

## 3D Projection and Clipping



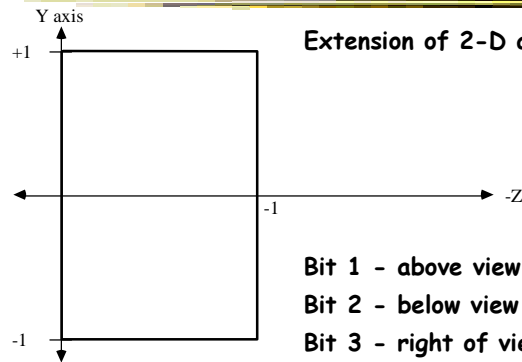
## Midterm Course Surveys



[www.coursesurvey.gatech.edu/student\\_login.cfm](http://www.coursesurvey.gatech.edu/student_login.cfm)

- Monday, Oct 4 - Sunday, Oct 10
  - Use banner ID and password
- Only I will receive the info. Let me know
  - How you think it's going
  - How you'd like me to improve things
- It's ANONYMOUS

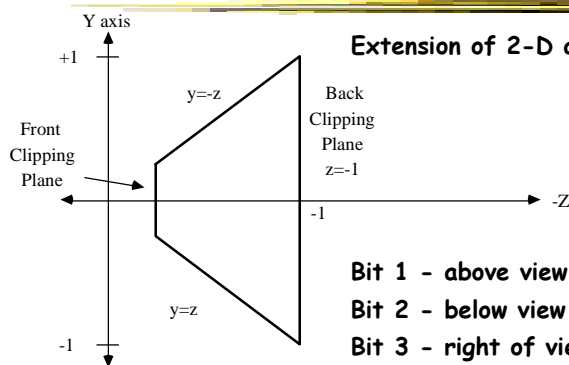
# Canonical Parallel View Volume



Extension of 2-D algorithm, 6-bit outcode

- |                              |          |
|------------------------------|----------|
| Bit 1 - above view volume    | $y > 1$  |
| Bit 2 - below view volume    | $y < -1$ |
| Bit 3 - right of view volume | $x > 1$  |
| Bit 4 - left of view volume  | $x < -1$ |
| Bit 5 - behind view volume   | $z < -1$ |
| Bit 6 - front of view volume | $z > 0$  |

# Canonical Perspective View Volume

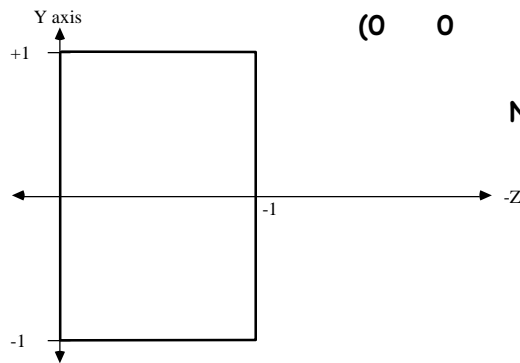


Extension of 2-D algorithm, 6-bit outcode

- |                              |               |
|------------------------------|---------------|
| Bit 1 - above view volume    | $y > -z$      |
| Bit 2 - below view volume    | $y < z$       |
| Bit 3 - right of view volume | $x > -z$      |
| Bit 4 - left of view volume  | $x < z$       |
| Bit 5 - behind view volume   | $z < -1$      |
| Bit 6 - front of view volume | $z > z_{min}$ |

## Transform Canonical Persp. Volume to Canonical Parallel Volume

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{1+z_{min}} & \frac{-z_{min}}{1+z_{min}} \\ 0 & 0 & -1 & 0 \end{pmatrix}$$



$$N'_{per} = M N_{per}$$

## Clip Against Parallel Volume

- One volume, provided by hardware
- Easy projection to screen space

## Clipping in Homogeneous Coordinates

---



- Two reasons:
  - Efficiency
  - Correctness

## Parallel Volume in Homegeneous Space

---



- 3D
  - $x = -1, x = 1, y = -1, y = 1, z = 0, z = -1$
- Homegeneous
  - $y = w, y = -w, x = w, x = -w, z = -w, z = 0$
- Clip as we did for persp volume

## Picking

---



- Goal: To use the mouse (2D) to select 3D objects
- Analytical method
  - `gluUnproject`
  
- expensive

## What are we trying to find?

---



- The objects that lie on the line that projects to the mouse position

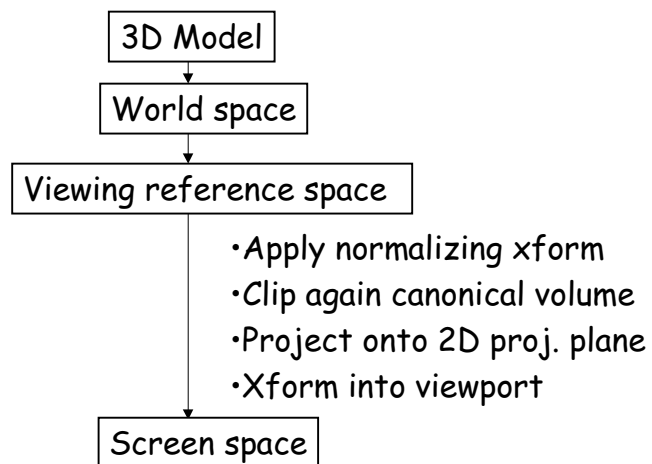
## Screen corresponds to Canonical View Volume

---

- What sliver lies under the mouse?

## 3D viewing process, again

---



## Scale Sliver to Screen: gluPickMatrix

---



- After Viewing Xform, before clipping

## How to know what gets drawn?

---



- OpenGL Selection Modes (Picking and Feedback) (chapter 13)
  
- Add "names" to rendering stream