

MCA Part II
Paper X: Computer Graphics and Multimedia

Topic: Polygon Filling Algorithm

Compiled by: Dr. Kiran Pandey
School of Computer Science
Email-id: kiranpandey.nou@gmail.com

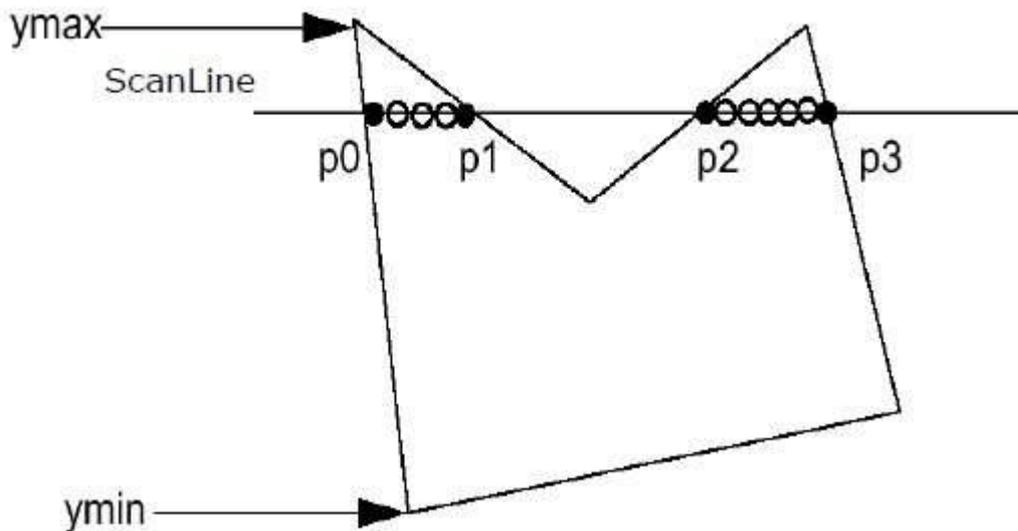
Polygon is an ordered list of vertices as shown in the following figure. For filling polygons with particular colors, you need to determine the pixels falling on the border of the polygon and those which fall inside the polygon. In this chapter, we will see how we can fill polygons using different techniques.



Scan Line Algorithm

This algorithm works by intersecting scanline with polygon edges and fills the polygon between pairs of intersections. The following steps depict how this algorithm works.

Step 1 – Find out the Ymin and Ymax from the given polygon.



Step 2 – ScanLine intersects with each edge of the polygon from Ymin to Ymax. Name each intersection point of the polygon. As per the figure shown above, they are named as p0, p1, p2, p3.

Step 3 – Sort the intersection point in the increasing order of X coordinate i.e. p0,p1, p1,p2, and p2,p3.

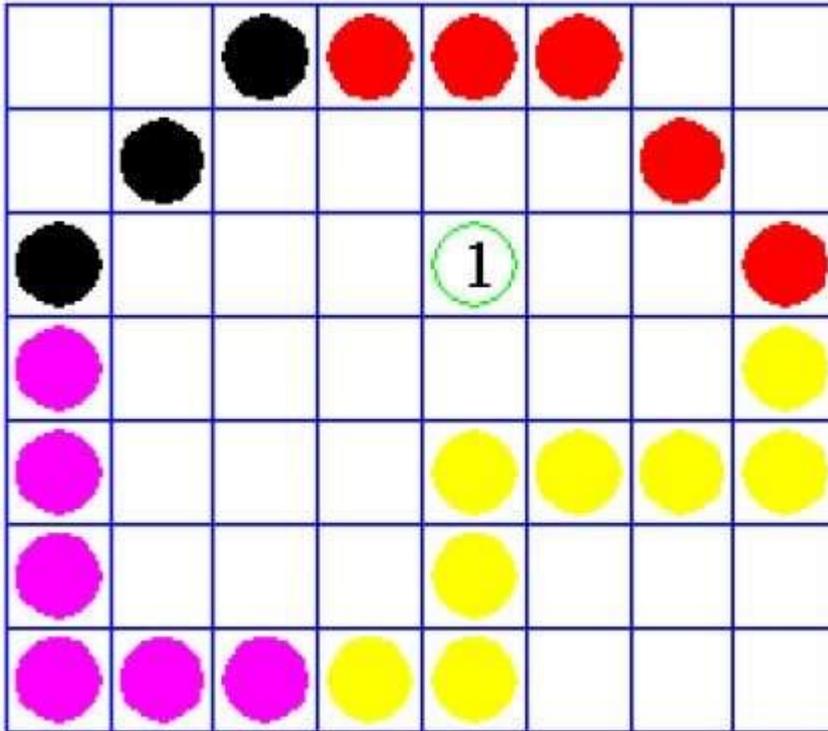
Step 4 – Fill all those pair of coordinates that are inside polygons and ignore the alternate pairs.

