

## Unit - 2 | Clipping

### Clipping:

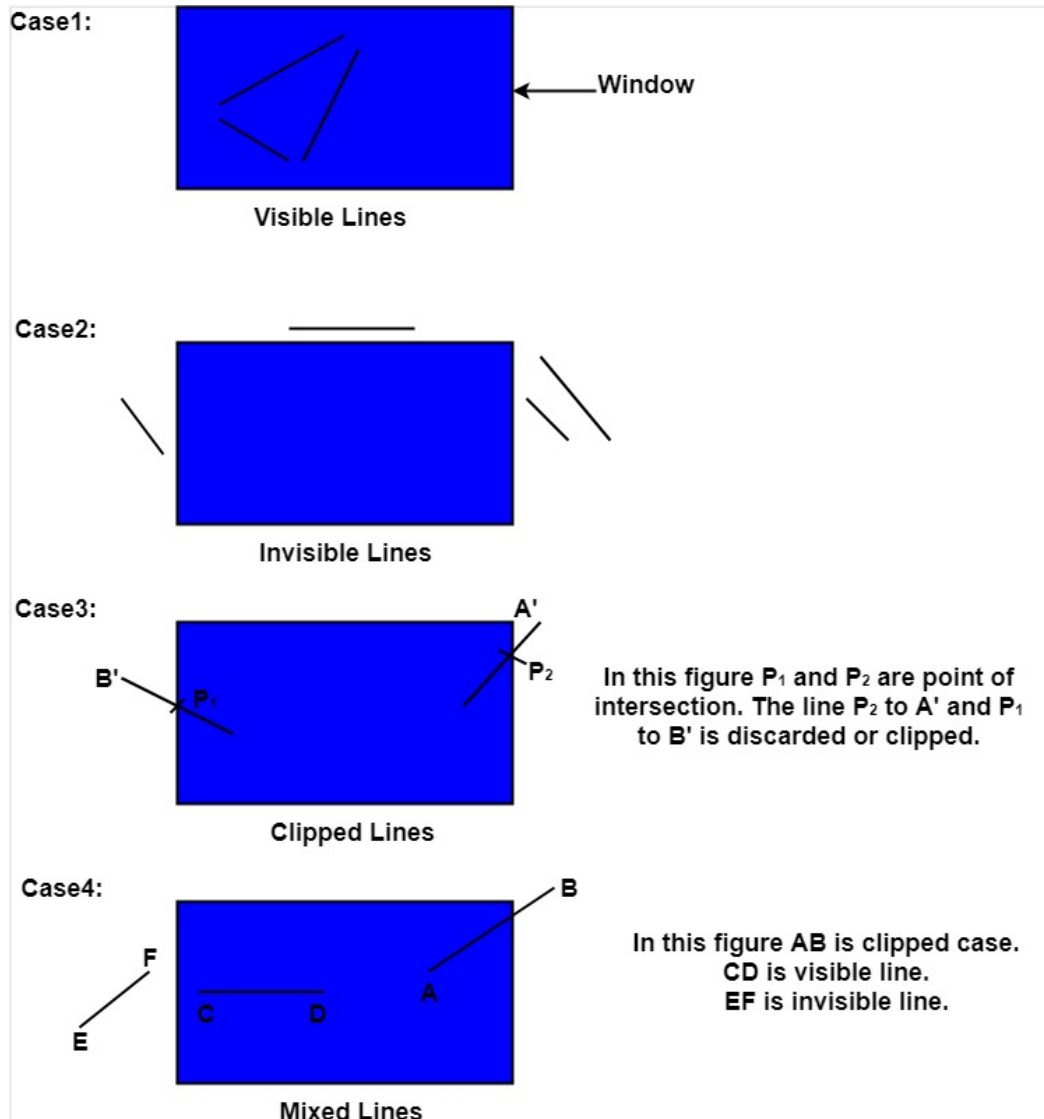
When we have to display a large portion of the picture, then not only scaling & translation is necessary, the visible part of picture is also identified. This process is not easy. Certain parts of the image are inside, while others are partially inside. The lines or elements which are partially visible will be omitted.

For deciding the visible and invisible portion, a particular process called clipping is used. Clipping determines each element into the visible and invisible portion. Visible portion is selected. An invisible portion is discarded.

