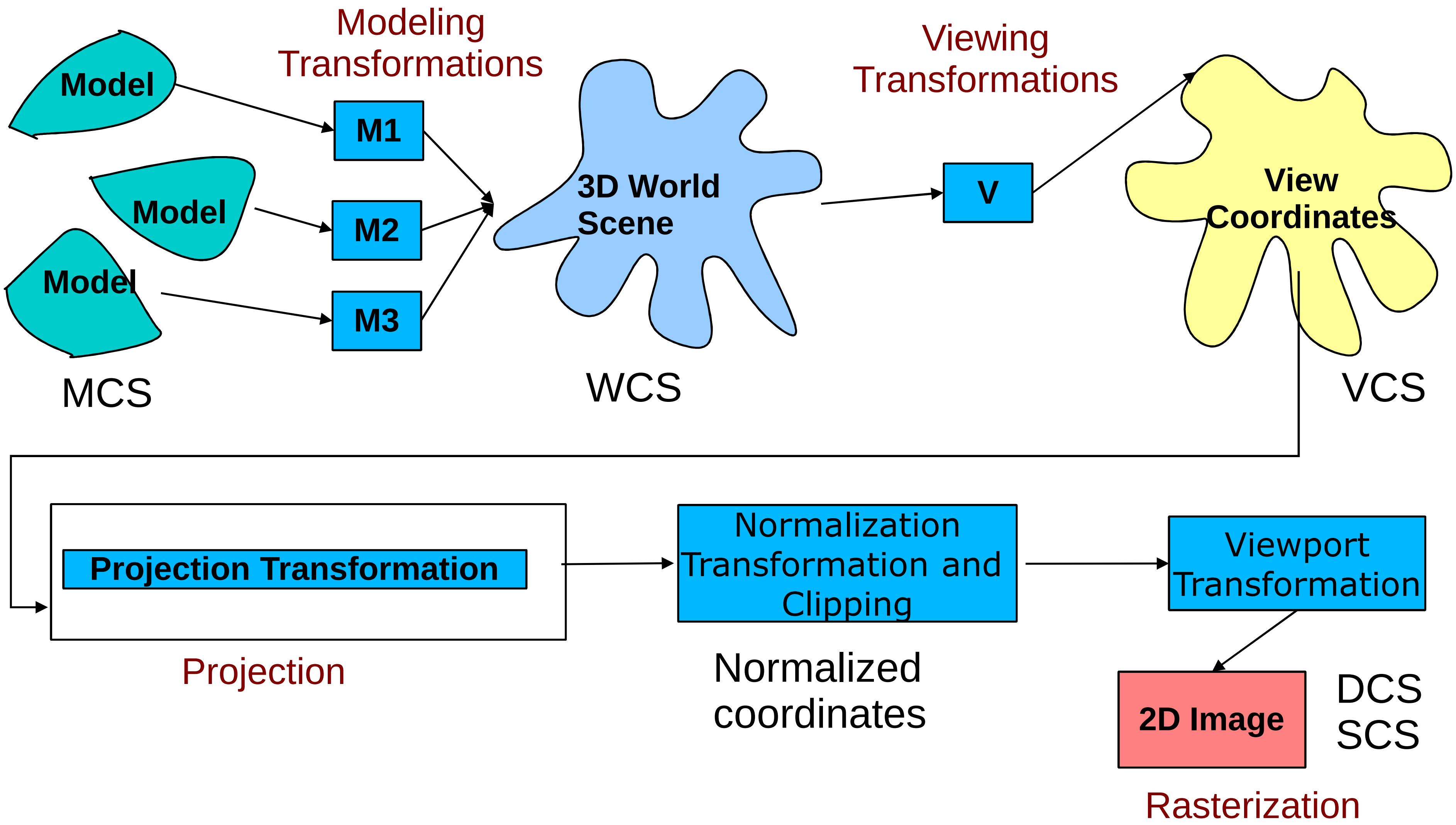
# 3 DIMENSIONAL VIEWING

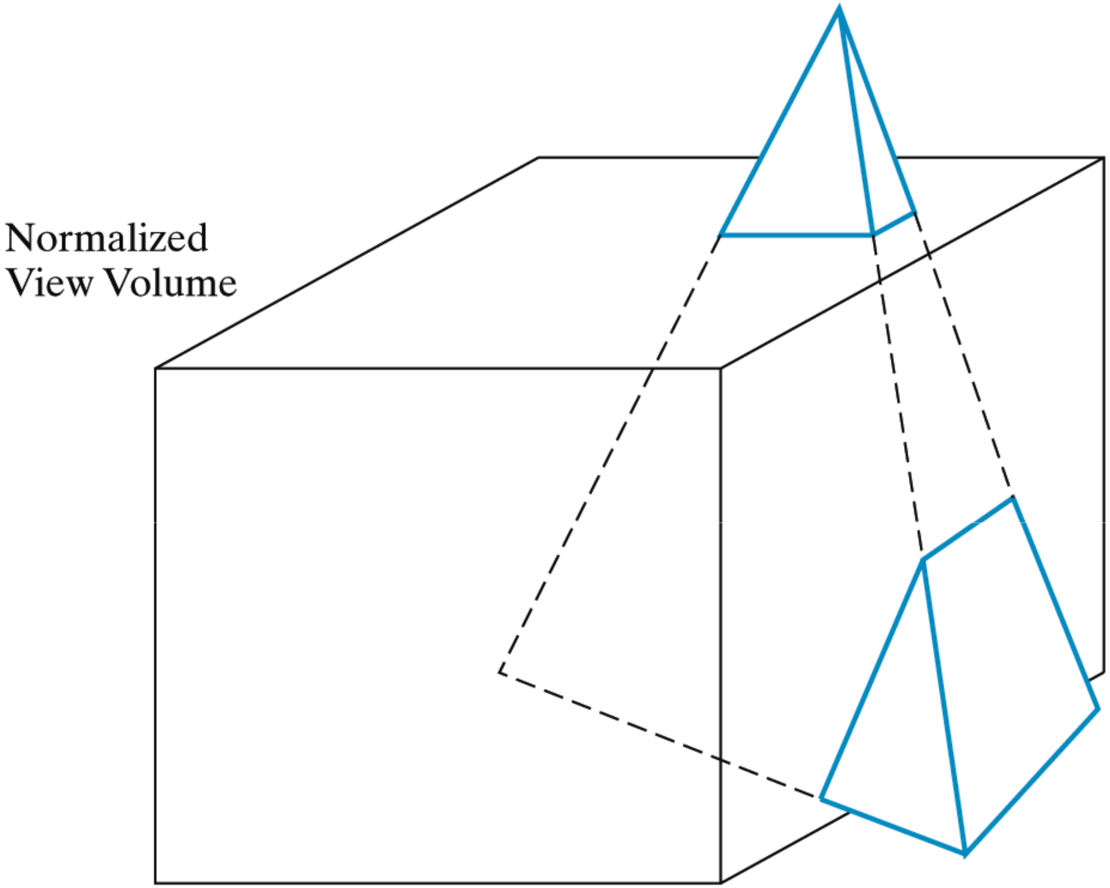
CLIPPING

# 3D Viewing Pipeline



# Clipping

- Clipping: Finding parts of the objects in the viewing volume.
- Algorithms from 2D clipping can easily be applied to 3D and used to clip objects against faces of the normalized view volume.

