https://dx.doi.org/10.22161/ijaers.4.2.15 ISSN: 2349-6495(P) | 2456-1908(O)

A Review: Polygon Filling Algorithms Using Inside-Outside Test

Nisha, Sapna Varshney

Assistant Professor, Department of computer science, University of Delhi, India

Abstract— *In computer graphics we have many polygon filling algorithms. In this paper I review scan line polygon filling algorithms using inside-outside test, boundary fill algorithm and flood fill algorithm. Scan line filling algorithm, finds an intersection of the scan line with polygon edges and inside-outside test is used to find the inside and outside region of a polygon. Boundary fill is a recursive algorithm. This paper has a review on various polygon filling algorithms.*

Keywords— Polygon filling, scan line, boundary fill, computer graphics.

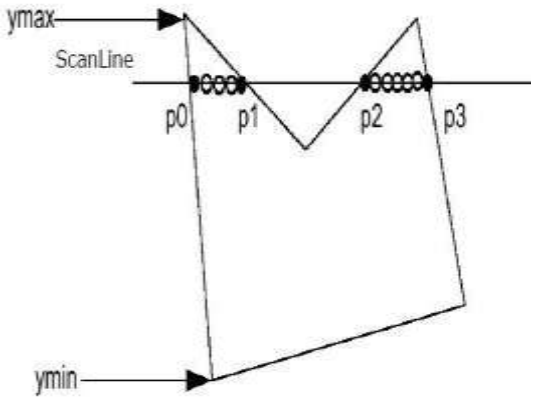
I. INTRODUCTION

In computer graphics image are created using computers with specialized graphics hardware and software. Images are composed of basic structures like points, lines, curves, polygons, etc. A polygon is a closed image. Polygons are classified as convex and concave. Convex polygons are those for which the line joining any two points within the polygon also lies completely within the polygon. In computer graphics, vector method and rotational method are used to determine whether the polygon is concave or convex. If a polygon is concave, these methods split such polygons into a pair of convex polygons. Polygon filling is the process of coloring the pixels of a polygon. In computer graphics there are many polygon fill algorithms Scan Line Polygon Fill, Seed Fill and Edge Fill. Such algorithms can be applied to both concave and convex polygons but is generally more complicated for concave polygons. In computer graphics we have many polygon fill algorithms. For filling polygons with particular colors, we need to determine the pixels falling on the border of the polygon and those which fall inside the polygon, for identifying polygon interior pixels inside-outside test is used. [1]

II. SCAN LINE FILLING ALGORITHM

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

a. Find out the Ymin and Ymax from the given polygon.



b. Scan Line 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.

c. Sort the intersection point in the increasing order of X coordinate i.e. (p0, p1), (p1, p2), and (p2, p3).

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

Originally parity is odd, when scan line first intersects the polygon edge at that time parity becomes odd and this algorithm starts filling the polygon. But after second intersection parity becomes even and algorithm stops filling the polygon, likewise this algorithm fills whole polygon. [2]

Inside- Outside Test

This method is also known as counting number method. While filling an object, we often need to identify whether particular point is inside the object or outside it. There are two methods by which we can identify whether particular point is inside an object or outside.

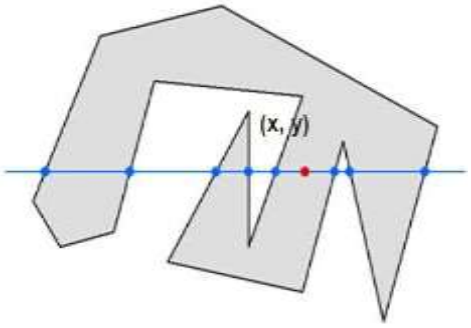
- Odd-Even Rule
- Nonzero winding number rule

Odd-Even Rule

In this technique, we will count the edge crossing along the line from any point (x,y) to infinity. If the number of interactions is odd, then the point (x,y) is an interior point; and if the number of interactions is even, then the point

https://dx.doi.org/10.22161/ijaers.4.2.15 *ISSN: 2349-6495(P) | 2456-1908(O)*

(x,y) is an exterior point. The following example depicts this concept.



From the above figure, we can see that from the point (x,y), the number of interactions point on the left side is 5 and on the right side is 3. From both ends, the number of interaction points is odd, so the point is considered within the object. [2]

Nonzero Winding Number Rule

This method is also used with the simple polygons to test the given point is interior or not. It can be simply understood with the help of a pin and a rubber band. Fix up the pin on one of the edge of the polygon and tie-up the rubber band in it and then stretch the rubber band along the edges of the polygon.

When all the edges of the polygon are covered by the rubber band, check out the pin which has been fixed up at the point to be test. If we find at least one wind at the point consider it within the polygon, else we can say that the point is not inside the polygon. [2]

In another alternative method, give directions to all the edges of the polygon. Draw a scan line from the point to be test towards the left most of X direction.

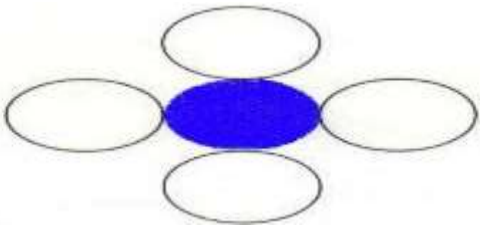
- Give the value 1 to all the edges which are going to upward direction and all other -1 as direction values.
- Check the edge direction values from which the scan line is passing and sum up them.
- If the total sum of this direction value is non-zero, then this point to be tested is an **interior point,** otherwise it is an **exterior point**.
- In the above figure, we sum up the direction values from which the scan line is passing then the total is 1 – 1 + 1 = 1; which is non-zero. So the point is said to be an interior point.

