三、Liang-Barsky算法

在Cohen-Sutherland算法提出后，梁友栋和Barsky又针对标准矩形窗口提出了更快的Liang-Barsky直线段裁剪算法。

上世纪80年代，梁友栋先生提出了著名的Liang-Barsky算法，至今仍是计算机图形学中最经典的算法之一，也是写进国内外主流《计算机图形学》教科书里的唯一一个以中国人命名的算法。

You-Dong Liang; Barsky, B.A. A new concept and method for line clipping, ACM Transactions on Graphics, Vol.3 1-22,1984

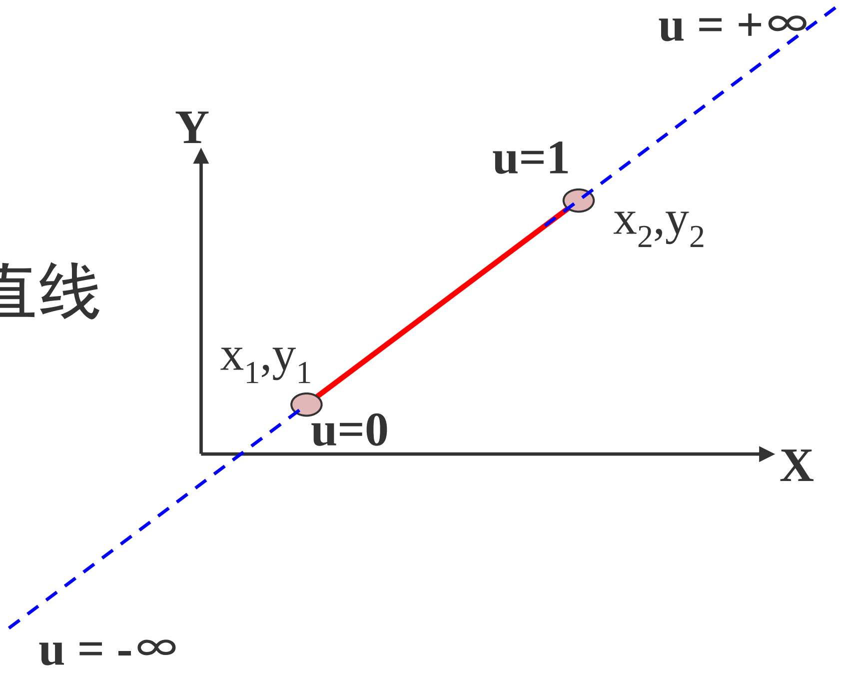
A New Concept and Method for Line Clipping

YOU-DONG LIANG and BRIAN A. BARSKY
University of California, Berkeley

A new concept and method for line clipping is developed that describes clipping in an exact and mathematical form. The basic ideas form the foundation for a family of algorithms for two-dimensional, three-dimensional, and four-dimensional (homogeneous coordinates) line clipping. The

梁算法的主要思想：

（1）用参数方程表示一条直线



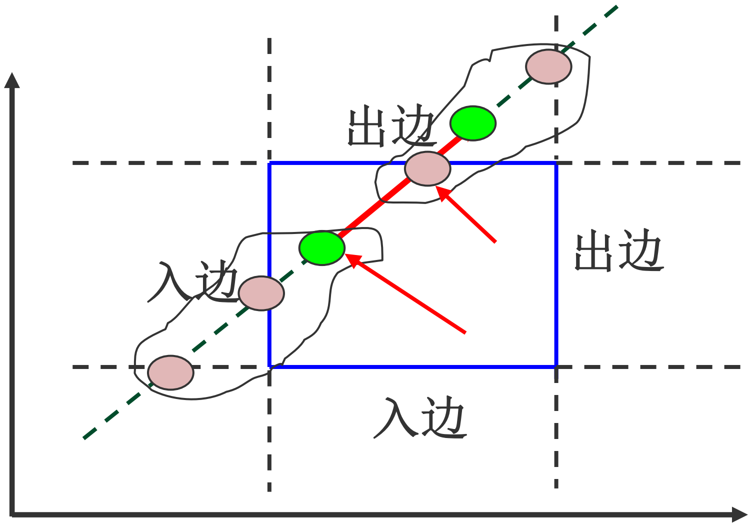
$$x = x_1 + u \cdot (x_2 - x_1) = \underline{x_1 + \Delta x \cdot u}$$
$$y = y_1 + u \cdot (y_2 - y_1) = \underline{y_1 + \Delta y \cdot u}$$

