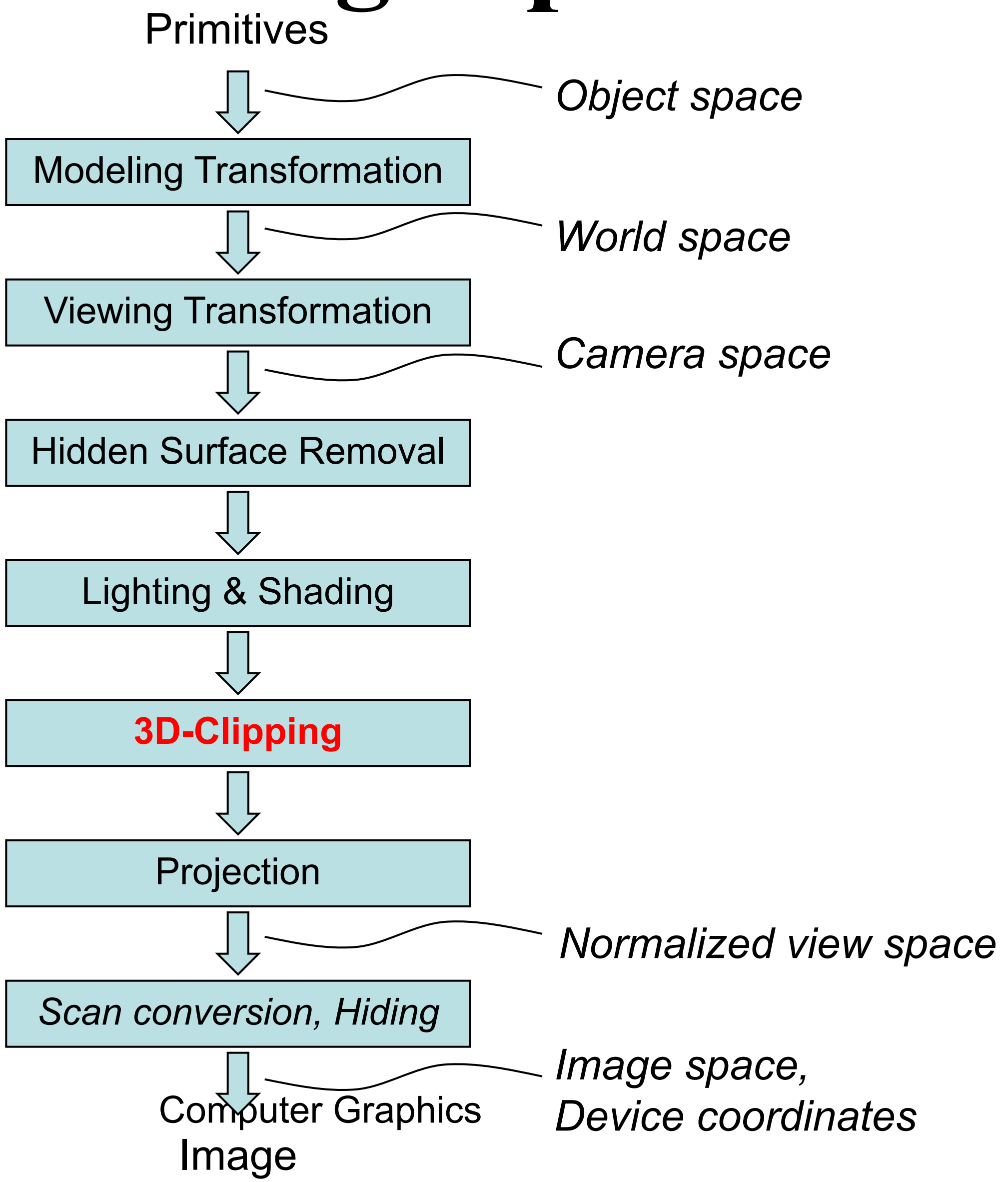
Chapter 11

3D Clipping

3D Viewing Pipeline



Contents

3D Clipping

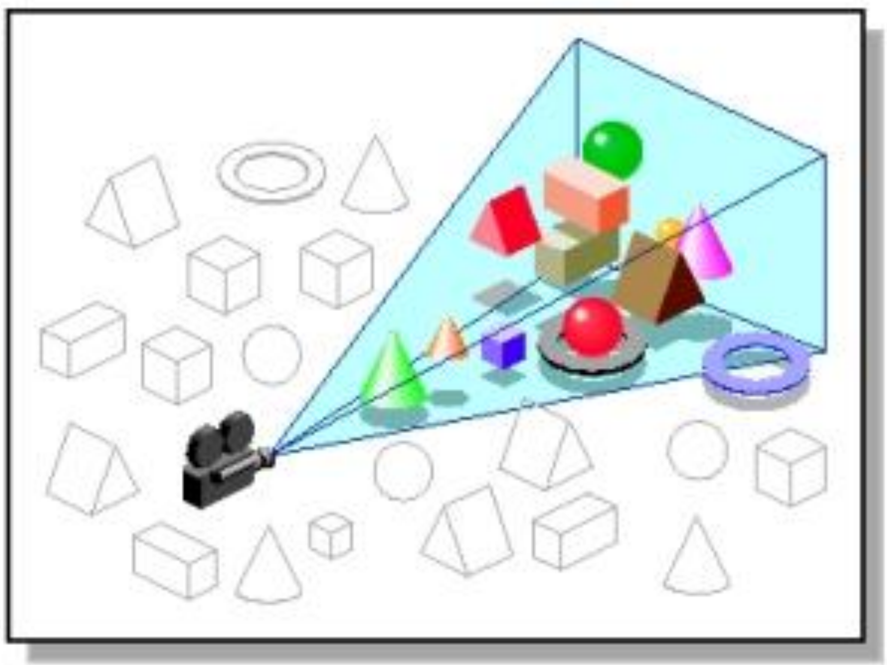
- Just like the case in two dimensions, clipping removes objects that will not be visible from the scene

