Clipping

Cohen-Sutherland and Cyrus-Beck algorithms

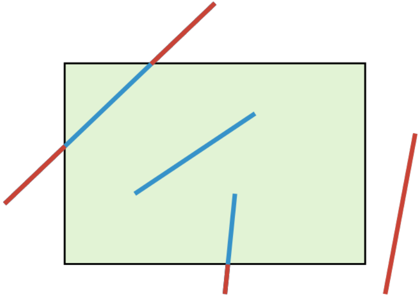
Marcel Makovník

Department of Algebra and Geometry
Faculty of Mathematics, Physics and Informatics
Comenius University in Bratislava, Slovakia

12 December 2019

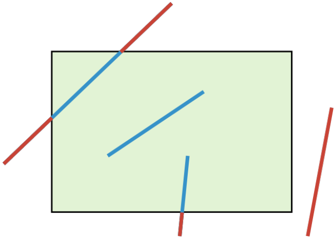
Motivation

- The aim is to remove each (part of) object which is outside the viewing area.
- Only the line segments need to be clipped.
  - The boundary of a polygon is also created by line segments.
- Usually we need to process the large number of line segments $\longrightarrow$ small number of operations
  - How to find the intersection of a line and the window efficiently?

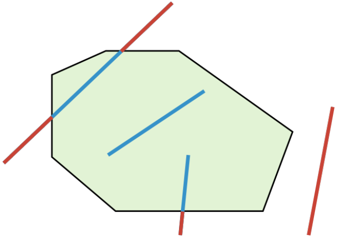


Algorithms

The algorithms may be classified with respect to the shape of the viewing area (clip window):



RECTANGLE
Cohen-Sutherland, Liang-Barsky, Nichol-Lee-Nichol



CONVEX POLYGON
Cyrus-Beck

Cohen-Sutherland

- Works only for the rectangular (axis-aligned) clip window

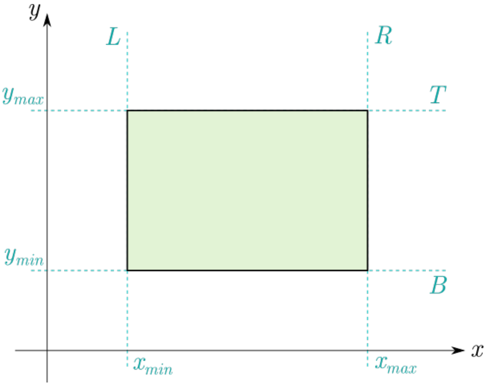
INPUT:
- clip window (given by values $x_{min}, x_{max}, y_{min}, y_{max}$,
- list of line segments (given by endpoints).

PROCESSING: For each line segment AB:

1. encode the points A and B,
2. decide if the line segment is **accepted / rejected / clipped.**

OUTPUT:
- list of the clipped line segments (their endpoints), drawing.

Cohen-Sutherland – INPUT – clip window



- Lines $x = x_{min}$, $x = x_{max}$, $y = y_{min}$ and $y = y_{max}$ divide the plane into 9 regions.
- Let us introduce the notation:

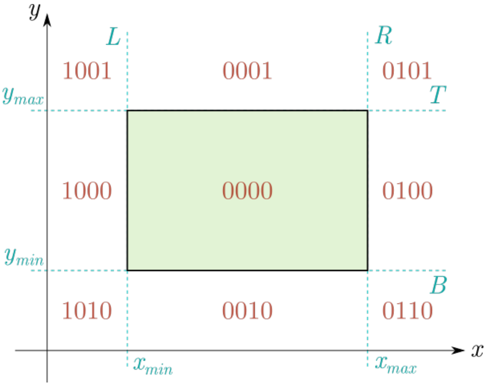
  if $x < x_{min}$, then $L = 1$; else $L = 0$

  if $x > x_{max}$, then $R = 1$; else $R = 0$

  if $y < y_{min}$, then $B = 1$; else $B = 0$

  if $y > y_{max}$, then $T = 1$; else $T = 0$

Cohen-Sutherland – INPUT – clip window



- Using this notation we can encode each of the region using only 4 bits.
  - We need to choose the order of bits, e.g. $LRBT$.
- Each region has unique encoding, i.e. to determine the position of a point with respect to the regions, we need to determine its $LRBT$ code.

