ICT 4203
Computer Graphics and Animation

Lecture 15

Three-Dimensional Viewing & Clipping

Md. Mahmudur Rahman
Lecturer
Institute of Information Technology

Contents

- Three-Dimensional Viewing
- Clipping
- Viewing Transformation
- A 3D Graphics Pipeline

Three-Dimensional Viewing

- For three-dimensional applications, First of all, we can view an object from any spatial position: from the front, from above, or from the back.
- *Or* 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.
- Additionally, three-dimensional descriptions of objects must be projected onto the flat viewing surface of the output device.

Three-Dimensional Viewing

- **Viewing-Coordinates:** Generating a view of an object in three dimensions is similar to photographing the object.
- We can walk around and take its picture from any angle, at various distances, and with varying camera orientations.
- Whatever appears in the viewfinder is projected onto the flat film surface. The type and size of the camera lens determines which parts of the scene appear in the final picture.
- These ideas are incorporated into three dimensional graphics packages so that views of a scene can be generated, given the spatial position, orientation, and aperture size of the "camera".

Continue…

- **Specifying the View-plane:** We choose a particular view for a scene by first establishing the viewing- coordinate system, also called the view reference coordinate system.
- A view plane, or **projection plane**, is then set up perpendicular to the viewing *z*-axis.
- World-coordinate positions in the scene are transformed to viewing coordinates, then viewing coordinates are projected onto the view plane.
- To establish the viewing-coordinate reference frame, we first pick a world coordinate position called the view reference point.

Clipping

- Two different strategies have been invented to deal with the extraordinary computational effort required for three-dimensional clipping:

1. **Direct Clipping:** In this method, clipping is done directly against the view volume.
2. **Canonical Clipping:** In this method, normalizing transformations are applied which transform the original view volume into a canonical view volume. Clipping is then performed against the canonical view volume.
    - The canonical view volume for perspective projections is defined by planes: $x=z$, $x=-z$, $y=z$, $y=-z$, $z=z_f$ and $z=1$.

Clipping Algorithm

- Three-dimensional clipping algorithms are direct adaptations of their two-dimensional counterpart.
- The modification only arise from the fact that we are now clipping against six faces of view volume, which are planes, as opposed to the four edges of the two-dimensional window, which are lines.

The technical differences involve:

1. Finding the intersection of a line and a plane.
2. Assigning region codes to the endpoints of line segments for the Cohen-Sutherland algorithm.
3. Deciding when a point is to the right or to the left of a plane for Sutherland-Hodgman algorithm.
4. Determining the inequalities for points inside the view plane for Liang-Barsky algorithm.

Three-dimensional region codes.

