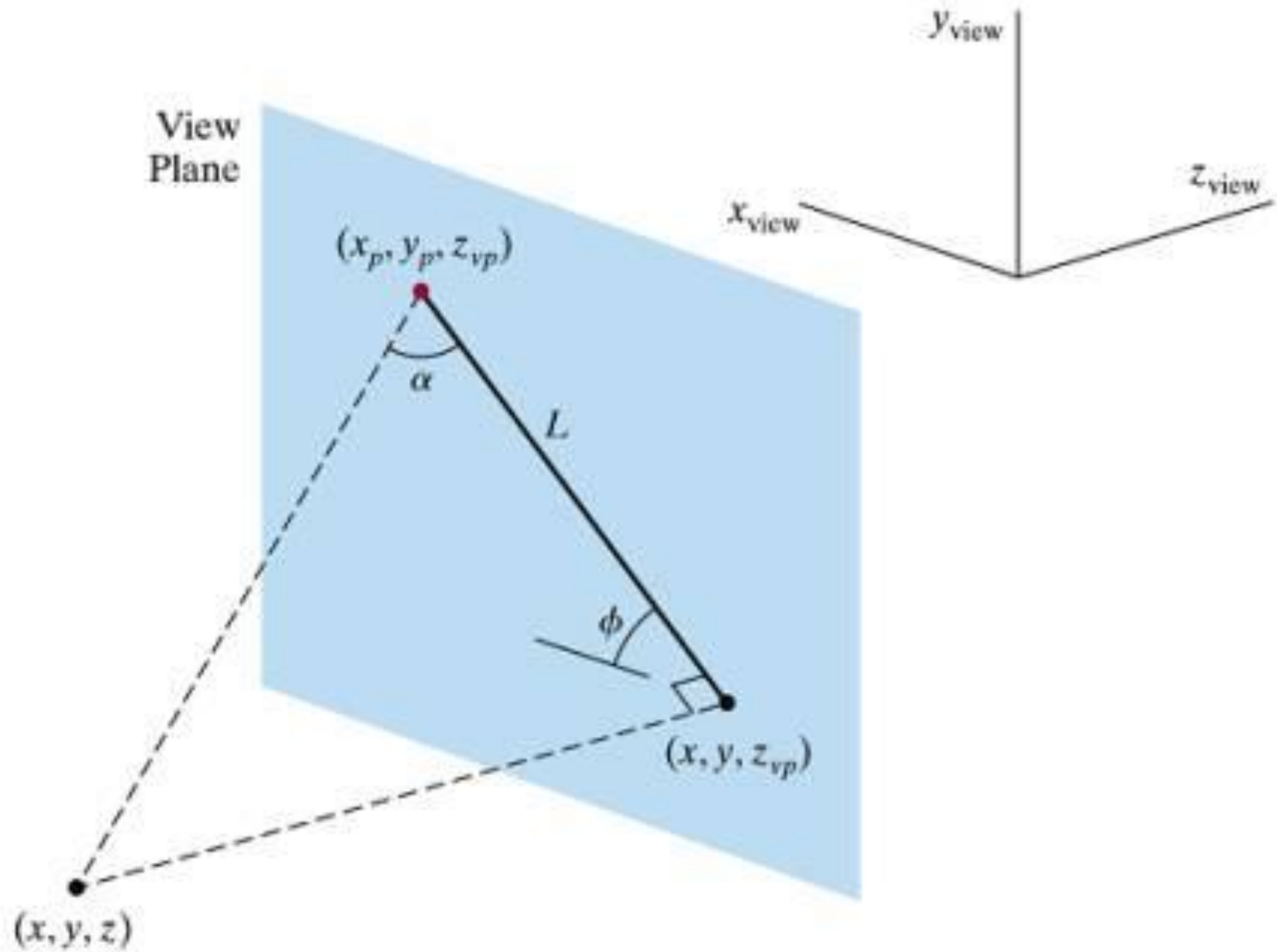
- *Projectors* are parallel to each other but not perpendicular to *projection plane*
- An oblique projection shows front and top surfaces that include the three dimensions of height, width, and depth.
- The front or principal surface of an object (the surface toward the plane of projection) is parallel to the plane of projection.
- Effective in pictorially representing objects



OBLIQUE PROJECTIONS

# Transformation Equation of Oblique Parallel Projection

- Oblique projection vector is specified by two angles: $\alpha$ and $\phi$
- Point (x, y, z) is projected at a position ($x_p$, $y_p$) on view plane
- Orthographic projection coordinates on the plane are (x, y)
- The oblique projection line from (x, y, z) to ($x_p$, $y_p$) makes an angle $\alpha$ with the line on the projection plane that joins ($x_p$, $y_p$) and (x, y)
- This line of length L, is at an angle $\phi$ with the horizontal direction on the projection plane
- Projection coordinates are defined in terms of x, y, L, $\phi$ as
- **$x_p = x + L \cos \phi$ and $y_p = y + L \sin \phi$**
- **$\tan \alpha = z/L$; $L = z/\tan \alpha$; $L = zL_1$ ($L_1 = 1/\tan \alpha$; if $z=1$, $L_1 = L$)**
- **$x_p = x + z(L_1 \cos \phi)$ and $y_p = y + z(L_1 \sin \phi)$**




Oblique parallel projection of coordinate position $(x, y, z)$ to position $(x_p, y_p, z_{vp})$ on a projection plane at position $z_{vp}$ along the $z_{\text{view}}$ axis.

# Transformation Equation of Oblique Parallel Projection

- The transformation matrix for producing any parallel projection onto the $x_v y_v$ plane is

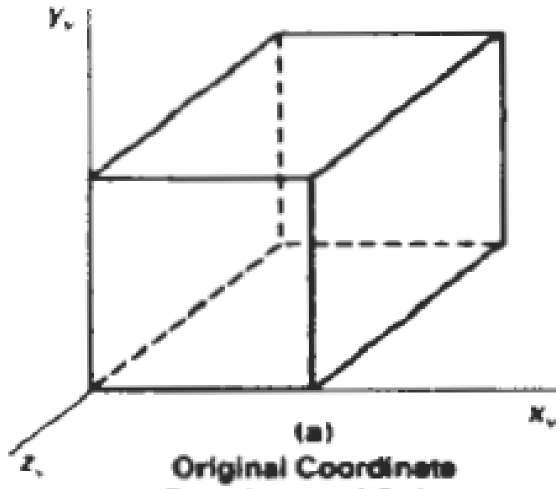
$$M_{parallel} = \begin{bmatrix} 1 & 0 & L_1 \cos\phi & 0 \\ 0 & 1 & L_1 \sin\phi & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- **$L_1 = 1$, $\alpha = 90^0$ -> orthographic projection**
- **Non zero value of $L_1$ => oblique projection**
- **Structure of projection matrix is similar to z-axis shear (shear planes of constant z and project on view plane; the x-, y-coordinate values within each plane of constant z are shifted by an amount proportional to the z-value of the plane so that angle, distance, parallel lines in the plane are projected accurately)**

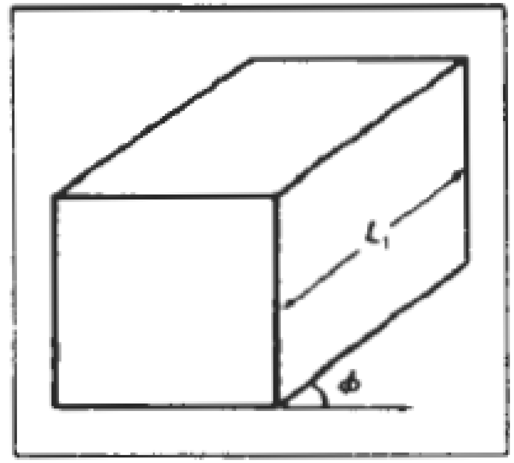
# Oblique Projection

- The back plane of the box is sheared and overlapped with the front plane in the projection to the viewing surface
- An edge of the box connecting the front and the back planes is projected into a line of length $L_1$ that makes an angle $\phi$ with a horizontal line in the projection plane
- Common choices for angle $\phi$: $45^0$ and $30^0$ (display combination of front side top/bottom of an object)

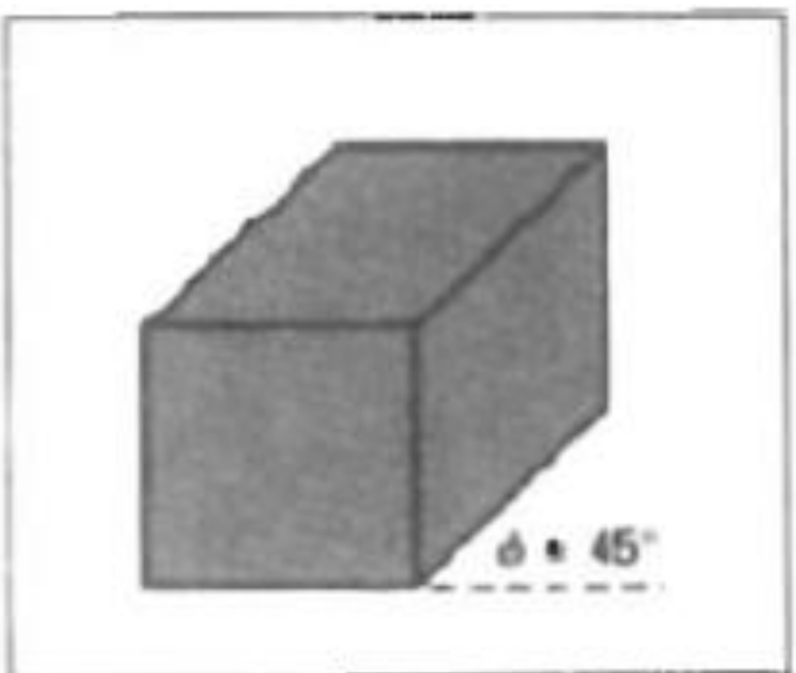
- Cavalier projection: **$\alpha$ = $45^0$, tan $\alpha$=1 (all line perpendicular to the projection plane are projected with no change in length)**
- Cabinet projection: **$\alpha$ = $63.4^0$, tan $\alpha$=2 (lines perpendicular to the viewing surface are projected one-half their length -> more realistic)**