$$0 \le u \le 1$$

梁算法的主要思想：

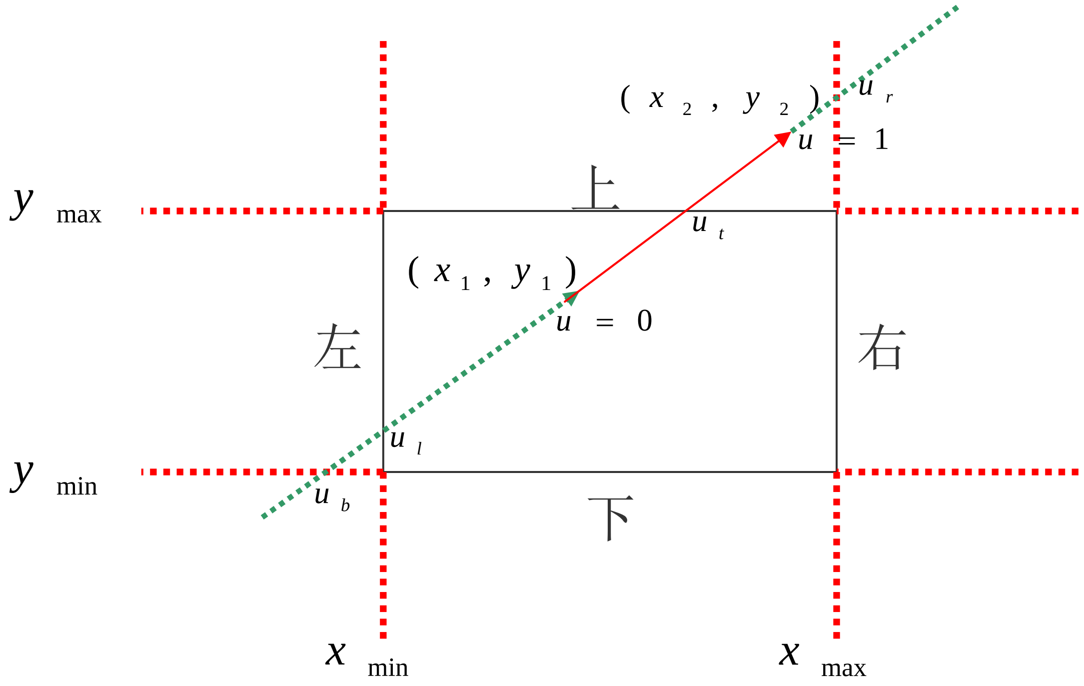
（2）把被裁剪的红色直线段看成是一条有方向的线段，把窗口的四条边分成两类：

入边和出边



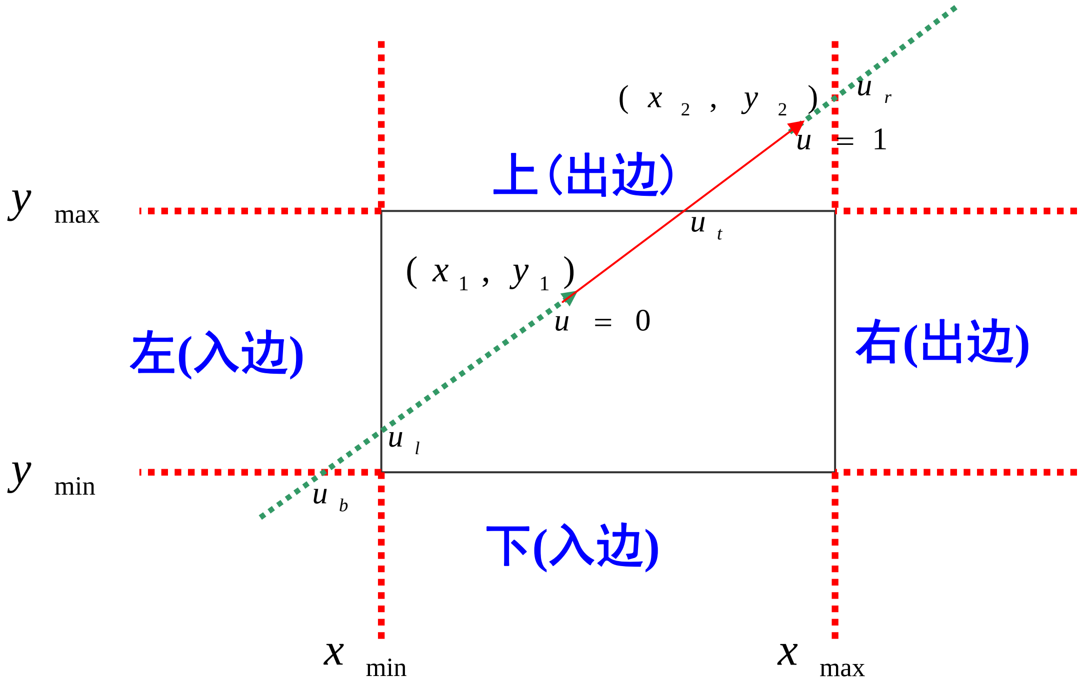
裁剪结果的线段起点是直线和两条入边的交点以及始端点三个点里最前面的一个点，即参数u最大的那个点；

裁剪线段的终点是和两条出边的交点以及端点最后面的一个点，取参数u最小的那个点。



值得注意的是，当u从-∞到+∞遍历直线时，首先对裁剪窗口的两条边界直线（下边和左边）从外面向里面移动，再对裁剪窗口两条边界直线（上边和右边）从里面向外面移动。

如果用u_1，u_2分别表示线段（$u_1 \leqslant u_2$）可见部分的开始和结束



$$\underline{u_1 = \max(0, u_l, u_b)}$$

$$\underline{u_2 = \min(1, u_t, u_r)}$$

这就是梁先生的重大发现！

Liang-Barsky算法的基本出发点是直线的参数方程

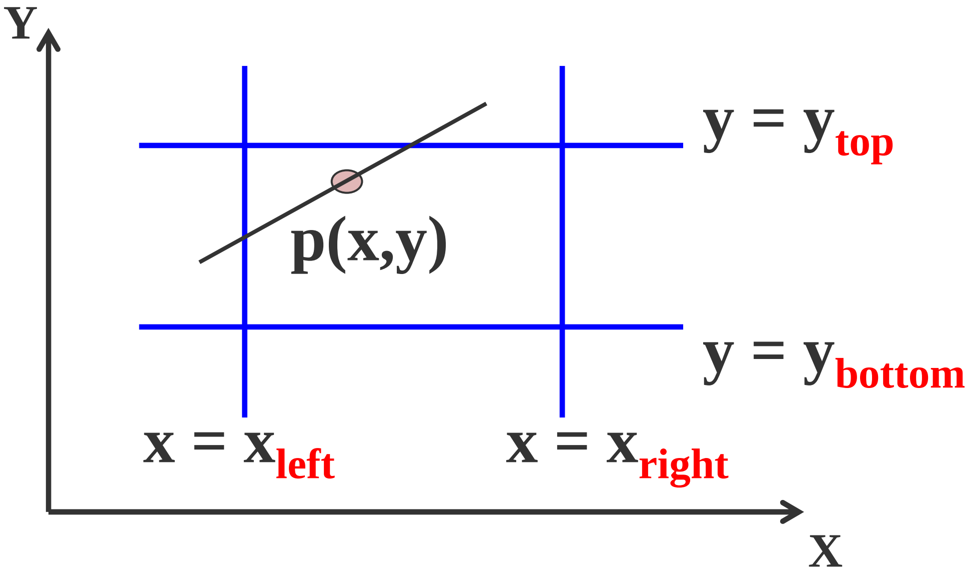
$$x = x_1 + u \cdot (x_2 - x_1)$$

$$y = y_1 + u \cdot (y_2 - y_1)$$

$$0 \le u \le 1$$

$$x_{left} \le x_1 + u \cdot \Delta x \le x_{right}$$

$$y_{bottom} \le y_1 + u \cdot \Delta y \le y_{top}$$



$$x_{left} \leq x_1 + u \cdot \Delta x \leq x_{right}$$

$$y_{bottom} \leq y_1 + u \cdot \Delta y \leq y_{top}$$

$$u \cdot (-\Delta x) \leq x_1 - x_{left}$$

$$u \cdot \Delta x \leq x_{right} - x_1$$

$$u \cdot (-\Delta y) \leq y_1 - y_{bottom}$$

$$u \cdot \Delta y \leq y_{top} - y_1$$

$$u \cdot (-\Delta x) \le x_1 - x_{left}$$

$$u \cdot \Delta x \le x_{right} - x_1$$

$$u \cdot (-\Delta y) \le y_1 - y_{bottom}$$

$$u \cdot \Delta y \le y_{top} - y_1$$

令：

$$p_1 = -\Delta x \qquad q_1 = x_1 - x_{left}$$

$$p_2 = \Delta x \qquad q_2 = x_{right} - x_1$$

$$p_3 = -\Delta y \qquad q_3 = y_1 - y_{bottom}$$

$$p_4 = \Delta y \qquad q_4 = y_{top} - y_1$$

令：

$$p_1 = -\Delta x \qquad q_1 = x_1 - x_{left}$$
$$p_2 = \Delta x \qquad q_2 = x_{right} - x_1$$
$$p_3 = -\Delta y \qquad q_3 = y_1 - y_{bottom}$$
$$p_4 = \Delta y \qquad q_4 = y_{top} - y_1$$