Cohen-Sutherland – PROCESSING

For each line segment AB:

1. get the $LRBT$ code for points A and B, i.e. code(A), code(B),
2. now, three situations may occur

Trivial accept – $\big(\text{code}(A) \mid\mid \text{code}(B)\big) = 0000$

both endpoints lie inside the region 0000,
$\hookrightarrow$ draw the line segment AB

Trivial reject – $\big(\text{code}(A) \;\&\; \text{code}(B)\big) \neq 0000$

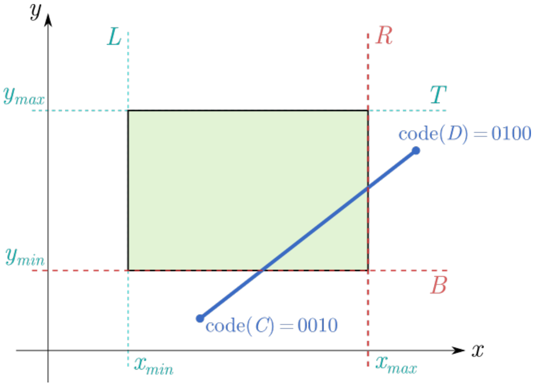
both endpoints lie outside the region 0000,
$\hookrightarrow$ drop the line segment AB

Test clipping – $\big(\text{code}(A) \;\&\; \text{code}(B)\big) = 0000$

the line segments intersects the region 0000,
$\hookrightarrow$ perform clipping

Cohen-Sutherland – PROCESSING – test clipping

The line segment AB is clipped by those lines, which have the bit value 1 (e.g. if one of the points A or B has L-bit equal to 1, we clip it by the respective line $x = x_{min}$).



Cohen-Sutherland – PROCESSING – test clipping

The coordinates of the endpoints of the clipped line:

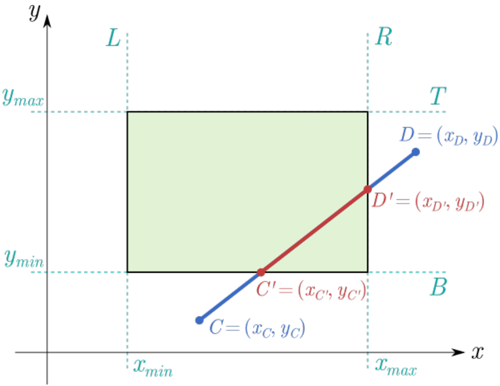
- Clipping by R:

$$x_{D'} = x_{max}$$
$$y_{D'} = y_D + \frac{y_C - y_D}{x_C - x_D}(x_{max} - x_D)$$

- Clipping by B:

$$x_{C'} = x_C + \frac{x_C - x_D}{y_C - y_D}(y_{min} - y_C)$$
$$y_{C'} = y_{min}$$

Clipping by L and T may be derived analogically.



Cyrus-Beck

- Works for any convex polygon (i.e. including rectangles)

INPUT:
- an oriented convex polygon given by its edges $\mathbf{e}_i$ and normal vectors,
- list of oriented line segments (given by endpoints).

PROCESSING: For each oriented line segment AB:

1. for each edge $\mathbf{e}_i$, clip AB by the edge $\mathbf{e}_i$.

OUTPUT:
- list of the clipped line segments (their endpoints), drawing.

Cyrus-Beck – input line segment

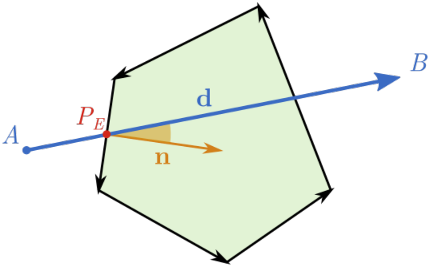
- The segment AB is expressed by its parametric equation

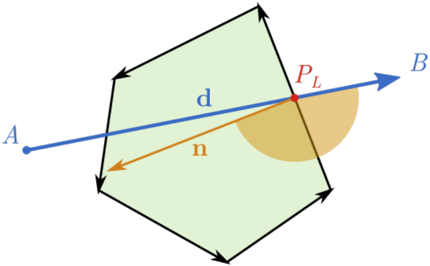
$$P(t) = A + \mathbf{d}t, \qquad \mathbf{d} = B - A, \qquad t \in \langle 0, 1 \rangle.$$

- We are searching for:
  - $\max\{t_E\} \in \langle 0, 1 \rangle$ – the maximal t value, where the line segments **enters** the clip window, i.e. the entering point P_E,
  - $\min\{t_L\} \in \langle 0, 1 \rangle$ – the minimal t value, where the line segments **leaves** the clip window, i.e. the leaving point P_L,

Cyrus-Beck – entering and leaving points

The entering and leaving points may be classified as follows:



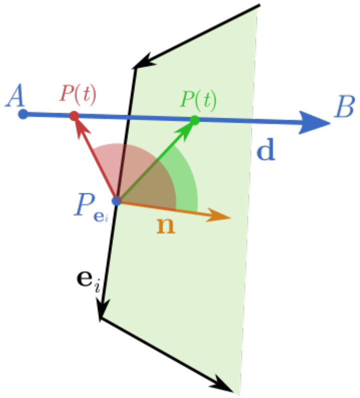


point P_E is entering if $\langle \mathbf{n}, \mathbf{d} \rangle \geq 0$

point P_L is leaving if $\langle \mathbf{n}, \mathbf{d} \rangle \leq 0$

Cyrus-Beck – position of a point with respect to the edge

The entering and leaving points may be classified as follows:



