

---

## ***Comparative Study of Region Filling Approaches in Computer Graphics***

***Thi Thi Soe\*, Zarni Sann\*\*, Zar Zar Hnin\****

*Faculty*

*Department of Computer Science\*, Department of Computer Systems and Technologies\*\**

*University of Computer Studies (Mandalay), Myanmar*

***Corresponding author's email id: hlaing.nn@gmail.com***

### ***Abstract***

*In computer graphics, region filling is a frontend operation to produce the realistic display scene on raster systems. There are two categories of approaches commonly used in filling the interior part of object namely scan-line approach and seed-fill approach. In this paper we study on the polygon area filling approaches among the various area primitives such as circles, ellipses, and other simple curves. The prime goal of this paper is to realize the fundamental aspects of area filling approaches in computer graphics and the second one is to provide the knowledge obtained from the study area in creating the pictures of entertainments, advertising, and medical application areas.*

***Keywords:*** *region filling, scan-line method, seed-fill method, computer graphics, raster system*

### **INTRODUCTION**

We are creating pictures in the world of scenes. There are various objects in a created scene such as building, tree, sky, person, and terrain. Some objects need to paint with specified color or pattern in their interior positions. Two fundamental approaches have been available in graphic

packages: scan-line method determining overlap interval of scan line and fill-methods starting from seed points [1]. In this paper, we consider for filling polygon area of several kinds of area primitives. Polygons have linear boundary in which boundary not intersect each other called standard polygon and having intersecting

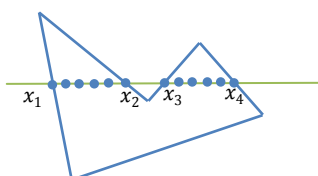
boundary called complex polygon. Besides some objects are bounded by single color and some are bounded by multiple colors. Following sections describe how to fill the interior part of these types of polygon by exploring the existing work. Region filling with patterned are outside of study area.

### SCAN-LINE APPROACH

For filling the area with solid color, scan-line method performs the following steps [1]:

- First locate the intersection pixel position of the scan line that cross the polygon boundary
- Sort these points in ascending order with respect to the  $x$  coordinates
- Paint with the specified color between each intersection pair

In fig. 1, four intersection points are  $x_1, x_2, x_3,$  and  $x_4$  and two intersection pairs are  $(x_1 \rightarrow x_2)$  and  $(x_3 \rightarrow x_4)$ .



*Fig.1 Interior Pixel Spans within Intersection Pairs*

Scan line approach is used to fill polygons, circles, ellipse, and other simple curves [1].

### *Problem with Vertices*

The next consideration for scan conversion algorithm to area filling is vertex-intersection problem. Some scan lines that intersect at the polygon vertex require special handling to ensure the vertex count. This can be accomplished by traversing around the edge clockwise/ counterclockwise direction. Some polygon edges splitting the vertex need to be shortened and so counted as single intersection [1].

Fig. 2 demonstrates the shortening of some edges. In fig. 2(a), endpoint  $y$  coordinates of the two edges is moving to upward direction. In this case,  $y$  value of the upper endpoint of the current edge is decreased by one. Endpoint  $y$  coordinates of the two edges is going downward direction in fig. 2(b). Therefore,  $y$  value of the upper endpoint of the next edge is decreased by one. In fig. 2(c) and fig. 2(d), two edges, they exist on the same side of the scan line and change their directions. So they have double intersection and there is no need to perform special handling for these vertices

