Liang–Barsky Line Clipping

Faster line clippers have been developed that are based on analysis of the parametric equation of a line segment, which we can write in the form

$$\begin{aligned} x &= x_1 + u\Delta x \\ y &= y_1 + u\Delta y, \qquad 0 \le u \le 1 \end{aligned} \tag{6-9}$$

where $\Delta x = x_2 - x_1$ and $\Delta y = y_2 - y_1$. Using these parametric equations, Cyrus and Beck developed an algorithm that is generally more efficient than the Cohen–Sutherland algorithm. Later, Liang and Barsky independently devised an even faster parametric line-clipping algorithm. Following the Liang–Barsky approach, we first write the point-clipping conditions 6-5 in the parametric form:

$$\begin{aligned} xw_{\min} &\le x_1 + u\Delta x \le xw_{\max} \\ yw_{\min} &\le y_1 + u\Delta y \le yw_{\max} \end{aligned} \tag{6-10}$$

Each of these four inequalities can be expressed as

$$up_k \le q_k, \qquad k = 1, 2, 3, 4 \tag{6-11}$$

where parameters p and q are defined as

$$\begin{aligned} p_1 &= -\Delta x, & q_1 &= x_1 - xw_{\min} \\ p_2 &= \Delta x, & q_2 &= xw_{\max} - x_1 \\ p_3 &= -\Delta y, & q_3 &= y_1 - yw_{\min} \\ p_4 &= \Delta y, & q_4 &= yw_{\max} - y_1 \end{aligned} \tag{6-12}$$

Any line that is parallel to one of the clipping boundaries has $p_k = 0$ for the value of k corresponding to that boundary ($k = 1$, 2, 3, and 4 correspond to the left, right, bottom, and top boundaries, respectively). If, for that value of k, we also find $q_k < 0$, then the line is completely outside the boundary and can be eliminated from further consideration. If $q_k \geq 0$, the line is inside the parallel clipping boundary.

When $p_k < 0$, the infinite extension of the line proceeds from the outside to the inside of the infinite extension of this particular clipping boundary. If $p_k > 0$, the line proceeds from the inside to the outside. For a nonzero value of p_k, we can calculate the value of u that corresponds to the point where the infinitely extended line intersects the extension of boundary k as

$$u = \frac{q_k}{p_k} \tag{6-13}$$

For each line, we can calculate values for parameters u_1 and u_2 that define that part of the line that lies within the clip rectangle. The value of u_1 is determined by looking at the rectangle edges for which the line proceeds from the outside to the inside ($p < 0$). For these edges, we calculate $r_k = q_k/p_k$. The value of u_1 is taken as the largest of the set consisting of 0 and the various values of r. Conversely, the value of u_2 is determined by examining the boundaries for which the line proceeds from inside to outside ($p > 0$). A value of r_k is calculated for each of these boundaries, and the value of u_2 is the minimum of the set consisting of 1 and the calculated r values. If $u_1 > u_2$, the line is completely outside the clip window and it can be rejected. Otherwise, the endpoints of the clipped line are calculated from the two values of parameter u.

This algorithm is presented in the following procedure. Line intersection parameters are initialized to the values $u_1 = 0$ and $u_2 = 1$. For each clipping boundary, the appropriate values for p and q are calculated and used by the function *clipTest* to determine whether the line can be rejected or whether the intersection parameters are to be adjusted. When $p < 0$, the parameter r is used to update u_1; when $p > 0$, parameter r is used to update u_2. If updating u_1 or u_2 results in $u_1 > u_2$, we reject the line. Otherwise, we update the appropriate u parameter only if the new value results in a shortening of the line. When $p = 0$ and $q < 0$, we can discard the line since it is parallel to and outside of this boundary. If the line has not been rejected after all four values of p and q have been tested, the endpoints of the clipped line are determined from values of u_1 and u_2.

```
procedure clipLiangBarsky (winMin, winMax : dcPt2; n : integer;
```