### Types of Lines:

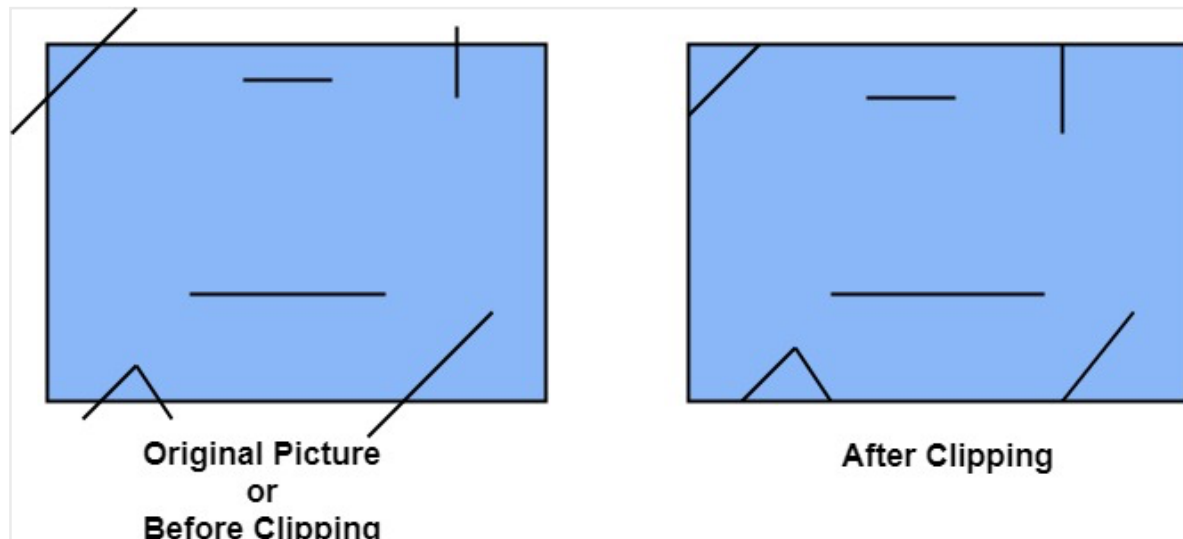
Lines are of three types:

1. **Visible:** A line or lines entirely inside the window is considered visible
2. **Invisible:** A line entirely outside the window is considered invisible
3. **Clipped:** A line partially inside the window and partially outside is clipped. For clipping point of intersection of a line with the window is determined.



Clipping can be applied through hardware as well as software. In some computers, hardware devices automatically do work of clipping. In a system where hardware clipping is not available software clipping applied.

Following figure show before and after clipping



The window against which object is clipped called a clip window. It can be curved or rectangle in shape.

### **Applications of clipping:**

1. It will extract part we desire.
2. For identifying the visible and invisible area in the 3D object.
3. For creating objects using solid modeling.
4. For drawing operations.
5. Operations related to the pointing of an object.
6. For deleting, copying, moving part of an object.

Clipping can be applied to world co-ordinates. The contents inside the window will be mapped to device co-ordinates. Another alternative is a complete world co-ordinates picture is assigned to device co-ordinates, and then clipping of viewport boundaries is done.

### **Types of Clipping:**

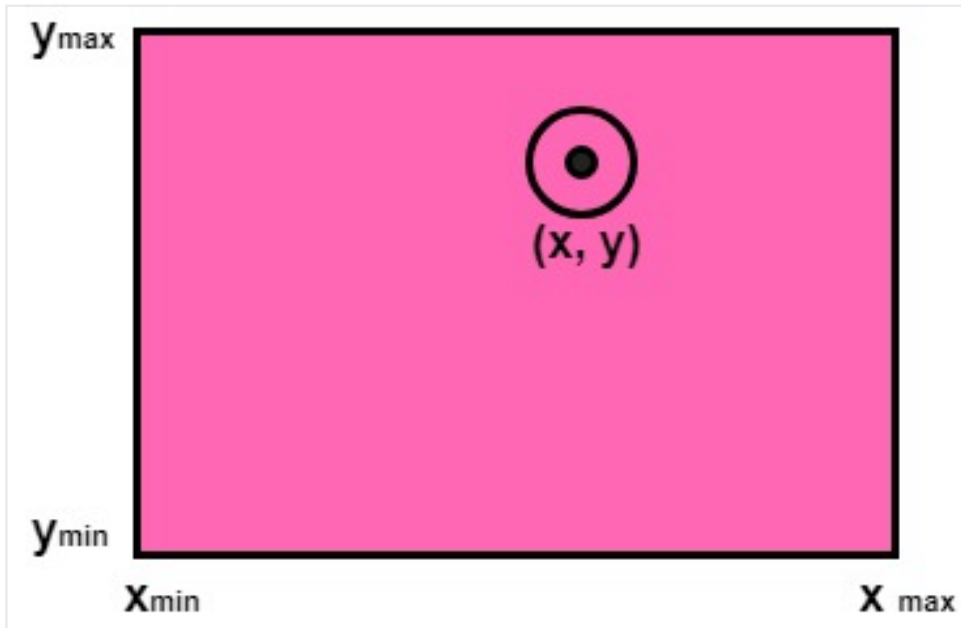
1. Point Clipping
2. Line Clipping
3. Area Clipping (Polygon)
4. Curve Clipping
5. Text Clipping
6. Exterior Clipping

