5. Two Dimensional Viewing

Contents

Introduction

- The window defines what is to be viewed the view port defines where it is to be displayed.
- The two dimensional viewing transformation is referred to as window to view port transformation of windowing transformation.
- 2D viewing consist of 3 things:-
    1. Terminology
    2. Viewing pipeline
    3. Clipping

Terminology

Window

- A world coordinate area selected for display is called a window.

Viewing Port

- An area on a display device to which a window is mapped is called a view port.

Viewing transformation

- The mapping of a part of a world coordinate scene to device coordinate is referred to as viewing transformation.

Viewing coordinate system

A coordinate system defined world coordinate frame.

Viewing transformation pipeline

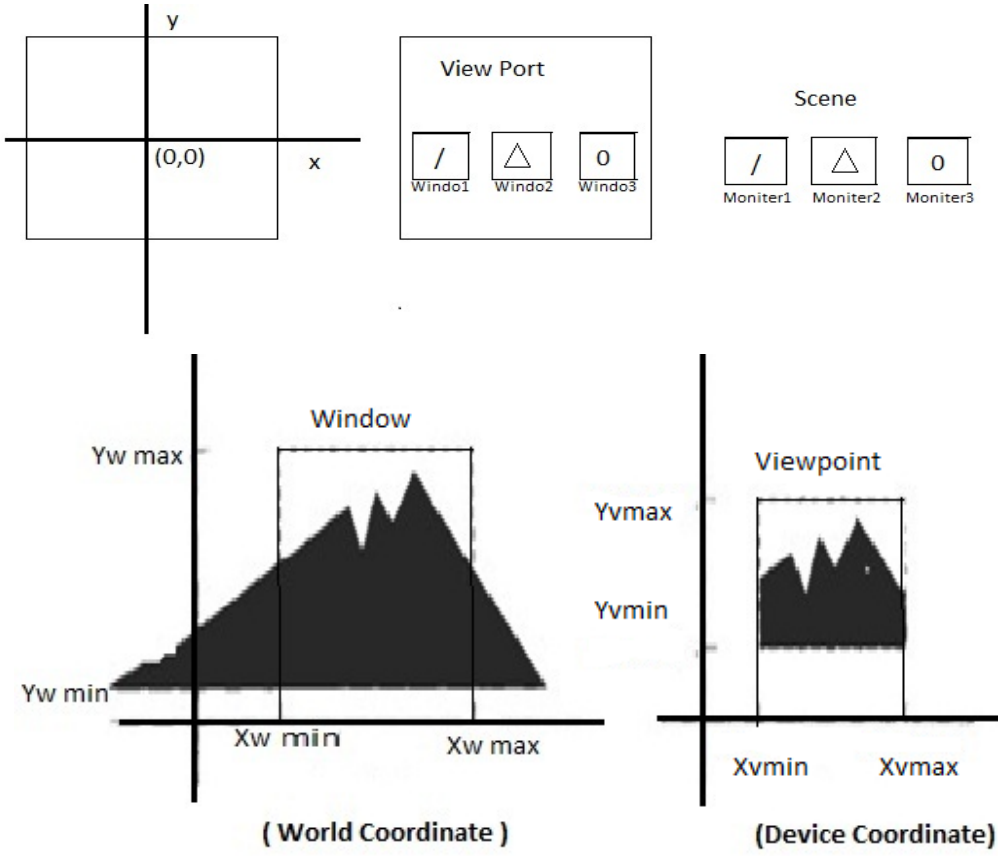
i. Construct world coordinate screen
ii. Convert world coordinate to view coordinate.
iii. Convert view coordinate to normalized view coordinate.
iv. Normalized view coordinate to device coordinate.