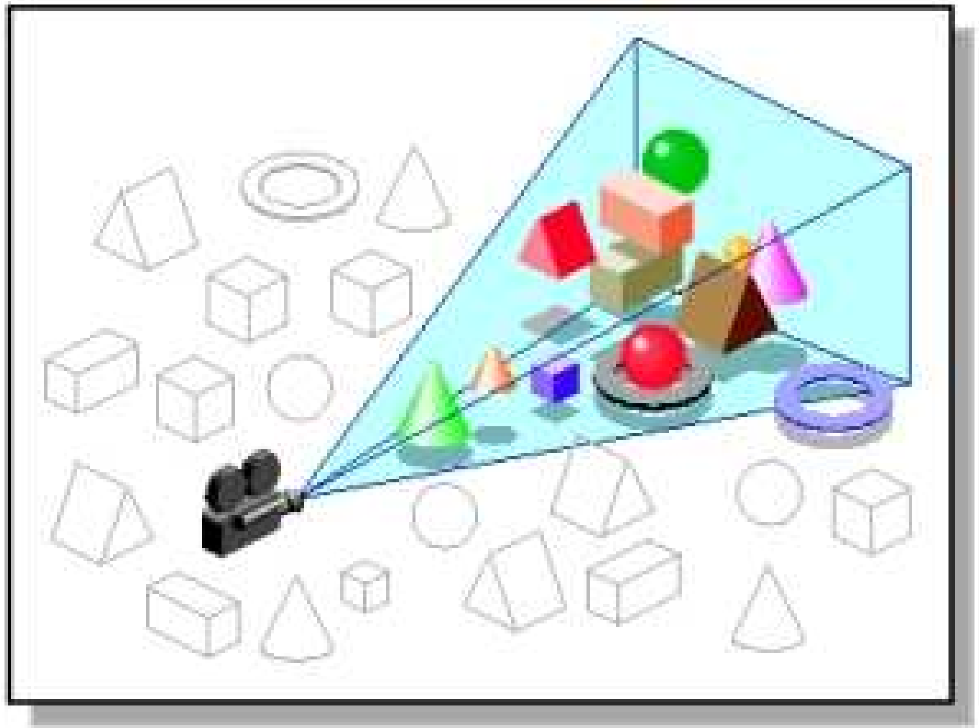


# 3-D Clipping

- Similar to two dimensions, clipping removes objects that will not be visible from the scene
- 3-D clipping is achieved in two basic steps
  - **Discard objects that can’t be viewed**
    - i.e. objects that are behind the camera, outside the field of view, or too far away
  - **Clip objects that intersect with any clipping plane**