III. BOUNDARY FILL ALGORITHM

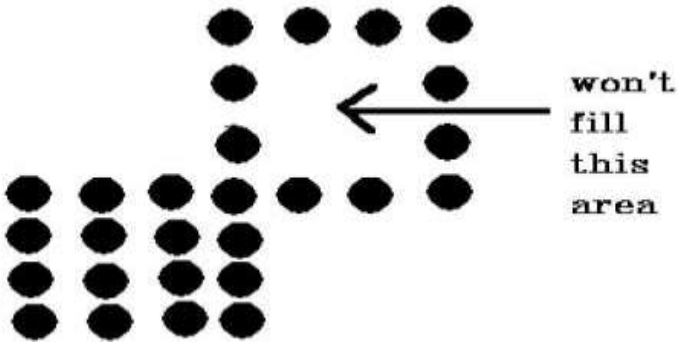
The boundary fill algorithm works as its name. This algorithm picks a point inside an object and starts to fill until it hits the boundary of the object. The color of the boundary and the color that we fill should be different for this algorithm to work. In this algorithm, we assume that color of the boundary is same for the entire object. The boundary fill algorithm can be implemented by 4-connected pixels or 8-connected pixels. [3]

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.



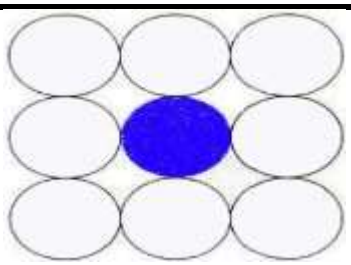
There is a problem with this technique. Consider the case as shown below where we tried to fill the entire region. Here, the image is filled only partially. In such cases, 4-connected pixels technique cannot be used.



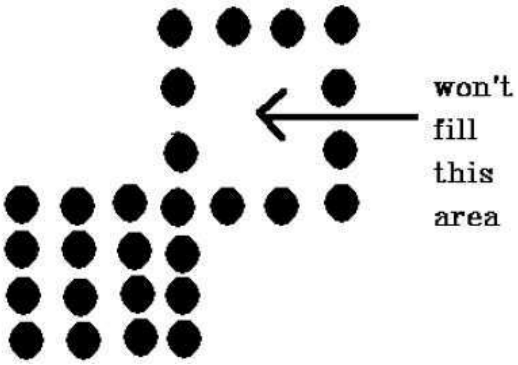
8-Connected Polygon

In this technique 8-connected pixels are used as shown in the figure. We are putting pixels above, below, right and left side of the current pixels as we were doing in 4-connected technique. In addition to this, we are also putting pixels in diagonals so that entire area of the current pixel is covered. This process will continue until we find a boundary with different color. [3]

https://dx.doi.org/10.22161/ijaers.4.2.15 *ISSN: 2349-6495(P) | 2456-1908(O)*

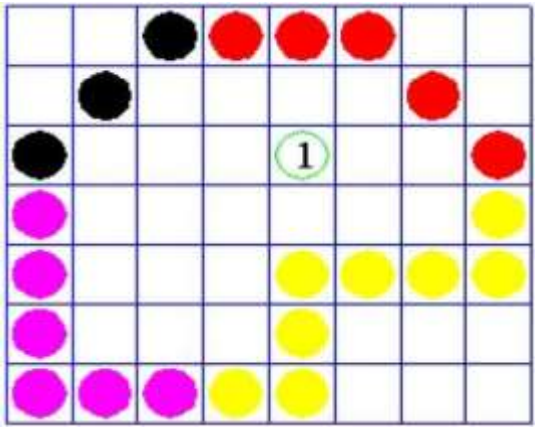


The 4-connected pixel technique failed to fill the area as marked in the following figure which won't happen with the 8-connected technique.



IV. 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. [2]



V. CONCLUSION

In this paper I reviewed different polygon filling algorithms. Boundary fill and flood fill algorithms have some disadvantage like if an inside pixel is in some other color then the fill terminates and the polygon remains unfilled. Boundary fill and flood fill method does not work for large polygons. Whereas scan line fill algorithm does not have these disadvantages. In scan line algorithm rightmost and the left most pixels of the seed pixel are marked and a horizontal line is drawn between them. The process is continued by locating more seed pixels upwards and downwards.

REFERENCES

[1] S.Anitha, D.Evangeline, "An Efficient Fence Fill Algorithm using Inside-Outside Test", International Journal of Advanced Research in Computer Science and Software Engineering Volume 3, Issue 11, November 2013.

[2] https://www.tutorialspoint.com/computer_graphics/polygon_filling_algorithm.htm

[3] Donald Hearn, and M. Pauline Baker, "Computer Graphics, C Version", 3 edition,, December 2004

[4] Rogers DF. "Procedural elements for computer graphics". New York: McGraw-Hill, 1985.

[5] https://www.techfak.unibielefeld.de/ags/wbski/lehre/digiSA/WS0607/3DVRCG/Vorlesung/13.RT3DCGVR-vertex-2-fragment.pdf