Transformation

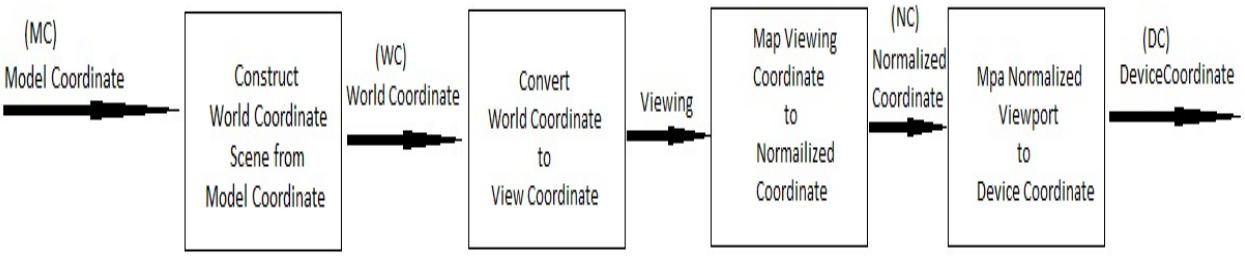
From world to device coordinate transformation, we need translation, rotation & Scaling operation.

5.1 Viewing pipeline

- A world coordinate area are selected for display window .
- An area on a display device to which a window is mapped is called a view port.
- The window defines what is to be view.
- View port define where it is to be display.
- Mapping of a world coordinate screen scale to a device coordinate is referred to as viewing transformation.
- Sometime 2D viewing transformation is simple referred to as window to view port transformation.



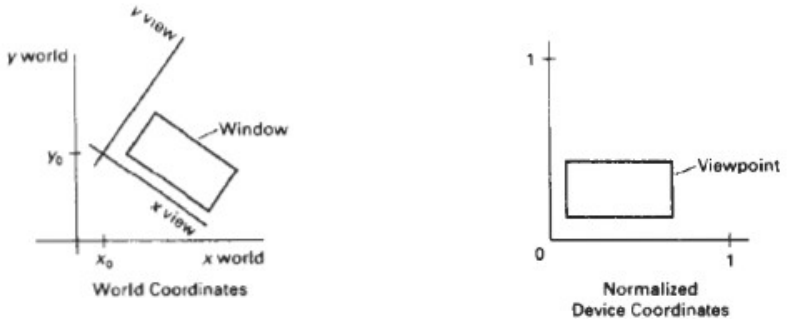
5.2 The two dimensional viewing transformation pipeline



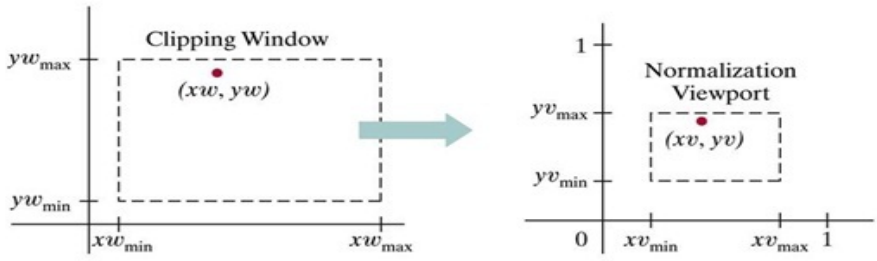
The viewing transformation in several steps, as indicated in above Fig.

- First, we construct the scene in world coordinates using the output primitives.

- Next to obtain a particular orientation for the window, we can set up a two-dimensional viewing- coordinate system in the world coordinate plane, and define a window in the viewing- coordinate system.
- The viewing- coordinate reference frame is used to provide a method for setting up arbitrary orientations for rectangular windows.
- Once the viewing reference frame is established, we can transform descriptions in world coordinates to viewing coordinates.
- We then define a viewport in normalized coordinates (in the range from 0 to 1) and map the viewing-coordinate description of the scene to normalized coordinates.
- At the final step all parts of the picture that lie outside the viewport are clipped, and the contents of the viewport are transferred to device coordinates.
- By changing the position of the viewport, we can view objects at different positions on the display area of an output device.



5.3 Window to Viewport coordinate Transformation



A point (xw, yw) in a world-coordinate clipping window is mapped to viewport coordinates (xv, yv), within a unit square, so that the relative positions of the two points in their respective rectangles are the same.