Flood Fill Algorithm

Sometimes we come across an object where we want to fill the area and its boundary with different colors. We can paint such objects with a specified interior color instead of searching for particular boundary color as in boundary filling algorithm.

Instead of relying on the boundary of the object, it relies on the fill color. In other words, it replaces the interior color of the object with the fill color. When no more pixels of the original interior color exist, the algorithm is completed.

Once again, this algorithm relies on the Four-connect or Eight-connect method of filling in the pixels. But instead of looking for the boundary color, it is looking for all adjacent pixels that are a part of the interior.



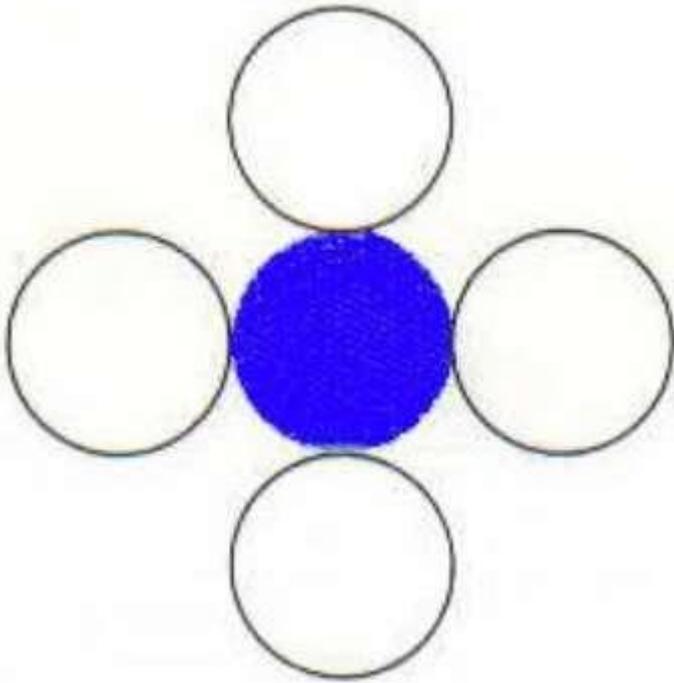
Boundary Fill Algorithm

The boundary fill algorithm works as its name. This algorithm picks a point inside an object and starts to fill until it hits the boundary of the object. The color of the boundary and the color that we fill should be different for this algorithm to work.

In this algorithm, we assume that color of the boundary is same for the entire object. The boundary fill algorithm can be implemented by 4-connected pixels or 8-connected pixels.

4-Connected Polygon

In this technique 4-connected pixels are used as shown in the figure. We are putting the pixels above, below, to the right, and to the left side of the current pixels and this process will continue until we find a boundary with different color.



Algorithm

Step 1 – Initialize the value of seed point $seedx, seedy$, $fcoll$ and $dcol$.

Step 2 – Define the boundary values of the polygon.

Step 3 – Check if the current seed point is of default color, then repeat the steps 4 and 5 till the boundary pixels reached.

If $getpixel(x, y) = dcol$ then repeat step 4 and 5

Step 4 – Change the default color with the fill color at the seed point.

$setPixel(seedx, seedy, fcol)$

Step 5 – Recursively follow the procedure with four neighborhood points.

$FloodFill(seedx - 1, seedy, fcol, dcol)$

$FloodFill(seedx + 1, seedy, fcol, dcol)$

$FloodFill(seedx, seedy - 1, fcol, dcol)$

$FloodFill(seedx - 1, seedy + 1, fcol, dcol)$

Step 6 – Exit

There is a problem with this technique. Consider the case as shown below where we tried to fill the entire region. Here, the image is filled only partially. In such cases, 4-connected pixels technique cannot be used.