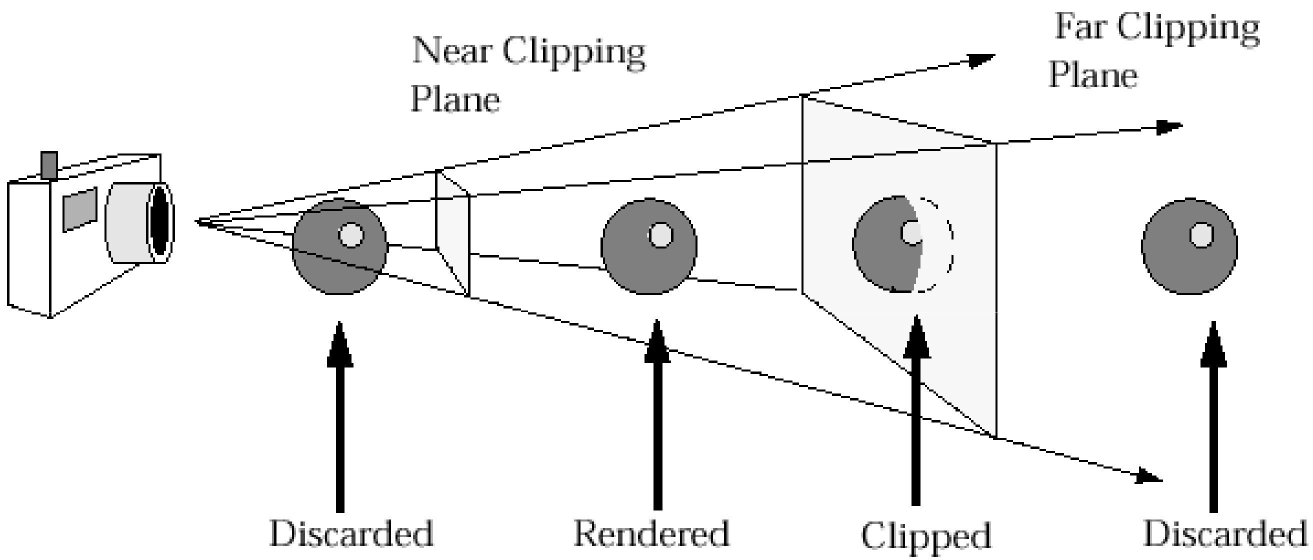
# Discard Objects

- Discarding objects involves comparing an object's bounding box/sphere against the dimensions of the view volume
  - **Can be done before or after projection**



# Clipping Objects

- **Objects that are partially within the viewing volume need to be clipped**



# Clipping

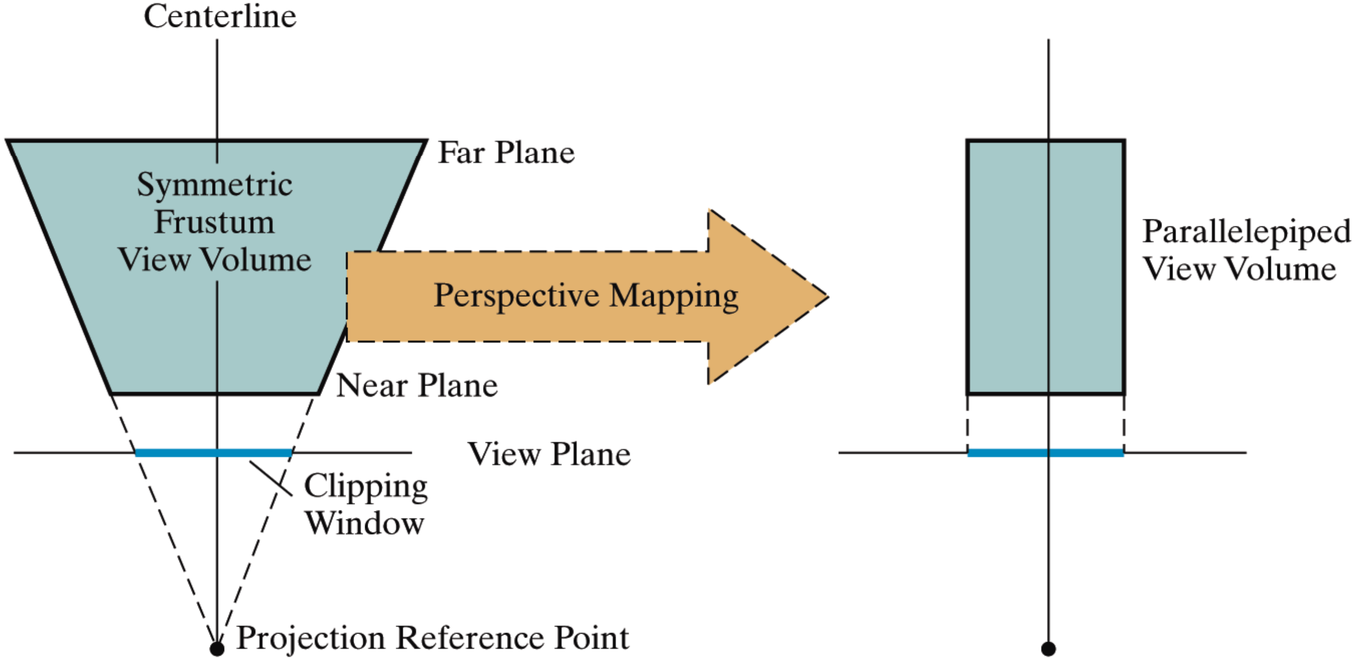
- Both the 2D algorithms, Cohen-Sutherland’s for line clipping and Sutherland-Hodgeman’s for polygon clipping, can easily be modified to 3D clipping.
- Clipping has to be performed against boundary planes instead of boundary edges
- Clipping in 3D generally needs to be done in homogeneous coordinates

# View volume

- View volume clipping boundaries are planes whose orientations depends on the **type of projection ,the projection window and position of the projection reference point.**
- The front and back clipping planes are parallel to the view plane each has constant z-coordinate value.
- The z coordinate of intersection of the lines with these planes is simply z coordinate of the corresponding plane.
- To find the intersection of line with one of the view volume boundaries, obtain the equation of the plane.