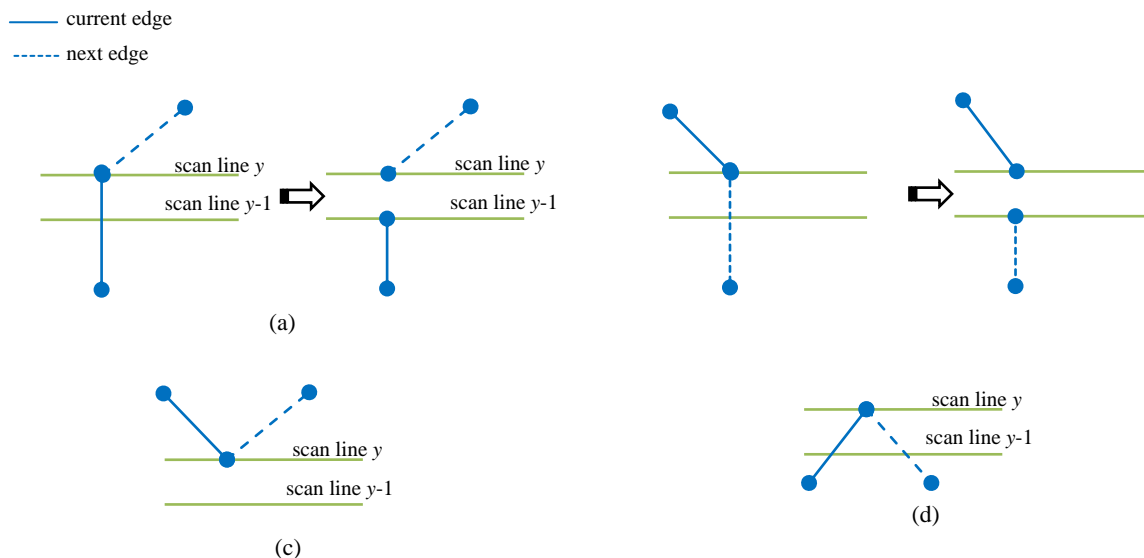


Figure:- 2 Illustrating the Special Handling for

### Coherence

To reduce processing performed in scan-conversion, many enhancements must be added to the basic scheme. One of these is to take the advantage of coherence properties of the object. Coherence means that the properties of one part of the object are related with the other in a way [1, 2]. This method includes incremental calculations along a scan line. In fig. 3, the value of the coordinate  $x_{k+1}$  on scan line  $y_{k+1}$  can be obtained from the  $x_k$  on scan line  $y_k$  by using the equation  $x_{k+1} = x_k + \frac{1}{m}$ .

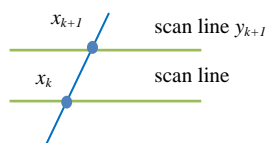


Fig. 3 Intersection Points on Successive Scan line

### Edge Table and Edge List

In order to fill the region efficiently by applying the characteristics mentioned above, it will need to initially set up a sorted edge table in which polygon edges are stored [ 1, 3].

#### Sorted Edge Table

- Make a clockwise/counterclockwise traversal around the edge
  - Start from smallest scan line number
  - For each scan line, edges are sorted in left to right order
  - Only non-horizontal edges are stored and also shorten the certain edge
- Linked list for edge entry is formed as

maximum y value	x - value of lower node	inverse slope	pointer to the next edge
-----------------	-------------------------	---------------	--------------------------

Fig. 4 Data Structure to Store Edges

In the following polygon, it needs to shorten the edge AB.

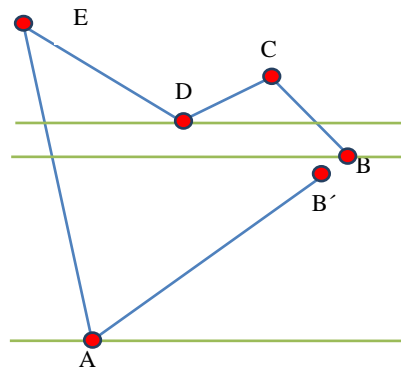


Fig. 5 A Polygon

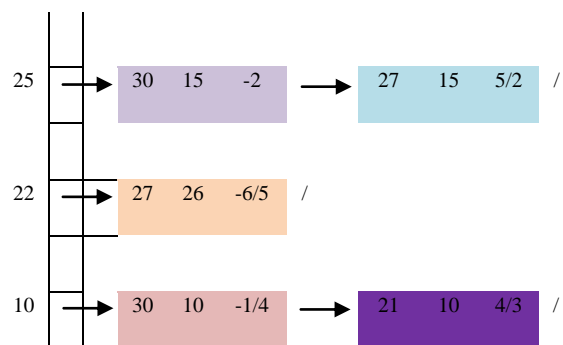
In second phase of producing active edge list, scan lines are processed from below to top [1, 3].

**Edge List**

- Contains all edges that are crossed by scan line
- Update the current x value of the edge based on m value
- Add the lists in the current table entry based on their x value
- Remove the completed edges

- Fill the specified color in the pixel spans within x intersection pairs

Edge table and edge list of scan-line method for filling a polygon with the vertices (10,10), (26,22), (15,25), (20,27), and (5,30), are described below. For clarity only x intersection points are described in edge list as in [3].