于是有：$u \cdot p_k \le q_k$ 其中，$k = 1, 2, 3, 4$

$$p_1 = -\Delta x \qquad q_1 = x_1 - x_{left} \qquad p_2 = \Delta x \qquad q_2 = x_{right} - x_1$$
$$p_3 = -\Delta y \qquad q_3 = y_1 - y_{bottom} \qquad p_4 = \Delta y \qquad q_4 = y_{top} - y_1$$

入边：左边和下边　　　　出边：右边和上边

Liang-Barsky算法的基本出发点是直线的参数方程

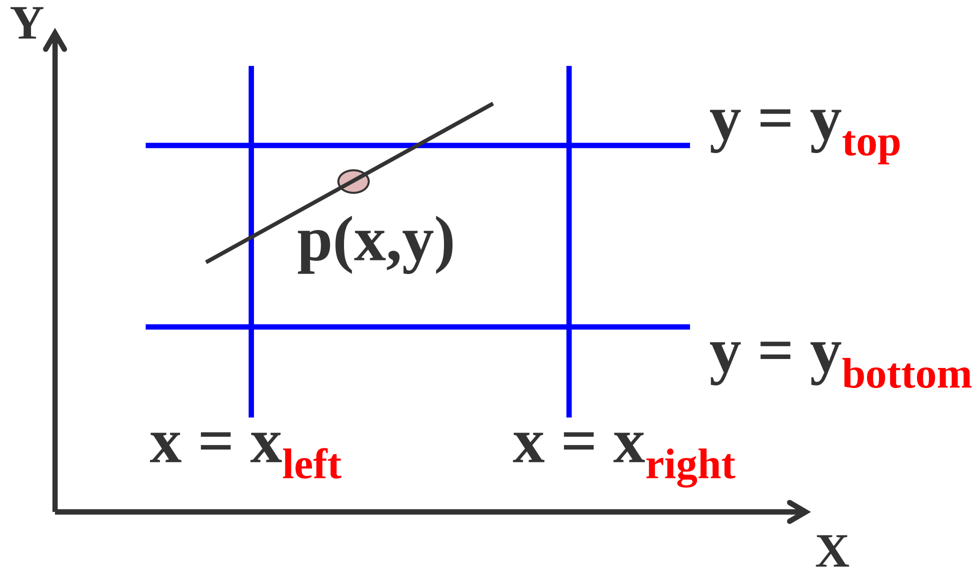
$$x = x_1 + u \cdot (x_2 - x_1)$$

$$y = y_1 + u \cdot (y_2 - y_1)$$

$$0 \le u \le 1$$

$$x_{left} \le x_1 + u \cdot \Delta x \le x_{right}$$

$$y_{bottom} \le y_1 + u \cdot \Delta y \le y_{top}$$



令：

$$p_1 = -\Delta x \qquad q_1 = x_1 - x_{left}$$
$$p_2 = \Delta x \qquad q_2 = x_{right} - x_1$$
$$p_3 = -\Delta y \qquad q_3 = y_1 - y_{bottom}$$
$$p_4 = \Delta y \qquad q_4 = y_{top} - y_1$$

于是有：$u \cdot p_k \leq q_k$ 其中，k = 1，2，3，4

$$p_1 = -\Delta x \qquad q_1 = x_1 - x_{left} \qquad p_2 = \Delta x \qquad q_2 = x_{right} - x_1$$
$$p_3 = -\Delta y \qquad q_3 = y_1 - y_{bottom} \qquad p_4 = \Delta y \qquad q_4 = y_{top} - y_1$$

入边：左边和下边

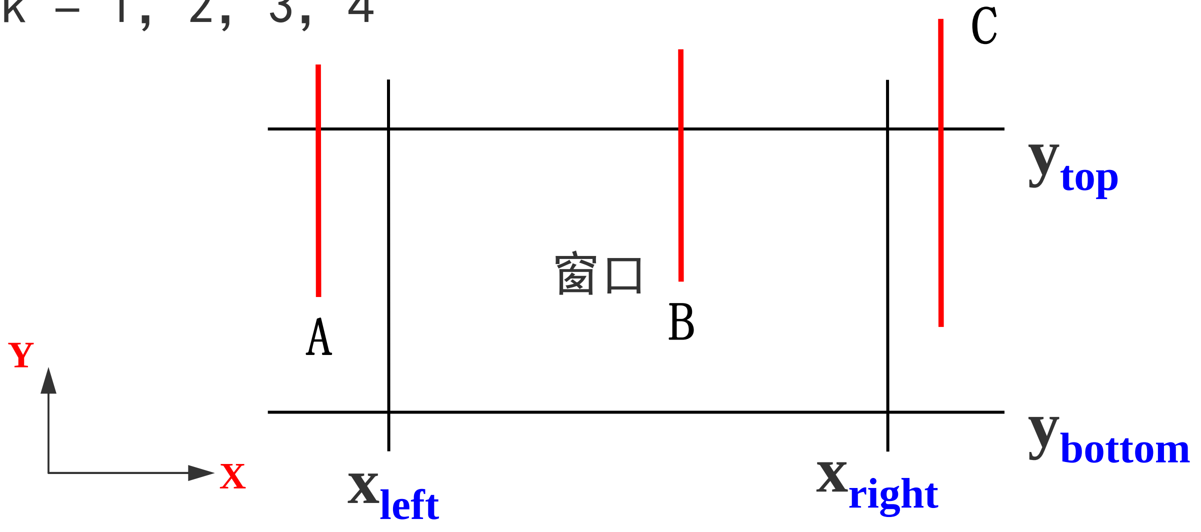
出边：右边和上边

$$p_1 = -\Delta x \qquad q_1 = x_1 - x_{left}$$
$$p_2 = \Delta x \qquad q_2 = x_{right} - x_1$$
$$p_3 = -\Delta y \qquad q_3 = y_1 - y_{bottom}$$
$$p_4 = \Delta y \qquad q_4 = y_{top} - y_1$$