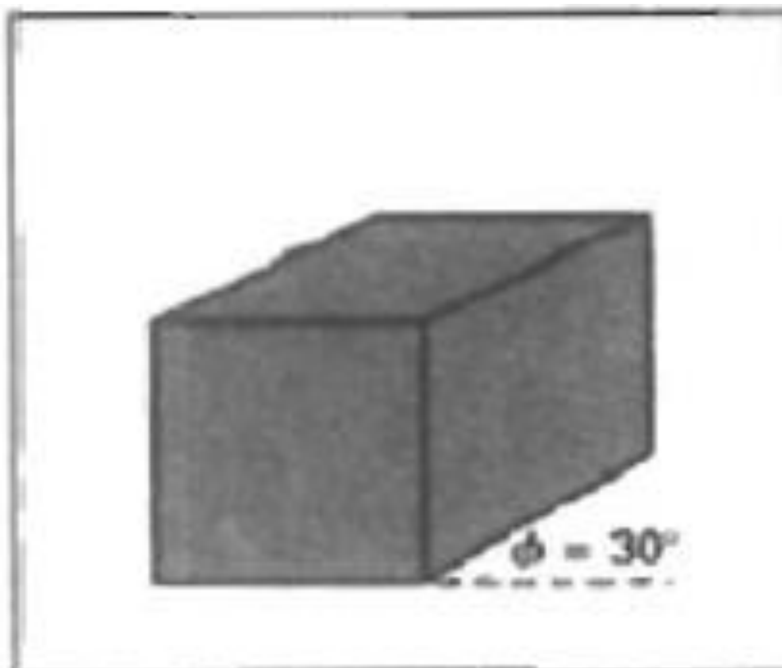


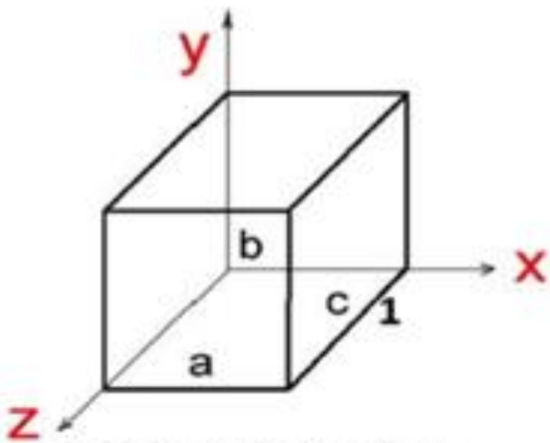

(a)
Original Coordinate
Description of Object



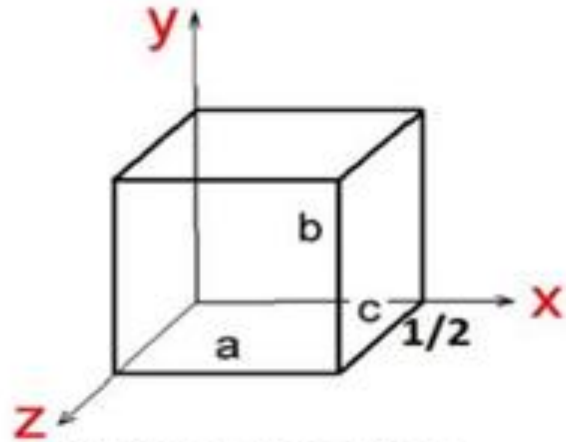

(b)
Projection on the
Viewing Plane



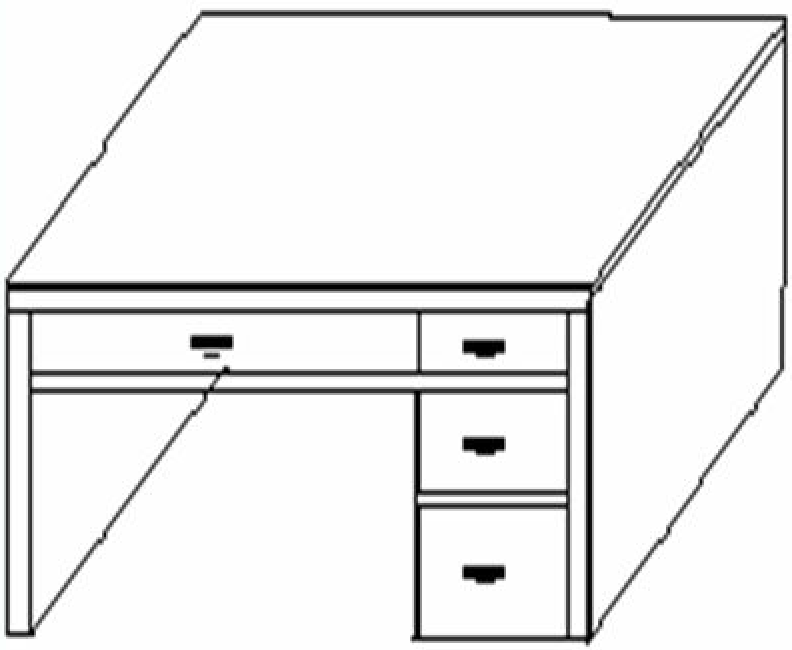



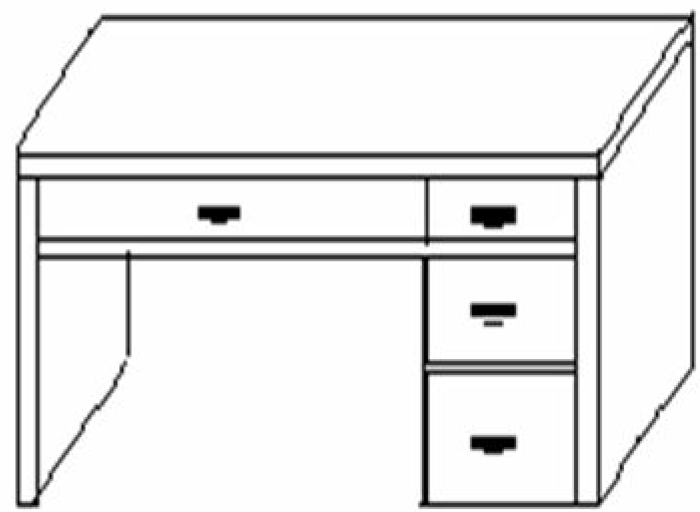



**Cavalier Projection**




**Cabinet Projection**





Cavalier

Cabinet

# Perspective Projection

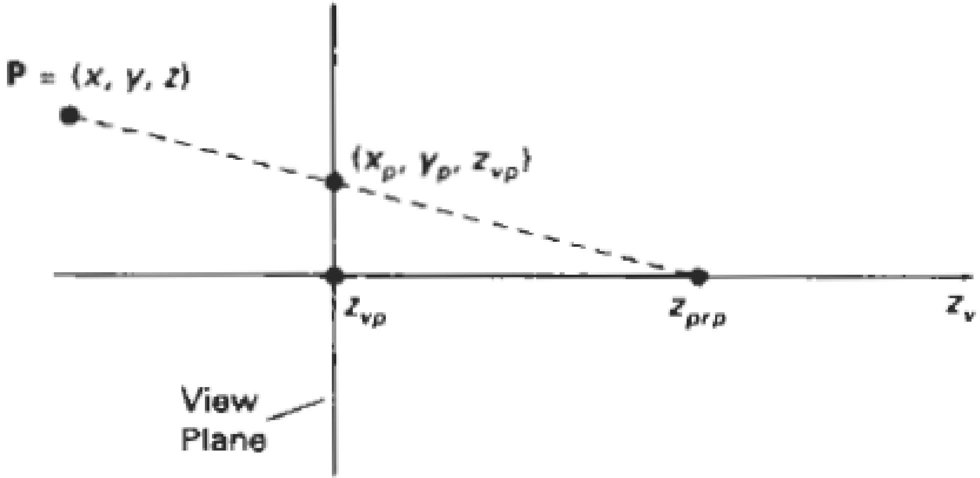
- Set projection reference point at position $z_{prp}$ along $z_v$ axis
- Place view plane at $z_{vp}$
- Coordinate positions along the perspective projection line in parametric form

$$x' = x - xu$$

$$y' = y - yu$$

$$z' = z - (z - z_{prp})u$$

- Parameter u takes value from 0 to 1
- Coordinate position (x', y', z') represents an point along projection line