- A point at position(x_W,y_W)in a designated window is mapped to viewport coordinates (x_V,y_V) ,so that the relative positions in the two areas are the same. The figure illustrates the window to viewport mapping.
- A point at position (x_W,y_W)in the window is mapped into position(x_V,y_V)in the associated viewport. To maintain the same relative placement in viewport as in window

$$\frac{Xv - Xvmin}{Xvmax - Xvmin} = \frac{Xw - Xwmin}{Xwmax - Xwmin}$$

$$\frac{Yv - Yvmin}{Yvmax - Yvmin} = \frac{Yw - Ywmin}{Ywmax - Ywmin}$$

$$Xv = Xw + (Xv\,min - Xw\min)Sx$$

$$= Xv\min + (Xw - Xwmin)\,Sx$$

$$Sx = \left(\frac{Xvmax - Xvmin}{Xwmax - Xwmin}\right)$$

$$Yv = Yw + (Yvmin - Ywmin)Sy$$

$$= Yvmin + (Yw - Ywmin)Sy$$

$$Sy = \left(\frac{Yvmax - Yvmin}{Ywmax - Ywmin}\right)$$

5.1 Clipping

Clipping Operation

- Any procedure that identifies those positions of picture is either inside or outside of a specified reason of space is referred to as clipping algorithm or simply clipping.
- The reason against which an object is clip is called clipping windows.
- Application of clipping in to extracting the part of a defined seen for viewing.
- Depending on the application the clip window can be general polygon for it can even have curve boundary.
- For the viewing transformation we want to display only those picture parts that are within the window.
- Everything the outside the window is discarded.
- Clipping algorithm can be applied in world coordinate so that the contents of window integer are mapped to device coordinate.
- The clipped against the view port boundary.
- View port clipping can reduced calculation by allowing concatenation of viewing and geometric transformation matrix.
- There are number of clipping algorithm are used.
    I. Point Clipping Algorithm
    II. Line Clipping Algorithm
    III. Polygon Clipping Algorithm
    IV. Text Clipping Algorithm

V. Curve Clipping Algorithm

Clipping individual Point

- Clipping individual point we need two coordinate if X coordinate boundary of clipping rectangle are X_{max} and X_{min}. Y coordinate boundary of clipping rectangle are Y_{max} and Y_{min}
- Following inequalities must be satisfied for point at (X,Y) to be inside the clipping rectangle.

  $X_{min} < X < X_{max}$.

  $Y_{min} < Y < Y_{max}$

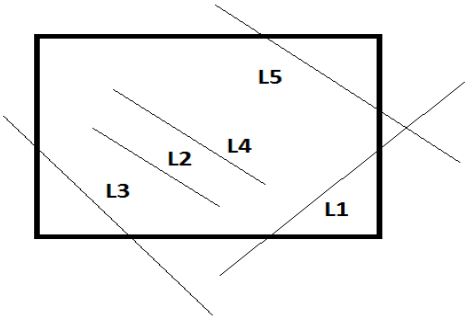
If any of the for inequalities doesn't hold then the point is outside the clipping rectangle

a. $X > X_{min}$
b. $X < X_{max}$
c. $Y > Y_{min}$
d. $Y < Y_{max}$

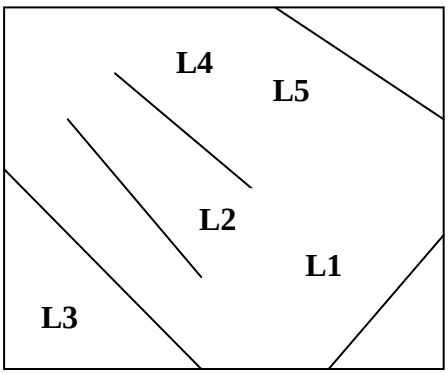
1. Line Clipping

- Line clipping algorithm involves several parts.
- First we can test a given line segment to determine whether the line completely inside the clipping window.
- If it doesn't, we try to determine whether the line completely outside the clipping window.
- Finally it can't identify a line as completely inside and outside, we must perform intersection, calculation with one or more clipping boundary.
- We process the line through the inside outside test by checking the line and point.
- Lines with both end points inside such lines are L_2 &L_4.

Before clipping



After Clipping



- A line with end point outside the boundaries is rejected which is outside the window.
- We use clipping algorithm that can efficiently identify outside line and reduce intersection calculation.