- 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

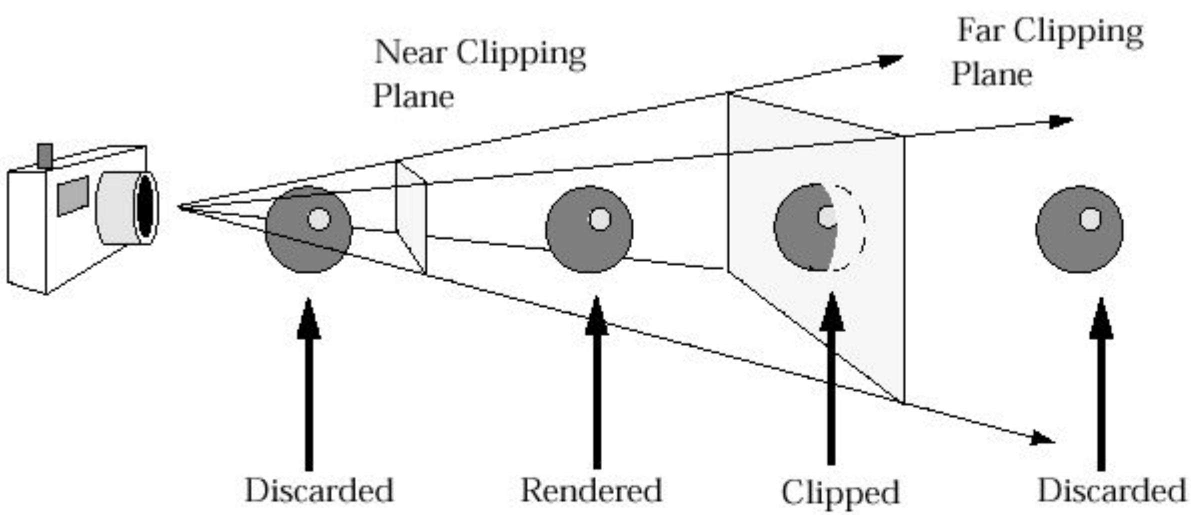
3D Clipping

- 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

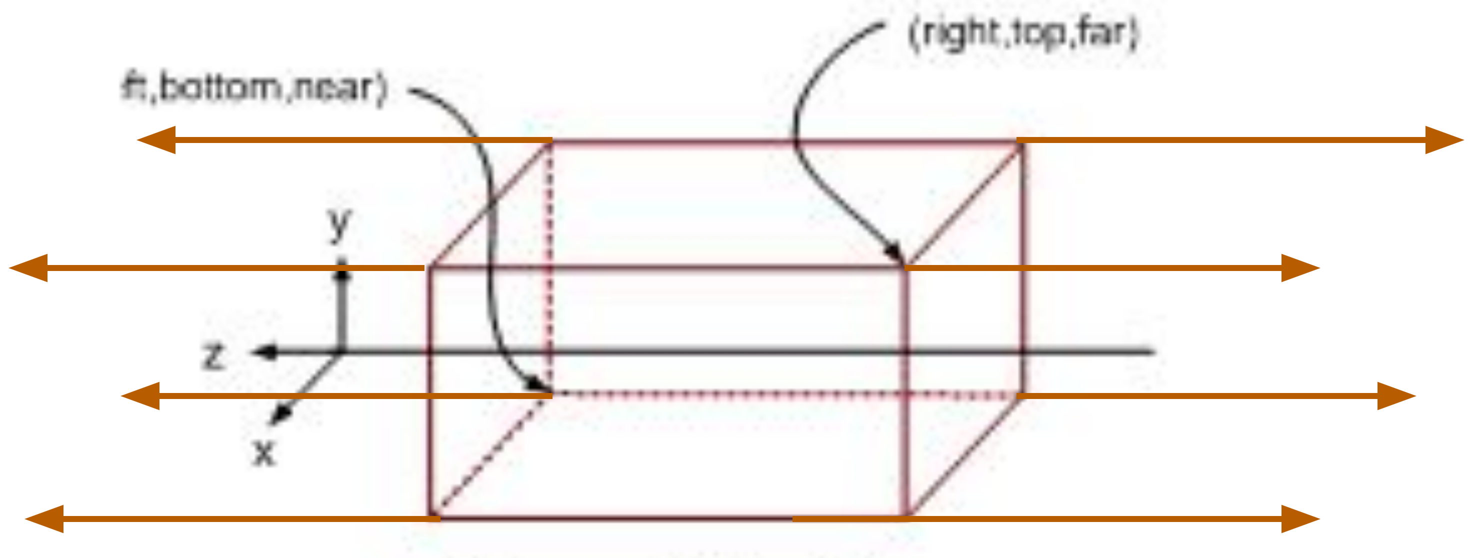
- Objects that are partially within the viewing volume need to be clipped – just like the 2D case



