

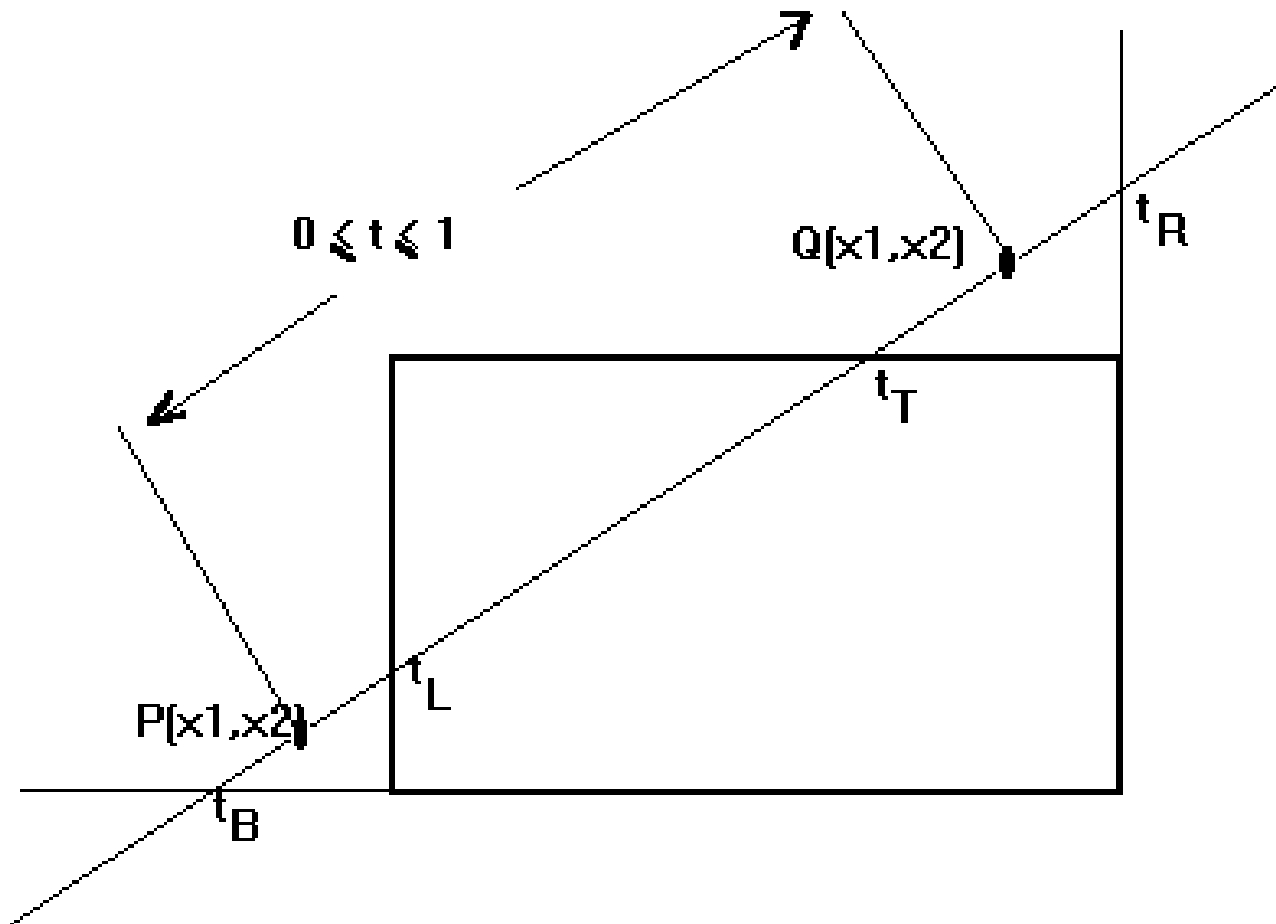
Introduction

Liang–Barsky algorithm uses the parametric equation of a line and inequalities describing the range of the clipping box to determine the intersections between the line and the clipping box. With these intersections it knows which portion of the line should be drawn. This algorithm is significantly more efficient than Cohen–Sutherland, but Cohen-Sutherland does trivial accepts and rejects much faster, so it should be considered instead if most of the lines you need to clip would be completely in or out of the clip window.

Parametric line clipping

Liang-Barsky Line Clipping

This algorithm uses the parametric equations for a line and solves four inequalities to find the range of the parameter for which the line is in the viewport.



