Clipping Algorithms

Clipping algorithms are designed to efficiently identify the portions of a scene (in viewing coordinates) that lie inside a given viewport. They are useful because they

- excludes unwanted graphics from the screen;
- improves efficiency, as the computation dedicated to objects that appear off screen can be significantly reduced;
- can be used in other ways (modelling of rectangular aperatures, for example).

Two possible ways to apply clipping in the viewport transformation:

1. Apply clipping in the world coordinate system: ignore objects (e.g., vertices, line segments, and polygons) that lie outside of the window.
2. Apply clipping in the device coordinate system:ignore objects that lie outside of the viewport.

Clipping Algorithms

Most often clipping is defined with respect to a rectangular window; though the algorithms can be extended to different geometries.

- Point clipping
- Line clipping
  - Cohen-Sutherland
  - Liang-Barsky
  - Nicholl-Lee-Nicholl
- Polygon clipping
  - Sutherland-Hodgeman
  - Weiler-Atherton

Point Clipping

Let W denote a clip window with coordinates $(x_{\text{min}}, y_{\text{min}})$, $(x_{\text{min}}, y_{\text{max}})$, $(x_{\text{max}}, y_{\text{min}})$, $(x_{\text{max}}, y_{\text{max}})$, then a vertex (x, y) is displayed only if *all* four of the following "point clipping" inequalities are satisfied:

$$x_{\text{min}} \leq x \leq x_{\text{max}}, \quad \text{and,} \quad y_{\text{min}} \leq y \leq y_{\text{max}}.$$

- Can be applied in viewing or device coordinates.
- Very simple and efficient!
- Only works for vertices.

Line Clipping

In computer graphics, a the term “line” usually refers to a line segment.

Basic principles (assuming the clip window W is convex):

- If both endpoints of a line segment fall within W, then display the line segment.
- If both endpoints of a line segment fall outside of W because they violate the same point clipping inequality, then do not display the line segment.
- If one endpoint falls within W and another falls outside, then part of the line segment is displayed.
- If both endpoints fall outside W, but do not violate a common point clipping inequality, then part of the line may be visible.

Intersection Test

Let (x_1, y_1) and (x_2, y_2) denote two endpoints of a given line segment S. Then the line segment can be described parametrically as

$$\begin{aligned} x &= x_1 + u(x_2 - x_1) \\ y &= y_1 + u(y_2 - y_1) \end{aligned}$$

where u varies over the interval $0 \leq u \leq 1$.

How can we use the above to determine if S intersects the boundary of W?

Cyrus-Beck and Liang-Barsky Line Clipping

Let $\Delta x = x_2 - x_1$ and $\Delta y = y_2 - y_1$. The parametrized segments are thus

$$\begin{aligned} x &= x_1 + u\Delta x \\ y &= y_1 + u\Delta y \end{aligned}$$

(for $0 \leq u \leq 1$). Whence the point-clipping inequalities become:

$$\begin{aligned} x_{\text{min}} &\leq x_1 + u\Delta x \leq x_{\text{max}} \\ y_{\text{min}} &\leq y_1 + u\Delta y \leq y_{\text{max}} \end{aligned}$$

These four inequalities can be expressed as

$$up_k \leq q_k$$

for $k = 1, 2, 3, 4$, where

$$\begin{aligned} p_1 &= -\Delta x, & q_1 &= x_1 - x_{\text{min}} \\ p_2 &= \Delta x, & q_2 &= x_{\text{max}} - x_1 \\ p_3 &= -\Delta y, & q_3 &= y_1 - y_{\text{min}} \\ p_4 &= \Delta y, & q_4 &= y_{\text{max}} - y_1 \end{aligned}$$

Cyrus-Beck and Liang-Barsky Line Clipping (cont.)

$$up_k \leq q_k$$

$$\begin{aligned} p_1 &= -\Delta x, & q_1 &= x_1 - x_{\text{min}} \\ p_2 &= \Delta x, & q_2 &= x_{\text{max}} - x_1 \\ p_3 &= -\Delta y, & q_3 &= y_1 - y_{\text{min}} \\ p_4 &= \Delta y, & q_4 &= y_{\text{max}} - y_1 \end{aligned}$$

Observations:

- If $p_k = 0$, then the line is parallel to a boundary; if $q_k < 0$ for the same k, then the line can be discarded. If $q_k \geq 0$ then the corresponding point-clipping inequality is satisfied.
- If $p_k < 0$ the segment potentially *enters* the clip rectangle across the k-th boundary (PE).
- If $p_k > 0$ the segment potentially *leaves* the clip rectangle across the k-th boundary (PL).