# Perspective Projection

- When u=0, we are at position P=(x,y,z)
- When u=1, we have the projection reference coordinates $(0,0,z_{prp})$
- On the view plane, $z'=z_{vp}$
- We can solve z' equation for parameter u at this position along the projection line

$$u = \frac{z_{vp} - z}{z_{prp} - z}$$

- After substitution:
- $d_p = z_{prp} - z_{vp}$, the distance of the view plane from the projection reference point

$$x_p = x\left(\frac{z_{prp} - z_{vp}}{z_{prp} - z}\right) = x\left(\frac{d_p}{z_{prp} - z}\right)$$

$$y_p = y\left(\frac{z_{prp} - z_{vp}}{z_{prp} - z}\right) = y\left(\frac{d_p}{z_{prp} - z}\right)$$

# Perspective Projection

- Using 3D homogeneous coordinate representations:

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

- Homogeneous factor is h: $h = \dfrac{z_{prp} - z}{d_p}$

- The projection coordinates on the view plane are calculated from homogeneous coordinates $x_p = x_h/h, \qquad y_p = y_h/h$
- Original z coordinate value is retained

# Perspective Projection

- In general, the projection reference point does not have to be along $z_v$-axis
- The coordinate position can be $(x_{prp}, y_{prp}, z_{prp})$
- View plane is taken to be uv plane: $z_{vp}=0$, the projection coordinates are:

$$x_p = x\left(\frac{z_{prp}}{z_{prp} - z}\right) = x\left(\frac{1}{1 - z/z_{prp}}\right)$$

$$y_p = y\left(\frac{z_{prp}}{z_{prp} - z}\right) = y\left(\frac{1}{1 - z/z_{prp}}\right)$$

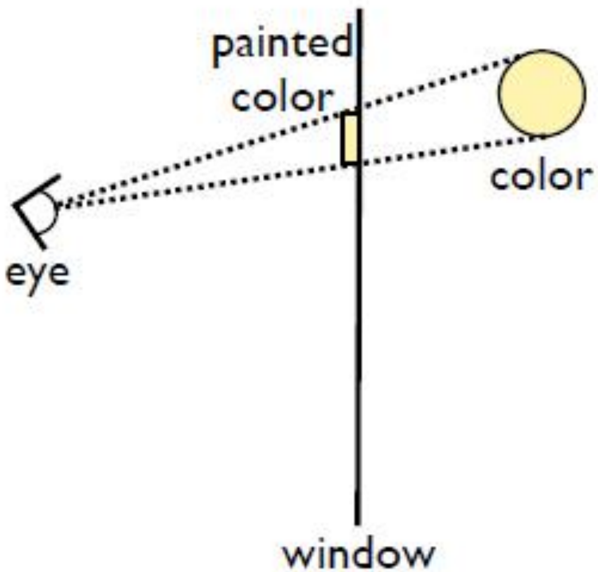
# Perspective Projection

- The projection reference point is taken to be $z_{prp}=0$ in some graphics packages, the projection coordinates are:

$$x_p = x\left(\frac{z_{vp}}{z}\right) = x\left(\frac{1}{z/z_{vp}}\right)$$

$$y_p = y\left(\frac{z_{vp}}{z}\right) = y\left(\frac{1}{z/z_{vp}}\right)$$

The simplest way to look at perspective projection is as painting on a window....




Paint on the window whatever color you see there.

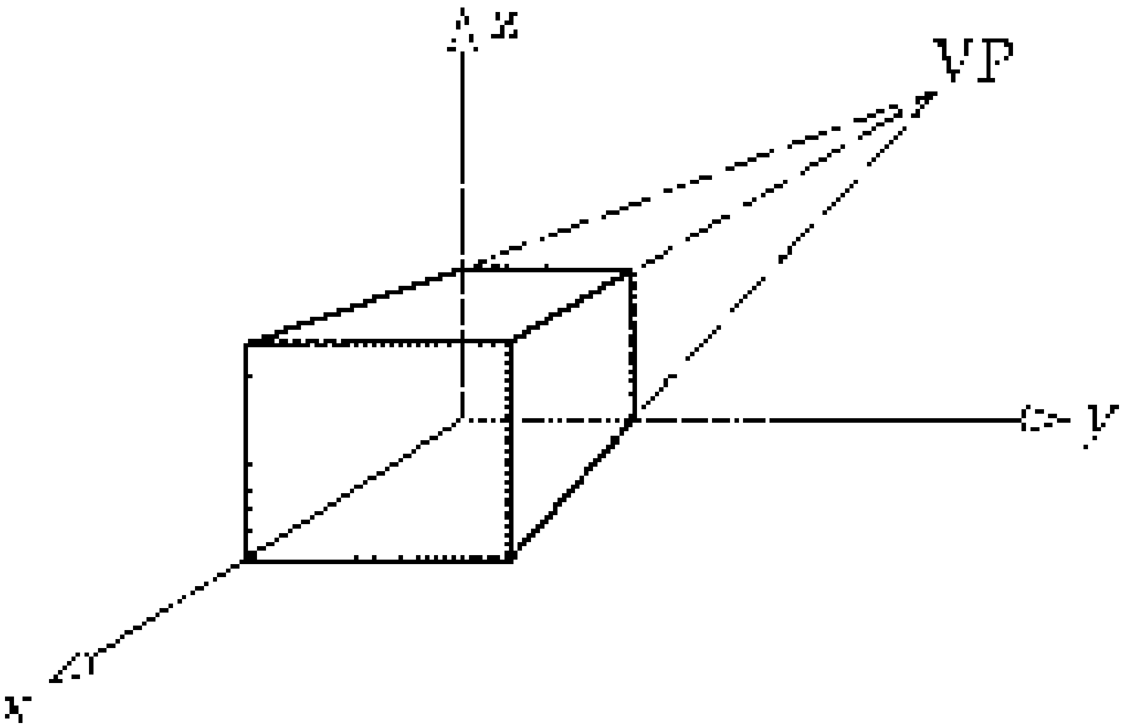
source: http://blog.mlive.com/flintjournal/newsnow/2007/11/WINDOW_PAINTING_02.jpg

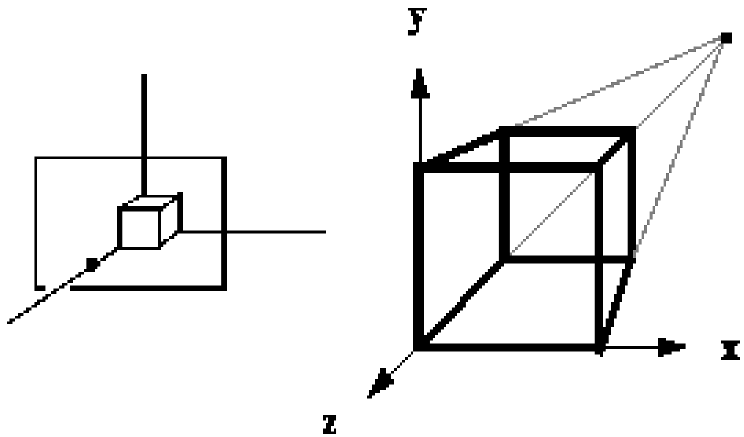
# Perspective Projection

- Set of parallel lines in the object are not parallel to the plane are projected into converging lines
- The point at which the set of projected lines appears to converge is called vanishing point
- Each such projected parallel lines will have a separate vanishing point
- A scene can have any number of vanishing points, depending on how many sets of parallel lines are there in the scene
- The vanishing point for any set of lines that are parallel to one of the principal axes of an object is referred to as principal vanishing point
- The number of principal vanishing points can be controlled -> 1, 2, or 3, with the orientation of the projection plane

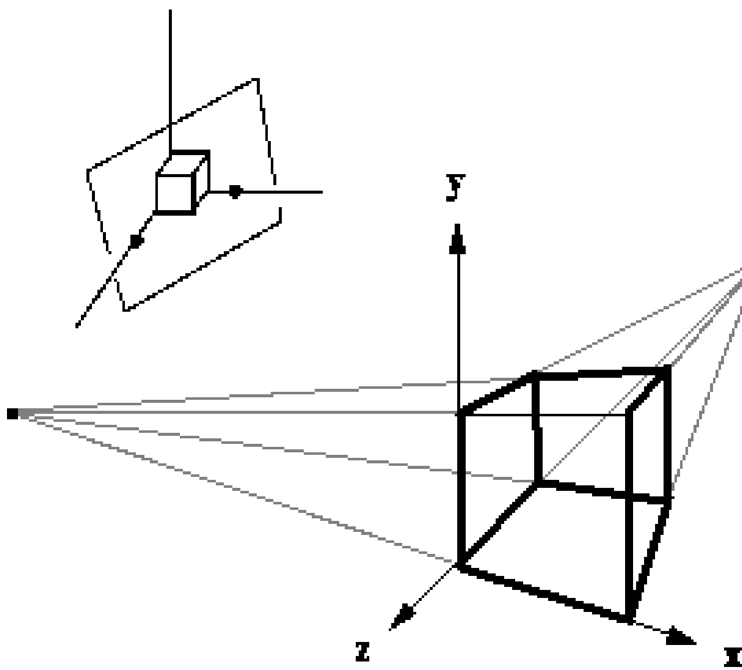
# Perspective Projection

- Accordingly, perspective projections are classified to
- 1-point, 2-point, or 3-point projections
- The number of principal vanishing points in a projection is determined by the number of principal axes intersecting the view plane

