Computer Graphics

Lecture-11
Two –Dimensional Viewing and Clipping

Md Imtiaz Ahmed
Lecturer
DIIT

Liang-Barsky Algorithm

- The following parametric equations represent a line from (x1,x2) to (x2,y2) along with its infinite extension:

$$x = x_1 + \Delta x.u$$

$$y = y_1 + \Delta y.u$$

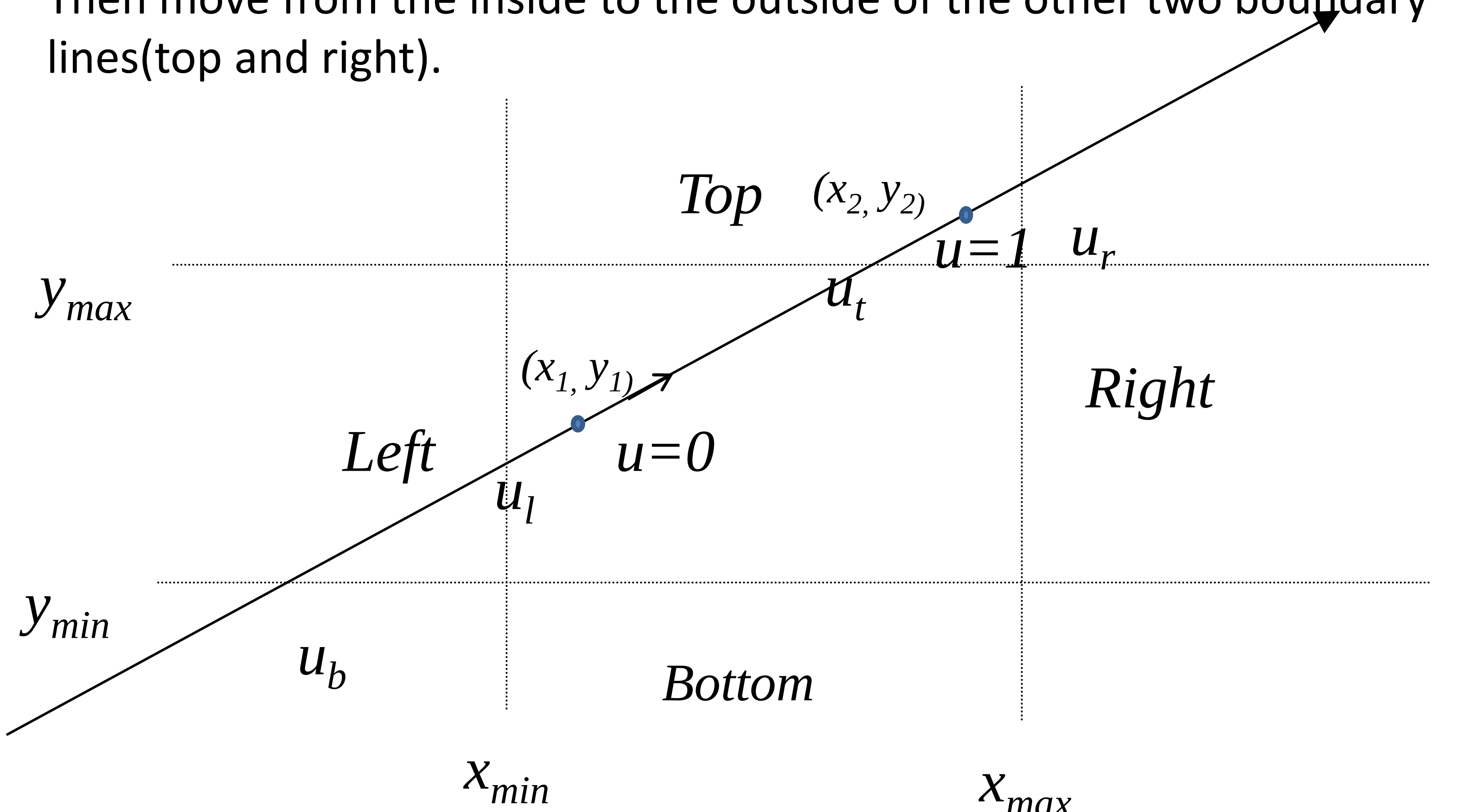
- Where

$$\Delta x = x_2 - x_1$$

$$\Delta y = y_2 - y_1$$

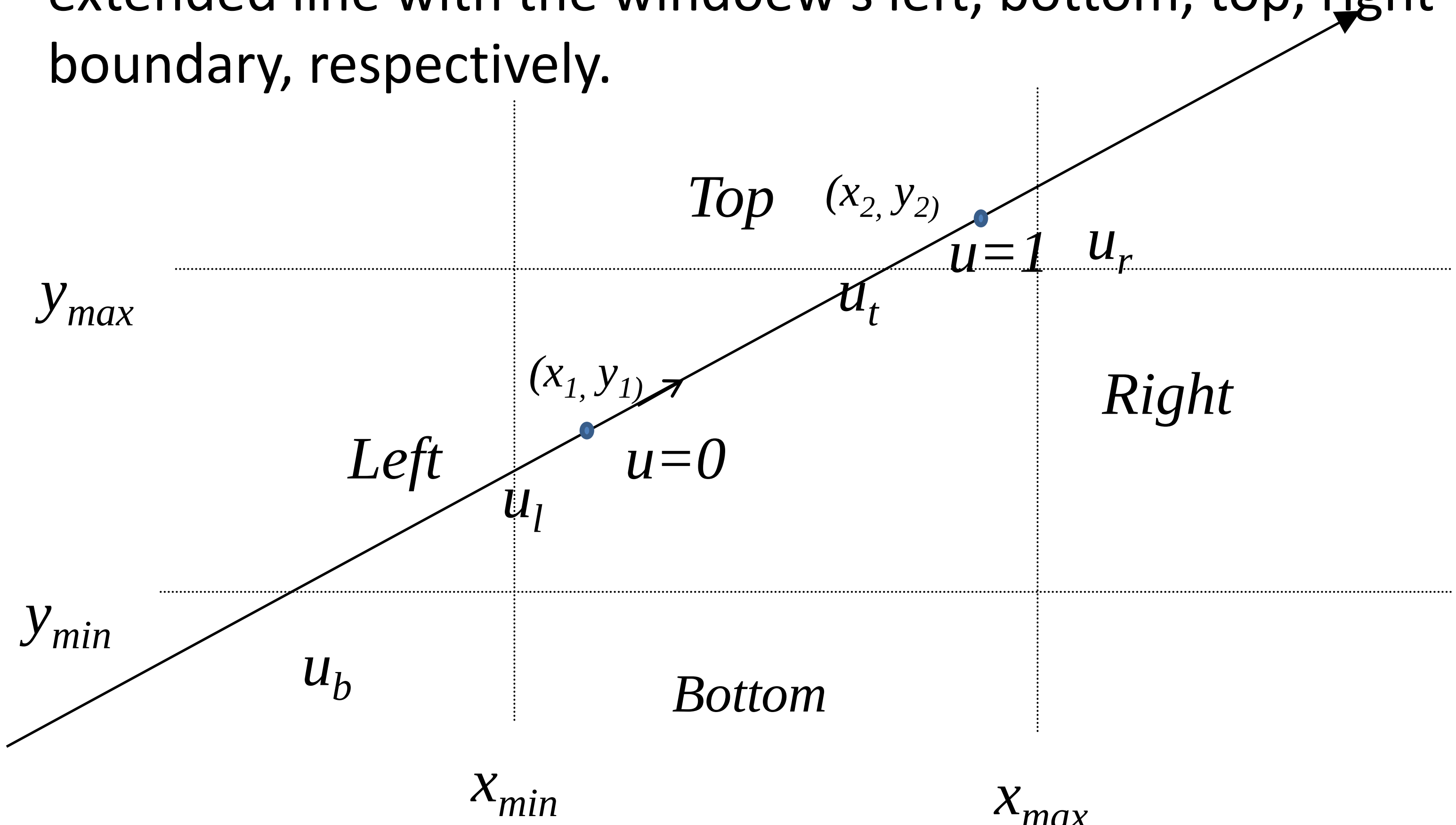
Liang-Barsky Algorithm

- The line itself corresponds to 0<=u<=1.
