Extraction of data/primitives inside a region of interest “window”

=> Discard (parts of ) primitives outside window.

**Point Clipping:** Remove points outside window.

- A point is either entirely inside the window or not.

\[ (x_L, y_B) \leq (x, y) \leq (x_R, y_T) \]

- **Q** is inside.
- **P** is outside.
Line Clipping: Remove portion of line segment outside window

• Can we use point clipping for the end points?

Point clipping works
Line Clipping: Remove portion of line segment outside window

• How about these lines?

Point clipping does not work
Clipping

Cohen and Sutherland
Cohen and Sutherland

4 bit code to indicate the zone of end points of line with respect to window
Clipping

Cohen and Sutherland

4 bit code to indicate the zone of end points of line with respect to window
Cohen and Sutherland

Trivially accept case
- line is totally visible
- if both ends of the line have outcode as 0000
Cohen and Sutherland

Trivially reject case
- line is totally invisible
- logical AND of the two end points outcodes
Cohen and Sutherland

If not trivially reject and accept case
• line is potentially visible
Cohen and Sutherland

If potentially visible
  • subdivide into segments and apply trivial acceptance and rejection test
  • segments by intersection with window edges
  • edges in any order but consistent (e.g., top-bottom, right-left)
Cohen and Sutherland

- simple, still popular
- limited to rectangular region
- extension to 3D clipping using 3D orthographic view volume is straightforward
Cyrus Beck Line Clipping
(Liang and Barsky)

• any convex region

Parametric line (input line AB):

\[ L(t) = A + (B - A)t; t \in (0,1) \]
Cyrus Beck Line Clipping
(Liang and Barsky)

Implicit line (window edge):

\[ l(Q) = (Q - P) \cdot n \]

Tells us on which side of the line the point Q is.
Cyrus Beck Line Clipping
(Liang and Barsky)

Evaluate

\[ I(Q) = (Q - P).n \]

If > 0 inside halfspace of line (plane)
If < 0 outside halfspace of line (plane)
If = 0 on the line

Should give indications for trivial accept and reject cases.
Cyrus Beck Line Clipping
(Liang and Barsky)

Window edge: \( I(Q) = (Q - P) \cdot n \)

Line segment: \( L(t) = A + t(B - A) \)

Trivial Reject: \( I(A) < 0 \) AND \( I(B) < 0 \)

Trivial Accept: \( I(A) > 0 \) AND \( I(B) > 0 \)
Cyrus Beck Line Clipping
(Liang and Barsky)
Cyrus Beck Line Clipping (Liang and Barsky)

\[ L(t) = A + (B - A)t \]
\[ I(Q) = (Q - P).n \]
\[ I(L(t)) = 0; \text{solve for } t \]
\[ (L(t) - P).n = 0 \]
\[ (A + t(B - A) - P).n = 0 \]
\[ (A - P).n + t(B - A).n = 0 \]
Cyrus Beck Line Clipping
(Liang and Barsky)

\[ t = \frac{(A - P).n}{(B - A).n} \]

\[ t = \frac{(A - P).n}{(A - P).n - (B - P).n} \]
Cyrus Beck Line Clipping (Liang and Barsky)

Which ‘t’ to select?
Cyrus Beck Line Clipping (Liang and Barsky)

\[ t = \frac{(A - P).n}{(B - A).n} \]

\[ D = (B - A).n \]

\[ D > 0 \text{ label } t \text{ as } t_E \]

Entering
Cyrus Beck Line Clipping (Liang and Barsky)

\[ t = \frac{(A - P) \cdot n}{(B - A) \cdot n} \]

\[ D = (B - A) \cdot n \]

If \( D < 0 \) label \( t \) as \( t_L \), and Leaving.
Cyrus Beck Line Clipping
(Liang and Barsky)

A (t=0)

B (t=1)

t of interest:
largest $t_E$
smallest $t_L$

$\max t_E$
$\min t_L$
Cyrus Beck Line Clipping
(Liang and Barsky)

If \( t_{E}^{\text{max}} > t_{L}^{\text{min}} \)

Reject
Cyrus Beck Line Clipping (Liang and Barsky)

Arbitrary Convex Window
Cyrus Beck Line Clipping (Liang and Barsky)

Arbitrary Convex Window

Polygon is convex if for all adjacent edges the sign of cross product is same.
Cyrus Beck Line Clipping (Liang and Barsky)

Arbitrary Window

\[ E_1 \times E_2 : \text{positive} \]
\[ E_2 \times E_3 : \text{positive} \]
\[ E_3 \times E_4 : \text{negative} \]

Polygon is non-convex
Cyrus Beck Line Clipping
(Liang and Barsky)

Arbitrary Window

Make the polygon convex by adding the edge $V_3 V_5$

Clip against the convex polygon => $P_3 P_4$
Cyrus Beck Line Clipping
(Liang and Barsky)

Arbitrary Window

Clip against the triangle
\( \Rightarrow P_5P_4 \)

Subtract \( P_5P_4 \) from \( P_3P_4 \)
\( \Rightarrow P_3P_5 \)