# The Clipping Volume

- Clipping against a regular parallelepiped is simpler because each surface is now perpendicular to one of the coordinate axes.
- After the perspective transformation is complete the frustum shaped viewing volume has been converted to a parallelepiped.
- The symmetric perspective transformation will map the objects into a parallelepiped view volume



# Normalization Transformation

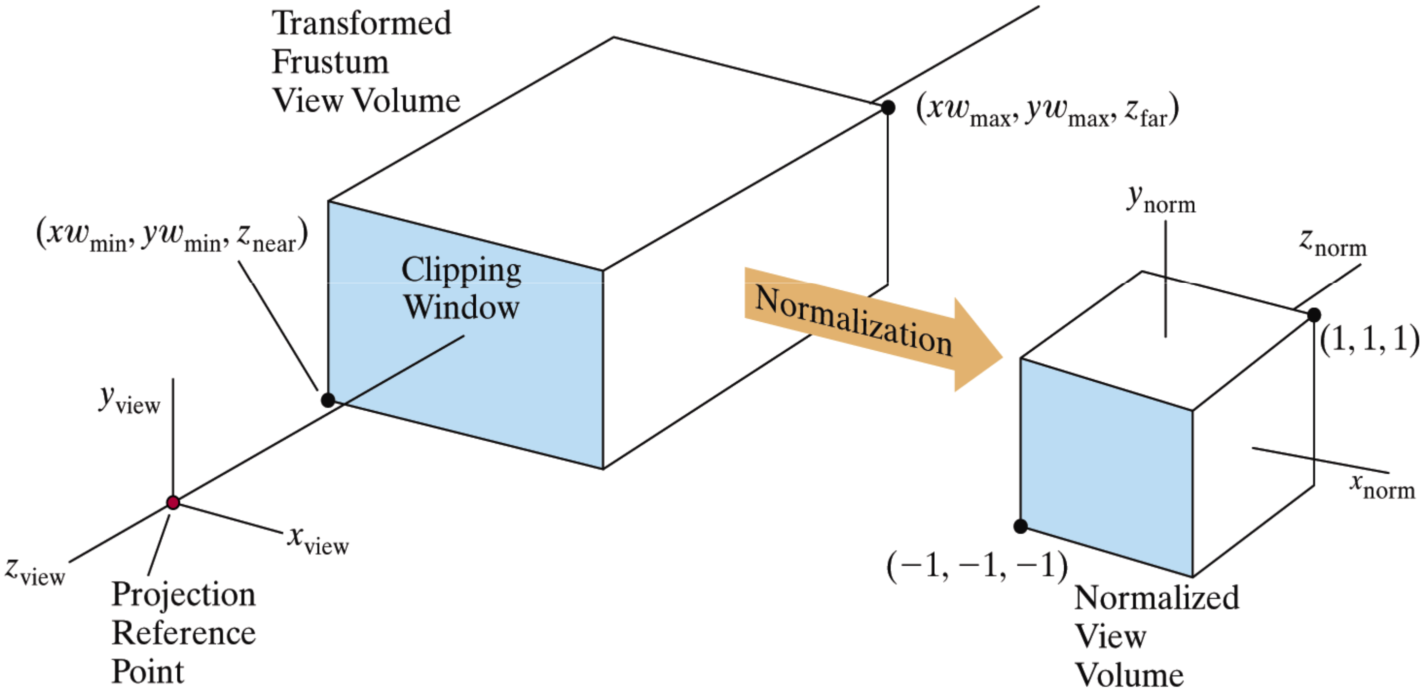
- The rectangular parallelepiped is mapped into **unit cube,** a normalized view volume called normalized projection coordinate system.