- U increasing from - ∞ to ∞
- First move from the outside to the inside of the clipping window's two boundary llines(bottom and left)
- Then move from the inside to the outside of the other two boundary lines(top and right).

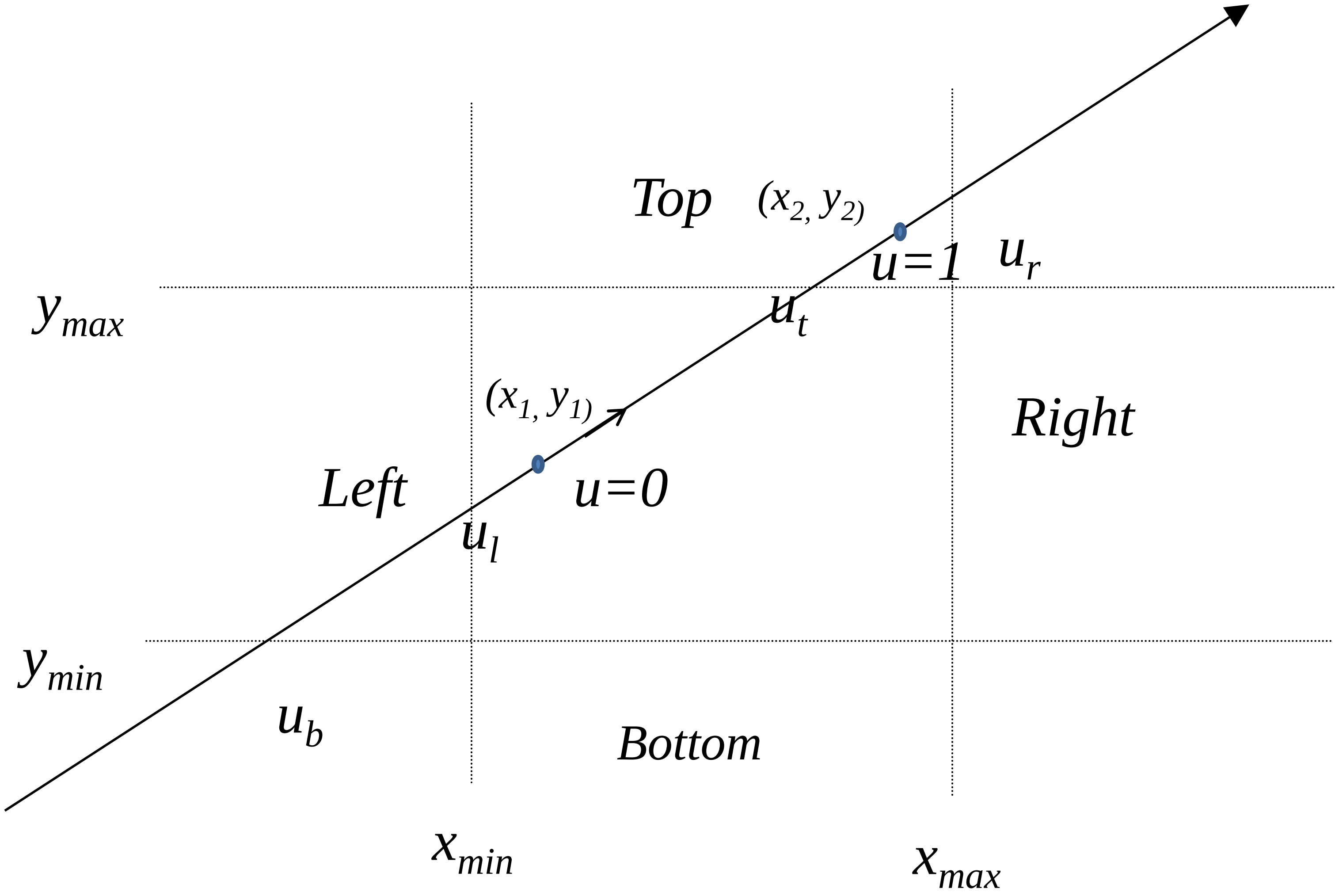


Liang-Barsky Algorithm

- u_1=maximum($0,u_l,u_b$) and u_2=minimum($1,u_t,u_r$)
- u_l,u_b,u_t,u_r correspond to the intersection point of the extended line with the windoew's left, bottom, top, right boundary, respectively.