Cyrus-Beck and Liang-Barsky Line Clipping (cont.)

Let

$$K_{\mathrm{PE}} = \{k : p_k < 0\}$$

denote the set of indices of boundaries across which the line *potentially enters* the clip rectangle, and let

$$K_{\mathrm{PL}} = \{k : p_k > 0\}$$

denote the set of indices of boundaries across which the line *potentially leaves*. Let

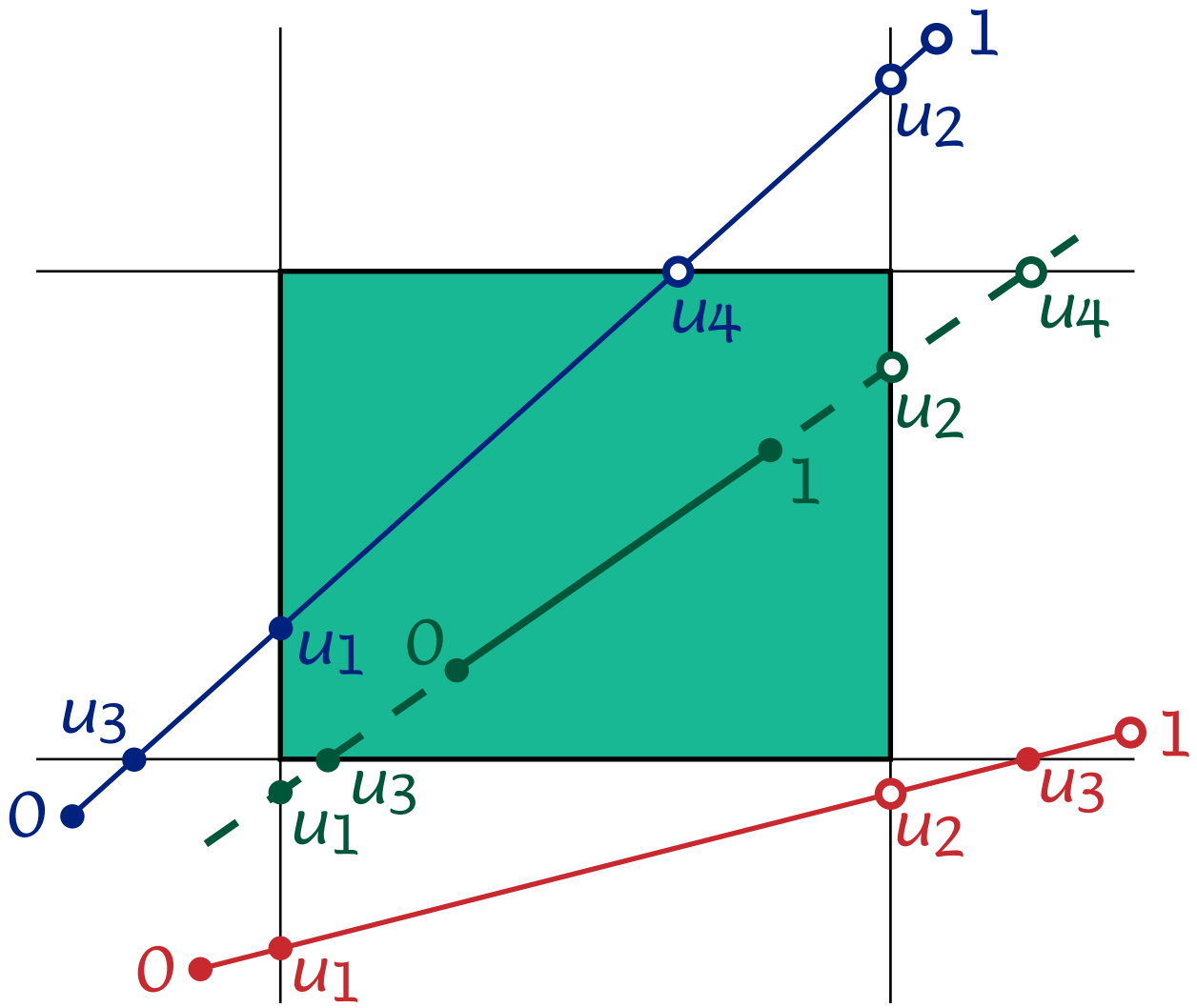
$$u_k = \frac{q_k}{p_k}$$

denote the parameter value at the k-th boundary crossing. Then the visible portion of the segment is in the parameter range $u_{\min} \le u \le u_{\max}$, where,

$$u_{\min} = \max_{k \in K_{\mathrm{PE}}} [0, u_k], \qquad u_{\max} = \min_{k \in K_{\mathrm{PL}}} [1, u_k]$$

If $u_{\min} > u_{\max}$ then the entire segment is clipped.

Example



Liang-Barsky Line Clipping — The Code

```
int clipTest (float p, float q, float * u1, float * u2) {
   float r;
   int retVal = TRUE;

   if (p < 0.0) {
      r = q / p;
      if (r > *u2)
         retVal = FALSE;
      else if (r > *u1)
         *u1 = r;
   } else if (p > 0.0) {
      r = q / p;
      if (r < *u1)
         retVal = FALSE;
      else if (r < *u2)
         *u2 = r;
   } else if (q < 0.0)
        /* p = 0, so line is parallel to this clipping edge */
        /* Line is outside clipping edge */
         retVal = FALSE;

   return (retVal);
}
```

Liang-Barsky Line Clipping — The Code

```
void clipLine (dcPt winMin, dcPt winMax, wcPt2 p1, wcPt2 p2) {
   float u1 = 0.0, u2 = 1.0, dx = p2.x - p1.x, dy;