$u \cdot p_k \leq q_k$　其中，k = 1，2，3，4

（1）分析P_k=0的情况

若P_1= P_2 = 0



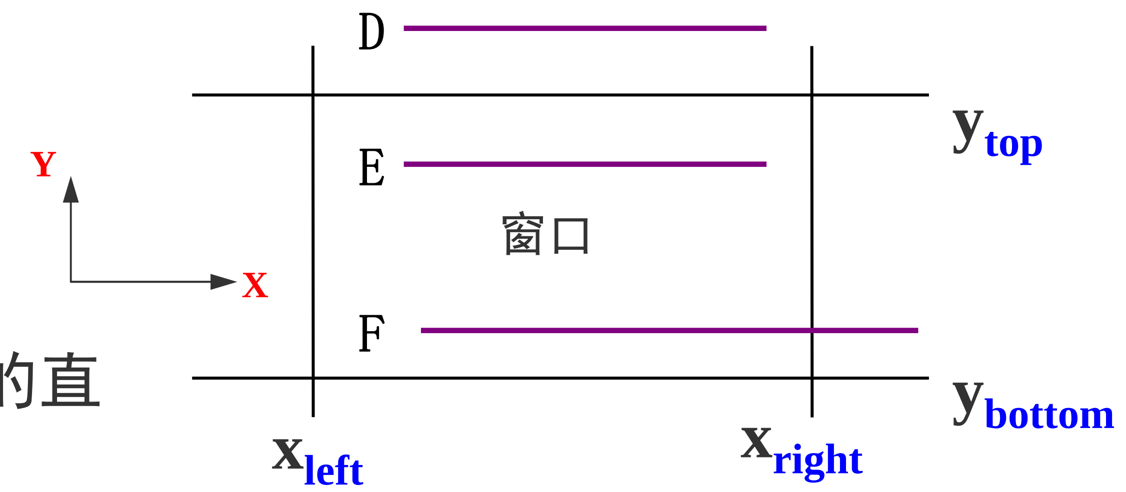
$$p_1 = -\Delta x \qquad q_1 = x_1 - x_{left}$$
$$p_2 = \Delta x \qquad q_2 = x_{right} - x_1$$
$$p_3 = -\Delta y \qquad q_3 = y_1 - y_{bottom}$$
$$p_4 = \Delta y \qquad q_4 = y_{top} - y_1$$

$u \cdot p_k \le q_k$　其中，k = 1，2，3，4

（1）分析P_k=0的情况

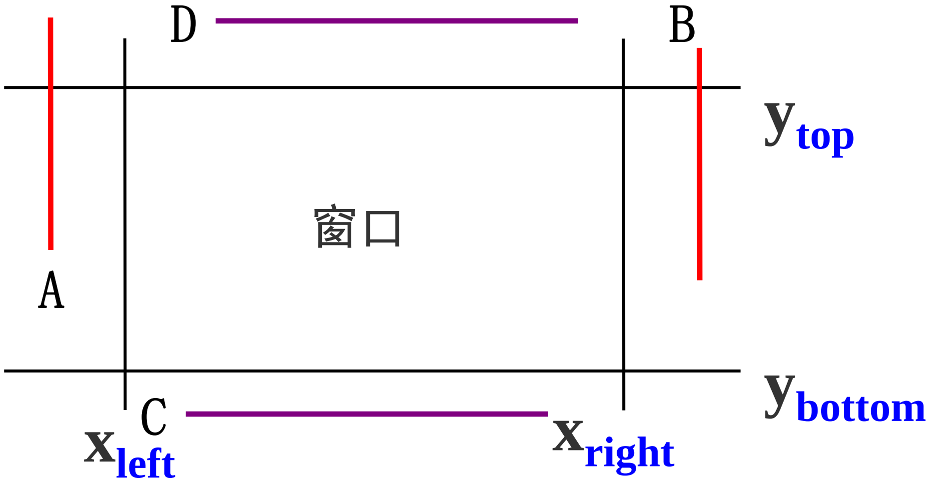
若P_3=P_4=0

任何平行于窗口某边界的直线，其p_k=0



$$p_1 = -\Delta x \qquad q_1 = x_1 - x_{left}$$
$$p_2 = \Delta x \qquad q_2 = x_{right} - x_1$$
$$p_3 = -\Delta y \qquad q_3 = y_1 - y_{bottom}$$
$$p_4 = \Delta y \qquad q_4 = y_{top} - y_1$$

$u \cdot p_k \le q_k$ 其中，k = 1，2，3，4



（1）分析P_k=0的情况

如果还满足q_k<0

则线段完全在边界外，应舍弃该线段

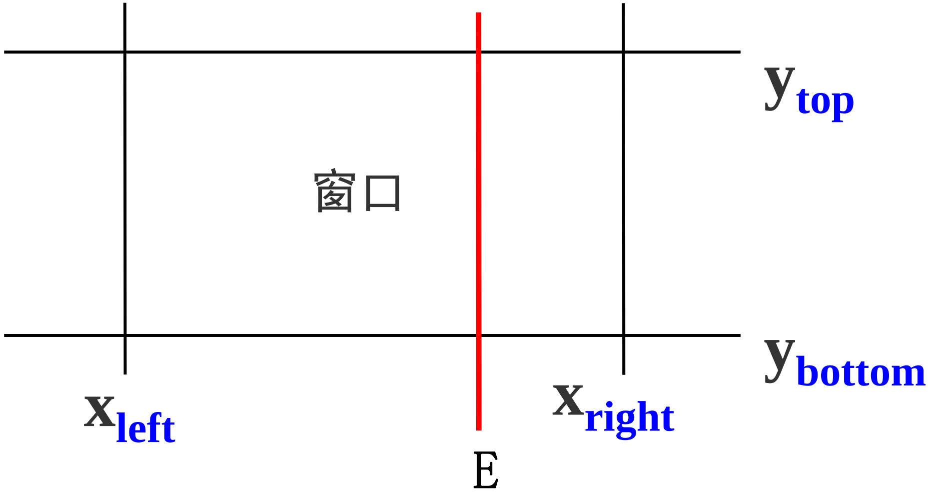
$$p_1 = -\Delta x \qquad q_1 = x_1 - x_{left}$$
$$p_2 = \Delta x \qquad q_2 = x_{right} - x_1$$
$$p_3 = -\Delta y \qquad q_3 = y_1 - y_{bottom}$$
$$p_4 = \Delta y \qquad q_4 = y_{top} - y_1$$