Parametric Intersection



Liang-Barsky Algorithm

- For point (x,y) inside the clipping window, we have

$$x_{\min} \le x_1 + u\Delta x \le x_{\max}$$
$$y_{\min} \le y_1 + u\Delta y \le y_{\max}$$

- Rewrite the four inequalities as

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

- Where

$p_1 = -\Delta x,$	$q_1 = x_1 - x_{\min}$	Left
$p_2 = \Delta x,$	$q_2 = x_{\max} - x_1$	Right
$p_3 = -\Delta y,$	$q_3 = y_1 - y_{\min}$	Buttom
$p_4 = \Delta y$	$q_4 = y_{\max} - y_1$	Top

Observation

- If $p_k = 0$, the line is parallel to the corresponding boundary and

  $q_k < 0,$ the line is completely outside the boundary and can be eliminated

  $q_k \geq 0,$ the line is inside the boundary and needs further consideration,

- If $p_k < 0,$ the extended line proceeds from the outside to the inside of the corresponding boundary line
- If $p_k > 0,$ the extended line proceeds from the inside to the outside of the corresponding boundary line
- When $p_k \neq 0,$ the value of u that corresponds to the intersection point is q_k / p_k

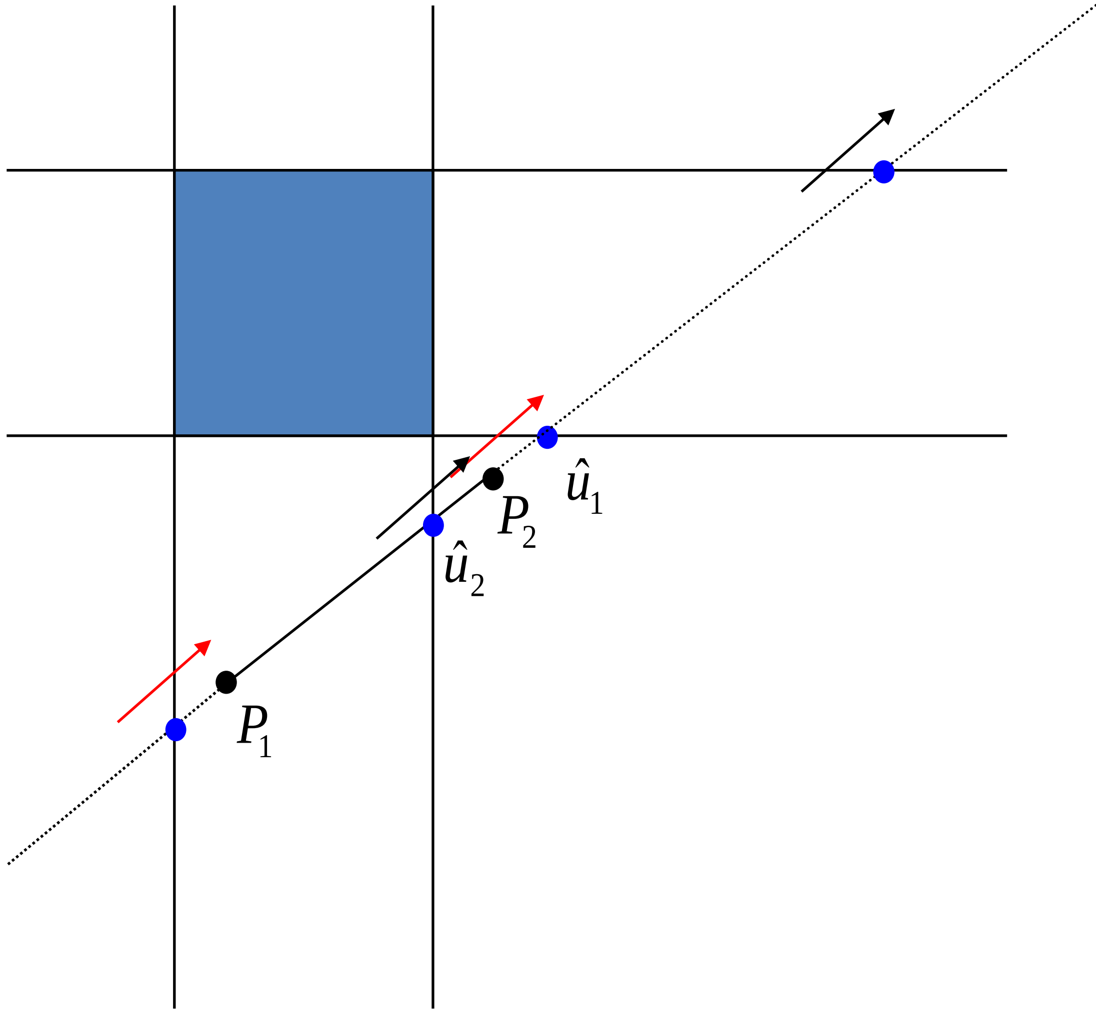
Liang-Barsky - Algorithm

- *If* p_k=0 and q_k<0 for any k, eliminate the line and stop. Otherwise proceed to the next step.
- For all k such that p_k<0, calculate $r_k = q_k/p_k$. Let u_1 *be the maximum of the set containing 0 and the calculated r values.*
- For all k such that p_k>0, calculate $r_k = q_k/p_k$. Let u_2 *be the minimum of the set containing 1 and the calculated r values.*
- *If* $u_1 > u_2$, eliminate the line since it is completely outside the clipping window. Otherwise , use u_1 *and* u_2 *to calculate the end points of the clipped line.*

Line Clipping – Liang-Barsky

- If u1 > u2, the line lies completely outside of the clipping area.
- Otherwise the segment from u1 to u2 lies inside the clipping window.

if $\hat{u}_1 > \hat{u}_2$, rejected.