   if (clipTest (-dx, p1.x - winMin.x, &u1, &u2))
      if (clipTest (dx, winMax.x - p1.x, &u1, &u2)) {
         dy = p2.y - p1.y;
         if (clipTest (-dy, p1.y - winMin.y, &u1, &u2))
            if (clipTest (dy, winMax.y - p1.y, &u1, &u2)) {
               if (u2 < 1.0) {
                  p2.x = p1.x + u2 * dx;
                  p2.y = p1.y + u2 * dy;
               }
               if (u1 > 0.0) {
                  p1.x += u1 * dx;
                  p1.y += u1 * dy;
               }
               lineDDA (ROUND(p1.x), ROUND(p1.y), ROUND(p2.x), ROUND(p2.y));
            }
      }
}
```

Liang-Barsky Line Clipping – The Code

Bibliographical Note: The preceding code was extracted from *Computer Graphics: C Version*, Second Edition, by Donald Hearn and M. Pauline Baker, Prentice-Hall, 1994, pp. 231–232. A similar pair of subroutines appears in *Computer Graphics: Principles and Practice*, Second Edition in C, by James D. Foley, Andries van Dam, Stephen K. Feiner, and John F. Hughes, Addison-Wesley, 1996, pp. 122–123.

Nicholl-Lee-Nicholl Line Clipping

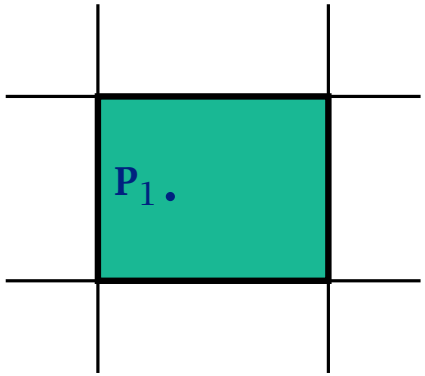
In general the line that supports a given line segement $\overline{\mathbf{P}_1\mathbf{P}_2}$ intersects all four window boundaries; at most, two of these intersections are relevant.

Principle: Assuming that most of the computational overhead involves finding these intersections, one should avoid computing irrelevant intersections.