- For a line segments with two end points (X_1,Y_1) and (X_2,Y_2)

  X = X1+U (X2-X1)

  Y = Y1+U (Y2-Y1) $0 \leq U \leq 1$

Cohen Sutherland line clipping algorithm

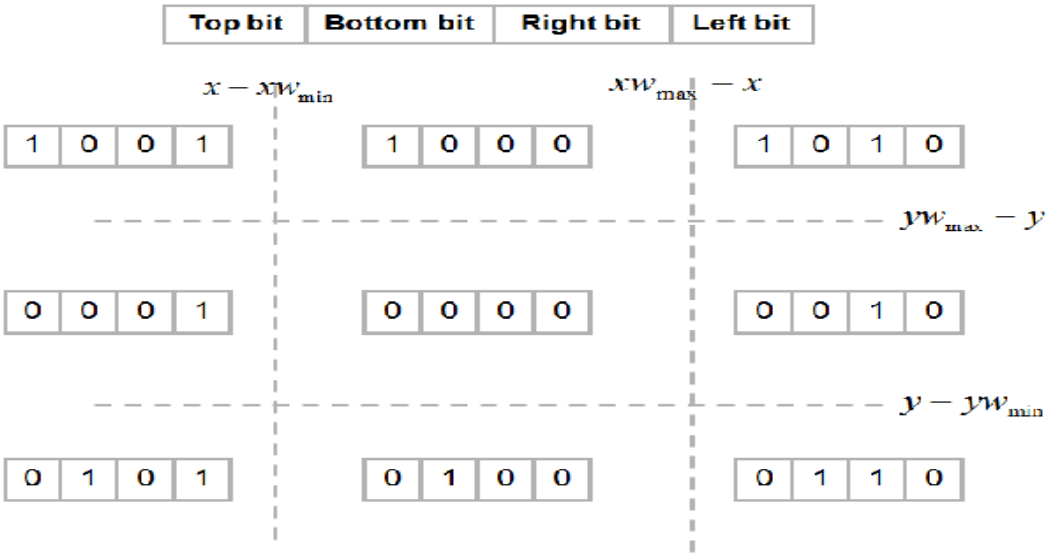
- This algorithm divides a 2D space into 9 parts, of which only the middle part (viewport) is visible.
- These 9 regions can be uniquely identified using a 4 bit code, often called an out code. The 9 regions are: top-left, top-centre, top-right, centre-left, centre, centre-right, bottom-left, bottom-centre, and bottom-right.
- If the beginning coordinate and ending coordinate both fall inside the viewport then the line is automatically drawn in its entirety.
- If both fall in the same region outside the viewport, it is disregarded and not drawn. If a line's coordinates fall in different regions, the line is divided into two, with a new coordinate in the middle.
- Bit1 is the sign bit of $x - xw_{min}$;bit 2 is the sign bit of $xw_{max} - x$;bit 3 is sign bit of $y - yw_{min}$;bit 4 is the sign of $yw_{max} - y$.
- The algorithm is repeated for each section; one will be drawn completely, and the other will need to be divided again, until the line is only one pixel
- The algorithm includes, excludes or partially includes the line based on where the two endpoints are:

  1).Both endpoints are in the viewport (bitwise OR of endpoints == 0000): trivial accept.

  2).Both endpoints are on the same side of the rectangle, which is not visible(bitwise AND of endpoints!= 0000): trivial reject.

  3).Both endpoints are in different parts: In case of this non trivial situation the algorithm finds one of the two points that are outside the viewport (there is at least one point outside). The intersection of the outpoint and extended viewport border is then calculated (i.e. with the parametric equation for the line) and this new point replaces the outpoint. The algorithm repeats until a trivial accept or reject occurs.
- An out code is computed for each of the two points in the line. The first bit is set to 1 if the point is above the viewport.
- The bits in the out code represent: Top, Bottom, Right, Left. For example the out code 1010 represents a point that is top-right of the viewport.

- Note that the out codes for endpoints must be recalculated on each iteration after the clipping occurs.



