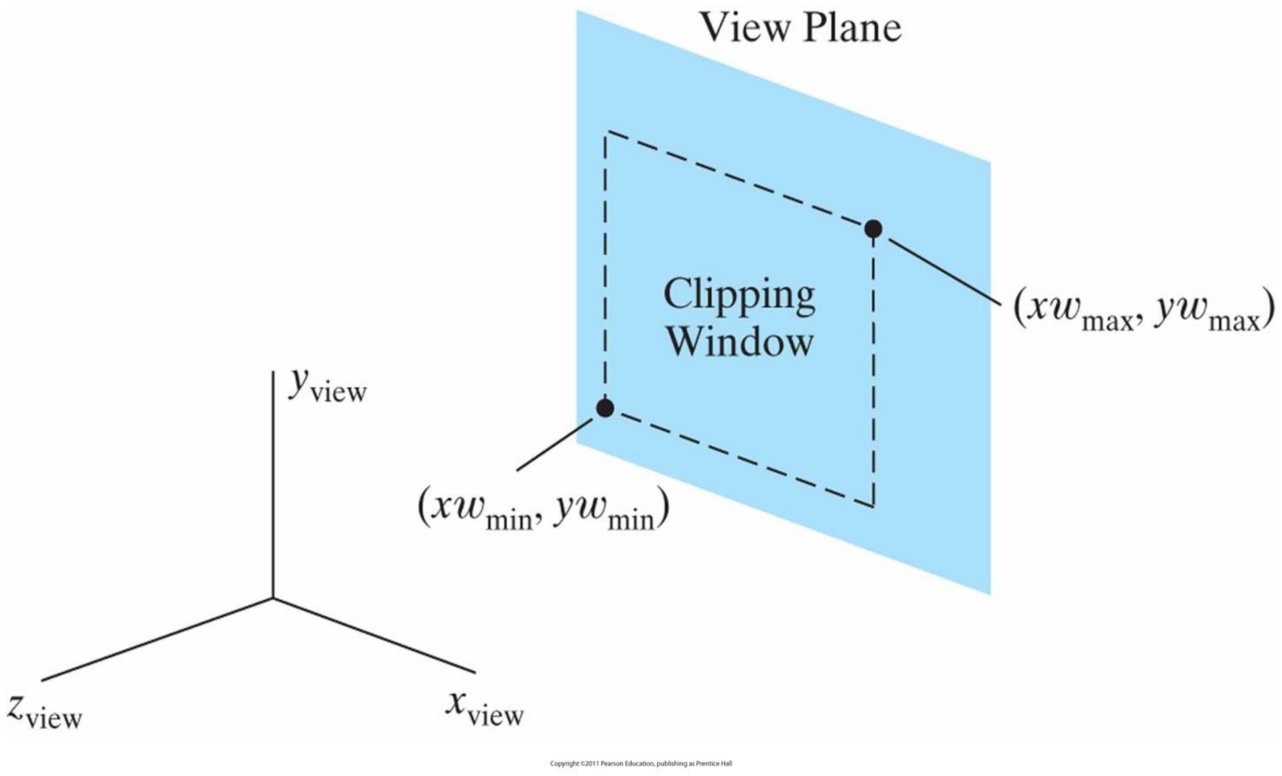
3D Clipping

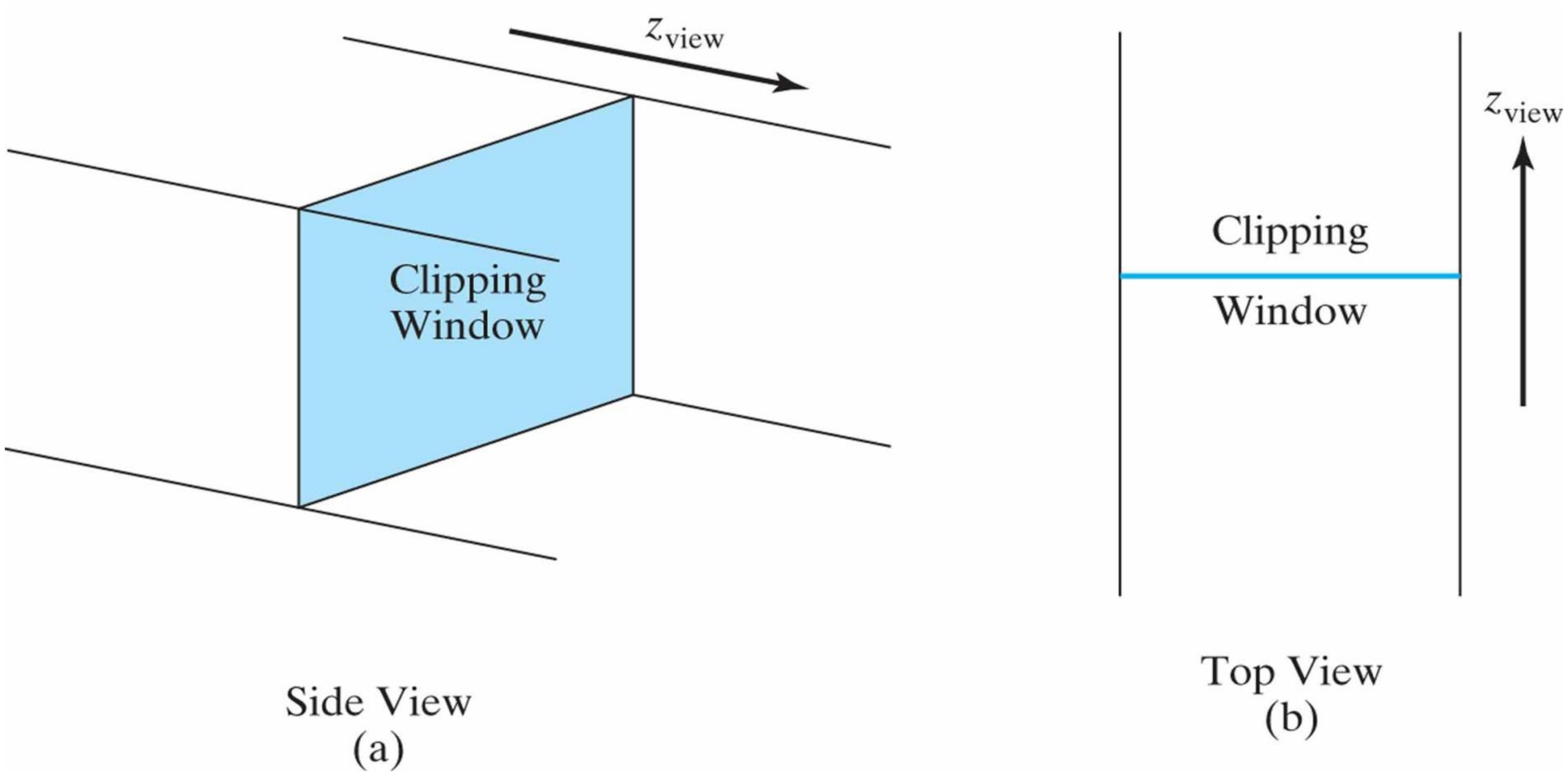
THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

- **An algorithm for 3D clipping identifies and saves all surface segments within the view volume for display.**
- **All parts of object that are outside the view volume are discarded.**