### **1) Point Clipping:**

Point Clipping is used to determining, whether the point is inside the window or not. For this following conditions are checked.

1.  $x \leq x_{\max}$

2.  $x \geq x_{\min}$
3.  $y \leq y_{\max}$
4.  $y \geq y_{\min}$



The  $(x, y)$  is coordinate of the point. If anyone from the above inequalities is false, then the point will fall outside the window and will not be considered to be visible.

## 2) Line Clipping:

It is performed by using the line clipping algorithm. The line clipping algorithms are:

1. Cohen Sutherland Line Clipping Algorithm
2. Midpoint Subdivision Line Clipping Algorithm
3. Liang-Barsky Line Clipping Algorithm

### Cohen Sutherland Line Clipping Algorithm:

In the algorithm, first of all, it is detected whether line lies inside the screen or it is outside the screen. All lines come under any one of the following categories:

1. Visible
2. Not Visible
3. Clipping Case

**1. Visible:** If a line lies within the window, i.e., both endpoints of the line lies within the window. A line is visible and will be displayed as it is.

**2. Not Visible:** If a line lies outside the window it will be invisible and rejected. Such lines will not display. If any one of the following

inequalities is satisfied, then the line is considered invisible. Let A  $(x_1, y_1)$  and B  $(x_2, y_2)$  are endpoints of line.

$x_{\min}, x_{\max}$  are coordinates of the window.

$y_{\min}, y_{\max}$  are also coordinates of the window.