Example

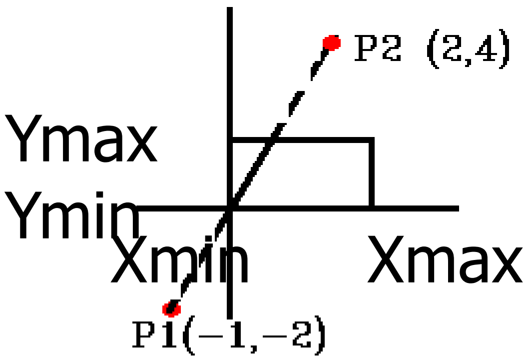
Let P1 (-1, -2), P2 (2, 4)

$X_{min} = 0$, $X_{max} = 1$, $Y_{min} = 0$, $Y_{max} = 1$

- dx = 2 - (-1) = 3; dy = 4 - (-2) = 6
- P1 = -dx = -3; q1 = x1 - X_{min} = -1 - 0 = -1; u_1 = q1 / P1 = 1/3 Left
- P2 = dx = 3; q2 = X_{max} - x1 = 1 - (-1) = 2; u_2 = q2 / P2 = 2/3 Right
- P3 = -dy = -6; q3 = y1 - Y_{min} = -2-0 = -2; u_3 = q3 / P3 = 1/3 Buttom
- P4 = dy = 6; q4 = Y_{max} - y1 = 1-(-2) = 3; u_4 = q4 / P4 = ½ Top
- for (Pk < 0) u’1 = MAX(1/3, 1/3, 0) = 1/3
- for (Pk> 0) u’2 = MIN(2/3, 1/2, 1) = 1/2 Since u’1 < u’2 there is a visible section

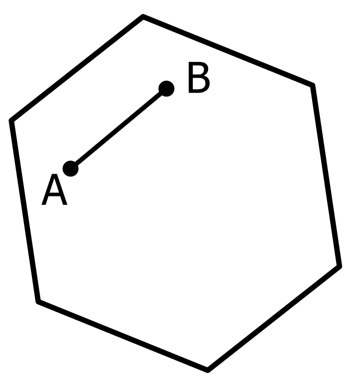
compute new endpoints

- U’1 = 1/3;
- x1' = x1 + dx.u’1 = -1 + (3 . 1 / 3) = 0
- y1' = y1 + dy.u’1 = -2 + (6 . 1 / 3) = 0
- U’2 = ½;
- x2' = x1 + dx.u’2 = -1 + (3 . 1 / 2) = 1 / 2
- y2' = y1 + dy.u’2 = -2 + (6 . 1 / 2) = 1

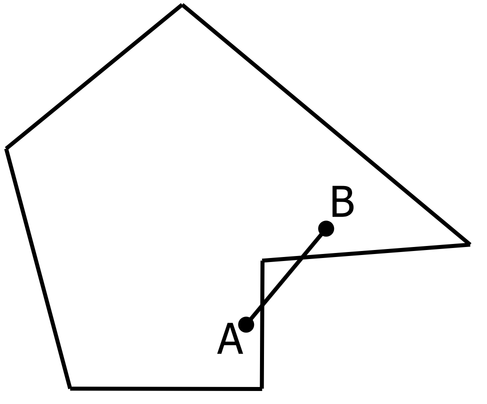


Polygon Clipping

- Convex Polygonal Clipping Windows:
  - A polygonal is called convex if the line joining any two interior points of the polygon lies completely inside the polygon
  - A non convex polygon is said to be concave



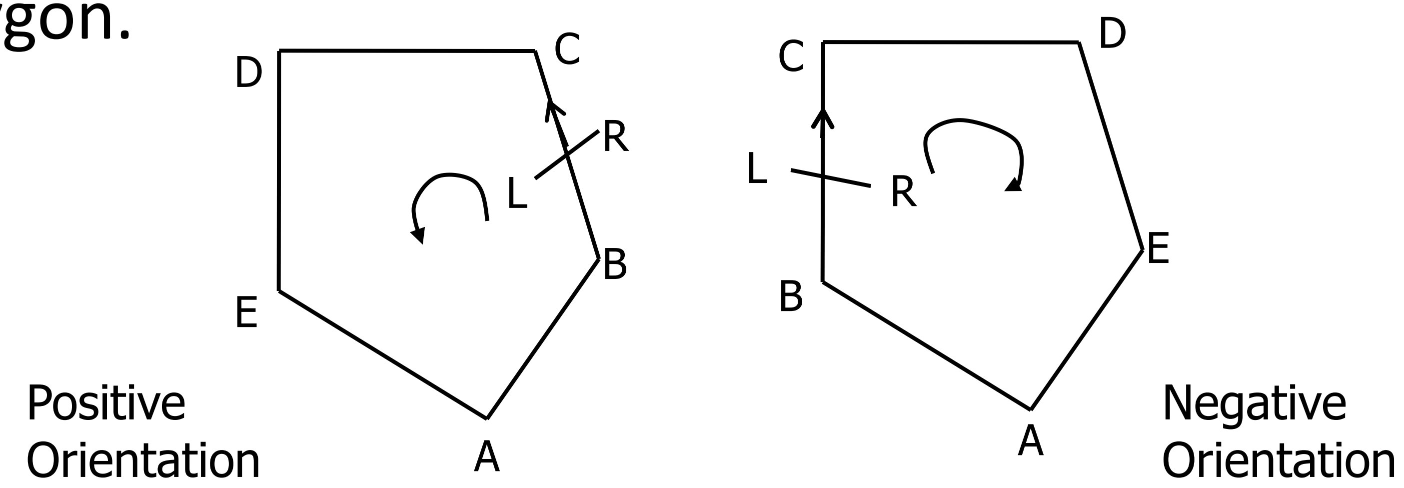
Convex Polygon



Concave Polygon

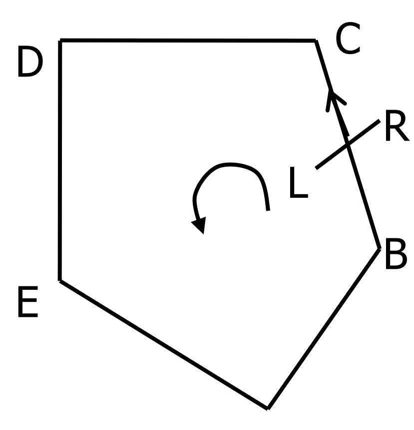
Polygon Clipping

- A Polygon with vertices P_1 .....P_N (and edges P_iP_{i-1} and P_1P_N) is said to be positively oriented if a tour of the vertices in the given order produces a counterclockwise circuit.
- The left hand of a person standing along any directed edge $\overline{P_iP_{i-1}}$ or $\overline{P_1P_N}$ would be pointing inside the polygon.