Clipping Window and Orthogonal Projection

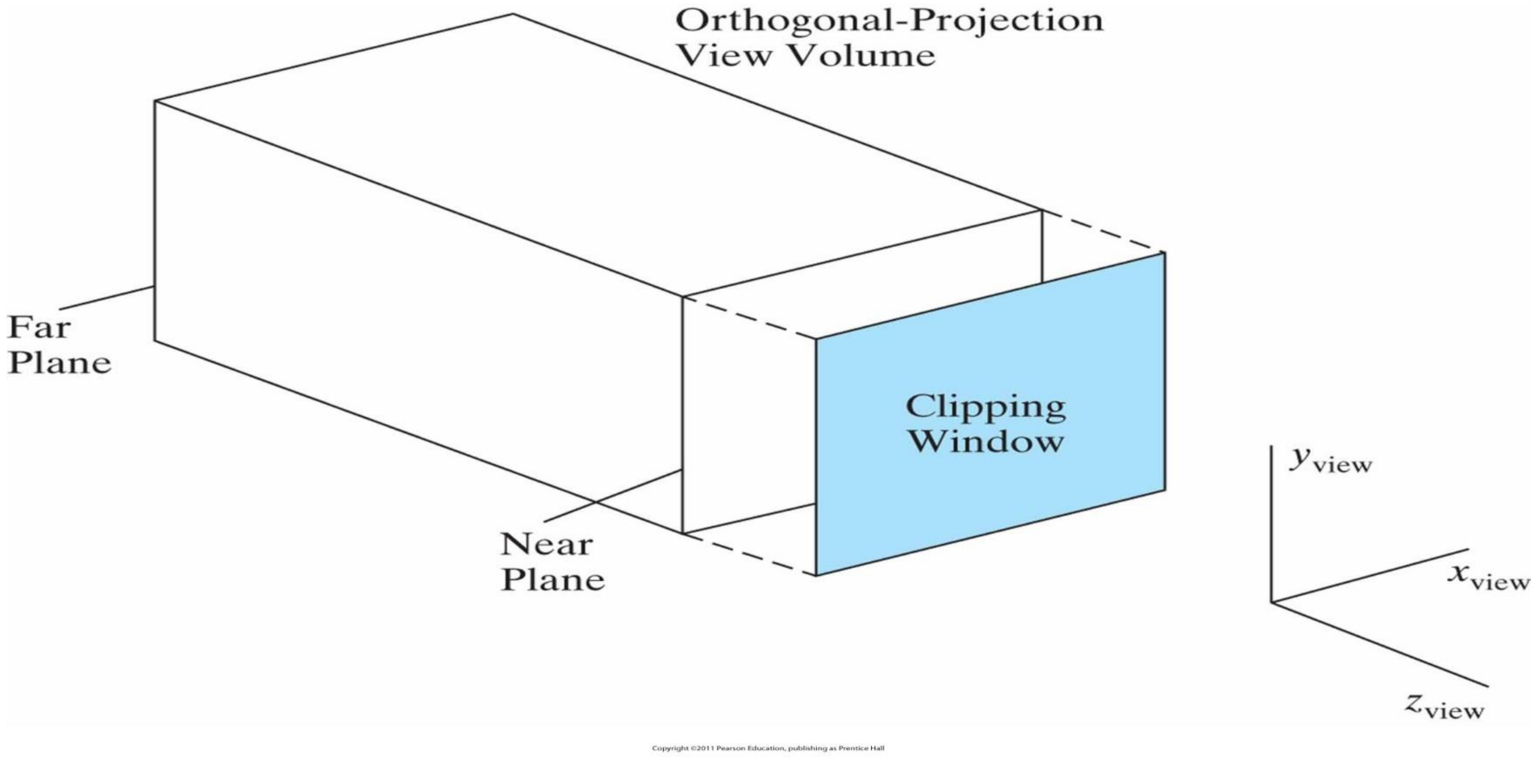


In CG generally rectangular clipping windows are used.



Limiting the volume with near-far or front-back clipping planes

Finite Orthogonal View Volume



with the view plane “in front” of the near plane.