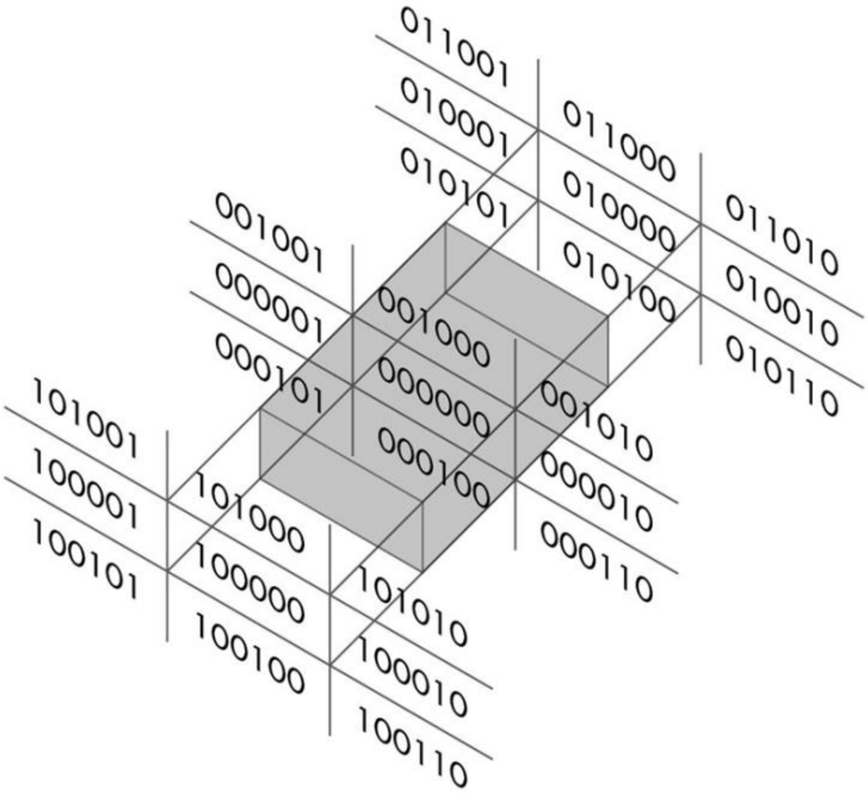


Figure: Three-dimensional six-bit region codes.

8.8: Show how region codes would be assigned to the endpoints of a line segment for the three-dimensional Cohen-Sutherland clipping algorithm for:

a) Canonical parallel view volume

b) Canonical perspective view volume

- The procedure follows the logic of the 2-D algorithm.
- For three dimensions, the planes describing the view volume divide three-dimensional space into six overlapping exterior regions (above, below, to right, to left, behind and in front of view volume), plus the interior of the view volume; thus 6-bit codes are used.
- Let P (x,y,z) be the coordinates of an endpoint.

For (a): Recall that sign (a) = 1 if a is positive, 0 otherwise.

- For the canonical parallel view volume, each bit is set to true (1) or false (0) according to the scheme:
- Bit 1 = endpoint is above view volume = sign (y-1)
- Bit 2 = endpoint is below view volume = sign (-y)
- Bit 3 = endpoint is to the right of view volume = sign (x-1)
- Bit 4 = endpoint is to the left of view volume = sign (-x)
- Bit 5 = endpoint is behind view volume = sign (z-1)
- Bit 6 = endpoint is in front of view volume = sign (-z)

8.8: Show how region codes would be assigned to the endpoints of a line segment for the three-dimensional Cohen-Sutherland clipping algorithm for:

a) Canonical parallel view volume

b) Canonical perspective view volume

- Let P (x,y,z) be the coordinates of an endpoint.

For (b): Recall that sign (a) = 1 if a is positive, 0 otherwise.

- For the canonical perspective view volume, each bit is set to true (1) or false (0) according to the scheme:
  - Bit 1 = endpoint is above view volume = sign (y-z)
  - Bit 2 = endpoint is below view volume = sign (-z-y)
  - Bit 3 = endpoint is to the right of view volume = sign (x-z)
  - Bit 4 = endpoint is to the left of view volume = sign (-z-x)
  - Bit 5 = endpoint is behind view volume = sign (z-1)
  - Bit 6 = endpoint is in front of view volume = sign (z_f-z)
  - **The category of line segment:**
  - Visible: if both region codes are 000000.
  - Not-visible: If bitwise logical AND of the region codes is not 000000.
  - Clipping-candidate: If bitwise logical AND of the region codes is 000000.

8.10: Determine the inequalities that are needed to extend the Liang-Barsky line-clipping algorithm to three-dimensions for:

a) **Canonical parallel view volume**
b) Canonical perspective view volume

- Let P_1 (x_1,y_1,z_1) and P_2 (x_2,y_2,z_2) be the coordinates of the endpoints of a line.
- $x = x_1+\Delta x.u$; $y = y_1+\Delta y.u$; $z = z_1+\Delta z.u$

Where, $0 \leq u \leq 1$, $\Delta x = x_2 - x_1$, $\Delta y = y_2 - y_1$ and $\Delta z = z_2 - z_1$. The infinite extension of the line corresponds to $u<0$ and $1<u$.

- **For (a):** points inside canonical parallel view volume satisfy-
  - $x_{min} \leq x_1+\Delta x.u \leq x_{max}$
  - $y_{min} \leq y_1+\Delta y.u \leq y_{max}$
  - $z_{min} \leq z_1+\Delta z.u \leq z_{max}$

Where $x_{min} = y_{min} = z_{min} = 0$ and $x_{max} = y_{max} = z_{max} = 1$.