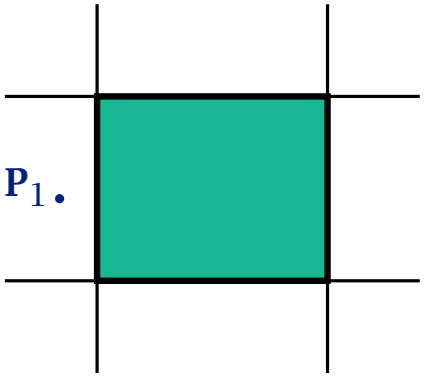
Symmetry reduces the problem to three basic cases. Slope calculations can be used to identify which intersections should be computed. Some intermediate values can be reused.

Nicholl-Lee-Nicholl Line Clipping

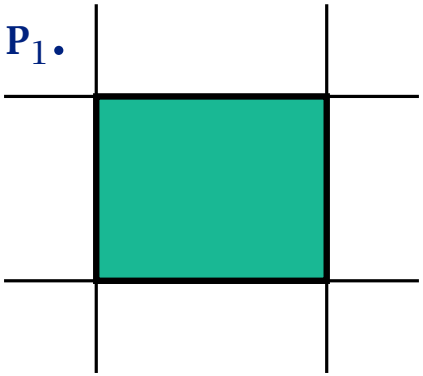
Using symmetry, a given endpoint $\mathbf{P}_1 = (x_1, y_1)$ falls in one of three distinct regions:



P_1 in the clip window



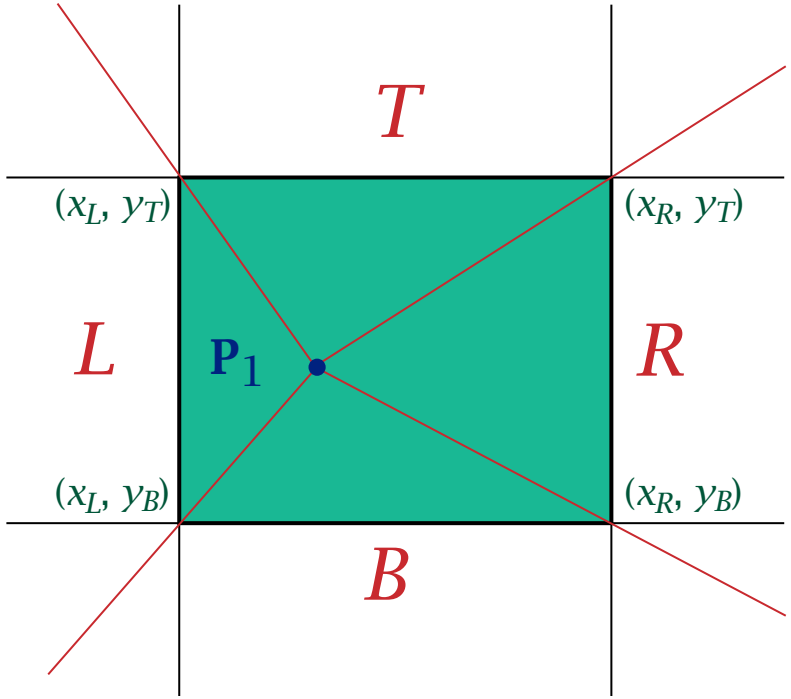
P_1 in an edge region



P_1 in a corner region

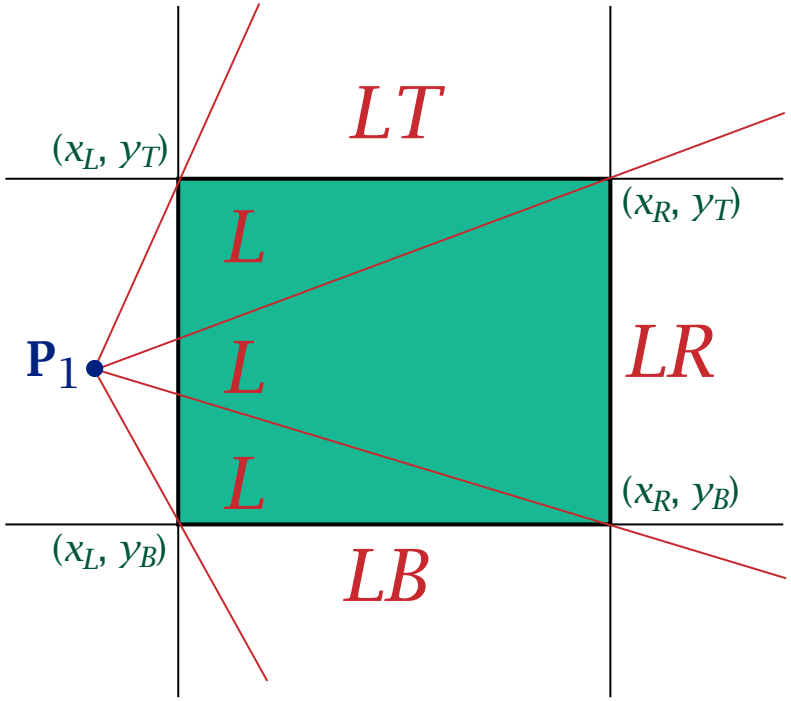
Nicholl-Lee-Nicholl Line Clipping (Case 1)

If $\mathbf{P}_1 = (x_1, y_1)$ falls in the clip rectangle, the ray from $\mathbf{P}_1$ through $\mathbf{P}_2$ will cross side L, B, R, or T.



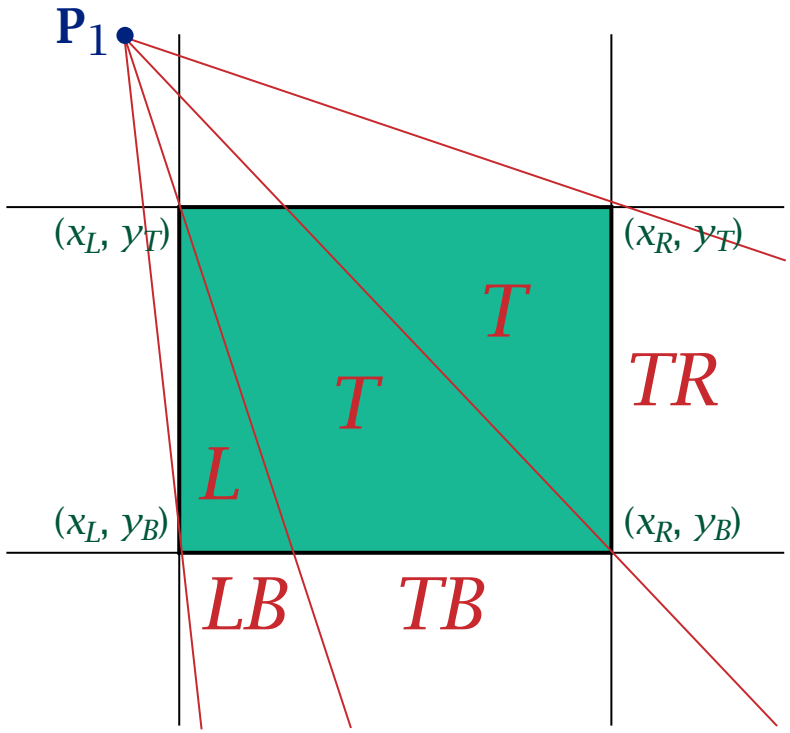
Nicholl-Lee-Nicholl Line Clipping (Case 2)

If $\mathbf{P}_1 = (x_1, y_1)$ falls in an outside edge region (shown as *left* WLOG), then part of the line may be visible if the ray from $\mathbf{P}_1$ through $\mathbf{P}_2$ enters region LB, LR, or LT.



Nicholl-Lee-Nicholl Line Clipping (Case 3A)

If $\mathbf{P}_1 = (x_1, y_1)$ falls in a corner region (shown as *upper left* WLOG), then two subcases are considered: If the ray from $\mathbf{P}_1$ through (x_L, y_T) intersects the bottom edge *before* the right edge, then part of the segment may be visible if the ray enters region LB, TB, or TR.



Nicholl-Lee-Nicholl Line Clipping (Case 3B)

If the ray from $\mathbf{P}_1 = (x_1, y_1)$ through (x_L, y_T) intersects the bottom edge *after* the right edge, then part of the segment may be visible if the ray enters region LB, LR, or TR.