Clipping Lines

Just like the case in two dimensions, clipping removes objects that will not be visible from the scene

The point of this is to remove computational effort 3-D clipping is achieved

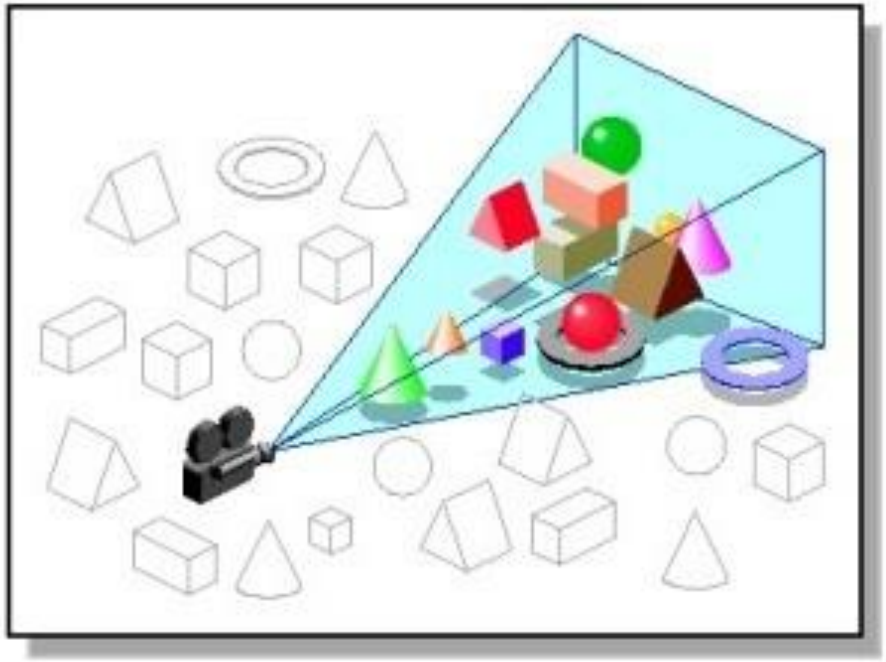
in two basic steps

- 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 that cannot possibly be seen involves comparing an objects bounding box/sphere against the dimensions of the view volume