$u \cdot p_k \le q_k$ 其中，k = 1，2，3，4

（1）分析P_k=0的情况

如果$q_k \geqslant 0$

则进一步判断



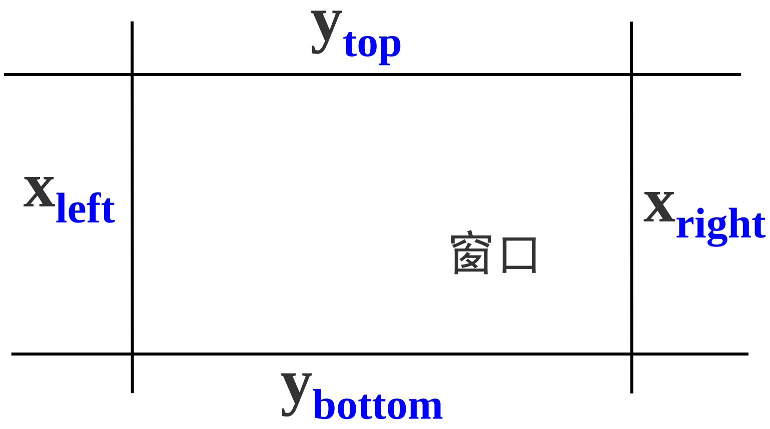
$$p_1 = -\Delta x \qquad q_1 = x_1 - x_{left}$$
$$p_2 = \Delta x \qquad q_2 = x_{right} - x_1$$
$$p_3 = -\Delta y \qquad q_3 = y_1 - y_{bottom}$$
$$p_4 = \Delta y \qquad q_4 = y_{top} - y_1$$

$u \cdot p_k \leq q_k$ 其中，k = 1，2，3，4

(2) 当$p_k \neq 0$时：

当$p_k < 0$时

线段从裁剪边界延长线的外部延伸到内部，是入边交点



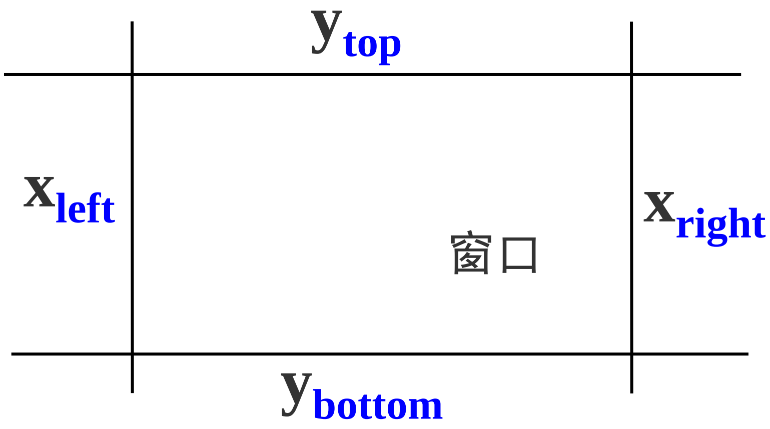
$$p_1 = -\Delta x \qquad q_1 = x_1 - x_{left}$$
$$p_2 = \Delta x \qquad q_2 = x_{right} - x_1$$
$$p_3 = -\Delta y \qquad q_3 = y_1 - y_{bottom}$$
$$p_4 = \Delta y \qquad q_4 = y_{top} - y_1$$

$u \cdot p_k \leq q_k$ 其中，k = 1，2，3，4

（2）当pk≠0时：

当p_k ＞ 0时

线段从裁剪边界延长线的内部延伸到外部，是出边交点



线段和窗口边界一共有四个交点，根据p_k的符号，就知道哪两个是入交点，哪两个是出交点

当$p_k < 0$时：对应入边交点

当$p_k > 0$时：对应出边交点

一共四个u值，再加上u=0、u=1两个端点值，总共六个值

把$p_k<0$的两个u值和0比较去找最大的，把$p_k>0$的两个u值和1比较去找最小的，这样就得到两个端点的参数值

$$u_k = \frac{q_k}{p_k} \qquad (p_k \neq 0, k = 1,2,3,4)$$

u_k是窗口边界及其延长线的交点的对应参数值

分别计算u_{max}和u_{min}：

$$u_{max} = \max(0, u_k|_{pk<0}, u_k|_{pk<0})$$

$$u_{min} = \min(u_k|_{pk>0}, u_k|_{pk>0}, 1)$$

注意：$p_k < 0$，代表入边；$p_k > 0$代表出边

若$u_{max}>u_{min}$，则直线段在窗口外，删除该直线

若$u_{max} \leqslant u_{min}$，将$u_{max}$和$u_{min}$代回直线参数方程，即求出直线与窗口的两实交点坐标。

$$x = x_1 + u \cdot (x_2 - x_1)$$
$$y = y_1 + u \cdot (y_2 - y_1)$$

注意：因为对于实交点$0 \leqslant u \leqslant 1$，因此$u_{max}$不能小于0，$u_{min}$不能大于1

下面写出Liang-Barsky裁剪算法步骤：

（1）输入直线段的两端点坐标（x_1, y_1）、（x_2, y_2），以及窗口的四条边界坐标：wxl、wxr、wyb和wyt

（2）若△X=0，则$p_1=p_2=0$，此时进一步判断是否满足$q_1<0$或$q_2<0$，若满足，则该直线段不在窗口内，算法转（7）-结束。否则，满足$q_1 \geq 0$且$q_2 \geq 0$，则进一步计算u_{max}和u_{min}：

$$u_{\max} = \max(0, \; u_k|_{pk<0})$$

$$u_{\min} = \min(u_k|_{pk>0}, 1)$$

其中，$u_k = \dfrac{q_k}{p_k} \quad (p_k \neq 0, k = 3,4)$。算法转（5）

（3）若△y=0，则$p_3=p_4=0$, 此时进一步判断是否满足$q_3<0$或$q_4<0$，若满足，则该直线段不在窗口内，算法转（7）。否则，满足$q_3 \geqslant 0$且$q_4 \geqslant 0$，则进一步计算u_{max}和u_{min}：

$$u_{\max} = \max(0, \left. u_k \right|_{pk<0})$$
$$u_{\min} = \min(\left. u_k \right|_{pk>0}, 1)$$

其中，$u_k = \dfrac{q_k}{p_k} \quad (p_k \neq 0, k = 3,4)$ 。算法转（5）

（4）若上述两条均不满足，则有$p_k \neq 0$（k=1, 2, 3, 4）， 此时计算u_{max}和u_{min}：

$$u_{max} = \max（0, u_k|_{pk<0}, u_k|_{pk<0}）$$

$$u_{min} = \min（u_k|_{pk>0}, u_k|_{pk>0}, 1）$$

其中，$u_k = \dfrac{q_k}{p_k} \quad (p_k \neq 0, k = 1,2,3,4)$

（5）求得u_{max}和u_{min}后，进行判断：若$u_{max}>u_{min}$，则直线段在窗口外，算法转（7）。若$u_{max}\leqslant u_{min}$，利用直线的参数方程：