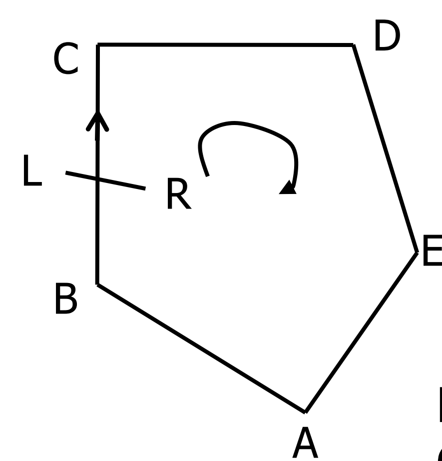


Polygon Clipping

- $A(x_1,y_1)$ and (x_2,y_2) be the end points of a directed line segment
- A point p(x,y) will be to the left of the line segment if the expression $C=(x_2-x_1)(y-y_1)-(y_2-y_1)(x-x_1)$ is positive.
- The point is to the right of the line segment if this quantity is negative.
- If a point p is to the right of any one edge of a positively oriented, convex polygon, it is outside the polygon
- If it is to the left of every edge of the polygon, it is inside the polygon.



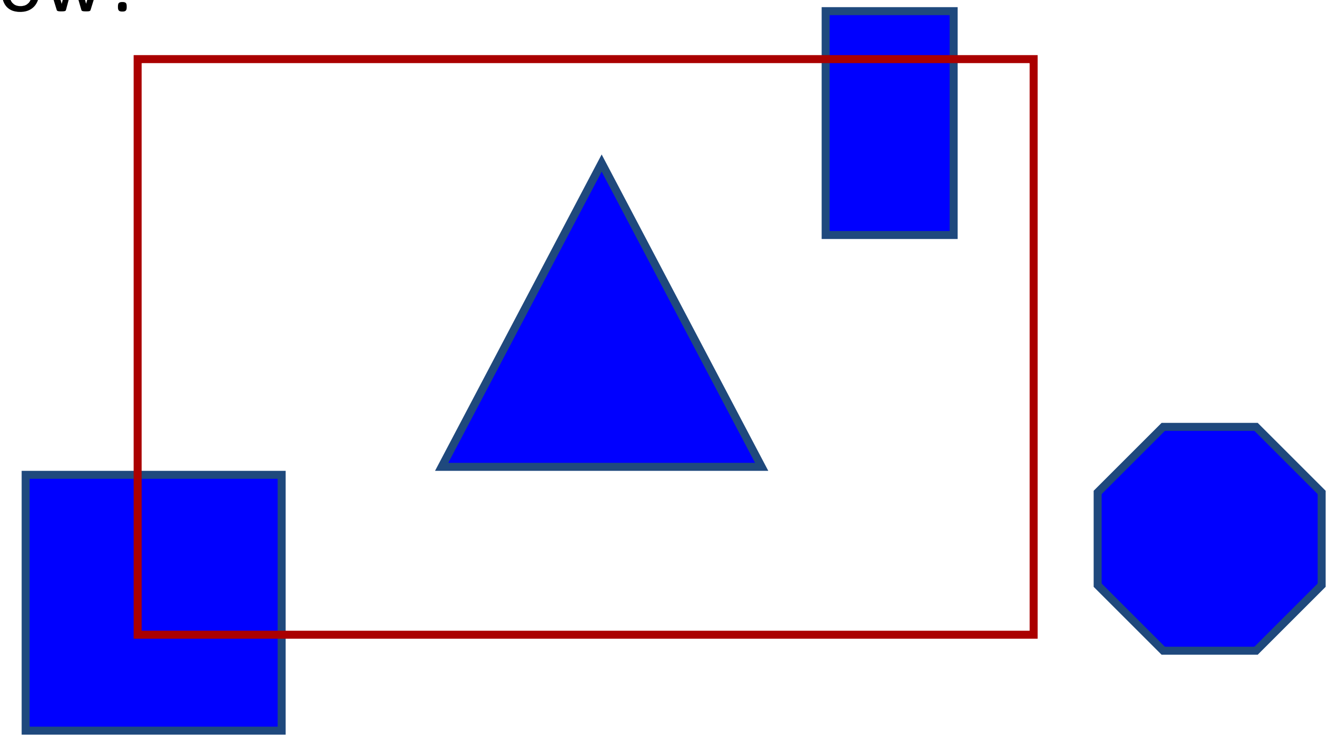
Positive Orientation



Negative Orientation

Polygon Clipping

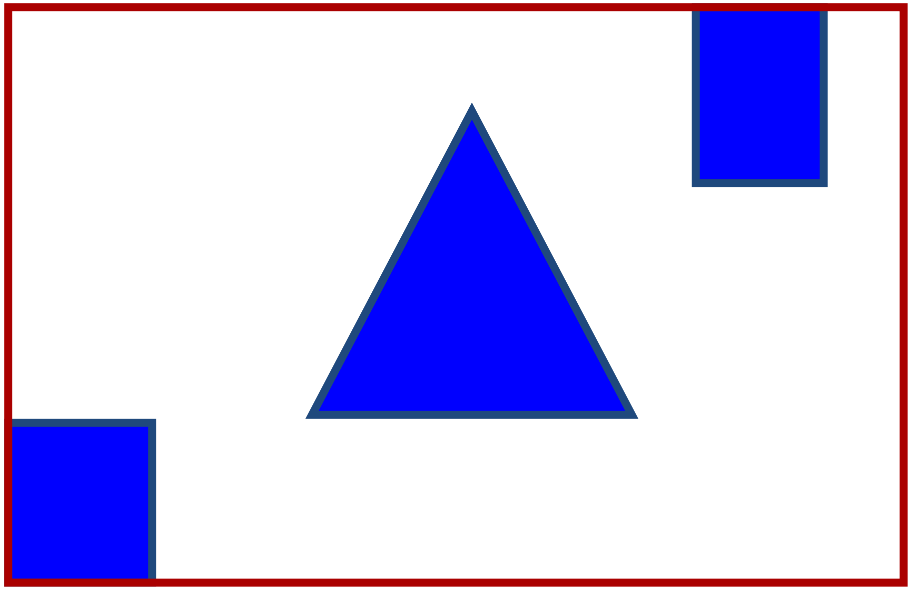
- Find the Part of a Polygon Inside the Clip Window?