Contents

The Clipping Volume

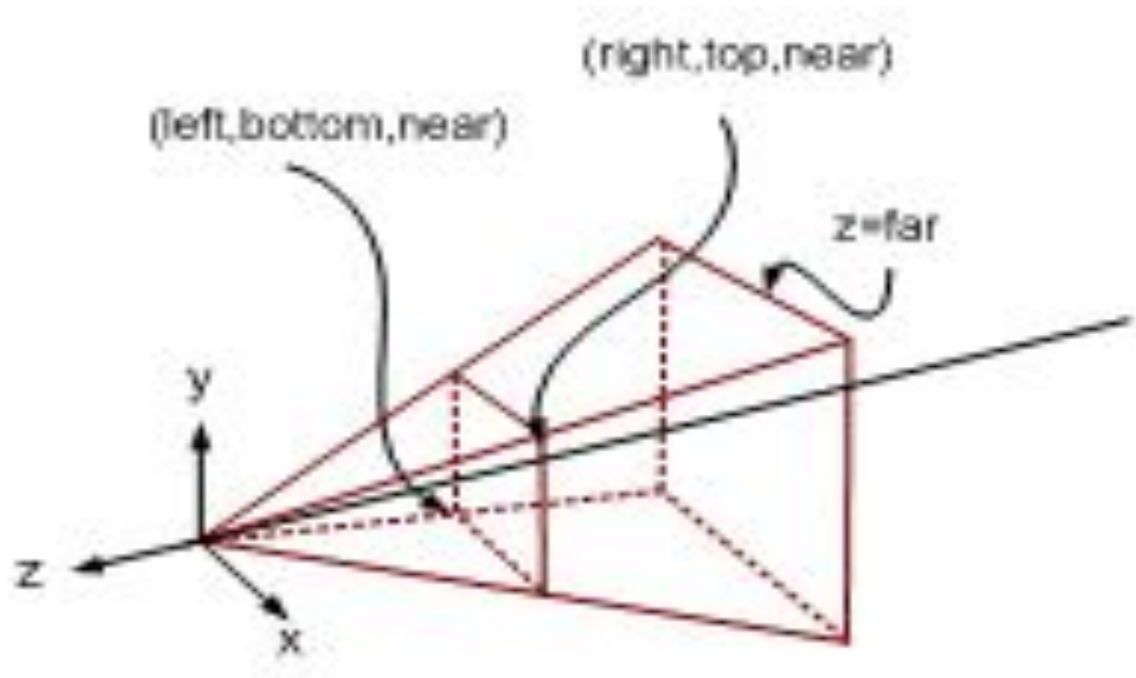
- In case of Parallel projections the infinite Parallelepiped is bounded by Near/front/hither and far/back/yon planes for clipping.



Orthographic View Volume

The Clipping Volume

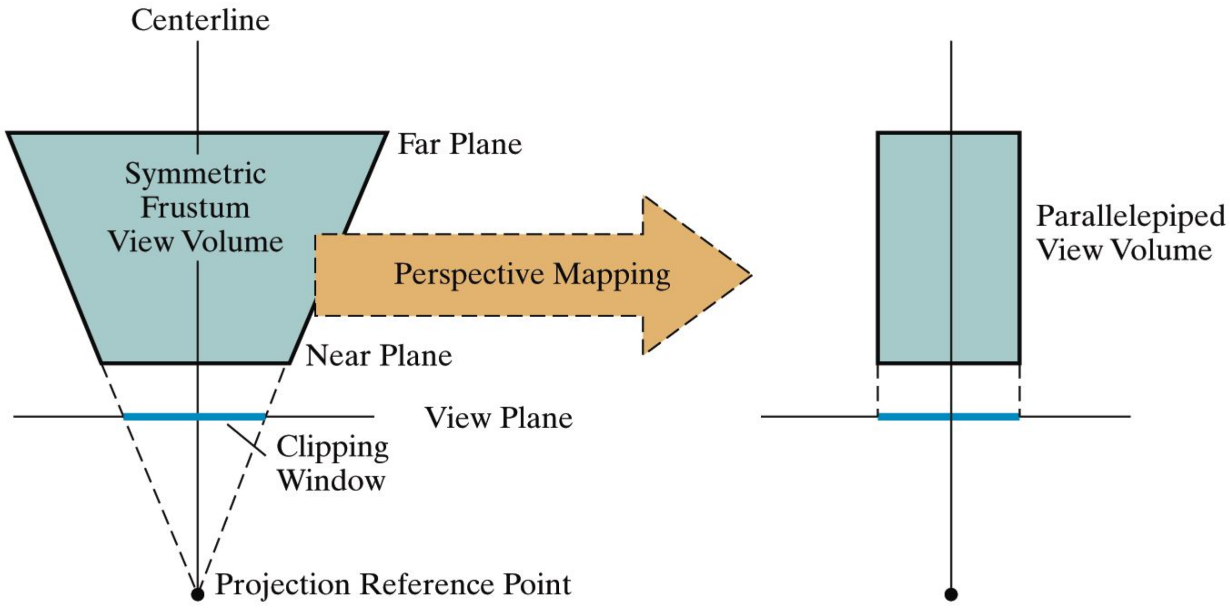
- In case of Perspective projections the semi Infinite Pyramid is also bounded by Near/front/hither and far/back/yon planes for cl