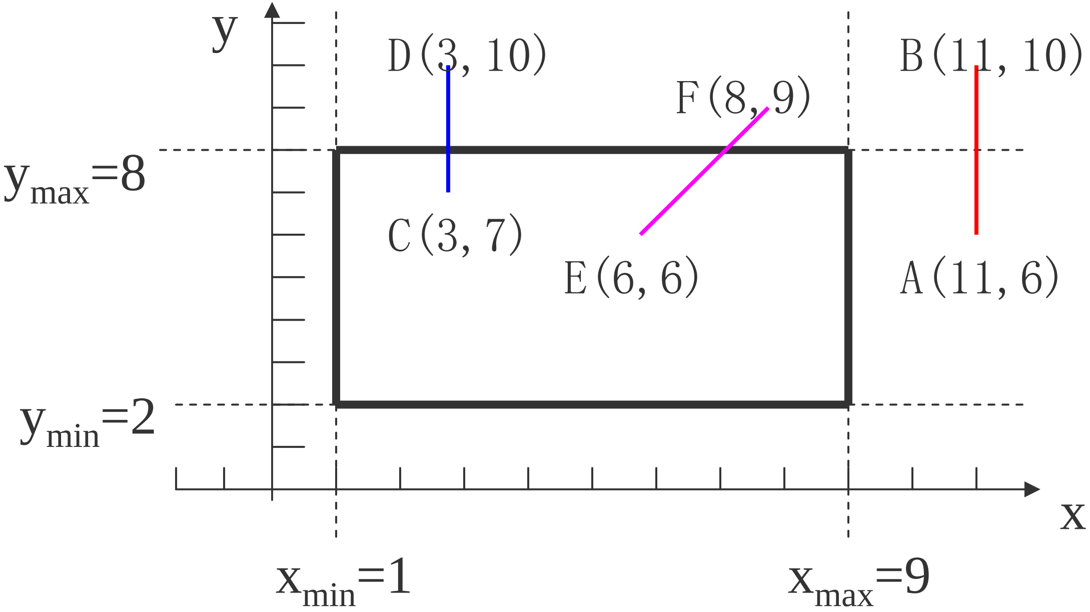
$$x = x_1 + u \cdot (x_2 - x_1)$$

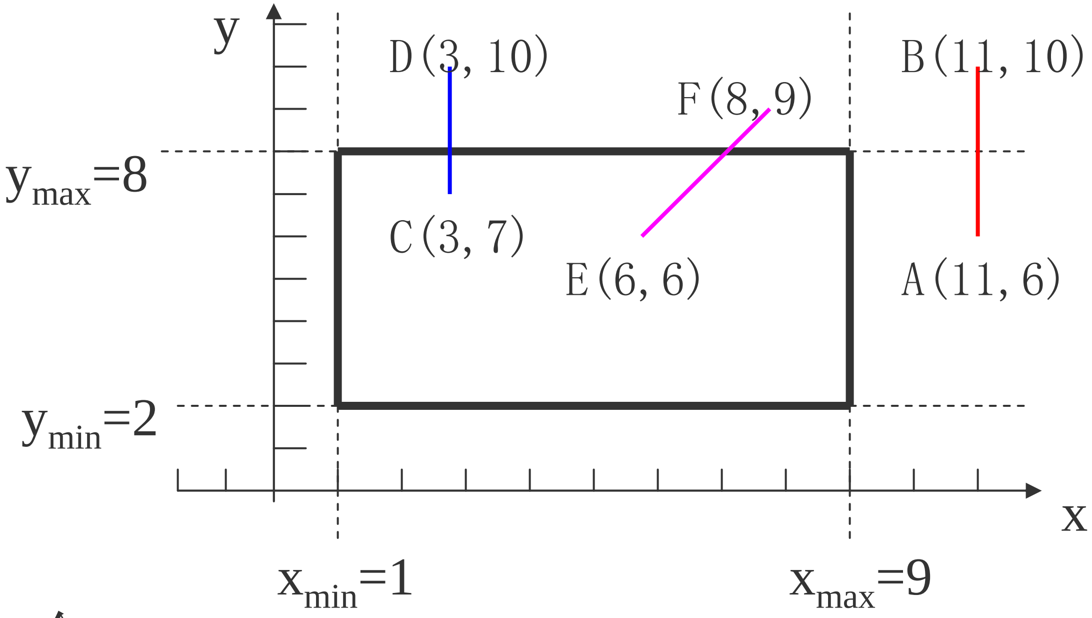
$$y = y_1 + u \cdot (y_2 - y_1)$$

（6）利用直线的扫描转换算法绘制在窗口内的直线段。

（7）算法结束。

用Liang-Barsky算法裁减直线段举例





令：

$$p_1 = -\Delta x \qquad q_1 = x_1 - x_{left}$$

$$p_2 = \Delta x \qquad q_2 = x_{right} - x_1$$

$$p_3 = -\Delta y \qquad q_3 = y_1 - y_{bottom}$$

$$p_4 = \Delta y \qquad q_4 = y_{top} - y_1$$

对于直线AB，有：

$$p_1 = 0 \qquad q_1 = 10$$

$$p_2 = 0 \qquad q_2 = -2$$

$$p_3 = -4 \qquad q_3 = 4$$

$$p_4 = 4 \qquad q_4 = 2$$

AB完全在右边界之右

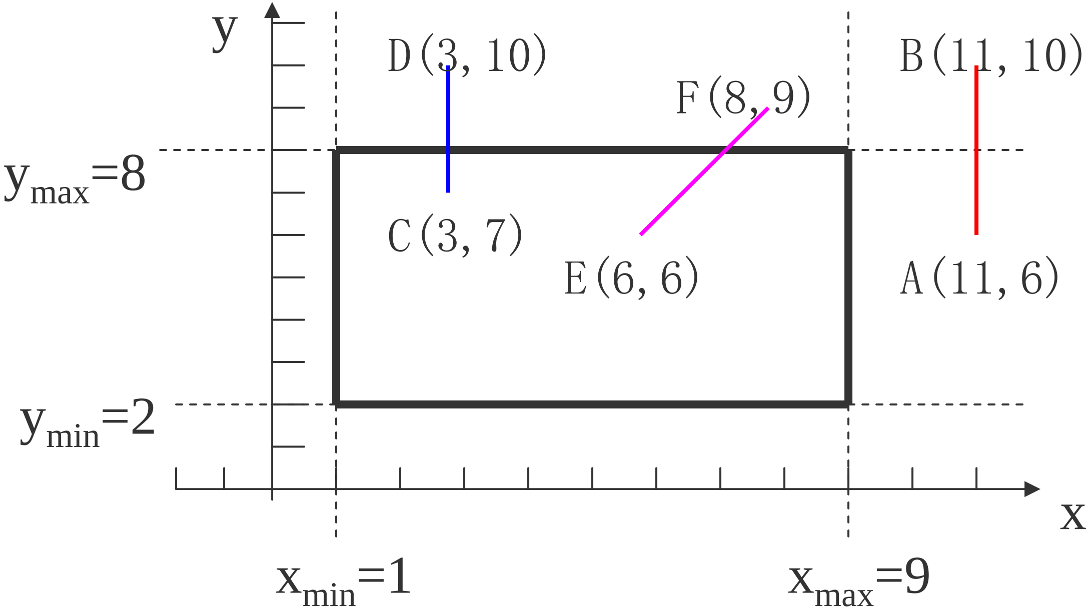
对于直线CD，有：

$p_1 = 0$　　$q_1 = 2$

$p_2 = 0$　　$q_2 = 6$

$p_3 = -3$　　$q_3 = 5$

$p_4 = 3$　　$q_4 = 1$



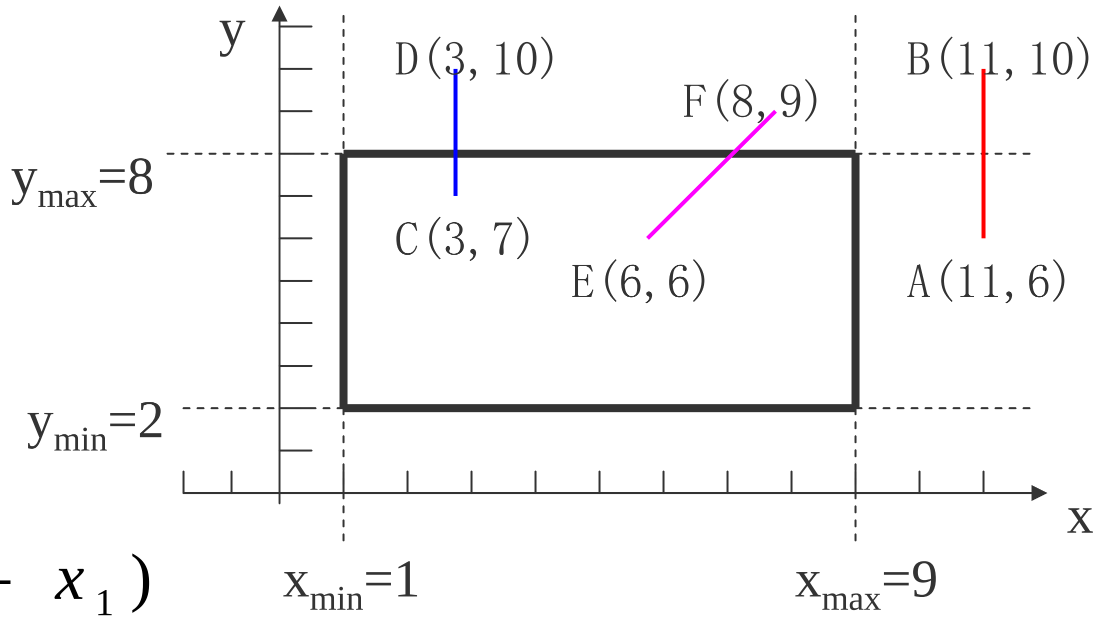
$u_3 = -5/3$　　$u_4 = 1/3$

$$u_{max} = \max(0, -5/3) = 0$$

$$u_{min} = \min(1, 1/3) = 1/3$$

$u_{max} = 0$