- The coordinates within the view volume are normalized to the range [0,1]

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

# Normalization

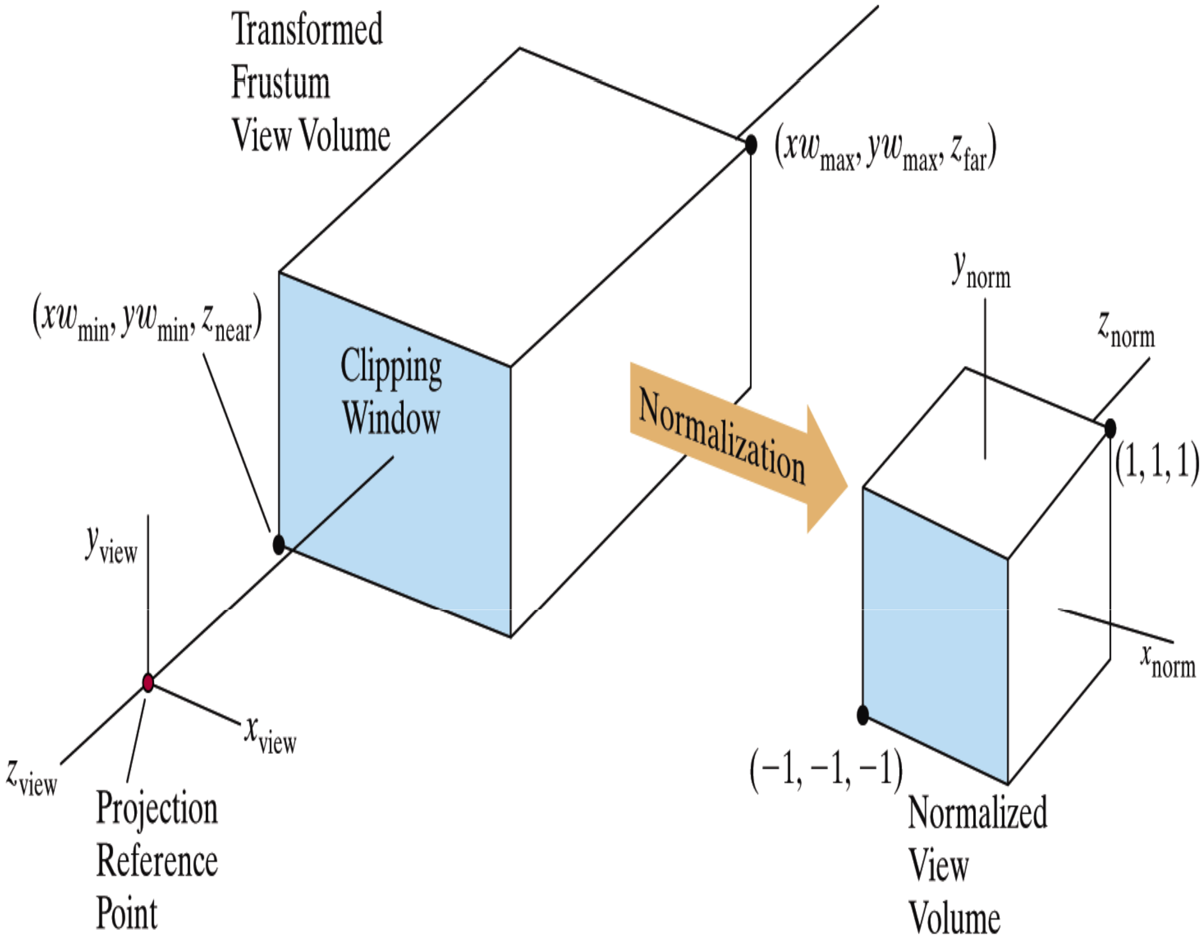
- The transformed volume is then *normalized* around position (0, 0, 0) and the z axis is reversed



# Advantages of clipping against unit cube

- Normalized view volume provides standard shape for representing any sized view volume. – can then be mapped into workstation of any size.
- Clipping procedures are simplified and standardized with unit clipping planes.
- Depth cueing and visible surface determination are simplified , since the z axis always points toward the viewer.