The point $P(t) \in AB$ is:

- exterior, if $\sphericalangle(P(t) - P_{\mathbf{e}_i}, \mathbf{n}) > {}^{\pi}/_{2}$, i.e. $\langle P(t) - P_{\mathbf{e}_i}, \mathbf{n} \rangle < 0$
- interior, if $\sphericalangle(P(t) - P_{\mathbf{e}_i}, \mathbf{n}) < {}^{\pi}/_{2}$, i.e. $\langle P(t) - P_{\mathbf{e}_i}, \mathbf{n} \rangle > 0$

Cyrus-Beck – a point lying on the edge

The point $P(t)$ lies on the edge $\mathbf{e}_i$, if $\sphericalangle(P(t) - P_{\mathbf{e}_i}, \mathbf{n}) = {}^{\pi}/_{2}$, i.e. $\langle P(t) - P_{\mathbf{e}_i}, \mathbf{n}\rangle = 0$. By expanding the expression we can express the parameter t and differentiate between three situations:

- $\langle \mathbf{d}, \mathbf{n}\rangle \neq 0$ and $t = \dfrac{\langle P_{\mathbf{e}_i} - A, \mathbf{n}\rangle}{\langle \mathbf{d}, \mathbf{n}\rangle} \rightsquigarrow AB$ and $\mathbf{e}_i$ are intersecting $\rightsquigarrow$ CLIP,
- $\langle \mathbf{d}, \mathbf{n}\rangle = 0$ and $\langle P_{\mathbf{e}_i} - A, \mathbf{n}\rangle = 0 \rightsquigarrow \mathbf{e}_i$ belongs to the **line** $AB \rightsquigarrow$ NO CLIP,
- $\langle \mathbf{d}, \mathbf{n}\rangle = 0$ and $\langle P_{\mathbf{e}_i} - A, \mathbf{n}\rangle \neq 0 \rightsquigarrow \mathbf{e}_i$ is parallel with $AB \rightsquigarrow$ NO CLIP,

Cyrus-Beck – clipping

Knowing, that $P(t) \in \mathbf{e}_i \Leftrightarrow t = \dfrac{\langle P_{\mathbf{e}_i} - A, \mathbf{n} \rangle}{\langle \mathbf{d}, \mathbf{n} \rangle}$, we say:

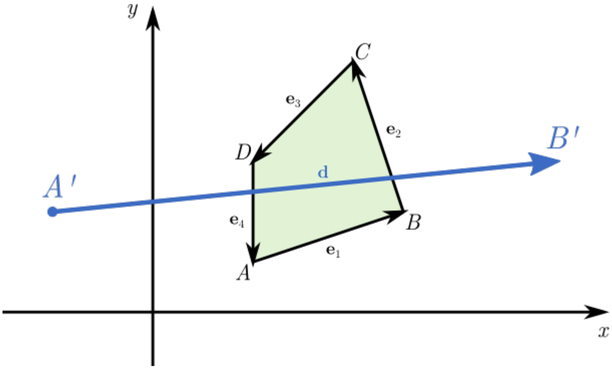
- if $\langle \mathbf{d}, \mathbf{n} \rangle > 0,\ t$ is the *entering parameter candidate*, denoted by t_E, and the point $P(t_E)$ is the *entering point candidate*,
- if $\langle \mathbf{d}, \mathbf{n} \rangle < 0,\ t$ is the *leaving parameter candidate*, denoted by t_L, and the point $P(t_L)$ is the *leaving point candidate*,

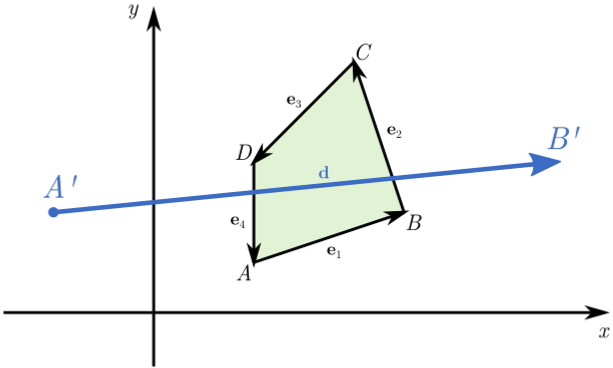
After we finished the clipping, we need to check if $t_E < t_L$, so the clipped line actually lies in the clip window

Clip the line segment $A'B'$ into the clip window, given by vertices

$$A = (2, 1)^\top,\ B = (5, 2)^\top,\ C = (4, 5)^\top,\ D = (2, 3)^\top,$$

where the endpoints of the segment are $A' = (-2, 2)^\top$ and $B' = (8, 3)^\top$. The clipping is performed using the Cyrus-Beck algorithm.