Let $P(x_1, y_1)$, $Q(x_2, y_2)$ be the line which we want to study. The **parametric equation of the line segment** from gives x-values and y-values for every point in terms of a **parameter t** that ranges from 0 to 1. The equations are

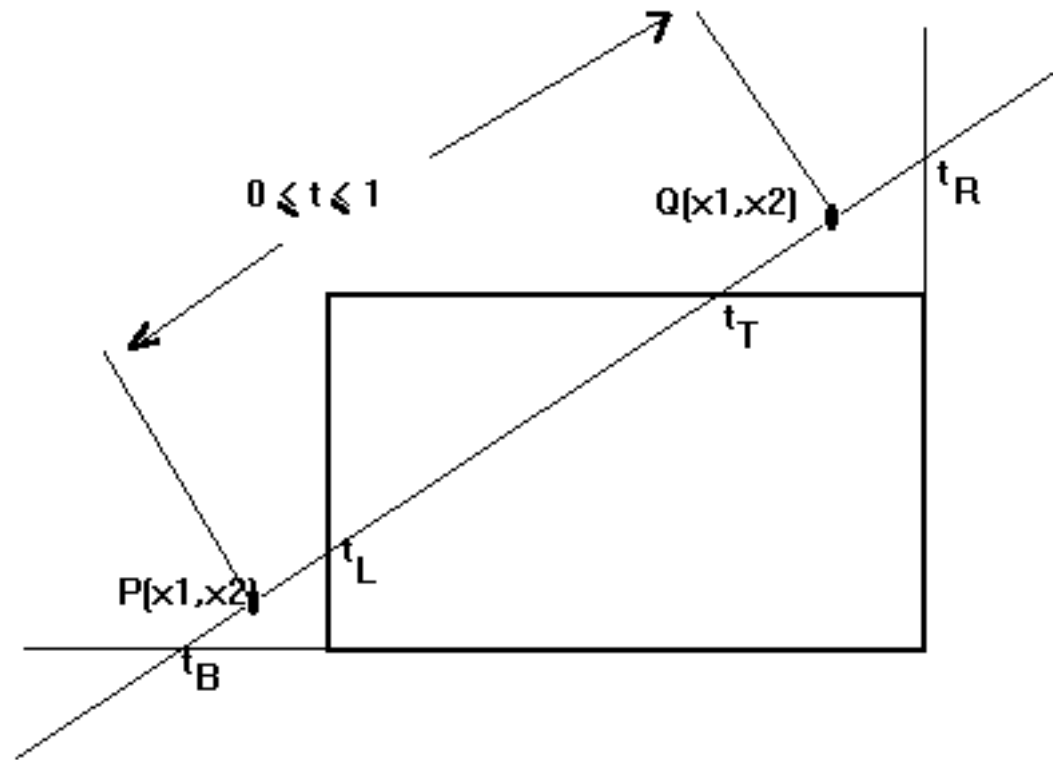
$$x = x_1 + (x_2 - x_1)*t = x_1 + dx*t \quad \text{and} \quad y = y_1 + (y_2 - y_1)*t = y_1 + dy*t$$

We can see that when $t = 0$, the point computed is $P(x_1, y_1)$; and when $t = 1$, the point computed is $Q(x_2, y_2)$.

Algorithm

1. Set $t_{\min} = 0$ and $t_{\max} = 1$
2. Calculate the values of t_L , t_R , t_T , and t_B (tvalues).
 - if $t < t_{\min}$ or $t > t_{\max}$ ignore it and go to the next edge
 - otherwise classify the tvalue as entering or exiting** value (using inner product to classify)
 - if t is entering value set $t_{\min} = t$; if t is exiting value set $t_{\max} = t$
3. If $t_{\min} < t_{\max}$ then **draw a line** from $(x1 + dx*t_{\min}, y1 + dy*t_{\min})$ to $(x1 + dx*t_{\max}, y1 + dy*t_{\max})$
4. If the line crosses over the window, you will see $(x1 + dx*t_{\min}, y1 + dy*t_{\min})$ and $(x1 + dx*t_{\max}, y1 + dy*t_{\max})$ are intersection*** between line and edge.

*Tvalue



Let $P(x_1, y_1)$, $Q(x_2, y_2)$ be the line that across window

From parametric equation $x = x_1 + (x_2 - x_1) * t$, $y = y_1 + (y_2 - y_1) * t$

Intersection between line and edge

From the **parametric equation** and the [t values](#), we can calculate the intersection between a line and an edge as follows

Intersection on left edge

$$\left(L, y_1 + \frac{(L - x_1)(y_2 - y_1)}{(x_2 - x_1)} \right)$$

Intersection on right edge

$$\left(R, y_1 + \frac{(R - x_1)(y_2 - y_1)}{(x_2 - x_1)} \right)$$

Intersection on top edge

$$\left(x_1 + \frac{(T - y_1)(x_2 - x_1)}{(y_2 - y_1)}, T \right)$$

Intersection on bottom edge

$$\left(x_1 + \frac{(B - y_1)(x_2 - x_1)}{(y_2 - y_1)}, B \right)$$

Application

In 2D graphics , if the user of an image editing program is modifying an image and has "zoomed in" the view to display only the top half of the image, there is no need for the program to spend any CPU time doing any of the calculations or memory moves needed to display the bottom half. By clipping the bottom half of the image and avoiding these calculations, the program runs faster.