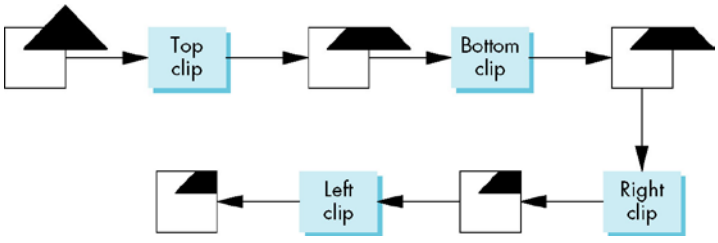
5.5 Polygon clipping

- Clipping polygons is more complex than clipping the individual lines.
- Input: polygon
- Output: polygon, or nothing
- Not as simple as line segment clipping
- Clipping a line segment yields at most one line segment
- Clipping a polygon can yield multiple polygons

However, clipping a convex polygon can yield at most one other polygon.

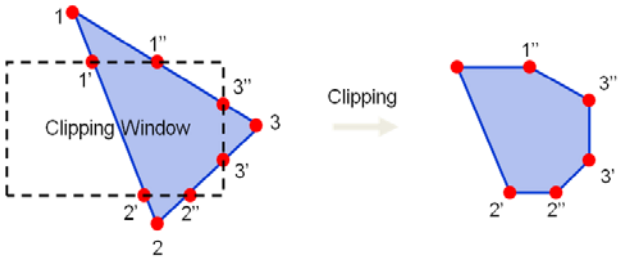
Pipeline Clipping of Polygons

- Three dimensions: add front and back clippers
- Strategy used in SGI Geometry Engine
- Small increase in latency

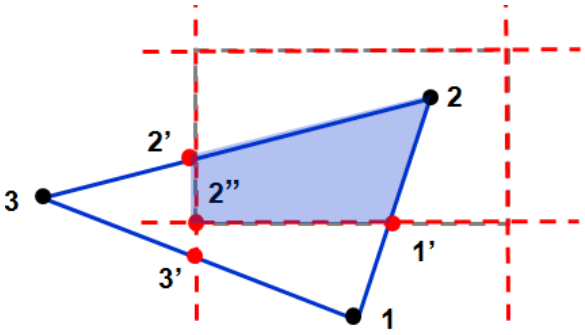


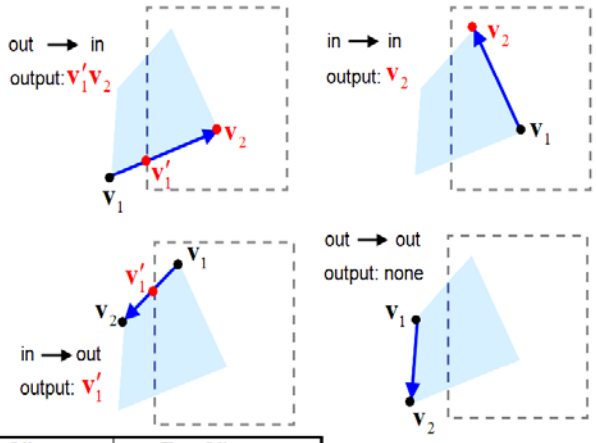
Sutherland-Hodgman Polygon Clipping

- It is efficient algorithm for clipping convex polygons.
- Edges are clipped against every border line of clipping window. Edges are processed successively.
- Allows pipelining of edge clipping of polygons, as well as pipelining of different polygons.



- The four possible outputs generated by the left clipper, depending on the relative position of pair of edge endpoints.





Input	Left Clipper	Right Clipper	Bottom Clipper	Top Clipper
[1,2]:	(in-in)>{2}			
[2,3]:	(in-out)>{2'}	[2,2']:(in-in)>{2'}		
[3,1]:	(out-in)>{3',1}	[2',3']:(in-in)>{3'}	[2',3']:(in-out)>{2''}	
		[3',1]:(in-in)>{1}	[3',1]:(out-out)>{}	
		[1,2]:(in-in)>{2}	[1,2]:(out-in)>{1',2}	[2'',1']:(in-in)>1'}
			[2,2']:(in-in)>{2'}	[1',2]:(in-in)>{2}
				[2,2']:(in-in)>{2'}
				[2',2'']:(in-in)>{2''}

The four clippers can work in parallel.

- Once a pair of endpoints it output by the first clipper, the second clipper can start working.
- The more edges in a polygon, the more effective parallelism is.

- Processing of a new polygon can start once first clipper finished processing.
  - No need to wait for polygon completion.