# Dimensions of the view volume and 3-D viewport




View volume boundaries are established by the window limits(XWmin,XWmax,

YWmin,Ywmax , Znear,Zfar).

View port boundaries are established (XVmin,XVmax,YVmin,

YVmax, ZVmin,ZVmax,).

The additive translation factors are Kx,Ky,Kz in the transformation

# Window to viewport

## three-dimensional window-to-viewport mapping

- similar to two-dimensional window-to-viewport mapping

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

where

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

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

$$D_z = \frac{zv_{\max} - zv_{\min}}{d_f - d_n}$$

and

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

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

$$K_z = zv_{\min} - d_n \cdot D_z$$

# 3D Cohen-Sutherland Region Codes in 3d

- 2d concepts of region codes can be extended to three dimensions by considering front and back planes
- Simply use 6 bits instead of 4.

| 6 | 5 | 4 | 3 | 2 | 1 |
|---|---|---|---|---|---|
| Far | Near | Top | Bottom | Right | Left |

# 3D Cohen-Sutherland Region Codes in 3d

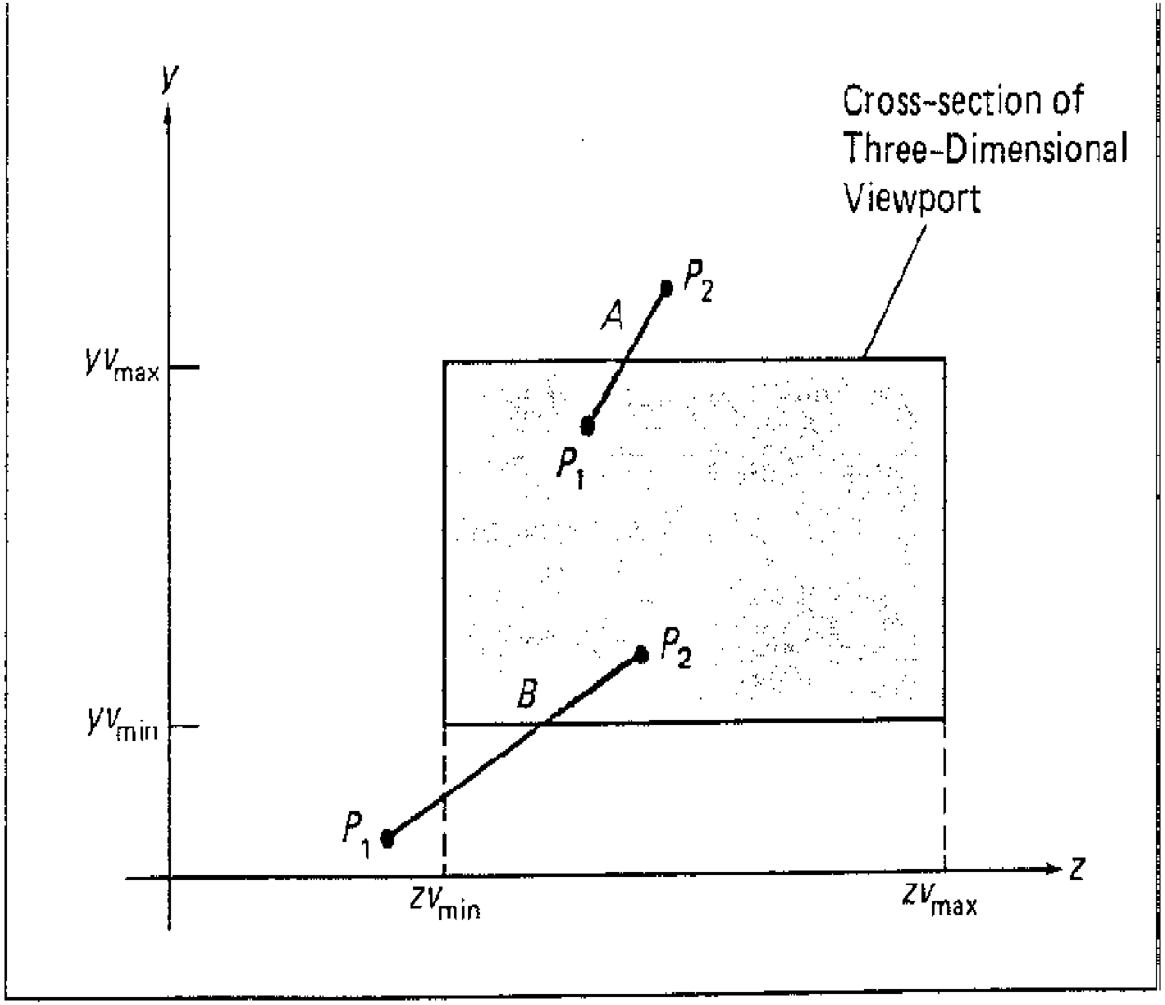
## clipping against a normalized view volume

