Computer Graphics

P. Healy

CS1-08
Computer Science Bldg.
tel: 202727
patrick.healy@ul.ie

Spring 2021–2022

Outline

Announcements

- Mid-term: ?

Liang-Barsky Line Clipper

- Liang-Barsky algorithm is a better approach by avoiding repeated shortening of segment $s - e$; like C-S also works for 3-D clipping regions
- Idea is to return to parametric form of line (see prev. lecture)

$$
\begin{aligned}
x &= x_s + u\Delta x \\
y &= y_s + u\Delta y, \qquad 0 \leq u \leq 1
\end{aligned}
$$

where

$$
\begin{aligned}
\Delta x &= x_e - x_s, \\
\Delta y &= y_e - y_s
\end{aligned}
$$

Liang-Barsky Line Clipper

- Liang-Barsky algorithm is a better approach by avoiding repeated shortening of segment $s - e$; like C-S also works for 3-D clipping regions
- Idea is to return to parametric form of line (see prev. lecture)

$$x = x_s + u\Delta x$$
$$y = y_s + u\Delta y, \qquad 0 \leq u \leq 1$$

where

$$\Delta x = x_e - x_s,$$
$$\Delta y = y_e - y_s$$

Liang-Barsky Line Clipper

- Applying "window" test

$$
\begin{aligned}
x_{bl} &\leq x_s + u\Delta x \leq x_{ur} \\
y_{bl} &\leq y_s + u\Delta y \leq y_{ur}
\end{aligned}
$$

- We break this in to four inequalities of the form $u_k p_k \leq q_k$

$$
\begin{aligned}
p_1 &= -\Delta x, & q_1 &= x_s - x_{bl} \text{ (left border)} \\
p_2 &= \Delta x, & q_2 &= x_{ur} - x_s \text{ (right border)} \\
p_3 &= -\Delta y, & q_3 &= y_s - y_{bl} \text{ (bot border)} \\
p_4 &= \Delta y, & q_4 &= y_{ur} - y_s \text{ (top border)}
\end{aligned}
$$

- We can now evaluate the four u_ks with $u_k = q_k / p_k$
- Liang-Barsky: it's all about finding the 4 u_ks

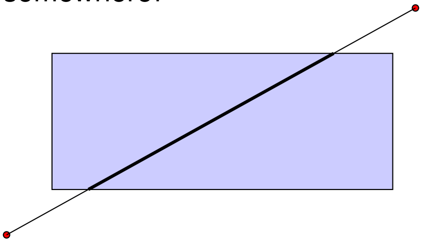
Liang-Barsky Line Clipper

- We want to find the points where line segment crosses borders
- Extension of line segment will cross all four borders – somewhere!

- We do this by working with the u_ks

Liang-Barsky Line Clipper

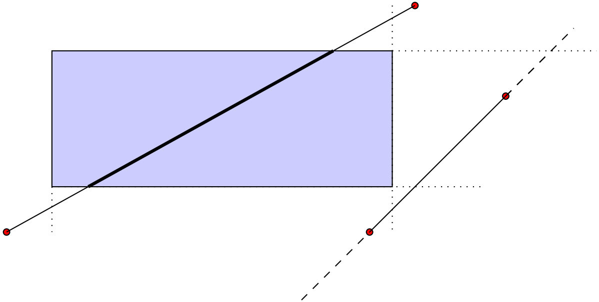
- We want to find the points where line segment crosses borders
- Extension of line segment will cross all four borders – somewhere!



- We do this by working with the u_ks

Liang-Barsky Line Clipper

- We want to find the points where line segment crosses borders
- Extension of line segment will cross all four borders – somewhere!



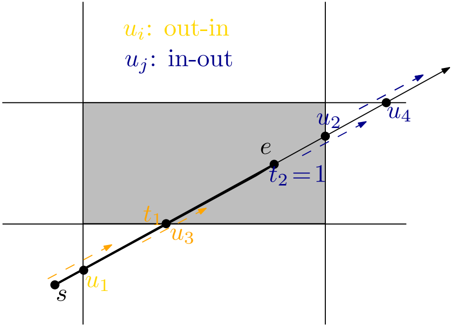
- We do this by working with the u_ks

Liang-Barsky Line Clipper (contd.)

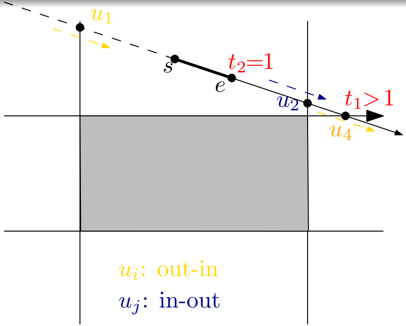
- If line is parallel to window border then either $\Delta x = 0$ or $\Delta y = 0$; so, two of p_is will be 0, also (either p_1, p_2 or p_3, p_4)
  - For each p_i that is 0, if $q_i < 0$ then line is entirely outside that clipping boundary; o/w, $q_i \geq 0$ and line is inside
- Proceeding from $u = -\infty$ to $u = +\infty$ the infinite extension of **every** non-parallel (to borders) line will cut 2 borders from outside to inside and 2 borders from inside to out
- If $p_k < 0$, with respect to that border, the infinite extension of the line moves from outside to inside; $p_k > 0$ means the line proceeds from inside to outside
- So two p_is will be positive and two negative
- The value of $u_k = q_k/p_k$ tells us critical information about the line

Liang-Barsky Line Clipper (contd.)

- Instead of maintaining four u_is, we only maintain 2:
  - t_1 for the "outside to inside cases" ($p_k < 0$)
  - t_2 for the "inside to outside cases" ($p_k > 0$)
- t_1 is initialised to 0 (corresponds to s) and is taken to be the **larger** of the two cases where $p_k < 0$
- t_2 is initialised to 1 (corresponds to e) and is taken to be the **smaller** of the two cases where $p_k > 0$



Liang-Barsky Line Clipper (contd.)



- if $t_1 > t_2$ line is completely outside the clipping region and it can be discarded
- Over Cohen-Sutherland, Liang-Barsky has been found to run 36% faster for 2-D lines and 40% faster for 3-D lines
- Note: alg doesn't depend on any "ordering" of start and end points of line
- See tutorial here