Perspective View Volume

The Clipping Volume

- After the perspective transformation is complete the frustum shaped viewing volume has been converted to a parallelepiped - remember we preserved all z coordinate depth information



Contents

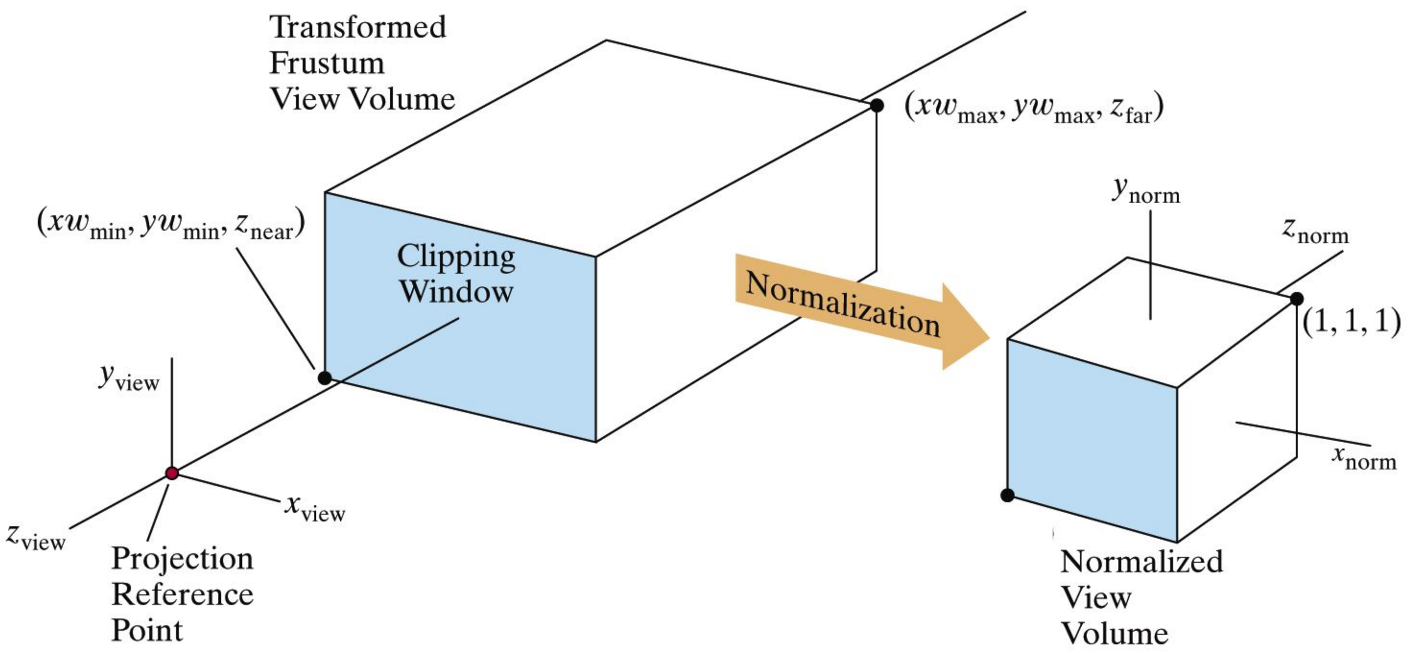
Clipping Strategies

- Because of the extraordinary computational effort required, two types of clipping strategies are followed:
  - **Direct Clipping:** The clipping is done directly against the view volume.
  - **Canonical Clipping:** Normalization transformations are applied which transform the original view volume into normalized (canonical) view volume. Clipping is then performed against canonical view volume.