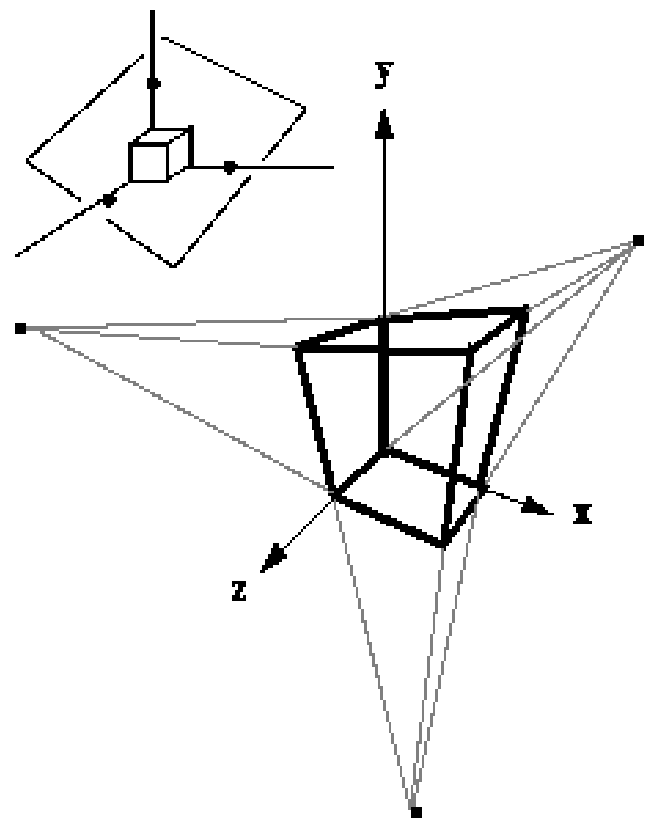




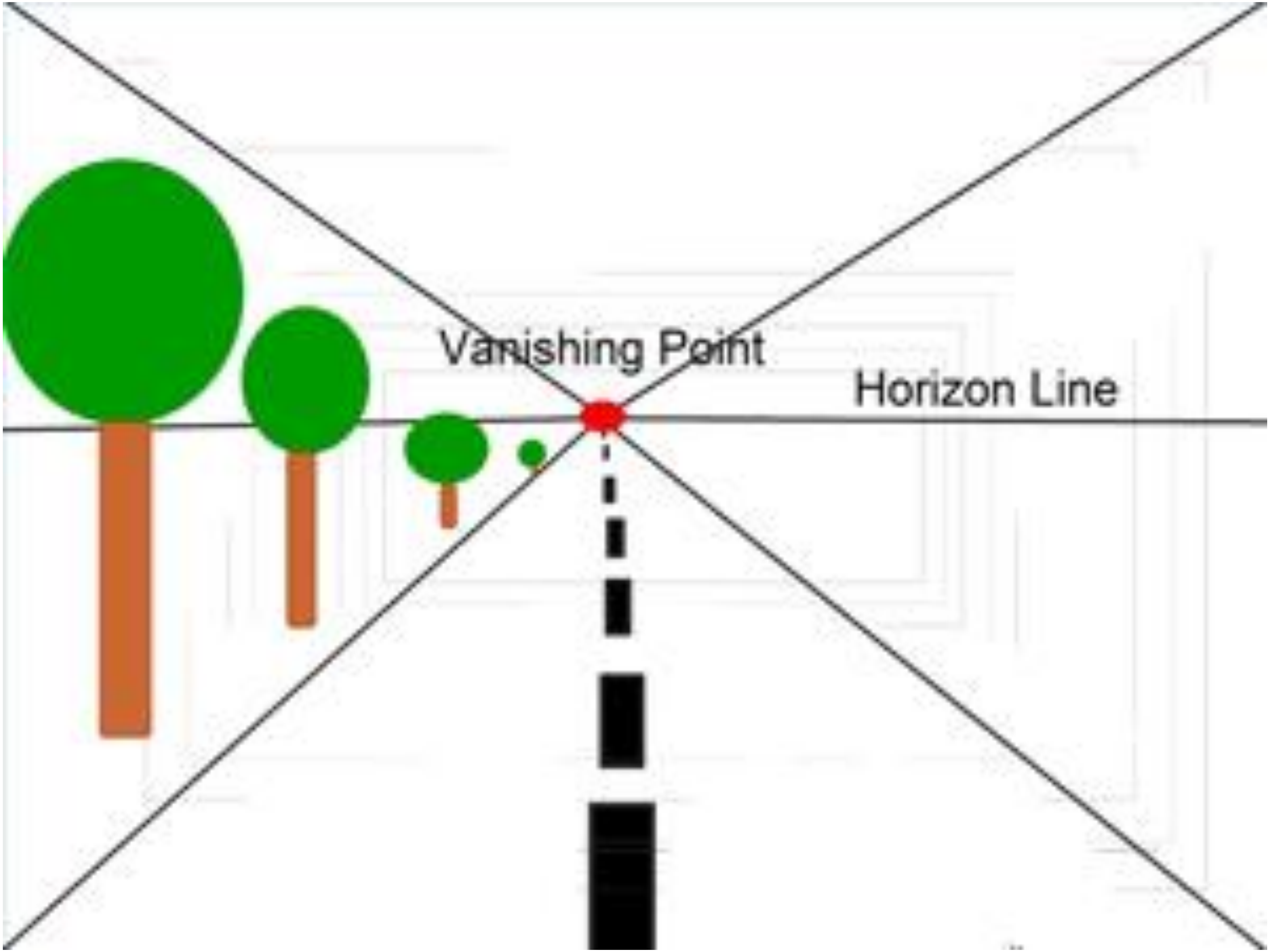

*One Point Perspective*
(z-axis vanishing point)