$u_{min} = 1/3$



$$x = x_1 + u \cdot (x_2 - x_1)$$

$$y = y_1 + u \cdot (y_2 - y_1)$$

裁减后直线的两个端点是（3，7）和（3，8）。

对于直线EF，有：

$p_1 = -2$　　$q_1 = 5$

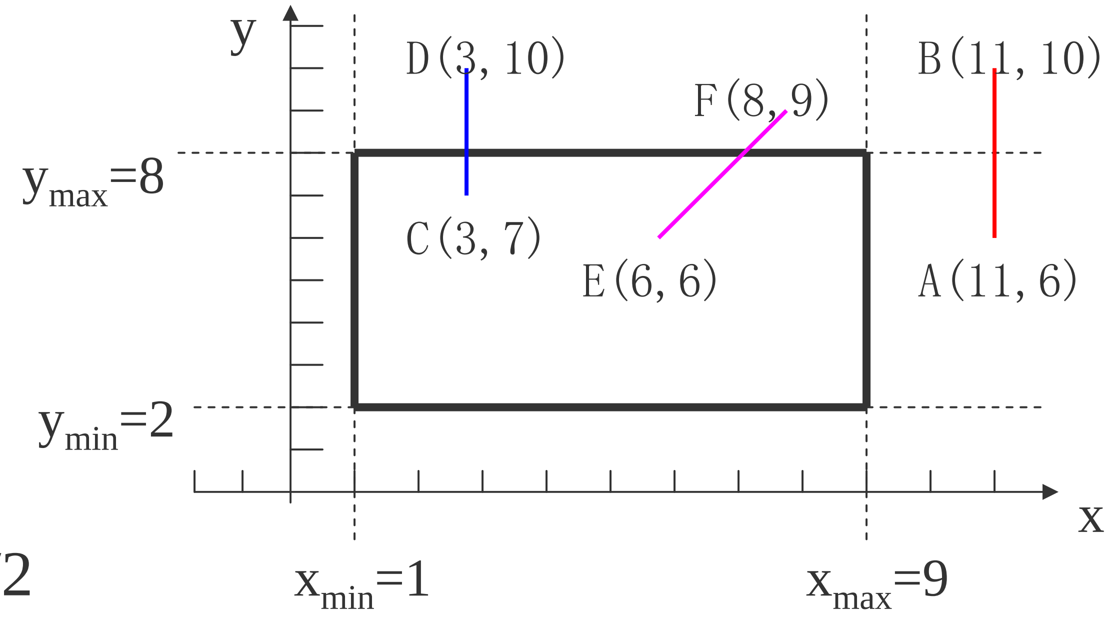
$p_2 = 2$　　$q_2 = 3$

$p_3 = -3$　　$q_3 = 4$

$p_4 = 3$　　$q_4 = 2$

$u_1 = -5/2$　　$u_2 = 3/2$

$u_3 = -4/3$　　$u_4 = 2/3$



因此：

$u_{max} = \max(0, -5/2, -4/3) = 0$

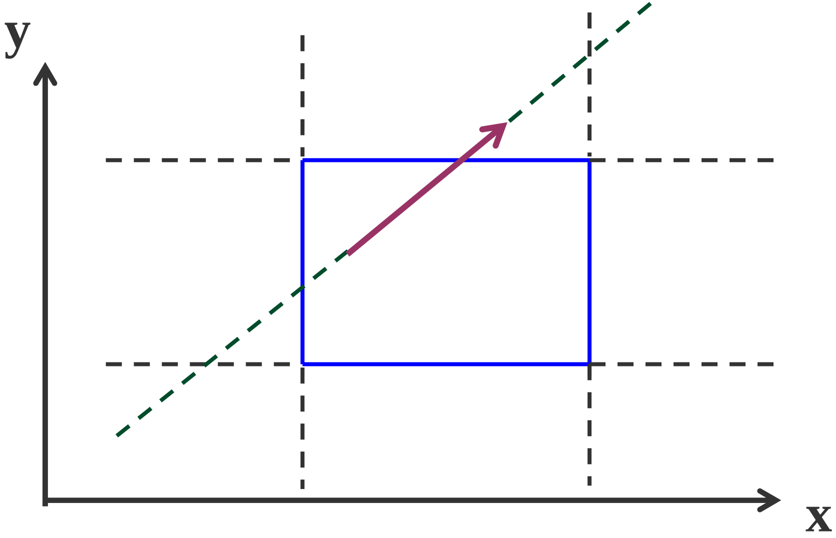
$u_{min} = \min(1, 3/2, 2/3) = 2/3$

$u_{max} < u_{min}$

裁剪后的直线的两个端点是（6，6）和（7.33，8）

Liang-Barsky算法小结

1、直线段看成是有方向的



2、直线参数化

$$x = x_1 + u \cdot (x_2 - x_1)$$

$$y = y_1 + u \cdot (y_2 - y_1)$$

3、判断线段上一点是否在窗口内，需满足下面两个不等式

$$x_{left} \le x_1 + u \cdot \Delta x \le x_{right}$$

$$y_{bottom} \le y_1 + u \cdot \Delta y \le y_{top}$$

$$u \cdot p_k \le q_k$$

4、线段和窗口边界一共有四个交点

$$u_k = \frac{q_k}{p_k} \qquad (p_k \neq 0, k = 1,2,3,4)$$