Clipping Strategies

- The canonical view volume for ***parallel projection*** is the unit cube whose faces are defined by planes

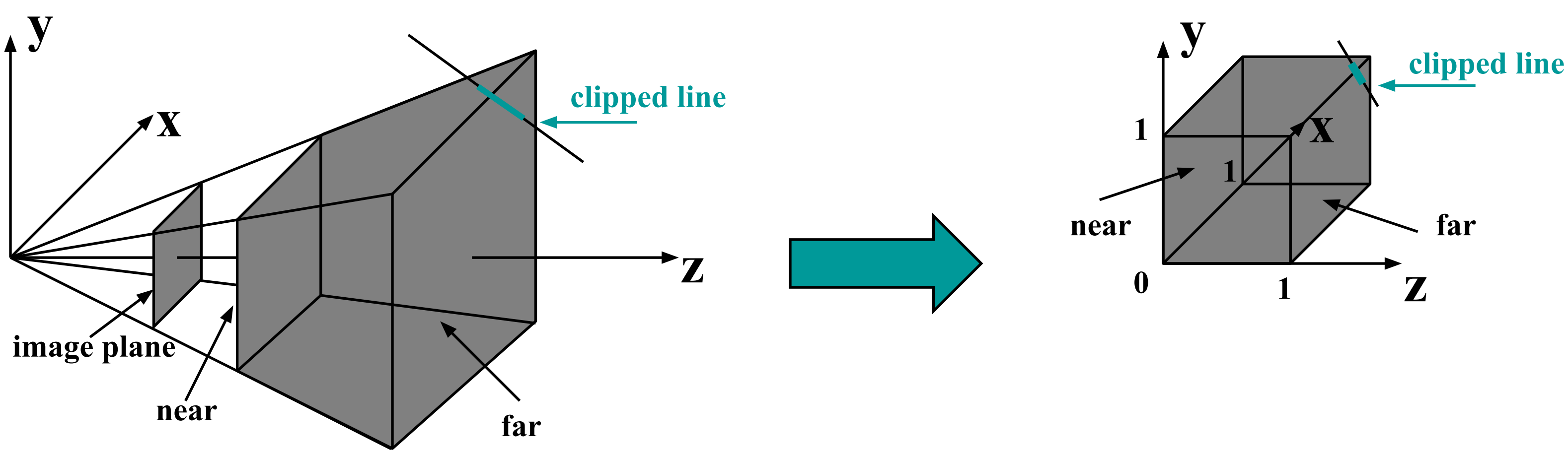
  $x = 0 \;;\; x = 1 \qquad y = 0;\; y = 1 \qquad z = 0;\; z = 1$



Clipping Strategies

- The canonical view volume for ***perspective projection*** is the truncated normalized pyramid whose faces are defined by planes

$x = z\ ;\ x = -z \qquad y = z;\ \ y = -z \qquad z = z_f;\ z = 1$



Clipping Strategies

- We perform clipping after the projection transformation and normalizations are complete.

- So, we have the following:

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

- We apply all clipping to these homogeneous coordinates

Clipping Strategies

- The basis of canonical clipping is the fact that intersection of line segments with the faces of canonical view volume require minimal calculations.

- For perspective views, additional clipping may be required to avoid perspective anomalies produced by the projecting objects that are behind view point.

Contents

Clipping Algorithms

3D clipping algorithms are direct adaptation of 2D clipping algorithms with following modifications:

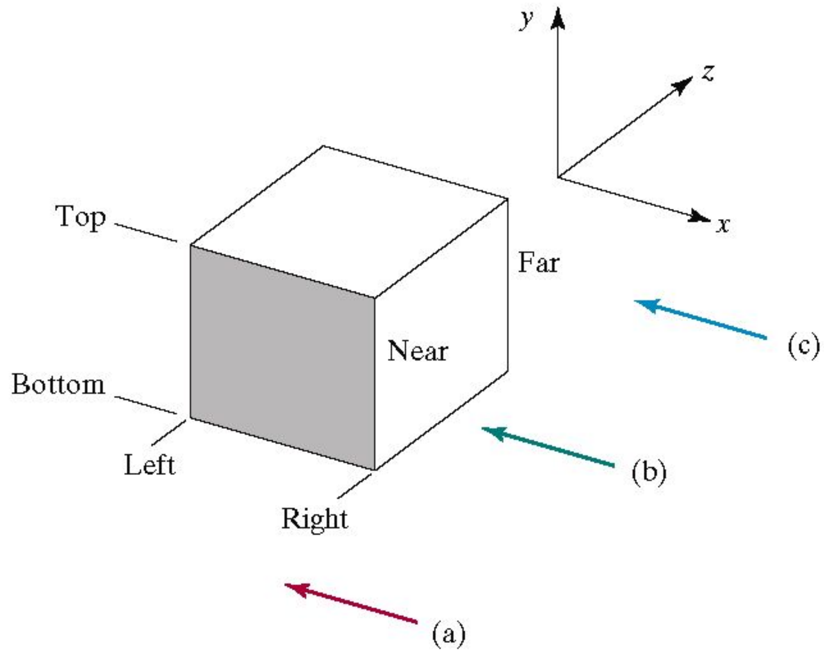
1. ***For Cohen Sutherland*****:** Assignment of out codes
2. ***For Liang-Barsky:*** Introduction of new equations
3. ***For Sutherland Hodgeman:*** Inside/Out side Test
4. ***In general:*** Finding the intersection of Line with plane.

