

Clipping and windowing

Introduction

Many graphics application programs give the user the impression of looking through a window at a very large picture.

To display an enlarged portion of a picture we must not only apply the appropriate scaling and translation but identify the visible parts of the picture for inclusion in the displayed image. The correct way to select visible information for display is to use **clipping** (a process which divides each element of the picture into its visible and invisible portions, allowing the invisible portion to be discarded) . Clipping can be applied to a variety of different types of picture elements:

vectors, curves of various kinds, and even polygons. The basis for these clipping operations is a simple pair of inequalities that determine whether a point (x,y) is visible or not:

$$x_{\text{left}} \leq x \leq x_{\text{right}} , \quad y_{\text{bottom}} \leq y \leq y_{\text{top}}$$

Where x_{left} , x_{right} , y_{bottom} , y_{top} are the positions of the edges of the screen. These inequalities provide us with a very simple method of clipping pictures on a point by point basis; we substitute the coordinates of each point for x and y and if the point fails to satisfy either inequality; it is invisible. It would be quite inappropriate to clip pictures by converting all picture elements into points and using these inequalities; the clipping process would take far too long and would leave the picture in a form no longer suitable for a line drawing display. We must attempt to clip larger elements of the picture. This involves developing more powerful clipping algorithms that can be determine the visible and invisible portions of such picture elements.

Clipping window

It is refer to a rectangular region whose sides are aligned with the coordinates axes. The x extent is measured from **xmin** to **xmax** and the y extent is measured from **ymin** to **ymax**.

Point clipping

The basis for these clipping operations is a simple pair of inequalities that determine whether a point (x,y) is visible or not:

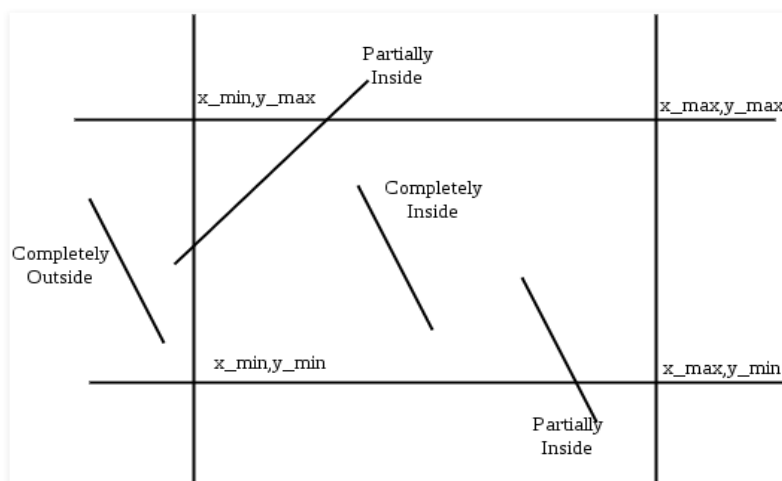
$$x_{min} \leq x \leq x_{max} , \quad y_{min} \leq y \leq y_{max}$$

Where **xmin**, **xmax**, **ymin**, **ymax** are the positions of the edges of the window.

Line clipping

Lines that do not intersect the clipping window are either completely inside the window or completely outside the window.

On the other hand a line that intersects the clipping window is divided by the intersection point (s) into segments that are either inside or outside the window. The following algorithm provide efficient way to decide the relationship between an arbitrary line and the clipping window to find intersection point (s).



The Cohen–Sutherland algorithm

The **Cohen–Sutherland algorithm** is a computer-graphics algorithm used for line clipping.

The Cohen–Sutherland algorithm can be used only on a rectangular clip window.

Given a set of lines and a rectangular area of interest, the task is to remove lines which are outside the area of interest and clip the lines which are partially inside the area.

Cohen-Sutherland algorithm divides a two-dimensional space into 9 regions and then efficiently determines the lines and portions of lines that are inside the given rectangular area.

The algorithm can be outlines as follows:-

Nine regions are created, eight "outside" regions and one "inside" region.