- extend region codes

$$
\begin{array}{lll}
\text{bit } 1 = 1 & \text{if } x < xv_{\min} & \text{(left)} \\
\text{bit } 2 = 1 & \text{if } x > xv_{\max} & \text{(right)} \\
\text{bit } 3 = 1 & \text{if } y < yv_{\min} & \text{(below)} \\
\text{bit } 4 = 1 & \text{if } y > yv_{\max} & \text{(above)} \\
\text{bit } 5 = 1 & \text{if } z < zv_{\min} & \text{(front)} \\
\text{bit } 6 = 1 & \text{if } z > zv_{\max} & \text{(back)}
\end{array}
$$

- trivial acceptance
- trivial rejection
- subdivision

# 3D Cohen-Sutherland Region Codes in 3d




101000->identifies as above and behind the viewport.

000000->point within the view volume

o1 = o2 = 000000: **accept**
o1 & o2 != 000000: **reject**
o1 = 000000, o2 != 000000 : **subdiv**
o1 != 000000, o2 = 000000 : **subdiv**
o1 & o2 = 000000 : **subdiv**

# Liang Barsky line clipping for 3D

- For clipping equations for three dimensional line segments are given in their parametric form
- For a line segment with end points $P_1(x1, y1, z1)$ and $P_2(x2, y2, z2)$ the parametric equation describing any point on the line is:

$$x = x1 + (x2 - x1)u$$
$$y = y1 + (y2 - y1)u$$
$$z = z1 + (z2 - z1)u$$
$$0 \le u \le 1$$

# Liang-Barsky

- Suppose we are testing a line against the ZVmin plane of the viewport

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

•If u is not in the range from 0 to 1 the line segment does not intersect the plane.

•If it is in the interval 0 to 1 calculate the line of intersection x and y coordinates.

x1 = x1 + (x2-x1) u

Substitute the value of u

Similarly for y

# Clipping in Homogenous coordinates

- Various transformations are represented by 4*4 matrices, concatenated for efficiency
- After the viewing, the homogenous coordinate positions are converted to 3d points.

  $(x,y,z) \rightarrow (x,y,z,1)$

- After geometric, viewing and projection transformations, each vertex is: $(x_h, y_h, z_h, h)$

# Clipping Homogeneous Coordinates in 3D

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

- Any homogeneous coordinate position ($x_h$, $y_h$, $z_h$, h) is inside the view volume if:

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

Thus, the homogeneous clipping limits are

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

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

- Thank you