3D Cohen-Sutherland Line Clipping

- Similar to the case in two dimensions, we divide the world into regions
- This time we use a 6-bit region code to give us **27 different region codes**
- The bits in these regions codes are as follows:

bit 1	bit 2	bit 3	bit 4	bit 5	bit 6
Above	Below	Right	Left	Behind	Front

3D Cohen-Sutherland Line Clipping



011001	011000	011010
010001	010000	010010
010101	010100	010110

Region Codes
In Front of Near Plane
(a)

001001	001000	001010
000001	000000	000010
000101	000100	000110

Region Codes
Between Near and Far Planes
(b)

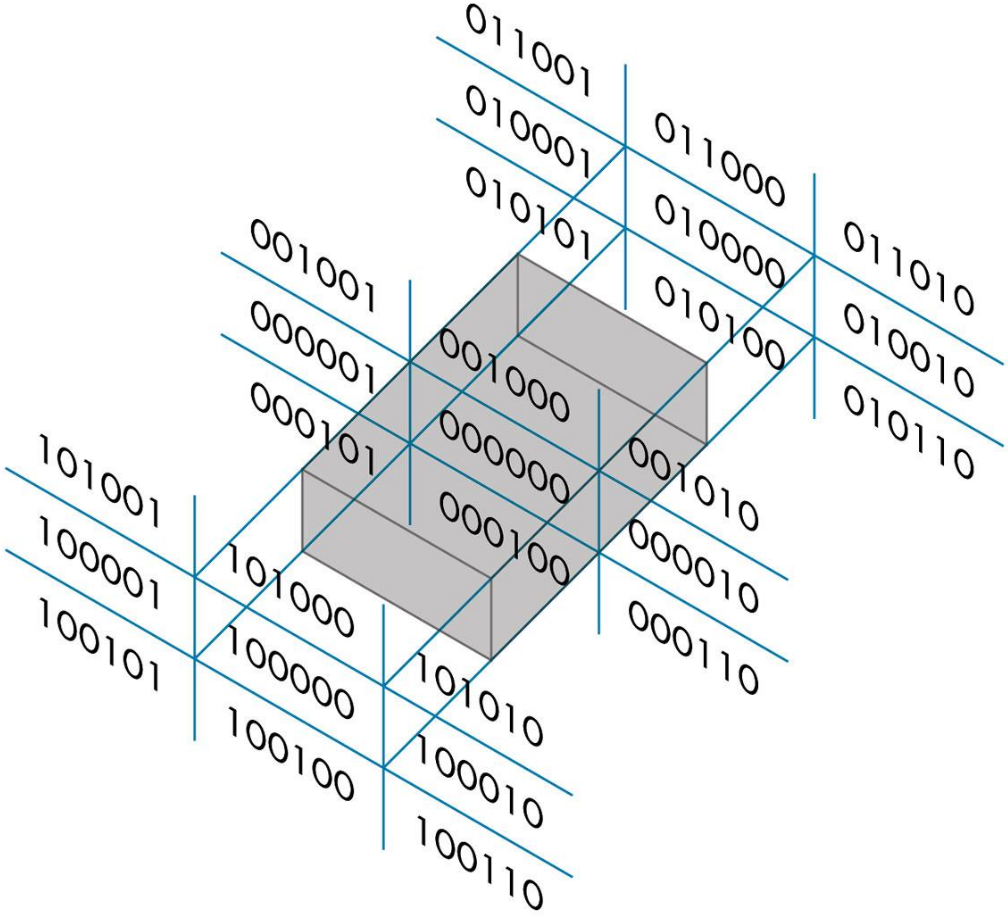
101001	101000	101010
100001	100000	100010
100101	100100	100110

Region Codes
Behind Far Plane
(c)

3D Cohen-Sutherland Line Clipping

Now we use a 6 bit out code to handle the near and far plane.

The testing strategy is virtually identical to the 2D case.