- Can be done before or after projection

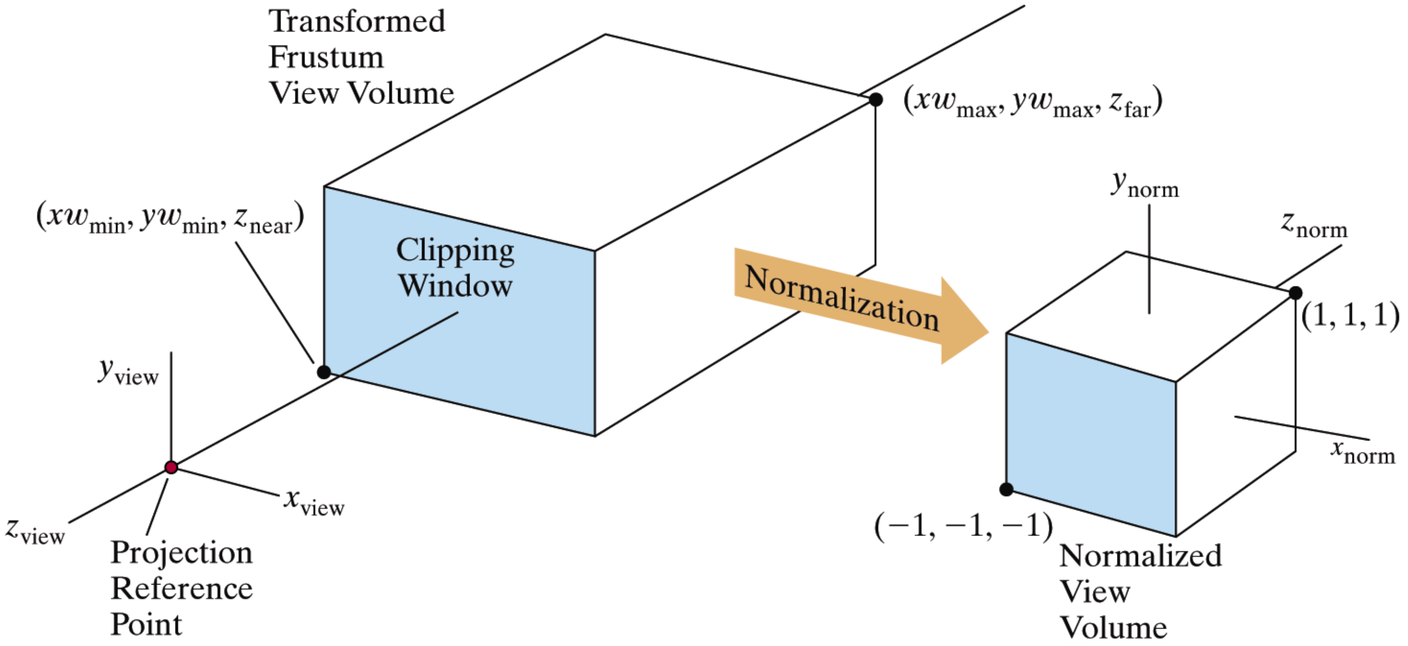


3D Clipping

- Main idea: Extending 2D thoughts into 3D
- Assumption: The boundaries are normalized
- View volume can be normalized to unit cube or to a symmetric cube with edge length 2 → clipping planes at 0 and 1 or -1 and 1
- If clipping window normalized to symmetric cube:
  - Clipping plane equations: $$\begin{array}{ll} xw_{min} = -1 & xw_{max} = 1 \\ yw_{min} = -1 & yw_{max} = 1 \\ zw_{min} = -1 & zw_{max} = 1 \end{array}$$

Normalisation

The transformed volume is then *normalised* around position (0, 0, 0) and



- Remember:

Homogeneous coordinate representation; *(x, y, z)* → *(x, y, z, 1)*

- So: $\begin{bmatrix} x_h \\ y_h \\ z_h \\ h \end{bmatrix} = M \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$

where M represents the concatenation of all various transformations from world to normalized, homogeneous projection coordinates.

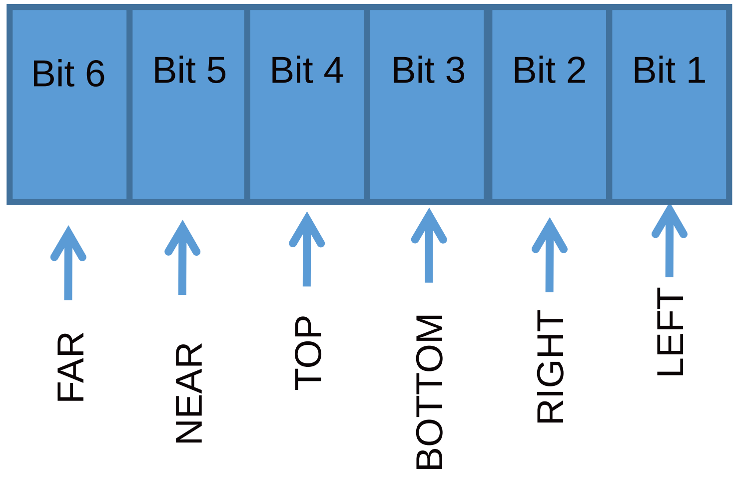
- Problem: h can be $\neq 1$

- If $h=1$ homogeneous parameters=Cartesian projection parameters, e.g. Parallel projection
- If $h \neq 1$ and clipping is done after division by h some coordinate information can be lost and objects may not be clipped correctly

- How to solve this problem?

Answer: Apply clipping to homogeneous coordinate representation of spatial positions

- 3D region codes: (Extending Cohen-Sutherland's idea to 3D)



- How to find out if a point is inside or outside the volume?