<i>y value</i>	<b>x intersection values</b>			
30	5	5		
29	5.25	7		
28	5.5	9		
27	5.75	11	20	20
26	6	13	17.5	21.2
25	6.25	15	15	22.4
24	6.5	23.6		
23	6.75	24.8		
22	7	26		
21	7.25	24.63		
20	7.5	23.3		
19	7.75	21.97		
18	8	20.64		
17	8.25	19.31		
16	8.5	17.98		
15	8.75	16.65		
14	9	15.32		
13	9.25	13.99		
12	9.5	12.66		
11	9.75	11.33		
10	10	10		

**Two Rules for Inside-Outside Region**

Triangles, rectangles, octagons, and decagons are standard polygons. For identifying the interior points of standard edges intersection. For such self-intersecting polygons, it is difficult to distinguish which points are interior of the object or not. Two rules could be identified interior region of the object [1].

**Odd-Even Rule** performs the following steps to distinguish the interior and exterior point:

- Draw a line from any point  $p$  to outside of object
- Count the number of edge crossing the line

Point  $p$  is  $\begin{cases} \text{interior} & \text{if edge count is odd} \\ \text{exterior} & \text{otherwise} \end{cases}$

**Nonzero-Winding Number Rule** process the below steps:

Initialize the winding number,  $w=0$

Draw a line from any point  $p$  to outside part not passing any vertices

Count the number of edge crossing the line

**Nonzero-Winding Number Rule** process the below steps:

- Initialize the winding number,  $w = 0$

polygons are simple processes because of edges of these objects having no self-intersection. Sometime in our graphics applications, some polygons having the

- Draw a line from any point  $p$  to outside part not passing any vertices
- Count the number of edge crossing the line

If summing of winder number,  $w$  is nonzero, point  $p$  is defined as interior point

Scheme for Directional Edge Crossing

Direction of edge crossing in nonzero-winding number rule could be determined by one of two ways: vector cross product or vector dot product [1].

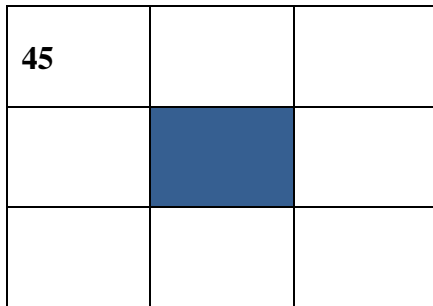
**SEED-FILLED APPROACH**

In filling the area with specified color or pattern, an approach starts at a given interior position (seed point). This approach so called seed-filled approach.

**Boundary Fill Algorithm**

Boundary fill is a kind of seed-fill approaches. Polygon area is bounded by single color and filling is accomplished based on the neighbor pixels' color condition [1]. In fig. 6, 4-neighbors of center pixel have two vertical and two horizontal neighbors. 4- neighbors and

diagonal neighbors together are called 8-neighbors of center pixel.



**Fig. 6 A Block with Center Pixel**

**Boundary fill algorithm works as follows:**

- Define starting interior position and filled color value
- Paint all neighbors (4- or 8- neighbors) which are not painted and are not boundary color points