3D Cohen-Sutherland Line Clipping

CASE – I Assigning region codes to endpoints for ***Canonical Parallel View Volume*** defined by:

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

The bit codes can be set to true(1) or false(0) for depending on the test for these equations as follows:

```
Bit 1 ≡ endpoint is Above view volume = sign (y-1)
Bit 2 ≡ endpoint is Below view volume = sign (-y)
Bit 3 ≡ endpoint is Right view volume = sign (x-1)
Bit 4 ≡ endpoint is Left view volume = sign (-x)
Bit 5 ≡ endpoint is Behind view volume = sign (z-1)
Bit 6 ≡ endpoint is Front view volume = sign (-z)
```

3D Cohen-Sutherland Line Clipping

CASE – II Assigning region codes to endpoints for ***Canonical Perspective View Volume*** defined by:

$x = -z\ ,\ x = z;\quad y = -z,\ \ y = z;\qquad z = z_f\ ,\ z = 1$

The bit codes can be set to true(1) or false(0) for depending on the test for these equations as follows:

Bit 1 ≡ endpoint is *Above* view volume = sign (y-z)

Bit 2 ≡ endpoint is *Below* view volume = sign (-z-y)

Bit 3 ≡ endpoint is *Right* view volume = sign (x-z)

Bit 4 ≡ endpoint is *Left* view volume = sign (-z-x)

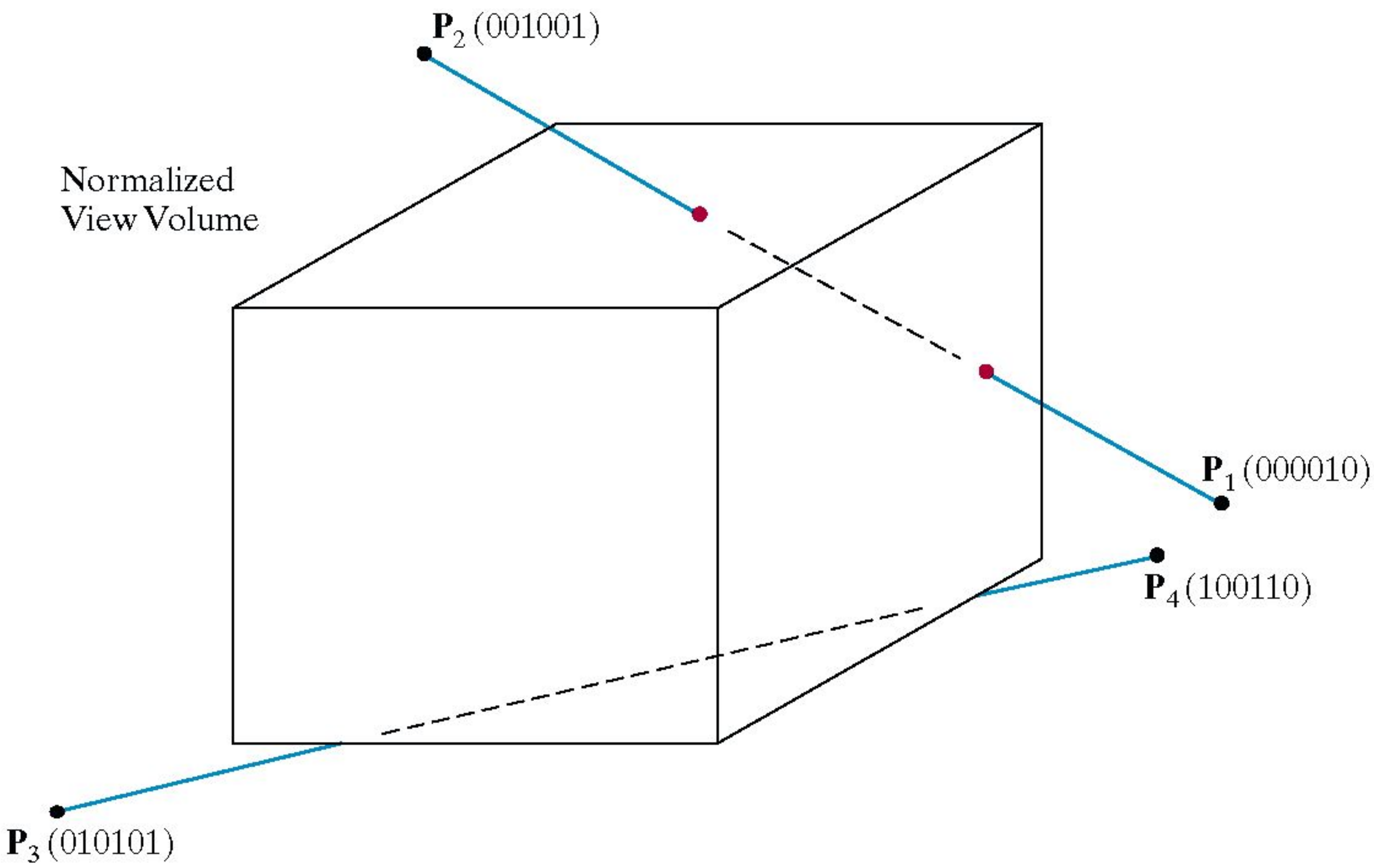
Bit 5 ≡ endpoint is *Behind* view volume = sign (z-1)

Bit 6 ≡ endpoint is *Front* view volume = sign (z_f-z)

3D Cohen-Sutherland Line Clipping

- To clip lines we first label all end points with the appropriate region codes.
- Classify the category of the Line segment as follows
  - ***Visible***: if both end points are 000000
  - ***Invisible***: if the bitwise logical AND is not 000000
  - ***Clipping Candidate***: if the bitwise logical AND is 000000
- We can trivially accept all lines with both end-points in the [000000] region.
- We can trivially reject all lines whose end points share a common bit in any position.