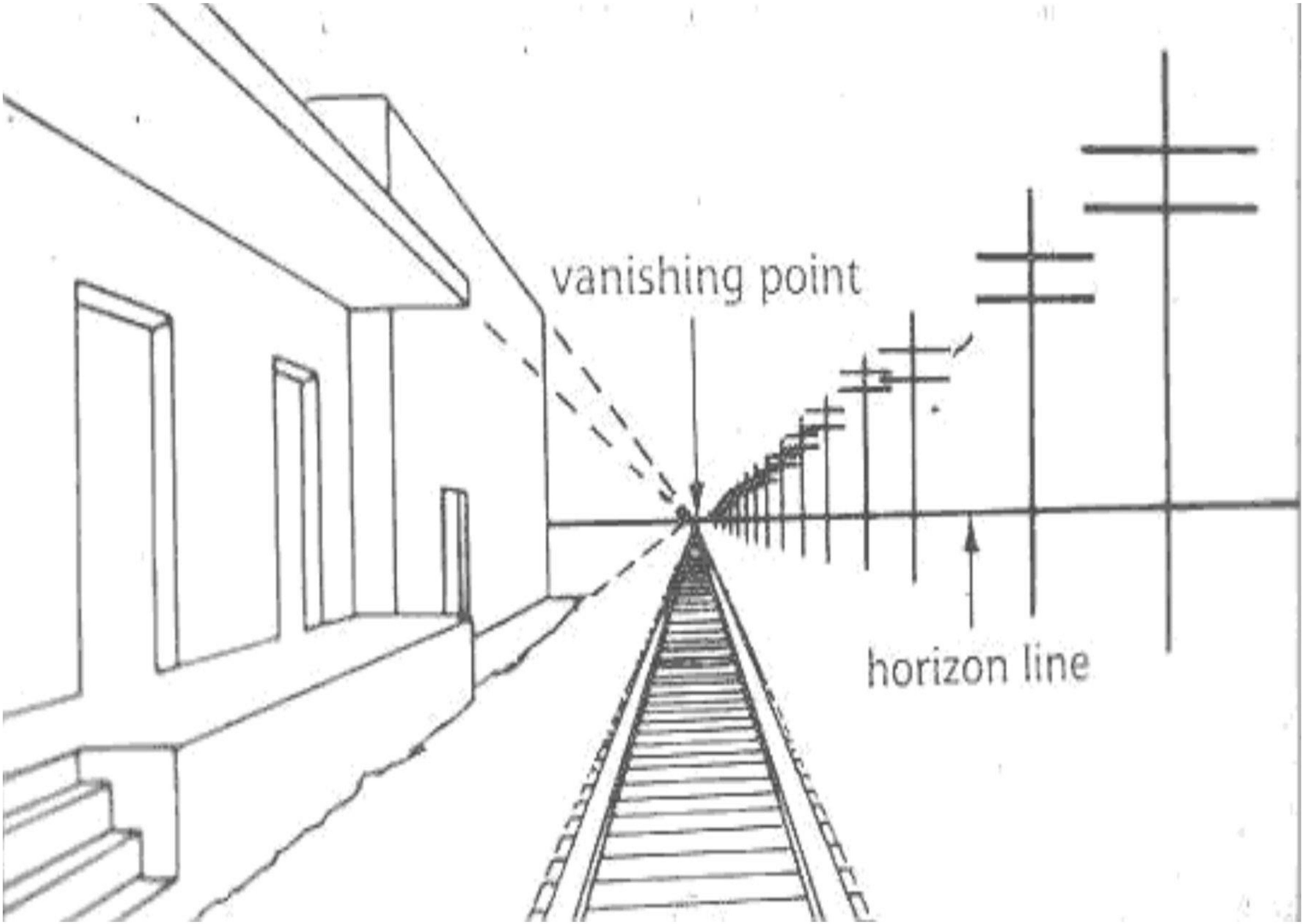


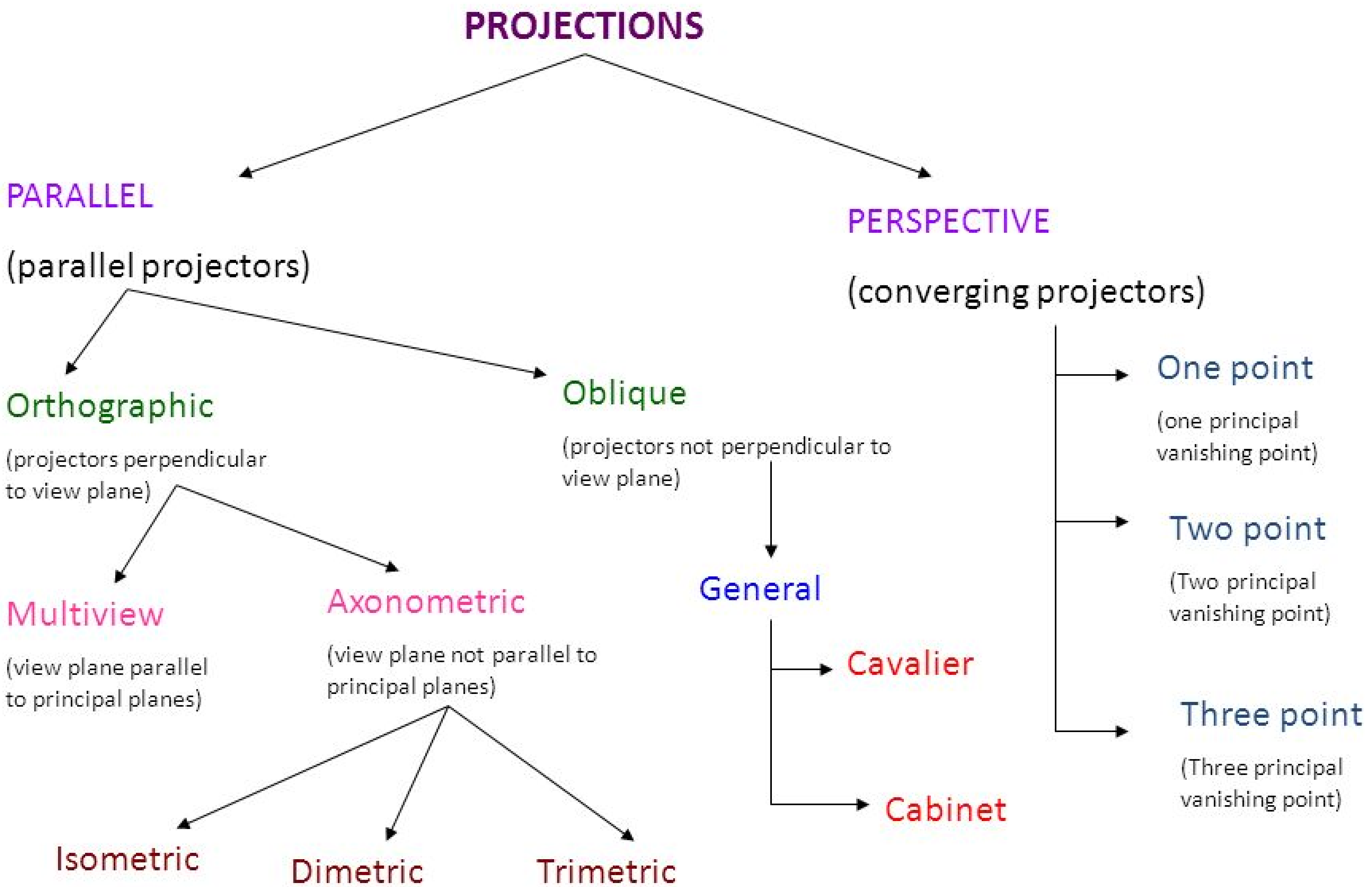

*Two Point Perspective*
z, and x-axis vanishing points




*Three Point Perspective*
(z, x, and y-axis
vanishing points)


Vanishing Point
Horizon Line


vanishing point
horizon line


PROJECTIONS
PARALLEL
(parallel projectors)
PERSPECTIVE
(converging projectors)
Orthographic
(projectors perpendicular to view plane)
Oblique
(projectors not perpendicular to view plane)
One point
(one principal vanishing point)
Two point
(Two principal vanishing point)
Three point
(Three principal vanishing point)
General
Cavalier
Cabinet
Multiview
(view plane parallel to principal planes)
Axonometric
(view plane not parallel to principal planes)
Isometric
Dimetric
Trimetric

# 3D Clipping

- Identifies and saves all surface segments within the view volume for display on the output device
- All parts that are outside the view volume are discarded
- Extension of 2D clipping method
- Clip against boundary plane of view volumes
- To clip a line segment: test the relative position of the line using the view volume's boundary plane equations
- Determine whether the end points are inside or outside
- Outside: $Ax+By+Cz+D > 0$
- Inside: $Ax+By+Cz+D < 0$
- A, B, C, D are plane parameters
- The line which is totally outside are discarded and totally inside are saved, otherwise intersection is calculated

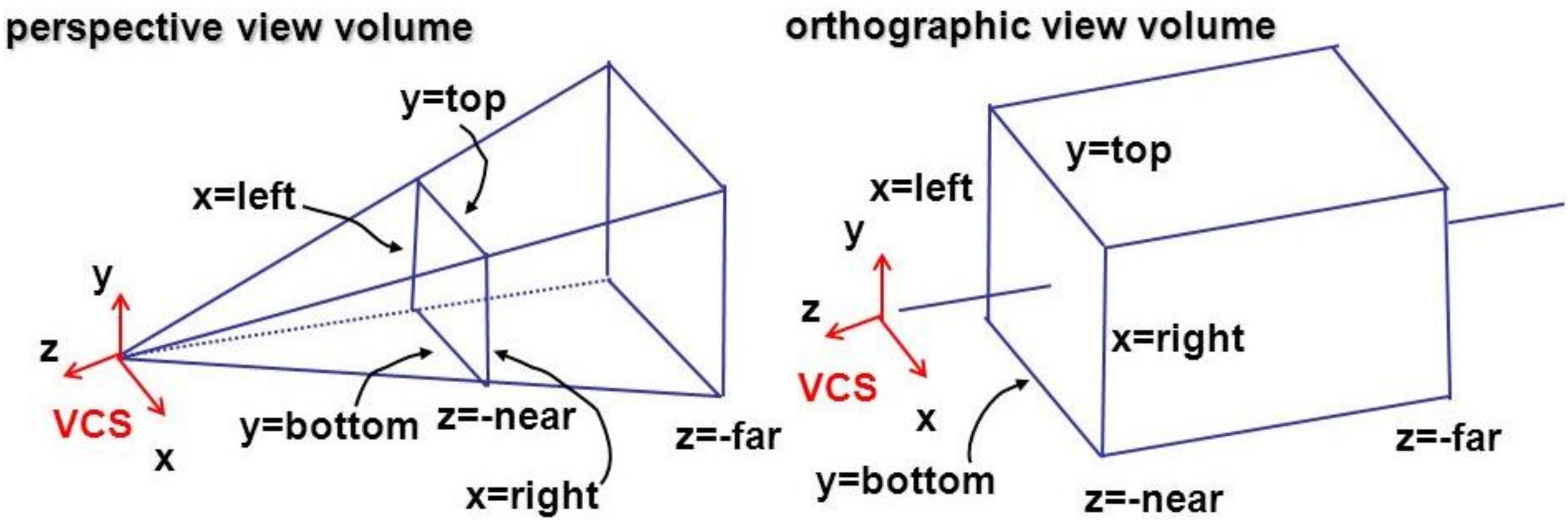
# 3D Clipping

- Let $(x_1, y_1, z_1)$ be the intersection point and lies on the line, then
- $Ax_1+By_1+Cz_1+D=0$
- To clip a polygon surface, we can clip the individual polygon edges