$$u_{max} = \max(0, u_k|_{pk<0}, u_k|_{pk<0})$$

$$u_{min} = \min(u_k|_{pk>0}, u_k|_{pk>0}, 1)$$

$$x = x_1 + u \cdot (x_2 - x_1)$$

$$y = y_1 + u \cdot (y_2 - y_1)$$

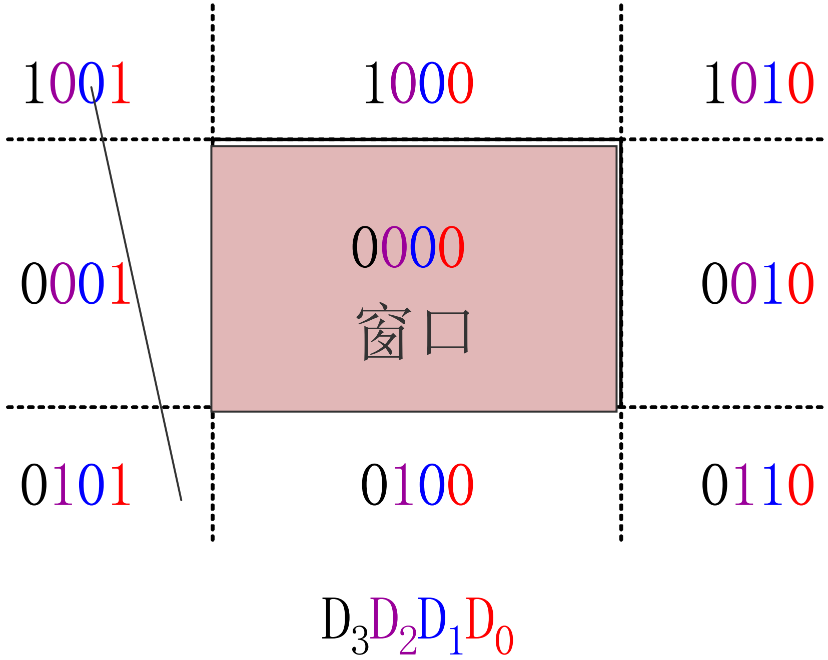
梁算法引进了一种新的思想，把窗口的4条边根据线段的方向走向分成入边和出边，裁剪的算法就变得比较简单。

它首先把线段的裁剪变成了射线的裁剪，也就是把线段赋以方向。发现最终裁剪结果的起始端点肯定是入边的两个交点和线段的起始点里面的一点；而裁剪结果的终点肯定是出边的两个交点和线段的终点里面的一点；前三个点取最大，后三个点取最小。这样就把一个经典的裁剪问题变成了解不等式。

这也是中国人的算法第一次出现在了所有图形学教科书都必须提的一个算法。这就叫原始创新！

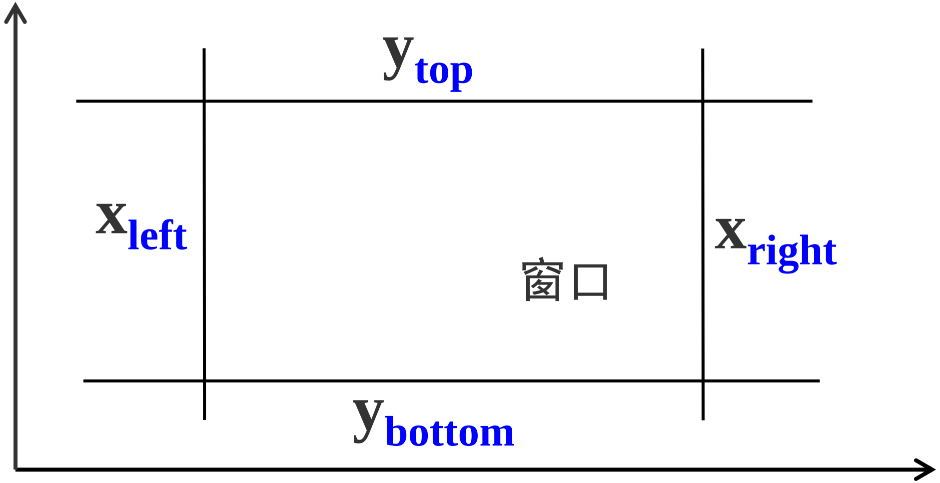
Cohen-Sutherland PK Liang-Barsky裁剪算法比较

1、Cohen-Sutherland算法的核心思想是编码



窗口及区域编码

2、如果被裁剪的图形大部分线段要么在窗口内或者要么完全在窗口外，很少有贯穿窗口的。Cohen-Sutherland算法效果非常好



3、在一般情况下，Liang-Barsky裁剪算法的效率则优于Cohen-Sutherland算法

4、Cohen-Sutherland和Liang-Barsky只能应用于矩形窗口

裁剪算法是非常底层的算法，任何一个图形显示的算法和软件都离不开这些底层算法

这些底层算法目前都已经固化到计算机硬件里了。现在做一个图形软件，不需要再研究裁剪算法

一是了解图形学底层算法的思想

另一方面，改进底层算法对提高图形显示和处理效率具有至关重要的作用