- Rewrite the six inequalities as: $p_k.u \leq q_k$
- k=1,2,3,4,5,6

Where:

$P_1 = -\Delta x$	$q_1 = x_1 - x_{min} = x_1$	Left
$P_2 = \Delta x$	$q_2 = x_{max} - x_1 = 1-x_1$	Right
$P_3 = -\Delta y$	$q_3 = y_1 - y_{min} = y_1$	Bottom
$P_4 = \Delta y$	$q_4 = y_{max} - y_1 = 1-y_1$	Top
$P_5 = -\Delta z$	$q_5 = z_1 - z_{min} = z_1$	Front
$P_6 = \Delta z$	$q_6 = z_{max} - z_1 = 1-z_1$	Back

8.10: Determine the inequalities that are needed to extend the Liang-Barsky line-clipping algorithm to three-dimensions for:

a) Canonical parallel view volume

b) Canonical perspective view volume

- **For (b):** points inside canonical perspective view volume satisfy-
  - $-z \le x \le z$
  - $-z \le y \le z$
  - $z_f \le z \le 1$

  i.e:
  - $(-z_1 - \Delta z.u) \le x_1 + \Delta x.u \le (z_1 + \Delta z.u)$
  - $(-z_1 - \Delta z.u) \le y_1 + \Delta y.u \le (z_1 + \Delta z.u)$
  - $z_f \le z_1 + \Delta z.u \le 1$
- Rewrite the six inequalities as: $p_k.u \le q_k$ k=1,2,3,4,5,6

Where:

$P_1 = -\Delta x - \Delta z$	$q_1 = x_1 + z_1$	Left
$P_2 = \Delta x - \Delta z$	$q_2 = z_1 - x_1$	Right
$P_3 = -\Delta y - \Delta z$	$q_3 = y_1 + z_1$	Bottom
$P_4 = \Delta y - \Delta z$	$q_4 = z_1 - y_1$	Top
$P_5 = -\Delta z$	$q_5 = z_1 - z_f$	Front
$P_6 = \Delta z$	$q_6 = 1 - z_1$	Back

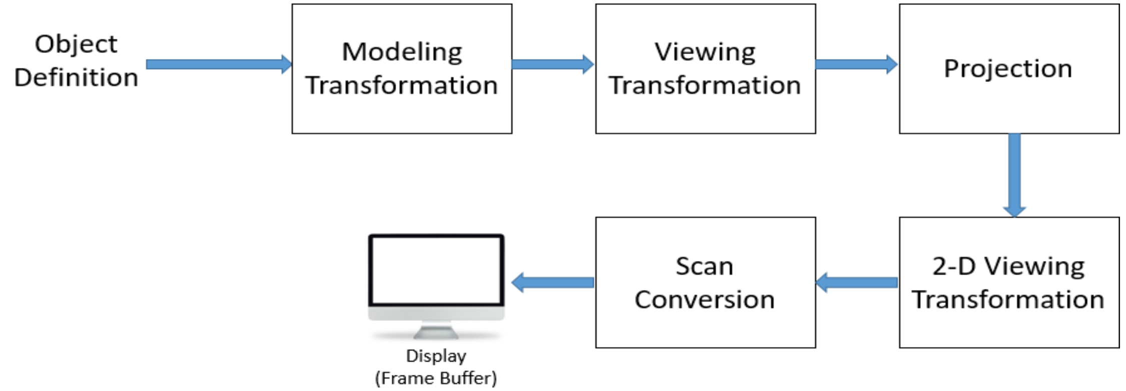
Viewing Transformation

Constructing a 3-D view: Complete 3-D viewing process steps:

- Transform from world coordinates to normalized viewing coordinates by applying the transformation.
- Clip in normalized viewing coordinates against the canonical clipping volume.
- Project onto the screen projection plane using the projections.
- Apply the appropriate viewing (2-D) viewing transformations.

A 3D Graphics Pipeline

- The 2-D graphics pipeline now be extended to 3-D, where modeling transformation first places individually defined objects into a common scene.
- Viewing transformation and projection are then carried out according to the viewing parameters set by the application. The result of projection in the view plane window is further mapped to the appropriate workstation viewpoint via 2D viewing transformation and scan-converted to a discrete image in the frame buffer for display.



Thank You