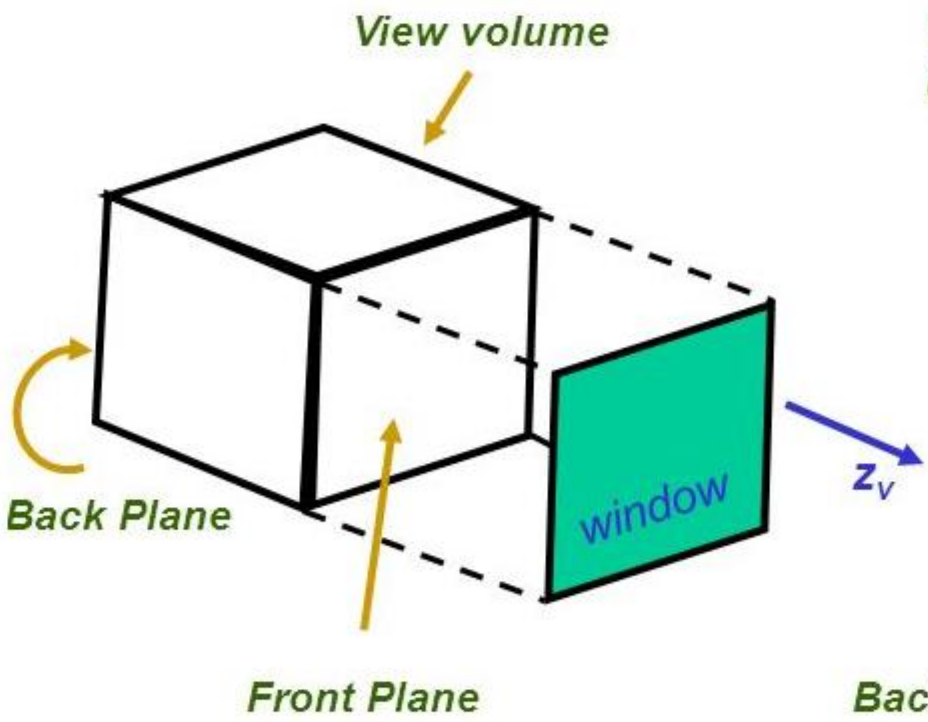
1. Test coordinate extents against each boundary of view volume to determine whether the object is completely inside or outside that boundary
2. If the coordinate extents of the object are inside all boundaries, save it
3. If outside discard it
4. Otherwise perform intersection calculations