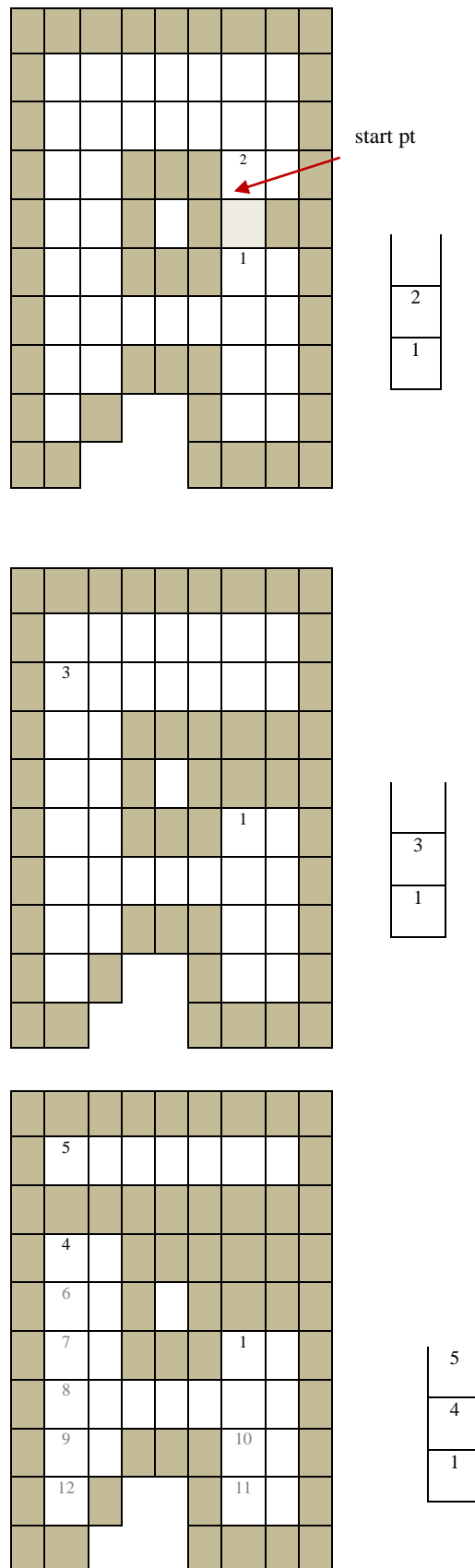
Although boundary fill algorithm with 4-neighbors unable to fill the interior parts of complex object completely, 8-neighbors based boundary fill algorithm produced entirely filled resultant region.

### **Horizontal Spans Filling**

To save the storage, boundary fill algorithm utilized horizontal pixel spans instead of stacking the neighbor points. In horizontal pixel spans filling, the algorithm only stacks a starting point of each horizontal pixel spans [1].

- Define starting interior position
- Fill the successive spans on the scan line
- Find out the leftmost starting points on below and above adjacent scan line
- Stack these points
- Unstack for next start position of horizontal spans filling
- Repeat the process until the stack is empty

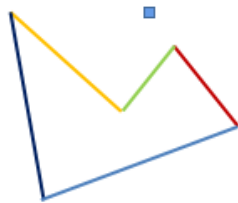
Fig. 7 illustrates the horizontal pixel spans filling approach. Some steps of filled spans and stacked position are shown on the left side and right side of the figure respectively.



**Fig. 7 Horizontal Pixel Spans Filling**

### ***Flood Fill Algorithm***

Flood fill algorithm is a good option for filling area with multiple color boundary, as in fig.8. It is also a seed-fill approach. An algorithm determines the filled area by matching with old colors instead of searching boundary color value. This algorithm also employed horizontal spans filling as in boundary fill algorithm.



***Figure:-8 A Poly with Multiple color Boundary***

Filled methods starting from an interior seed point are useful with more complex boundary and in interactive painting programs to fill connected similar color areas with other color. Methods are used in games such as Go and Minesweeper for determining which pieces are cleared [1, 4].

### **CONCLUSION**

This paper studied the region filling approaches: scan line approach and seed filled approach which are applied as initial steps in creating realistic scene on raster display. In scan line based approach, standard polygon filling is a simple

process and two rules are identified for determining the interior points in complex polygon. In seed fill based approach, boundary fill is suitable for filling region with single color boundary and flood fill is good choice to area filling with multiple color boundary. Horizontal pixel spans filling are used to reduce the utilization of storage space in both seed fill based approaches.

### **REFERENCES**

- I. D. Hearn, and M. Pauline Baker, "Computer Graphics, C Version (2nd Edition), May 1996, ISBN 0135309247 (ISBN13: 9780135309247), Prentice Hall.
- II. J. M. Lane, L. C. Carpenter, T. Whitted, and J. F. Blinn, "Scan Line Methods for Displaying Parametrically Defined Surfaces", Communications of the ACM, Volume 23, Number 1, January 1980, pp. 23-34.
- III. [http://ocw.metu.edu.tr/pluginfile.php/1103/mod-resource/content/0/documents/lecturenotes-2007/week3\\_filledAreaPrimitives.pdf](http://ocw.metu.edu.tr/pluginfile.php/1103/mod-resource/content/0/documents/lecturenotes-2007/week3_filledAreaPrimitives.pdf)
- IV. [https://en.m.wikipedia.org/wiki/Flood\\_fill](https://en.m.wikipedia.org/wiki/Flood_fill)