$$x_1 > x_{\max}$$

$$x_2 > x_{\max}$$

$$y_1 > y_{\max}$$

$$y_2 > y_{\max}$$

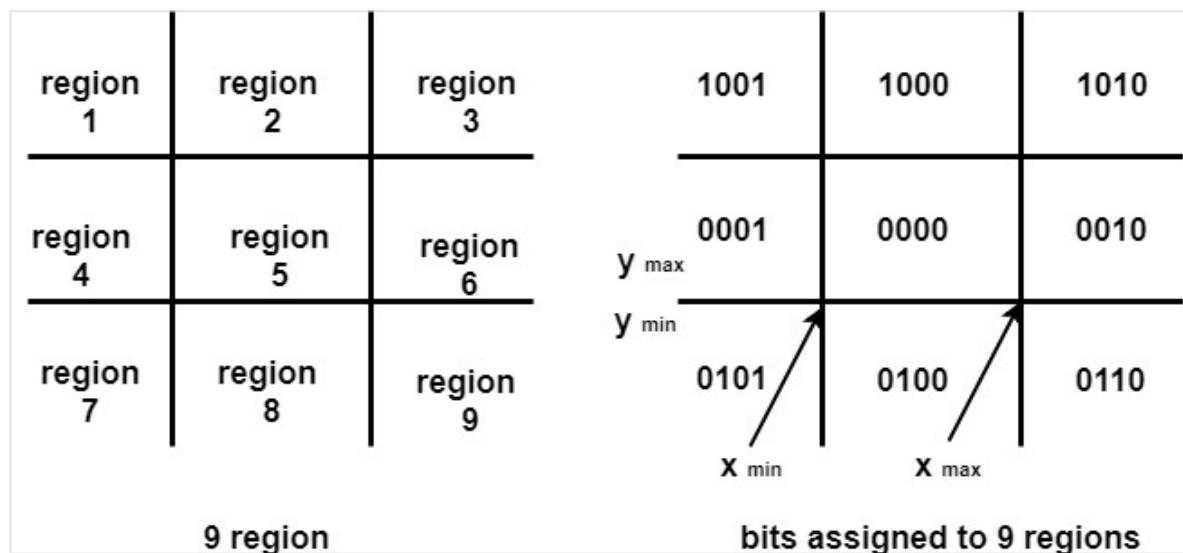
$$x_1 < x_{\min}$$

$$x_2 < x_{\min}$$

$$y_1 < y_{\min}$$

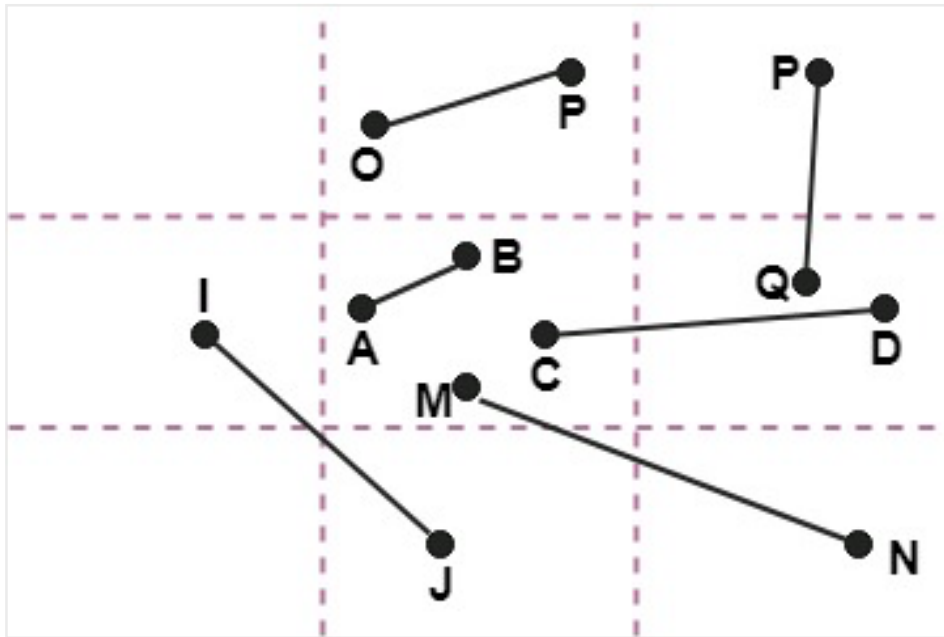
$$y_2 < y_{\min}$$

**3. Clipping Case:** If the line is neither visible case nor invisible case. It is considered to be clipped case. First of all, the category of a line is found based on nine regions given below. All nine regions are assigned codes. Each code is of 4 bits. If both endpoints of the line have end bits zero, then the line is considered to be visible.



The center area is having the code, 0000, i.e., region 5 is considered a rectangle window.

**Following figure show lines of various types**



Line AB is the visible case

Line OP is an invisible case

Line PQ is an invisible line

Line IJ are clipping candidates

Line MN are clipping candidate

Line CD are clipping candidate

### **Advantage of Cohen Sutherland Line Clipping:**

1. It calculates end-points very quickly and rejects and accepts lines quickly.
2. It can clip pictures much large than screen size.

### **Algorithm of Cohen Sutherland Line Clipping:**

**Step1:** Calculate positions of both endpoints of the line

**Step2:** Perform OR operation on both of these end-points

**Step3:** If the OR operation gives 0000

Then

line is considered to be visible

else

Perform AND operation on both endpoints

If And  $\neq$  0000

then the line is invisible

else

And=0000

Line is considered the clipped case.

**Step4:** If a line is clipped case, find an intersection with boundaries of the window

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

**(a)** If bit 1 is "1" line intersects with left boundary of rectangle window

$$y_3 = y_1 + m(x - X_1)$$

where  $X = X_{wmin}$

where  $X_{wmin}$  is the minimum value of X co-ordinate of

window

**(b)** If bit 2 is "1" line intersect with right boundary

$$y_3 = y_1 + m(X - X_1)$$

where  $X = X_{wmax}$

where X more is maximum value of X co-ordinate of the

window

**(c)** If bit 3 is "1" line intersects with bottom boundary

$$X_3 = X_1 + (y - y_1)/m$$

where  $y = y_{wmin}$

$y_{wmin}$  is the minimum value of Y co-ordinate of the window

**(d)** If bit 4 is "1" line intersects with the top boundary

$$X_3 = X_1 + (y - y_1)/m$$

where  $y = y_{wmax}$

$y_{wmax}$  is the maximum value of Y co-ordinate of the

window