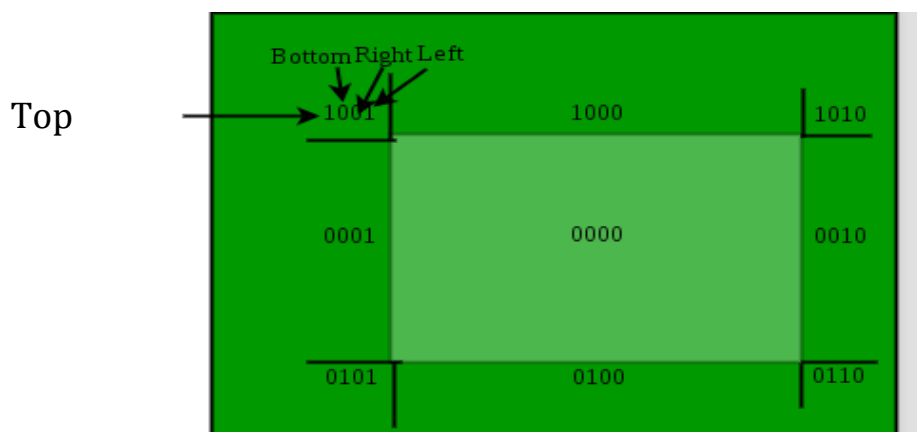
For a given line extreme point (x, y) , we can quickly find its region's four bit code. Four bit code can be computed by comparing x and y with four values (x_{\min} , x_{\max} , y_{\min} and y_{\max}).

If x is less than x_{\min} then bit number 1 is set.

If x is greater than x_{\max} then bit number 2 is set.

If y is less than y_{\min} then bit number 3 is set.

If y is greater than y_{\max} then bit number 4 is set



The diagram on the following page is associated with the checking order TBRL, corresponding to Top, Bottom, Right, Left. We assign a 1-bit where the region is strictly outside the boundary (in the half-plane not containing the window), and a 0-bit where the region is on the same side as the window. Thus, only the window itself is assigned all zeros. Since the high-order bit is associated with the top boundary for example, only the three regions above the window (outside the top boundary) have high-order bit equal to 1.

There are three possible cases for any given line.

1. **Completely inside the given rectangle** : Bitwise OR of region of two end points of line is 0 (Both points are inside the rectangle)
2. **Completely outside the given rectangle** : Both endpoints share at least one outside region which implies that the line does not cross the visible region. (bitwise AND of endpoints \neq 0).
3. **Partially inside the window** : Both endpoints are in different regions. In this case, the algorithm finds one of the two points that is outside the rectangular region. The intersection of the line from outside point and rectangular window becomes new corner point and the algorithm repeats

Intersection points

Intersection points with a clipping boundary can be calculated using the slop-intercept form of the line equation. For a line with endpoint coordinates (x_1, y_1) and (x_2, y_2) , the y coordinate of the intersection point with a **vertical boundary** can be obtained with the calculation

$$y = y_1 + m(x - x_1)$$

Where the x value is set either to x_{\min} or to x_{\max} , and the slop of the line is calculated as

$$m = (y_2 - y_1) / (x_2 - x_1).$$

Similarly, if we are looking for the intersection with a **horizontal boundary**, the x coordinate can be calculated as

$$x = x_1 + (y - y_1) / m$$

Not

1. If the boundary line is vertical then

$X=x_{\min}$ if the line is left

$X=x_{\max}$ if the line is right

$$y = y_1 + m(x - x_1)$$

2. If the boundary line is horizontal then

$Y=y_{\min}$ if the line is bottom

$Y=y_{\max}$ if the line is top

$$x = x_1 + (y - y_1) / m$$

Example1 :

Apply the Cohen Sutherland line clipping algorithm to clip the line segment with coordinates (30,60) and (60,25) against the window with $(X_{min}, Y_{min}) = (10, 10)$ and $(X_{max}, Y_{max}) = (50, 50)$.

Solution**Clip bit code**

AB 1000 AND

0010

0000 (Partially inside) (clipping)

First ,Find the slop of line AB from the equation:

$$m = (y_2 - y_1) / (x_2 - x_1)$$

$$m = (25 - 60) / (60 - 30)$$

$$= -35 / 30$$

$$= -1.16$$

Then ,We find the coordinate of intersection point from line A A'.

The boundary line A A' is horizontal ,so $Y_{max} = y = 50$ and calculate x value from this :

$$x = x_1 + (y - y_1) / m$$

$$= 30 + (50 - 60) / -1.16$$

$$= 30 + -10 / -1.16$$

$$= 30 + 8.6$$

$$= 38.6$$

the coordinate of intersection point is **A(38.6,50)**.

We find the coordinate of intersection point from line BB' .

The boundary line BB' is vertical, so $x_{max}=x=50$ and calculate y value from this:

$$y = y_1 + m(x - x_1)$$

$$= 25 + (-1.16)(50 - 60)$$

$$= 25 + 11.6$$

$$= 36.6$$

The coordinate of intersection point is $B'(50, 36.6)$.