- Assumptions:
  - Each point in the 4-component representation: $P = (x_h, y_h, z_h, h)$
  - Clipping against normalized symmetric cube

Given these assumptions a point is inside the volume if the projection coordinates of the point satisfies the following equations:

$$-1 \leq \frac{x_h}{h} \leq 1, \qquad -1 \leq \frac{y_h}{h} \leq 1, \qquad -1 \leq \frac{z_h}{h} \leq 1$$

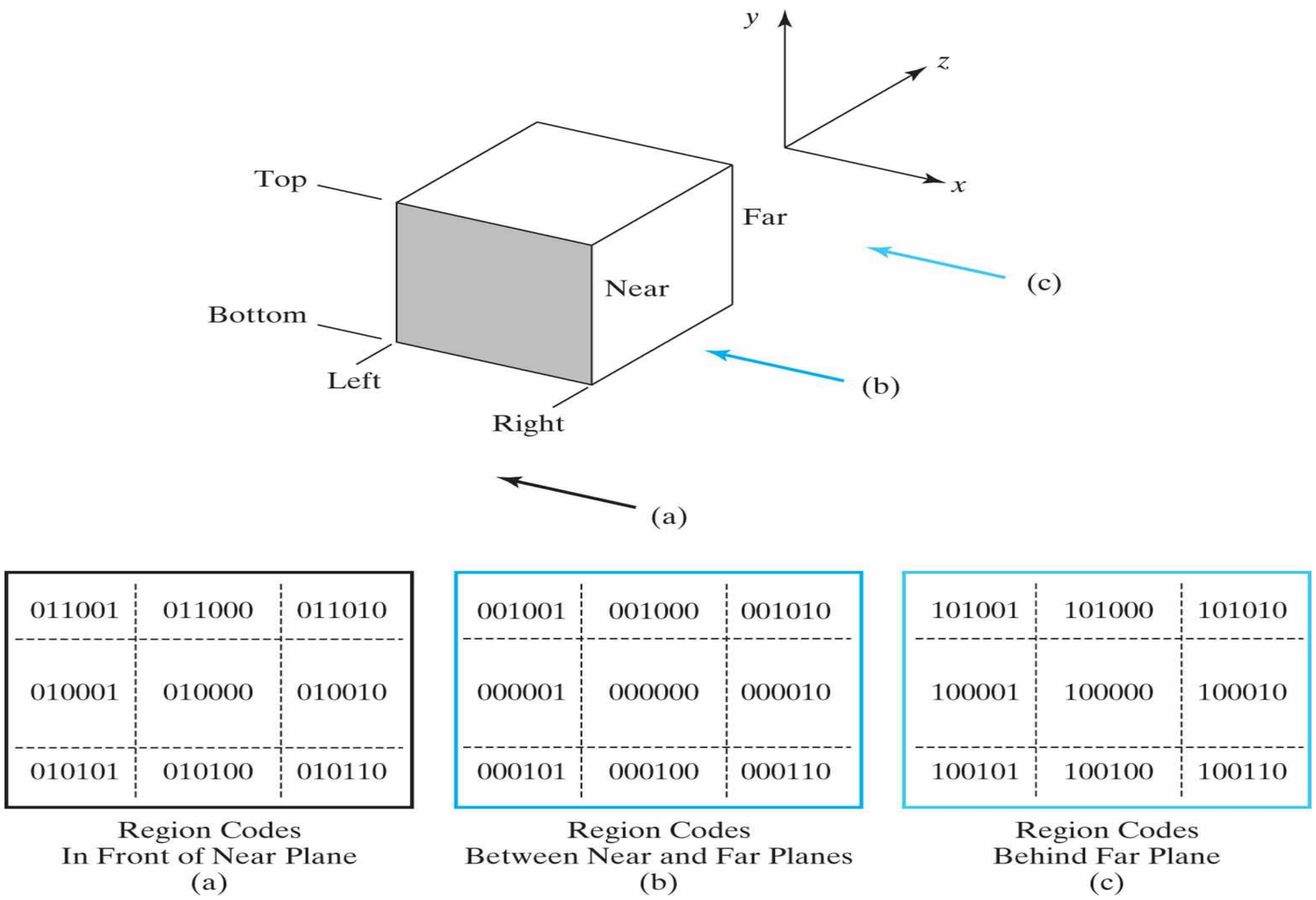
- Assuming $h \neq 0$

$$-h \leq x_h \leq h,\ \ -h \leq y_h \leq h, -h \leq z_h \leq h \quad \text{if } h>0$$

$$h \leq x_h \leq -h,\ \ h \leq y_h \leq -h, h \leq z_h \leq -h \quad \text{if } h<0$$

- Testing region codes (for $h > 0$):

$bit\ 1 = 1$	if $h + x_h < 0$	(left)
$bit\ 2 = 1$	if $h - x_h < 0$	(right)
$bit\ 3 = 1$	if $h + y_h < 0$	(bottom)
$bit\ 4 = 1$	if $h - y_h < 0$	(top)
$bit\ 5 = 1$	if $h + z_h < 0$	(near)
$bit\ 6 = 1$	if $h - z_h < 0$	(far)



Values for the three-dimensional, six-bit region code that identifies spatial positions relative to the boundaries of a view volume.

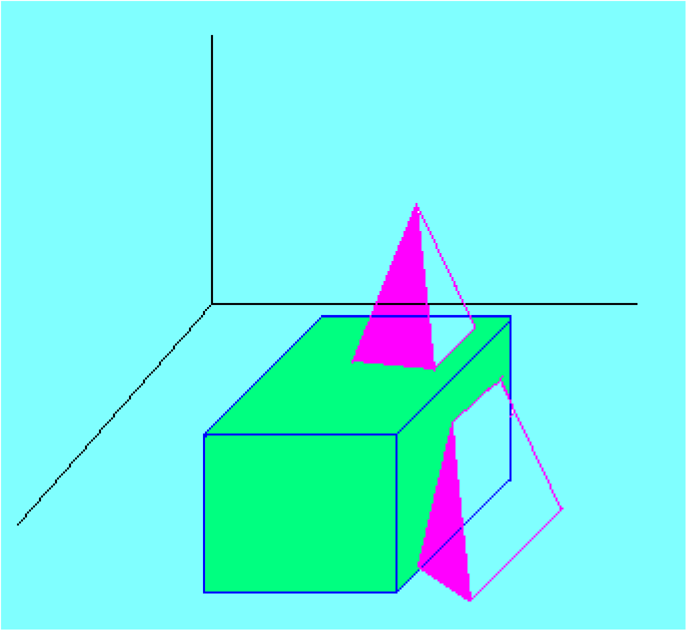
3D Clipping Algorithm

- For point clipping:
  - Eliminate all points with region code ≠ *000000*

- For line clipping:
  1. Test trivial accepts and trivial rejects using region codes, e.g. for trivial accepts use logical OR and for trivial rejects use logical AND
  2. For other cases, use methods of Cyrus-Beck or Liang-Barsky.

Clipping Polygon Surface

questions and concept



View volume

Clipping Polygon Surface

- To clip a polygon surface, we can clip the individual polygon edges.
- First we test the coordinate extends against each boundary of the view volume to determine whether the object is completely inside or completely outside of that boundary.
- If the object has intersection with the boundary then we apply intersection calculations.

Clipping Polygon Surface

- The projection operation can take place before the view- volume clipping or after clipping.
- All objects within the view volume map to the interior of the specified projection window.
- The last step is to transform the window contents to a 2D view port.

3D Polygon Clipping

- In this case we first try to eliminate the entire object using its bounding volume
- Next, we perform clipping on the individual polygons using the Sutherland-Hodgman algorithm we studied previously