Before Clipping

Polygon Clipping

- Find the Part of a Polygon Inside the Clip Window?



After Clipping

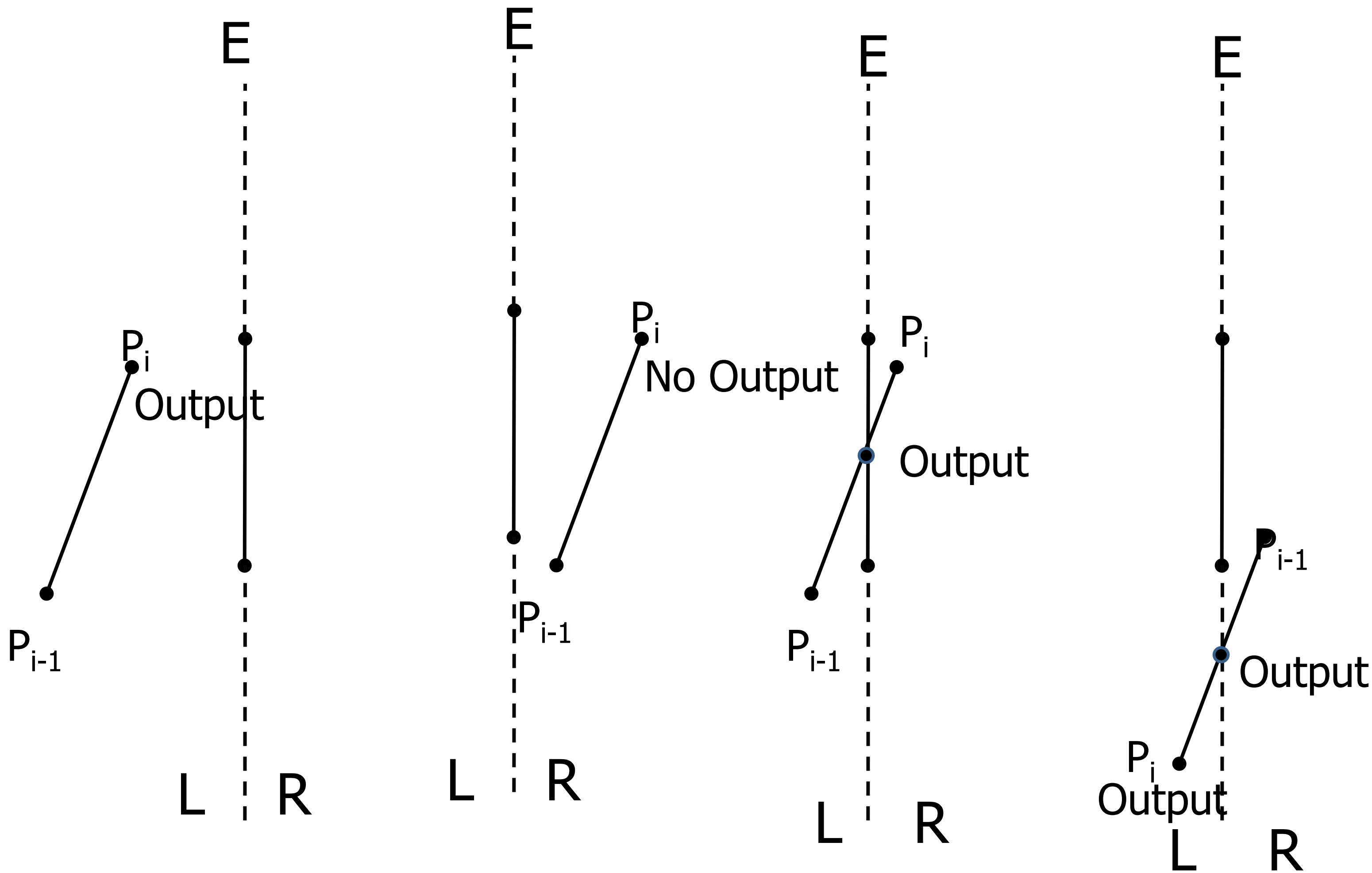
Sutherland-Hodgeman Polygon Clipping

- Let P_1 .....P_N be the vertex list of the polygon to be clipped. Let edge E, determined by endpoints A and B, be any edge of the positively oriented, convex clipping polygon.
- Clip each edge of the polygon in turn against the edge E of the clipping polygon, forming a new polygon whose vertices are determined as follows:

