Clipping

CS418 Computer Graphics
John C. Hart
Vertex Pipeline

Model Coords → Model Xform → World Coords → Viewing Xform → Viewing Coords → Projection

Homogeneous Divide → Still Clip Coords. → Clipping → Clip Coords. → Window Coordinates → Window to Viewport → Viewport Coordinates
Why Clip?

Why not just transform all triangles to the screen and just ignore pixels off the screen?

- Takes time to rasterize a triangle
- Very small number of triangles fall within the viewing frustum
- Output may not go directly to screen
Outcodes

- Cohen-Sutherland algorithm
- Assign segment endpoints a bitcode: \( b_3 b_2 b_1 b_0 \)
  \( b_0 = x < \text{left} \)
  \( b_1 = x > \text{right} \)
  \( b_2 = y < \text{bottom} \)
  \( b_3 = y > \text{top} \)
Outcodes

- Cohen-Sutherland algorithm
- Assign segment endpoints a bitcode: $b_3 b_2 b_1 b_0$
  - $b_0 = x < \text{left}$
  - $b_1 = x > \text{right}$
  - $b_2 = y < \text{bottom}$
  - $b_3 = y > \text{top}$
- Let $o_0 = \text{outcode}(x_0,y_0)$, $o_1 = \text{outcode}(x_1,y_1)$
Outcodes

• Cohen-Sutherland algorithm
• Assign segment endpoints a bitcode: $b_3b_2b_1b_0$
  
  \[ b_0 = x < \text{left} \]
  
  \[ b_1 = x > \text{right} \]
  
  \[ b_2 = y < \text{bottom} \]
  
  \[ b_3 = y > \text{top} \]
• Let $o_0 = \text{outcode}(x_0, y_0)$,
  \[ o_1 = \text{outcode}(x_1, y_1) \]
  
  $o_0 = o_1 = 0$: segment visible
Outcodes

- Cohen-Sutherland algorithm
- Assign segment endpoints a bitcode: $b_3 b_2 b_1 b_0$
  - $b_0 = x < \text{left}$
  - $b_1 = x > \text{right}$
  - $b_2 = y < \text{bottom}$
  - $b_3 = y > \text{top}$
- Let $o_0 = \text{outcode}(x_0,y_0)$,
  $o_1 = \text{outcode}(x_1,y_1)$
  - $o_0 = o_1 = 0$: segment visible
  - $o_0 = 0$, $o_1 \neq 0$: segment must be clipped
Outcodes

- Cohen-Sutherland algorithm

- Assign segment endpoints a bitcode: \( b_3 b_2 b_1 b_0 \)
  
  \( b_0 = x < \text{left} \)
  
  \( b_1 = x > \text{right} \)
  
  \( b_2 = y < \text{bottom} \)
  
  \( b_3 = y > \text{top} \)

- Let \( o_0 = \text{outcode}(x_0, y_0) \), \( o_1 = \text{outcode}(x_1, y_1) \)
  
  \( o_0 = o_1 = 0: \text{segment visible} \)
  
  \( o_0 = 0, o_1 \neq 0: \text{segment must be clipped} \)
  
  \( o_0 \land o_1 \neq 0: \text{segment can be ignored} \)
Outcodes

- Cohen-Sutherland algorithm
- Assign segment endpoints a bitcode: $b_3b_2b_1b_0$
  - $b_0 = x < \text{left}$
  - $b_1 = x > \text{right}$
  - $b_2 = y < \text{bottom}$
  - $b_3 = y > \text{top}$
- Let $o_0 = \text{outcode}(x_0, y_0)$, $o_1 = \text{outcode}(x_1, y_1)$
  - $o_0 = o_1 = 0$: segment visible
  - $o_0 = 0, o_1 \neq 0$: segment must be clipped
  - $o_0 \& o_1 = 0$: segment can be ignored
  - $o_0 \& o_1 = 0$: segment might need clipping
Outcodes

- Cohen-Sutherland algorithm
- Assign segment endpoints a bitcode: \( b_3b_2b_1b_0 \)
  \[ b_0 = x < \text{left} \]
  \[ b_1 = x > \text{right} \]
  \[ b_2 = y < \text{bottom} \]
  \[ b_3 = y > \text{top} \]
- Let \( o_0 = \text{outcode}(x_0,y_0) \), \( o_1 = \text{outcode}(x_1,y_1) \)
  \[ o_0 = o_1 = 0: \text{segment visible} \]
  \[ o_0 = 0, o_1 \neq 0: \text{segment must be clipped} \]
  \[ o_0 \& o_1 \neq 0: \text{segment can be ignored} \]
  \[ o_0 \& o_1 = 0: \text{segment might need clipping} \]
Serial Clipping

\[ y = \text{top} \]

\[ y = \text{bottom} \]

\[ x = \text{left} \]

\[ x = \text{right} \]
Serial Clipping

- First clip 0001
Serial Clipping

- First clip 0001
- Move \((x_0, y_0)\) to (left, …)
Serial Clipping

- First clip 0001
- Move \((x_0, y_0)\) to \((\text{left, ...})\)
- Then clip 0010
Serial Clipping

• First clip 0001
• Move \((x_0,y_0)\) to (left,\ldots)
• Then clip 0010
• Move \((x_1,y_1)\) to (right,\ldots)
Serial Clipping

- First clip 0001
- Move \((x_0, y_0)\) to (left, …)
- Then clip 0010
- Move \((x_1, y_1)\) to (right, …)
- Then clip 0100
Serial Clipping

• First clip 0001
• Move \((x_0, y_0)\) to (left,…)
• Then clip 0010
• Move \((x_1, y_1)\) to (right,…)
• Then clip 0100
• Move \((x_0, y_0)\) again, now to (…,bottom)
Serial Clipping

• First clip 0001
• Move \((x_0, y_0)\) to (left,…)
• Then clip 0010
• Move \((x_1, y_1)\) to (right,…)
• Then clip 0100
• Move \((x_0, y_0)\) again, now to (…,bottom)
• Finally clip 1000
Serial Clipping

• First clip 0001
• Move \((x_0, y_0)\) to (left, …)
• Then clip 0010
• Move \((x_1, y_1)\) to (right, …)
• Then clip 0100
• Move \((x_0, y_0)\) again, now to (…, bottom)
• Finally clip 1000
• Move \((x_1, y_1)\) again, now to (…, top)