PREPROCESSING:

1. choose the orientation of the clip polygon – **positive** (CCW) – inward-pointing normals
2. set the direction vector of $A'B'$ $\mathbf{d} := B' - A' = (10, 1)^\top$
3. initialize the entering and leaving parameter $t_E = 0,\ t_L = 1$.

CLIP $\mathbf{e}_1$:

1. Choose the normal $\mathbf{n}_1$ for $\mathbf{e}_1$:

$$\mathbf{n}_1 \sim B - A = (3, 1)$$
$$\mathbf{n}_1 = (-1, 3)$$

   Choose the (arbitrary) point P_1 belonging to $\mathbf{e}_1$:

$$P_1 = \frac{A + B}{2} = (^7/_2, ^3/_2)$$ (we choose the midpoint)

2. Classify the type of parameter t:

$$\langle \mathbf{d}, \mathbf{n}_1 \rangle = \langle (10, 1), (-1, 3) \rangle = -7 < 0$$

   t is the leaving candidate ($t \rightsquigarrow t_L$)

3. Compute the value of t:

$$t = \frac{\langle P_1 - A', \mathbf{n}_1 \rangle}{\langle \mathbf{d}, \mathbf{n}_1 \rangle} = \frac{-\frac{14}{2}}{-7} = 1.$$

4. Refresh the t_L value:

$$t_L = \min\{t_L, t\} = \min\{1, 1\} = 1$$

CLIP $\mathbf{e}_2$:

1. $\mathbf{n}_2 \sim C - B = (-1, 3)$
   $\mathbf{n}_2 = (-3, -1)$
   $P_2 = \dfrac{B + C}{2} = (^9/_2, ^7/_2)$
2. $\langle \mathbf{d}, \mathbf{n}_2 \rangle = \langle (10, 1), (-3, -1) \rangle = -31 < 0$
   $t \rightsquigarrow t_L$
3. $t = \dfrac{\langle P_2 - A', \mathbf{n}_2 \rangle}{\langle \mathbf{d}, \mathbf{n}_2 \rangle} = \dfrac{-\frac{42}{2}}{-31} = \dfrac{21}{31}.$
4. $t_L = \min\{t_L, t\} = \min\left\{1, \dfrac{21}{31}\right\} = \dfrac{21}{31}$

CLIP $\mathbf{e}_3$:

1. $\mathbf{n}_3 \sim D - C = (-2, -2)$
   $\mathbf{n}_3 = (2, -2)$
   $P_3 = \dfrac{C + D}{2} = (3, 4)$
2. $\langle \mathbf{d}, \mathbf{n}_3 \rangle = \langle (10, 1), (2, -2) \rangle = 18 > 0$
   $t \rightsquigarrow t_E$
3. $t = \dfrac{\langle P_3 - A', \mathbf{n}_3 \rangle}{\langle \mathbf{d}, \mathbf{n}_3 \rangle} = \dfrac{6}{18} = \dfrac{1}{3}.$
4. $t_E = \max\{t_E, t\} = \max\left\{0, \dfrac{1}{3}\right\} = \dfrac{1}{3}$

CLIP $\mathbf{e}_4$:

1. $\mathbf{n}_4 \sim A - D = (0, -2)$
   $\mathbf{n}_4 = (2, 0)$
   $P_4 = \dfrac{D + A}{2} = (2, 2)$
2. $\langle \mathbf{d}, \mathbf{n}_4 \rangle = \langle (10, 1), (2, 0) \rangle = 20 > 0$
   $t \rightsquigarrow t_E$
3. $t = \dfrac{\langle P_4 - A', \mathbf{n}_4 \rangle}{\langle \mathbf{d}, \mathbf{n}_4 \rangle} = \dfrac{8}{20} = \dfrac{2}{5}.$
4. $t_E = \max\{t_E, t\} = \max\left\{\dfrac{1}{3}, \dfrac{2}{5}\right\} = \dfrac{2}{5}$

After we finnished clipping by the edges, we need to check if

$$\langle 0, 1 \rangle \ni \frac{2}{5} = t_E < t_L = \frac{21}{31} \in \langle 0, 1 \rangle,$$

which is obviously true. Now we can finally compute the endpoints P_E, P_L of the clipped line segment by inserting it to the equation of the line segment $A'B'$:

$$P(t) = A' + \mathbf{d}t = (-2, 2) + (10, 1)t,$$

$$P_E = P(t_E) = (2, {}^{12}/_{5}),$$

$$P_L = P(t_L) = ({}^{148}/_{31}, {}^{83}/_{31}).$$