3D Cohen-Sutherland Line Clipping



3D Cohen-Sutherland Line Clipping

- For clipping equations for three dimensional line segments are given in their parametric form.
- For a line segment with end points $P_1(x1_h, y1_h, z1_h, h1)$ and $P_2(x2_h, y2_h, z2_h, h2)$ the parametric equation describing any point on the line is:

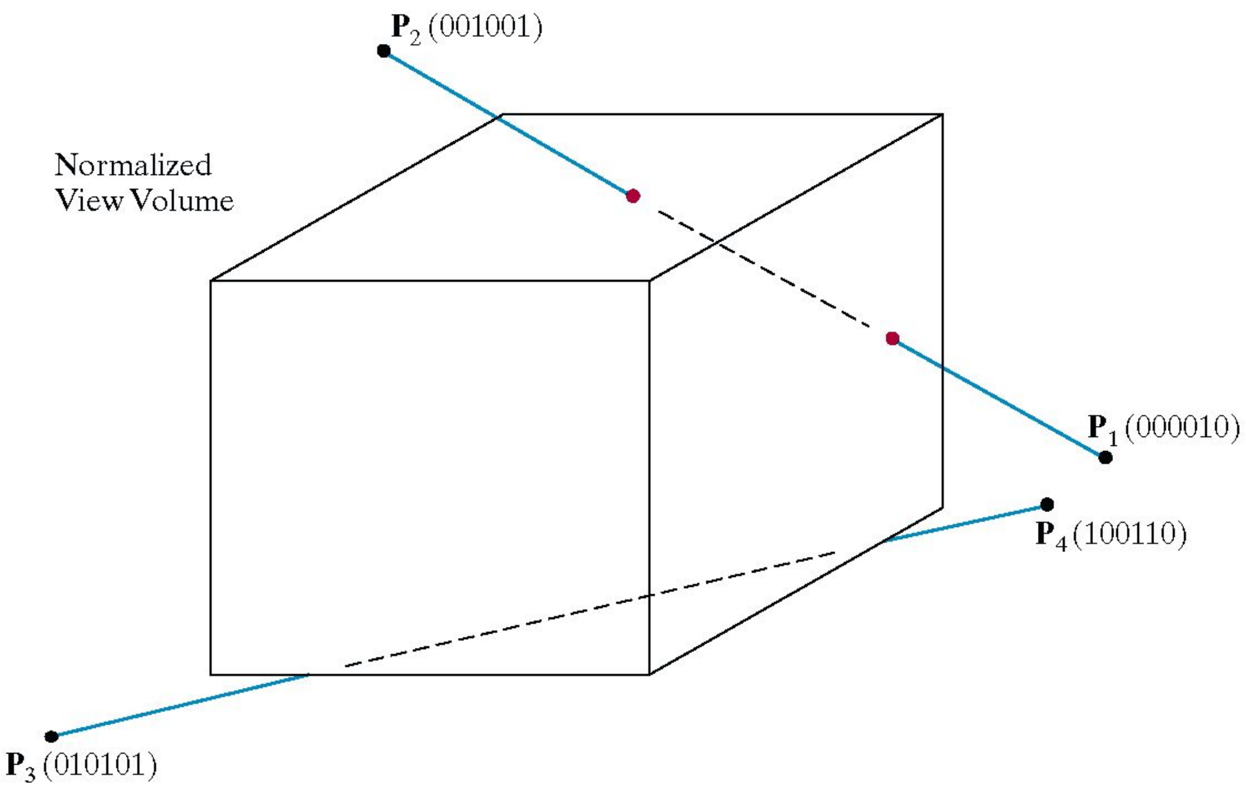
$$P = P_1 + (P_2 - P_1)u \qquad\qquad 0 \le u \le 1$$

- From this parametric equation of a line we can generate the equations for the homogeneous coordinates:

$$x_h = x1_h + (x2_h - x1_h)u$$
$$y_h = y1_h + (y2_h - y1_h)u$$
$$z_h = z1_h + (z2_h - z1_h)u$$
$$h = h1 + (h2 - h1)u$$

3D Cohen-Sutherland Line Clipping

- Consider the line P_1[000010] to P_2[001001]
- Because the lines have different values in bit 2 we know the line crosses the right boundary



3D Cohen-Sutherland Line Clipping

- Since the right boundary is at $x = 1$ we now know the following holds:

$$x_p = \frac{x_h}{h} = \frac{x1_h + (x2_h - x1_h)u}{h1 + (h2 - h1)u} = 1$$

- which we can solve for u as follows:

$$u = \frac{x1_h - h1}{(x1_h - h1) - (x2_h - h2)}$$

- using this value for u we can then solve for y_p and z_p similarly
- Then simply continue as per the two dimensional line clipping algorithm

Any Question !