Lecture 5

Line and Polygon Clipping

Computer Graphics & Visualization

Overview

Clipping

- Cohen-Sutherland line clipping algorithm
- Liang-Barsky line clipping algorithm
- Sutherland-Hogeman polygon clipping

Clipping Algorithms

Line Clipping:

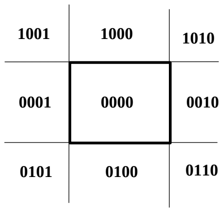
- Cohen-Suterland (encoding)
  - Oldest and most commonly used
- Nicholl-Lee-Nicholl (encoding) (more efficient)
- Liang-Barsky (parametric)
  - More efficient than Cohen-Sutherland

Polygon Clipping:

- Sutherland-Hodgeman (divide and conquer strategy)
- Weiler-Atherton (modified for concave polygons)

Cohen-Sutherland Line-Clipping

1. Encode end points
   Bit 0 = point is left of window
   Bit 1 = point is right of window
   Bit 2 = point is below window
   Bit 3 = point is above window
2. If $C_0 \wedge C_{end} \neq 0$ then
   P_0P_{end} is trivially rejected
3. If $C_0 \vee C_{end} = 0$ then
   P_0P_{end} is trivially accepted
4. Otherwise subdivide and
   go to step 1 with new segment.



C_0 = Bit code of P_0

C_{end} = Bit code of P_{end}

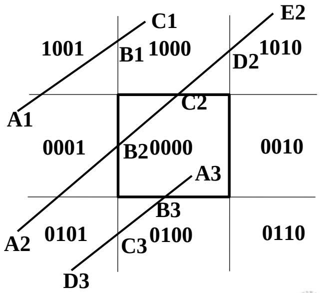
Cohen-Sutherland Line-Clipping

Clip order: Left, Right, Bottom, Top

1) A1C1
2) B1C1
3) reject

1) A2E2
2) B2E2
3) B2D2
4) B2C2
5) accept

1) A3D3
2) A3C3
3) A3B3
4) accept



Cohen-Sutherland Line-Clipping

- Will do unnecessary clipping.
- Not the most efficient.
- Clipping and testing are done in fixed order.
- Efficient when most of the lines to be clipped are either rejected or accepted (not so many subdivisions).
- Easy to program.
- Parametric clipping are more efficient.

Parametric form

- A line segment with endpoints

  (x_0, y_0) and (x_{end}, y_{end})

  we can describe in the parametric form

  $$x = x_0 + u\Delta x$$
  $$y = x_0 + u\Delta y \qquad 0 \le u \le 1$$

  where

  $$\Delta x = x_{end} - x_0$$
  $$\Delta y = y_{end} - y_0$$

Liang-Barsky Line-Clipping

- More efficient than Cohen-Sutherland
- A line is inside the clipping region for values of u such that:

  $$xw_{min} \le x_0 + u\Delta x \le xw_{max} \qquad \Delta x = x_{end} - x_0$$
  $$yw_{min} \le y_0 + u\Delta y \le yw_{max} \qquad \Delta y = y_{end} - y_0$$

- Can be described as

  $$u\, p_k \le q_k, \qquad k = 1, 2, 3, 4$$

Liang-Barsky Line-Clipping

The infinitely line intersects the clip region edges when:

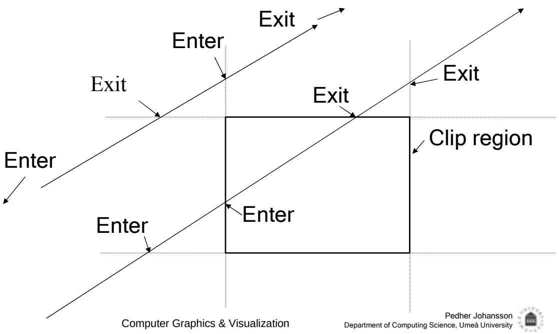
$u_k = \dfrac{q_k}{p_k}$ where

$p_1 = -\Delta x$	$q_1 = x_0 - xw_{min}$	Left boundary
$p_2 = \Delta x$	$q_2 = xw_{max} - x_0$	Right boundary
$p_3 = -\Delta y$	$q_3 = y_0 - yw_{min}$	Bottom boundary
$p_4 = \Delta y$	$q_4 = yw_{max} - y_0$	Top boundary

Liang-Barsky Line-Clipping

- When $p_k < 0$, as u increases
  - line goes from outside to inside - entering
- When $p_k > 0$,
  - line goes from inside to outside - exiting
- When $p_k = 0$,
  - line is parallel to an edge
- If there is a segment of the line inside the clip region, a sequence of infinite line intersections must go: entering, entering, exiting, exiting

Liang-Barsky Line-Clipping

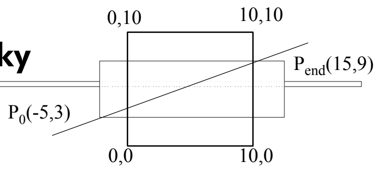


Liang-Barsky Line-Clipping

1. Set $u_{min} = 0$ and $u_{max} = 1$.
2. Calculate the u values:
3. If $u < u_{min}$ or $u > u_{max}$ ignore it. Otherwise classify the u values as entering or exiting.
4. If $u_{min} < u_{max}$ then draw a line from:

   $(x_0 + \Delta x \cdot u_{min},\ y_0 + \Delta y \cdot u_{min})$ to
   $(x_0 + \Delta x \cdot u_{max},\ y_0 + \Delta y \cdot u_{max})$

Example Liang-Barsky



$$u_{left} = \frac{q_1}{p_1} = \frac{x_0 - xw_{min}}{-\Delta x} = \frac{-5-0}{-(15-(-5))} = \frac{1}{4}$$ Entering ⇨ $u_{min} = 1/4$

$$u_{right} = \frac{q_2}{p_2} = \frac{xw_{max} - x_0}{\Delta x} = \frac{10-(-5)}{15-(-5)} = \frac{3}{4}$$ Exiting ⇨ $u_{max} = 3/4$

$$u_{bottom} = \frac{q_3}{p_3} = \frac{y_0 - yw_{min}}{-\Delta y} = \frac{3-0}{-(9-3)} = -\frac{1}{2}$$ $u < 0$ then ignore

$$u_{top} = \frac{q_4}{p_4} = \frac{yw_{max} - y_0}{\Delta y} = \frac{10-3}{9-3} = \frac{7}{6}$$ $u > 1$ then ignore

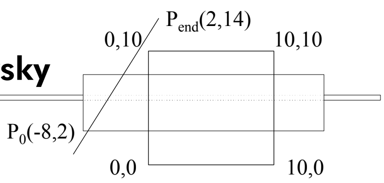
Liang-Barsky Line-Clipping

- We have $u_{min} = 1/4$ and $u_{max} = 3/4$

  $P_{end} - P_0 = (15+5, 9-3) = (20,6)$ (Δx, Δy)

- If $u_{min} < u_{max}$, there is a line segment
  - compute endpoints by substituting u values
- Draw a line from
  (-5+(20)·(1/4), 3+(6)·(1/4))
  to
  (-5+(20)·(3/4), 3+(6)·(3/4))

Example Liang-Barsky



$$u_{left} = \frac{q_1}{p_1} = \frac{x_0 - xw_{min}}{-\Delta x} = \frac{-8-0}{-(2-(-8))} = \frac{4}{5}$$ Entering ⇨ $u_{min} = 4/5$

$$u_{right} = \frac{q_2}{p_2} = \frac{xw_{max} - x_0}{\Delta x} = \frac{10-(-8)}{2-(-8)} = \frac{9}{5}$$ $u > 1$ then ignore

$$u_{bottom} = \frac{q_3}{p_3} = \frac{y_0 - yw_{min}}{-\Delta y} = \frac{2-0}{-(14-2)} = -\frac{1}{6}$$ $u < 0$ then ignore

$$u_{top} = \frac{q_4}{p_4} = \frac{yw_{max} - y_0}{\Delta y} = \frac{10-2}{14-2} = \frac{2}{3}$$ Exiting ⇨ $u_{max} = 2/3$

Liang-Barsky Line-Clipping

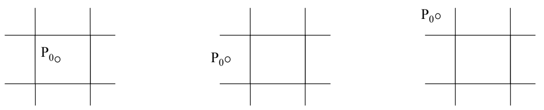
- We have $u_{min} = 4/5$ and $u_{max} = 2/3$

  $P_{end} - P_0 = (2+8, 14-2) = (10, 12)$

- $u_{min} > u_{max}$,
  there is no line segment do draw

Nicholl-Lee-Nicholl Line Clipping

- Avoids multiple clipping of an individual line by creating more regions.
- Only three regions need to be considered.

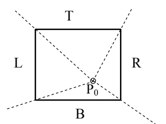


inside edge region corner region

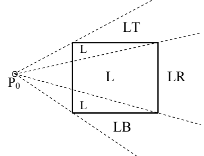
- Find position of P_{end} relative to P_0.

Nicholl-Lee-Nicholl Line Clipping

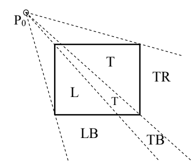
If P_0 inside and P_{end} outside:



If P_0 is to the left:



If P_0 is to the left and above:



Nicholl-Lee-Nicholl Line Clipping

To determine region

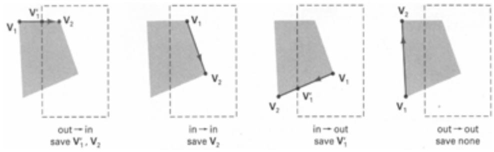
- compare slopes, and
- boundaries of the NLN region.

Pedher Johansson
Department of Computing Science, Umeå University

Suterland-Hodgeman Polygon Clipping

Four test cases:

1. First vertex inside and the second outside (in-out pair)
2. Both vertices inside clip window
3. First vertex outside and the second inside (out-in pair)
4. Both vertices outside the clip window



- Concave polygons may be displayed with extra lines.

Pedher Johansson
Department of Computing Science, Umeå University

Weiler-Atherton Polygon Clipping

- Clips concave polygons correctly.
- Instead of always going around the polygon edges, we also, want to follow window boundaries.
- For an outside-to-inside pair of vertices, follow the polygon boundary.
- For an inside-to-outside pair of vertices, follow the window boundary in a clockwise direction.

Pedher Johansson
Department of Computing Science, Umeå University