Example2:

Window is defined A(20,20),B(90,20),C(90,70),D(20,70)

Find visible portion of

line1 :P1(10,30),P2(80,90)

Line2: Q1(20,10) , Q2(70,60)

using Cohen Sutherland line clipping algorithm.

Solution

$X_{\min}=20$, $X_{\max}=90$, $y_{\min}=20$, $y_{\max}=70$

Clip bit code

P1P2 0001 AND

1000

0000 **(Partially inside) (clipping)**

Q1Q2 0101 AND

0000

0000 **(Partially inside) (clipping)**

First find the slop of line P1P2 from the equation:

$$m = (y_2 - y_1) / (x_2 - x_1)$$

$$=(90-30)/(80-10)$$

$$=60/70$$

$$=0.8$$

Then find the coordinate of intersection point from line P_1P_1' .

The boundary line P_1P_1' is vertical, so $X_{\min}=x=20$ and calculate y value from this :

$$y = y_1 + m(x - x_1)$$

$$= 30 + 0.8(20 - 10)$$

$$= 30 + 8 = 38$$

the coordinate of intersection point **$P_1'(20, 38)$** .

Then find the coordinate of intersection point from line P_2P_2' .

The boundary line P_1P_1' is horizontal, so $y_{\max}=y=70$ and find x from this equation:

$$x = x_1 + (y - y_1) / m$$

$$= 80 + (70 - 90) / 0.8$$

$$= 80 + (-20) / 0.8$$

$$= 80 + (-25) = 55$$

the coordinate of intersection point **$P_2'(55, 70)$** .

Find the slope of second line Q1Q2

$$\begin{aligned} m &= (y_2 - y_1) / (x_2 - x_1) \\ &= (60-10)/(70-20)=50/50=1 \end{aligned}$$

Then find the coordinate of intersection point from line Q1Q1'

The boundary line Q1Q1' is horizontal, so $y_{\min}=y=20$ and calculate x value from this :

$$\begin{aligned} x &= x_1 + (y - y_1) / m \\ &= 20 + (20-10)/1 \\ &= 20+10=30 \end{aligned}$$

The coordinate of intersection point is Q1' **(30,20)**

Example3:

Rectangular area of interest (defined by below four values which are coordinates of bottom left and top right)

$X_{\min}=4, y_{\min}=4, X_{\max}=10, y_{\max}=8$

A set of lines(defined by two corner coordinates)

Line 1: A(5,5), B(7,7)

Line 2: C(7,9), D(11,4)

Line 3: E(1,5), F(3,2)

Apply the Cohen Sutherland line clipping algorithm to clip the line segment.

Solution:

Clip bit code

AB 0000 AND

0000

0000 accept (inside)

CD 1000 AND

0110

0000 partially inside (clipping)

EF 0001 AND

0101

0001 reject (outside)

Find slop for line CD as follow:

$$m = (y_2 - y_1) / (x_2 - x_1)$$

$$=(4-9)/(11-7)$$

$$=-5/4=-1.25$$

We fined the coordinate of intersection point from line CC⁻,

The boundary line CC⁻ is horizontal ,so $Y_{\max}=y=8$ and find x as follow:

$$x = x_1 + (y - y_1) / m$$

$$= 7+(8-9)/-1.25$$

$$=7+-1/-1.25$$

$$7+0.8=7.8$$

The coordinate of intersection point is **C(7.8,8)**.

We fined the coordinate of intersection point form line DD⁻,

the boundary line DD⁻ is vertical ,so $x_{\max}=x=10$ and find y as follow:

$$y = y_1 + m(x - x_1)$$

$$=4+(-1.25)(10-11)$$

$$=4+1.25$$

$$=5.25$$

The coordinate of intersection point is **D(10,5.25)**.

Example4:

Window is defined A(10,20),B(20,20),C(20,10),D(10,10) Find visible portion of line P(15,15),Q(5,5) using Cohen Sutherland line clipping algorithm.

Solution**Clip bit code**

PQ 0000 AND

0101

0000 Partially inside (clipping)

Find the slop of line PQ

$$m = (y_2 - y_1) / (x_2 - x_1)$$

$$=(5-15)/(5-15)$$

$$=-10/-10=1$$

We find the coordinate of intersection point from line PP ;

The boundary line PP is horizontal ,so $Y_{min}=y=10$ and find x as follow:

$$x = x_1 + (y - y_1) / m$$

$$=15+(10-15)/1$$

$$=15-5$$

$$= 10$$

the coordinate of intersection point is **P(10,10)**.