Projection can occur before or after view volume clipping

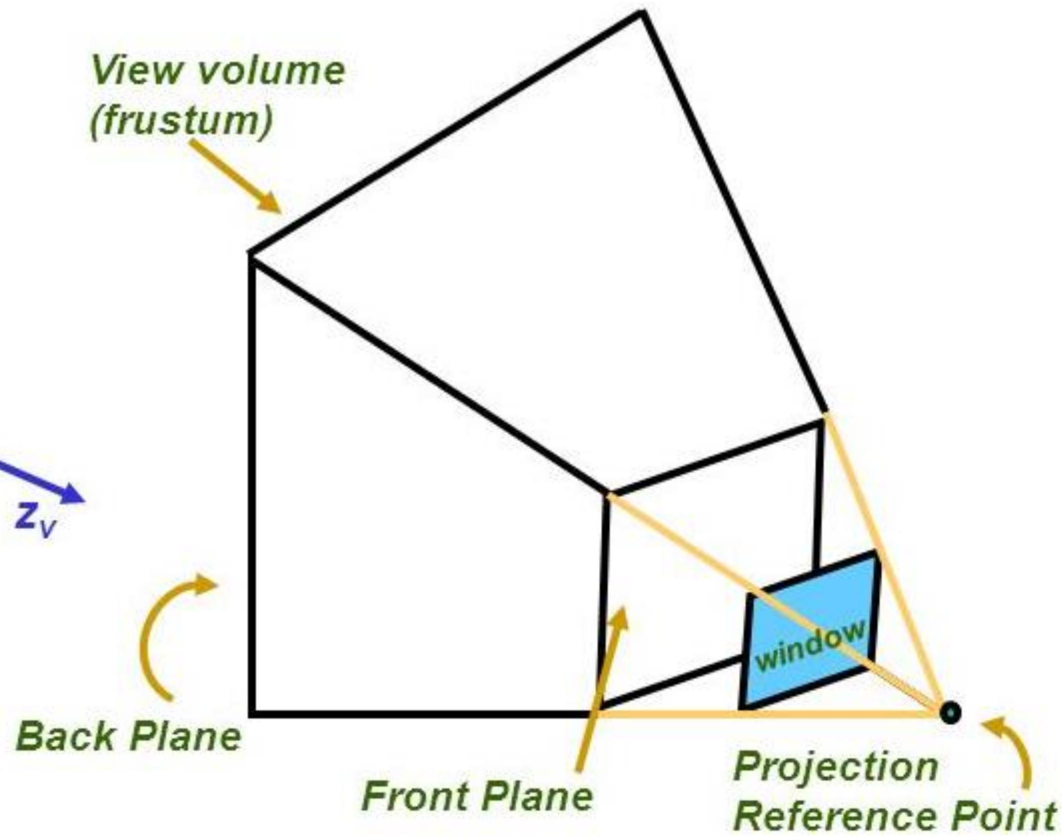
# View Volumes



# View Volumes

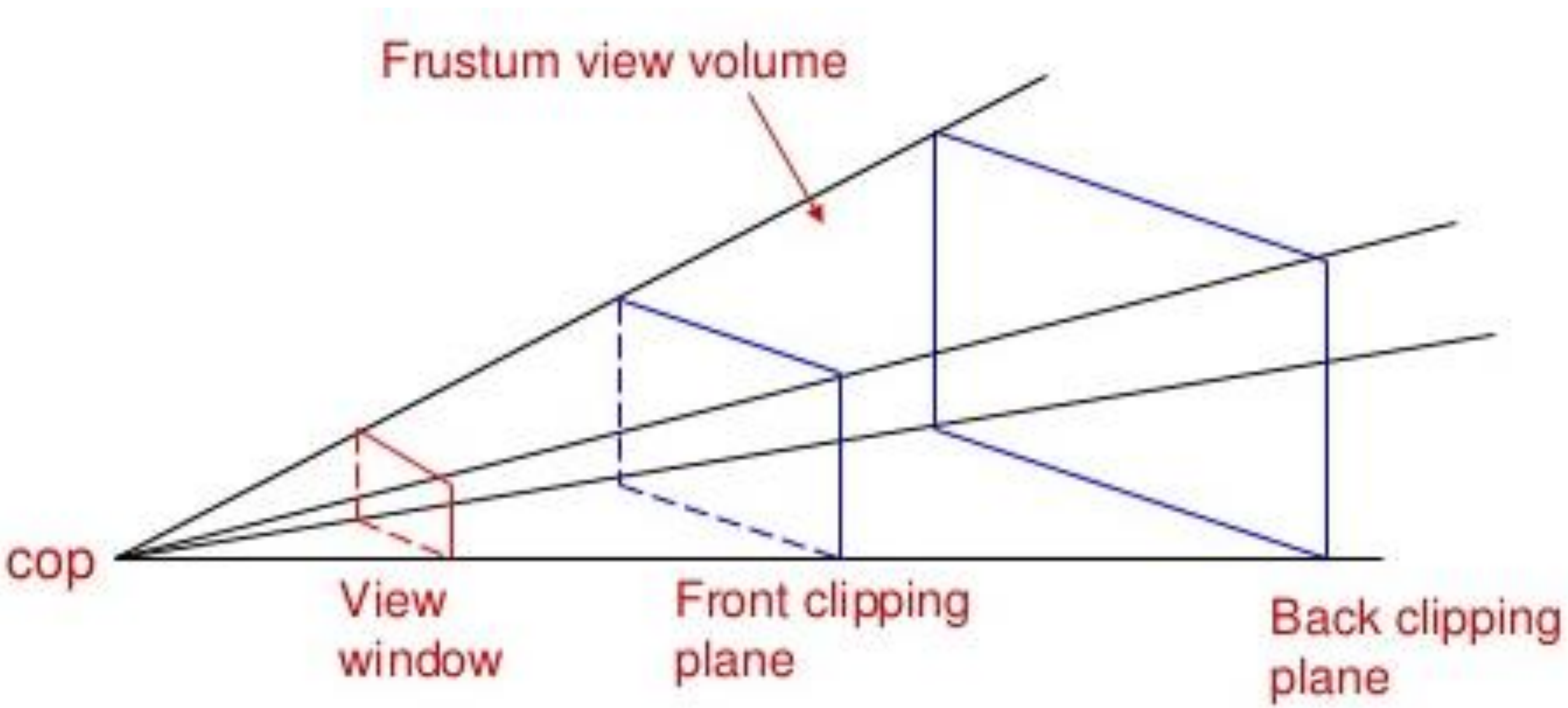



**Parallel Projection**

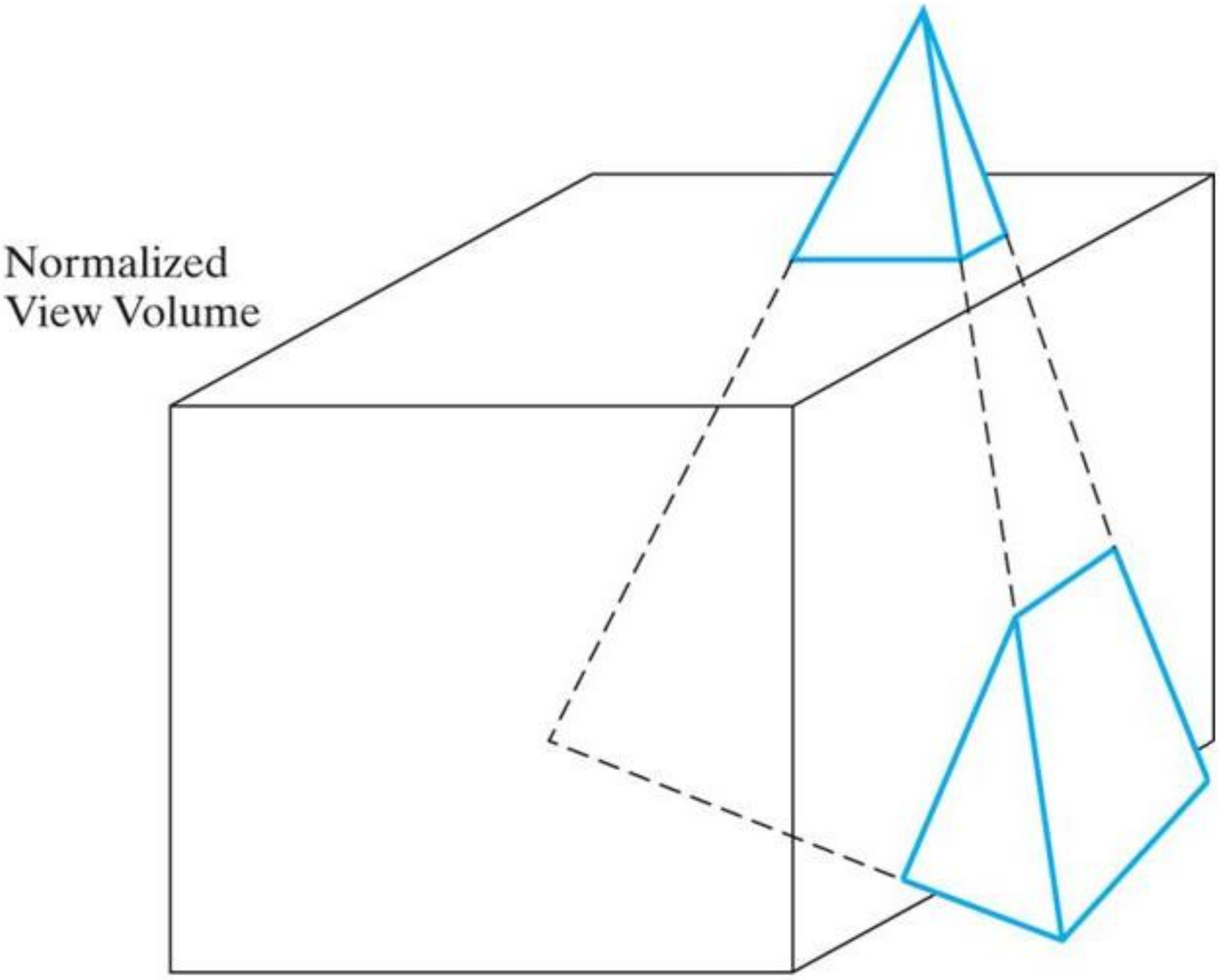



**Perspective Projection**

# View Volumes



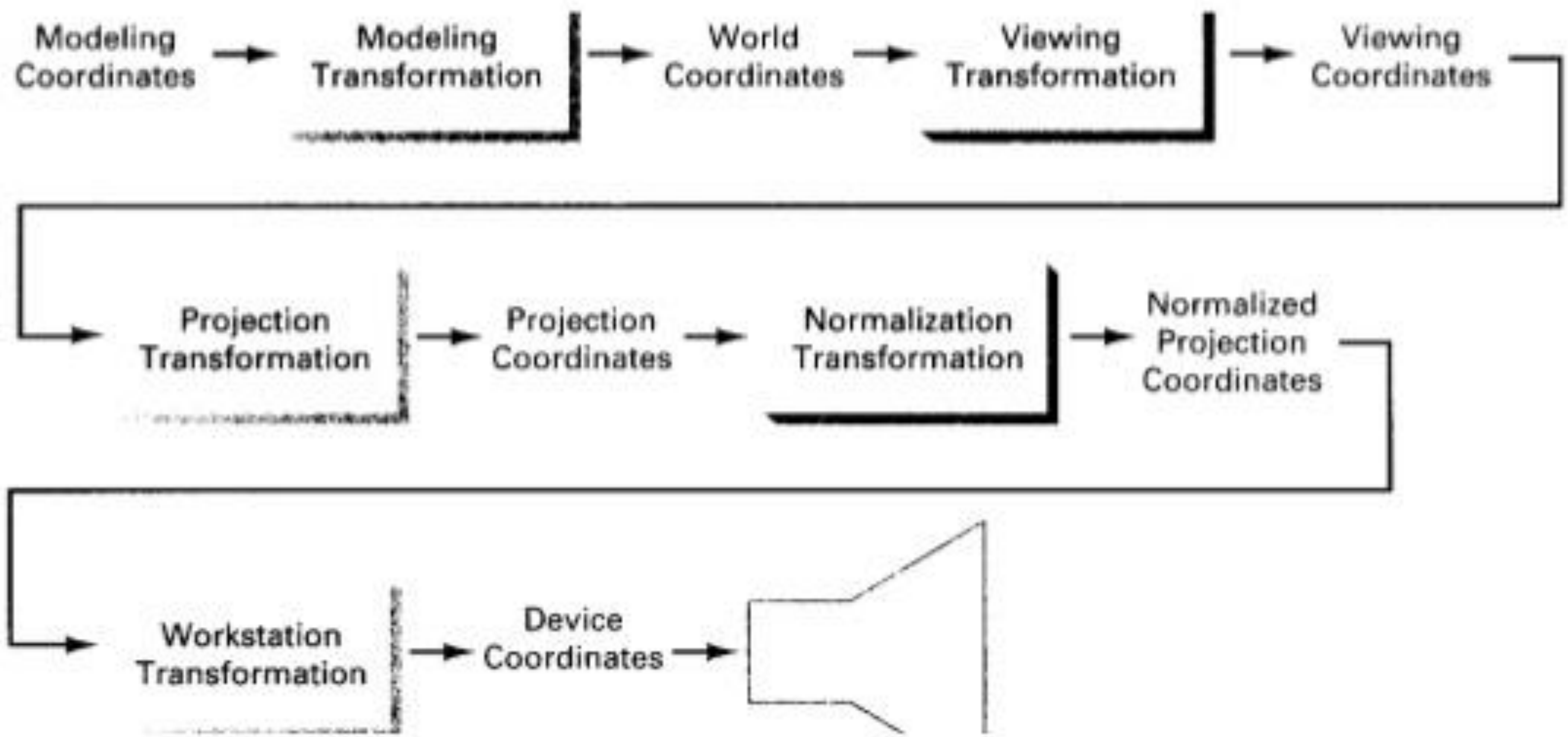
# 3D Clipping



# Normalized View Volume

- At projection stage, the viewing coordinates are transformed to projection coordinates, which effectively converts the view volume into a rectangular parallelepiped
- The parallelepiped is mapped into a unit cube, a normalized view volume, called the normalized projection coordinate system
- Accomplished by transforming points within the rectangular parallelepiped into a position within a specified 3D viewport, which occupies all or part of the unit cube
- Normalized view volume: x=0, x=1, y=1, y=1, z=0, z=1

# Transformation Pipeline



# Advantages of Clipping Against Unit Cube

- Normalized view volume provides a standard shape for representing any sized view volume
- It separates viewing transformation from any workstation consideration
- Unit cube can be mapped to a workstation of any size
- Clipping procedures are simplified and standardized with unit clipping planes or the viewport planes
- Additional clipping planes can be specified within the normalized space before transforming to device coordinates
- Depth-cueing and visible-surface determination are simplified (z axis is always point to the viewer)
- Prp has now been transformed to the z-axis
- Positive z-axis: front faces; Negative z-axis: back faces

# 3D Clipping

- Mapping positions within a rectangular view volume to a three dimensional rectangular viewport is achieved with a combination of scaling and translation

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

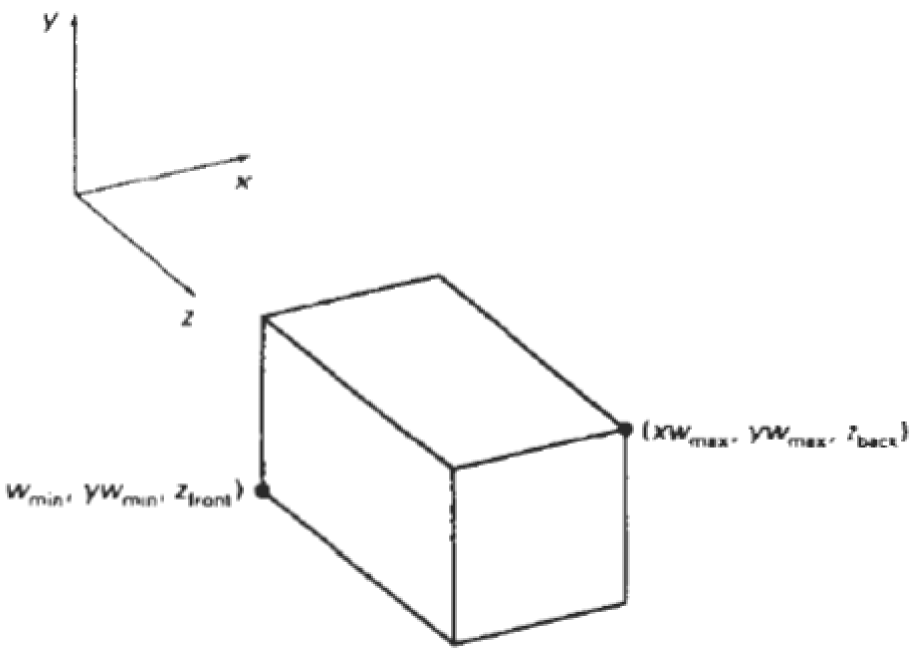
- $D_x$, $D_y$, $D_z$ are the ratios of the dimensions of the viewport and regular parallelepiped view volume in the x, y, and z directions