Sutherland-Hodgeman Polygon Clipping

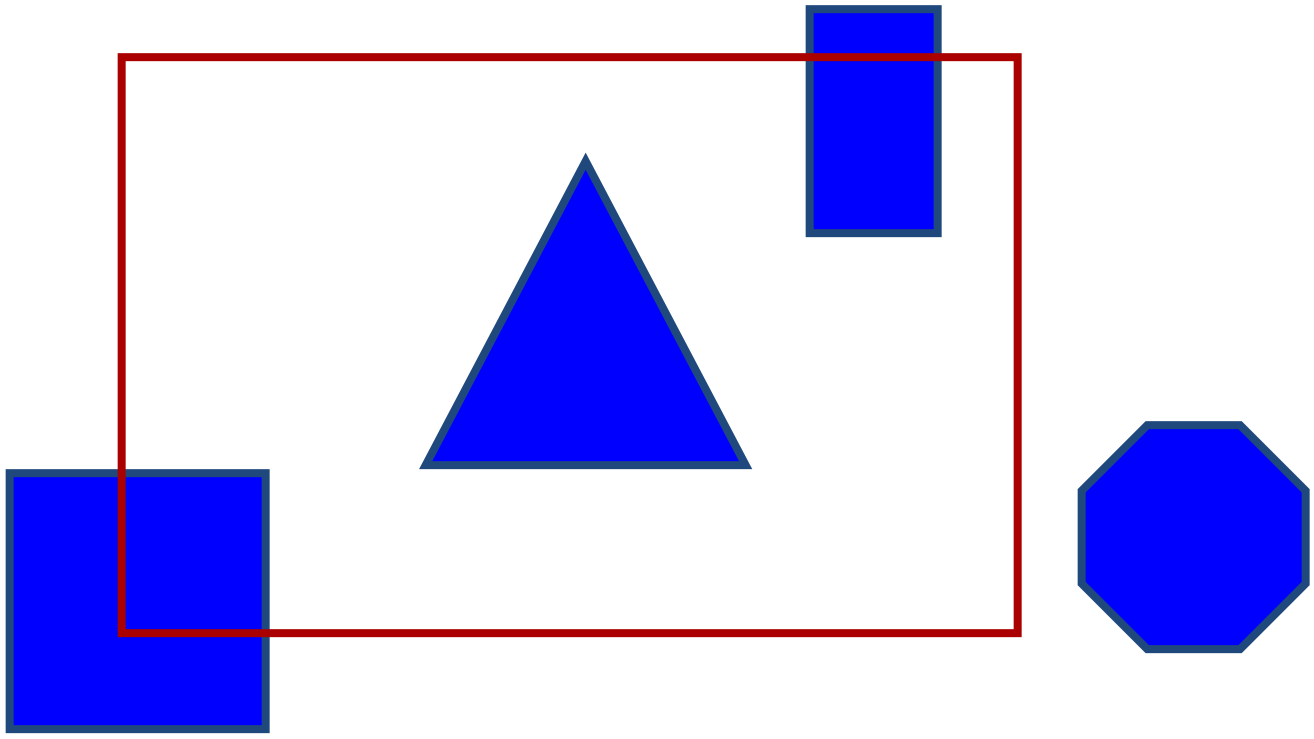
- Consider the edge $\overline{P_{i-1}P_i}$
- If both P_{i-1} and P_i are to the left of the edge, vertex P_i is placed on the vertex output list of the clipped polygon
- If both P_{i-1} and P_i are to the right of the edge, nothing is placed on the vertex output list of the clipped polygon
- If both P_{i-1} to the left and P_i is to the right of the edge E, the intersection point *I* of the line segment $\overline{P_{i-1}P_i}$ with the extended edge E is calculated and placed on the vertex output list.
- If both P_{i-1} to the right and P_i is to the left of the edge E, the intersection point *I* of the line segment $\overline{P_{i-1}P_i}$ with the extended edge E is calculated. Both *I* and P_i are placed on the vertex output list.