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

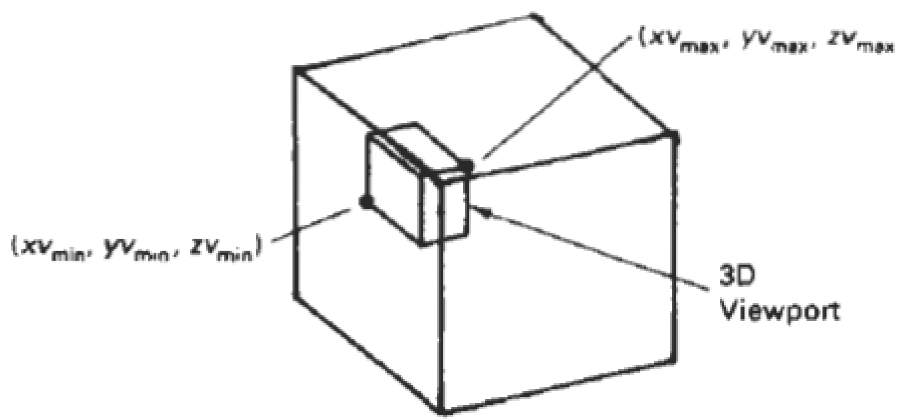
$$D_y = \frac{yv_{max} - yv_{min}}{yw_{max} - yw_{min}}$$

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

# 3D Clipping




Parallelepiped
View Volume




Unit Cube

# 3D Clipping

- Additive translation factors:

$$K_x = xv_{min} - xw_{min} D_x$$

$$K_y = yv_{min} - yw_{min} D_y$$

$$K_z = zv_{min} - z_{front} D_z$$

- Region code for a point (x,y,z)

$$\text{bit } 1 = 1, \quad \text{if } x < xv_{min} (\text{left})$$

$$\text{bit } 2 = 1, \quad \text{if } x > xv_{max} (\text{right})$$

$$\text{bit } 3 = 1, \quad \text{if } y < yv_{min} (\text{below})$$

$$\text{bit } 4 = 1, \quad \text{if } y > yv_{max} (\text{above})$$

$$\text{bit } 5 = 1, \quad \text{if } z < zv_{min} (\text{front})$$

$$\text{bit } 6 = 1, \quad \text{if } z > zv_{max} (\text{back})$$

# 3D Clipping

- Equations with parameters:

$$x = x_1 + (x_2 - x_1)u, \qquad 0 \le u \le 1$$

$$y = y_1 + (y_2 - y_1)u$$

$$z = z_1 + (z_2 - z_1)u$$

- $P_1(x_1,y_1,z_1)$ and $P_2(x_2,y_2,z_2)$ are two end points

(x,y,z) be any point on the line; u=0 -> $P_1$ and u=1 -> $P_2$

- Lets us test the line against the $zv_{min}$ plane of the viewport

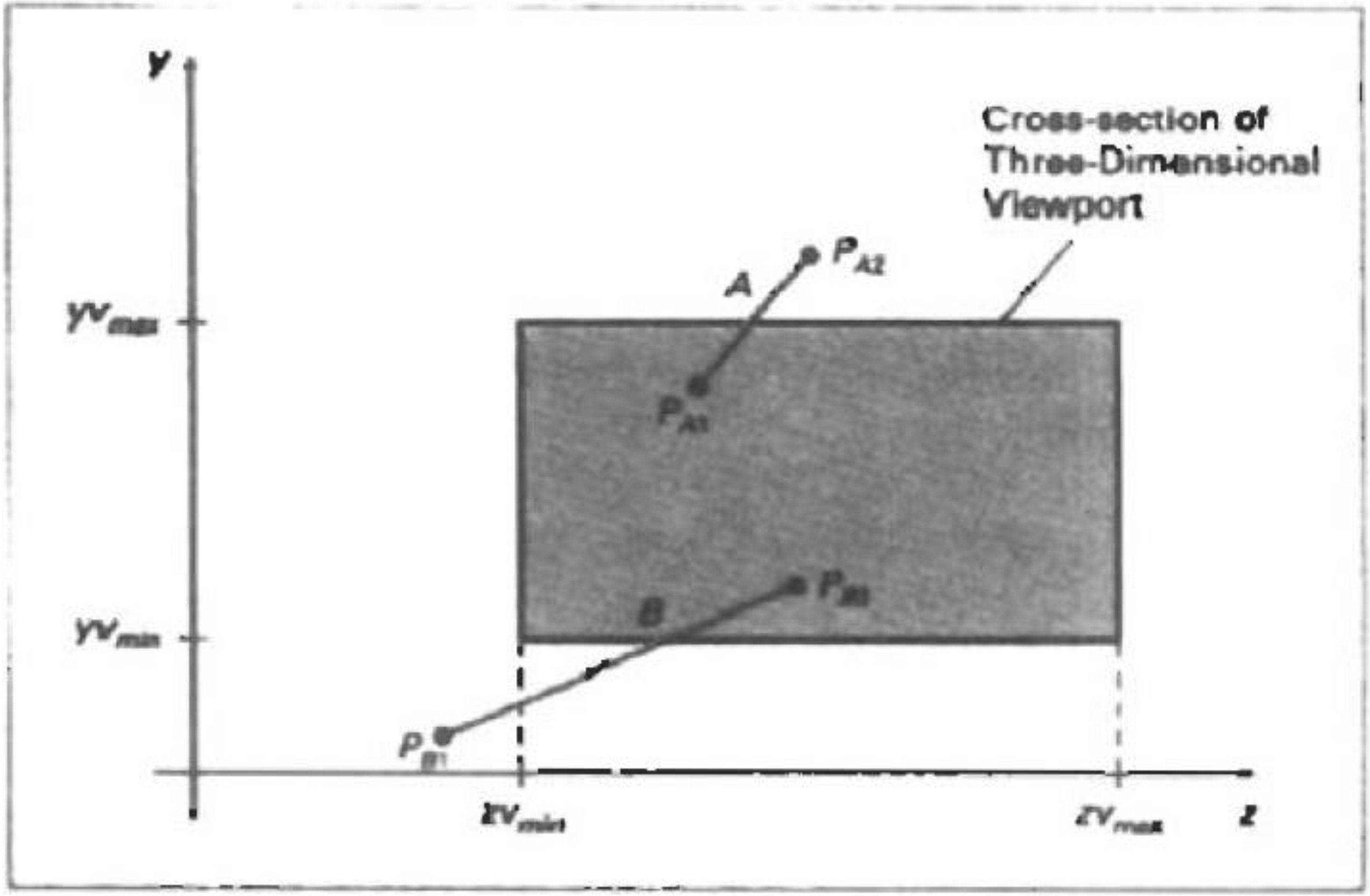
$$u = \frac{zv_{min} - z_1}{z_2 - z_1}$$

$$x_I = x_1 + (x_2 - x_1)\left(\frac{zv_{min} - z_1}{z_2 - z_1}\right)$$

$$y_I = y_1 + (y_2 - y_1)\left(\frac{zv_{min} - z_1}{z_2 - z_1}\right)$$

# 3D Clipping

- If either $x_1$ or $y_1$ is not in the range of the boundaries of the viewport, then this line intersects the front plane beyond the boundaries of the volume



# 3D Clipping

- The various transformations are applied and we obtain the final homogeneous point:
- (h can have any real value but not be zero or small value)
- After clipping homogeneous coordinates are converted to non homogeneous coordinates

$$\begin{bmatrix} x_h \\ y_h \\ z_h \\ h \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$x' = \frac{x_h}{h}, \qquad y' = \frac{y_h}{h}, \qquad z' = \frac{z_h}{h}$$

# 3D Clipping

- Mapping to device coordinates (z component is used for other purposes) -> xy position is plotted
- For parallel projection; h=1
- For perspective projection clip homogeneous coordinates to carry out clipping correctly
- Inside the viewport and clipping limits:

$$xv_{\min} \le \frac{x_h}{h} \le xv_{\max}, \qquad yv_{\min} \le \frac{y_h}{h} \le yv_{\max}, \qquad zv_{\min} < \frac{z_h}{h} \le zv_{\max}$$

$$h\,xv_{\min} \le x_h \le h\,xv_{\max}, \quad h\,yv_{\min} \le y_h \le h\,yv_{\max}, \quad h\,zv_{\min} \le z_h \le h\,zv_{\max}, \quad \text{if } h > 0$$

$$h\,xv_{\max} \le x_h \le h\,xv_{\min}, \quad h\,yv_{\max} \le y_h \le h\,yv_{\min}, \quad h\,zv_{\max} \le z_h \le h\,zv_{\min}, \quad \text{if } h < 0$$