Sutherland-Hodgeman Polygon Clipping



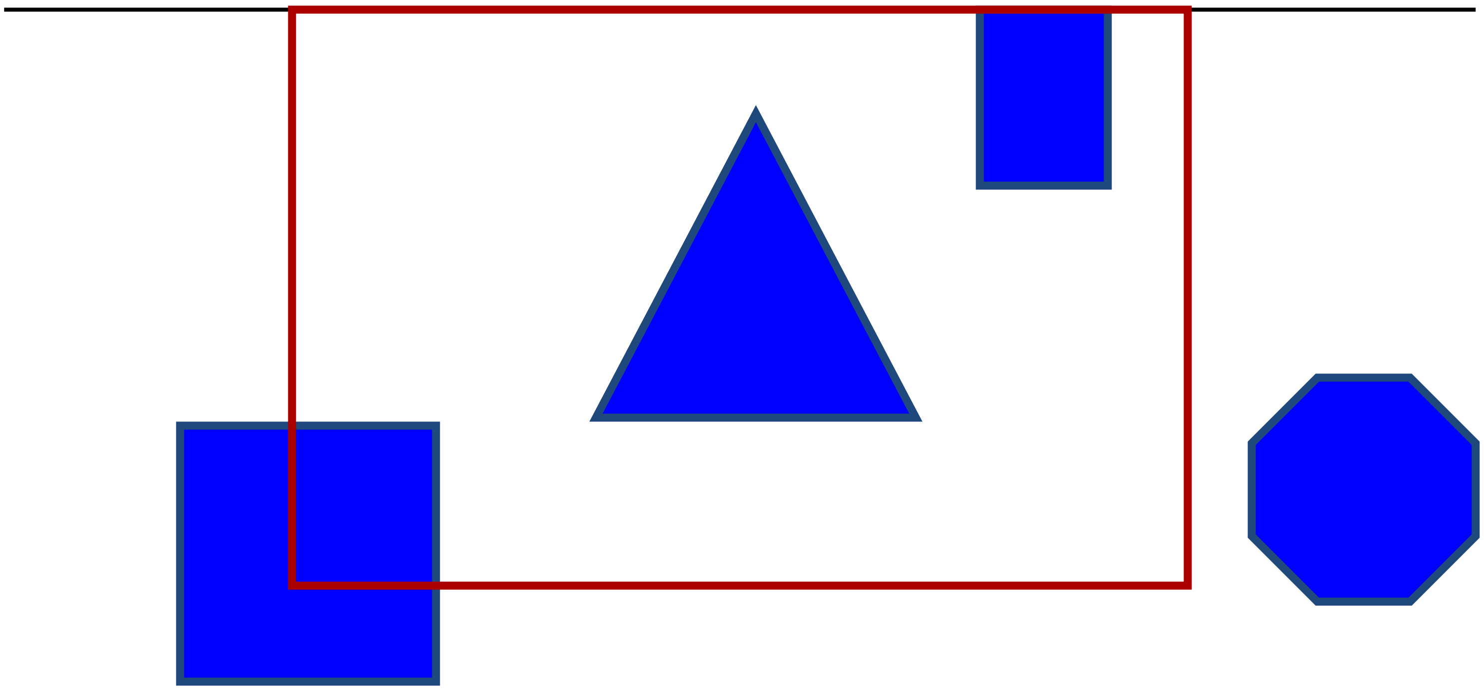
Sutherland-Hodgeman Polygon Clipping

- Clip to Each Window Boundary One at a Time



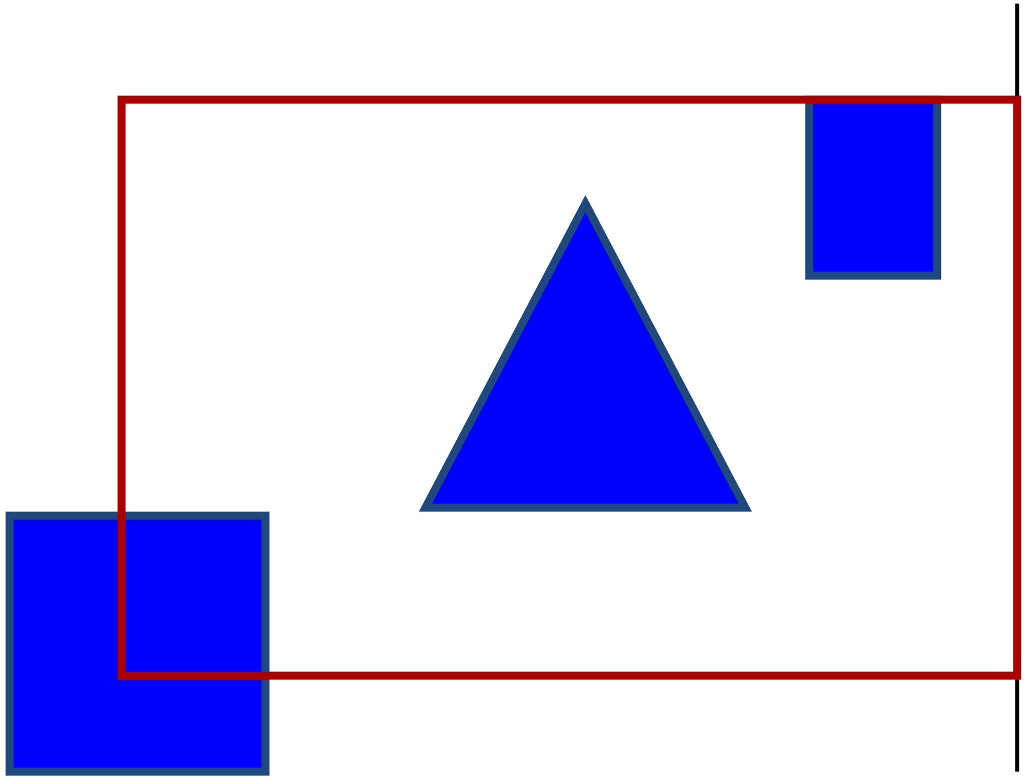
Sutherland-Hodgeman Polygon Clipping

- Clip to Each Window Boundary One at a Time



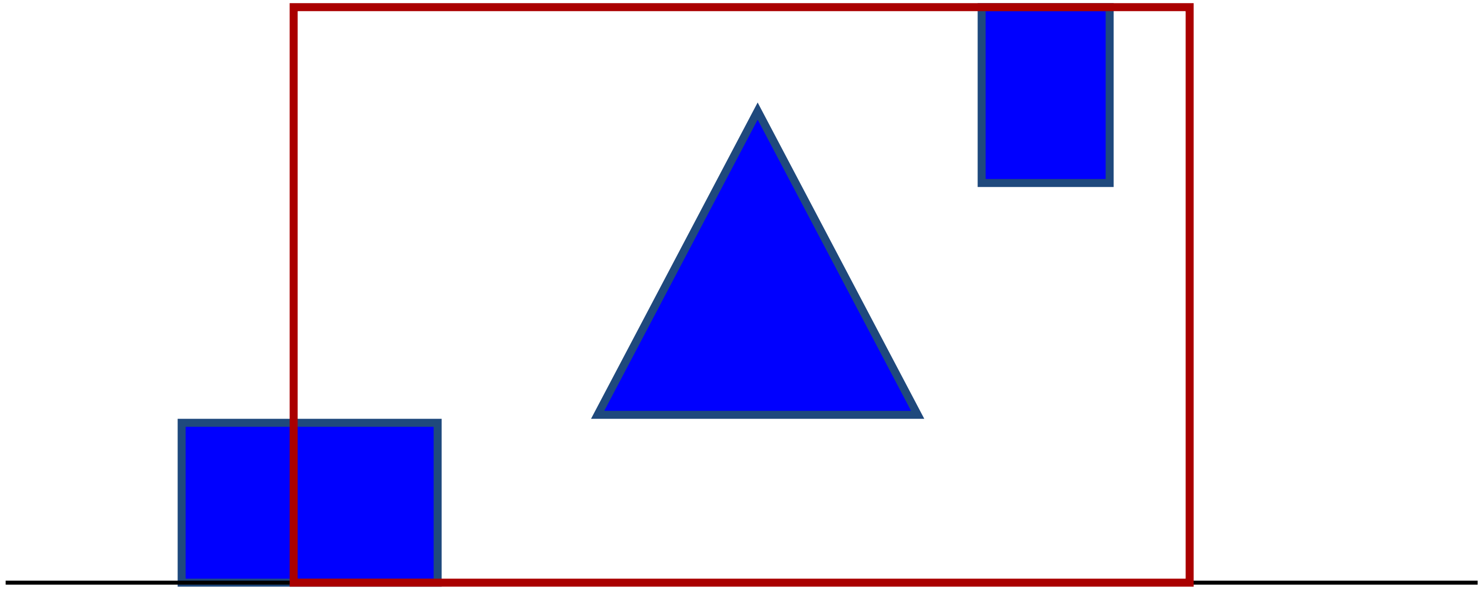
Sutherland-Hodgeman Polygon Clipping

- Clip to Each Window Boundary One at a Time



Sutherland-Hodgeman Polygon Clipping

- Clip to Each Window Boundary One at a Time



Sutherland-Hodgeman Polygon Clipping

- Clip to Each Window Boundary One at a Time