Binghamton University

EngiNet™

State University of New York

Thomas J. Watson

School of Engineering
and Applied Science

EngiNet™

WARNING

CS 560

Computer Graphics

Professor Richard Eckert

Lecture # 10

February 21, 2001

Area Fill Algorithms

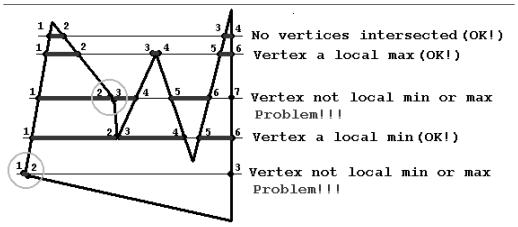
- Boundary/Flood Fill
- Scanline Polygon Fill
- Scanline Boundary Fill
- Pattern Fill

Scanline Polygon Fill Algorithm

- Look at individual scan lines
- Compute intersection points with polygon edges
- Fill between alternate pairs of intersection points

More specifically:

- For each scanline spanning the polygon:
  - Find int. pts. with all edges scanline cuts
  - Sort intersection points by increasing x
  - Turn on all pixels between alternate pairs of intersection points
- But--
  - Look at intersection points that are polygon vertices



Dealing With Vertex Intersection Points

Vertex intersection points that are not local max or min must be preprocessed!

Preprocessing non-max/min intersection points

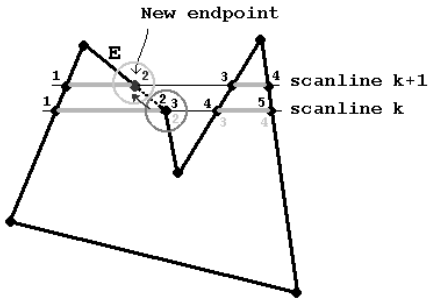
- Move lower endpoint of upper edge up by one pixel
- i.e., y <-- y + 1
- What about x?

  m = $\Delta y/\Delta x$, so $\Delta x = (1/m) * \Delta y$

  But $\Delta y = 1$, so:

  x <-- x + 1/m

Preprocessing



Preprocessing Edge E

Active Edge

- A polygon edge intersected by the current scanline
- As polygon is scanned, edges will become active and inactive.
- Criterion for activating an edge:
  ysl = ymin of the edge
  (Here ysl = y of current scanline)
- Criterion for deactivating an edge:
- ysl = ymax of the edge

Vertical & Horizontal Coherence

- Moving from one scanline to next:
- y = y + 1
- If edge remains active, new intersection point coordinates will be:
  ynew = yold + 1
  xnew = xold + 1/m
  (1/m = inverse slope of edge)

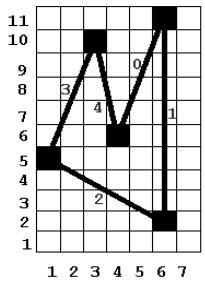
Scanline Polygon Fill Algorithm Input

- List of polygon vertices (xi,yi)

Scanline Polygon Fill Algorithm Data Structures

1. Edge table:
   - For each edge: edge #, ymin, ymax, x, 1/m
2. Activation Table:
   - (y, edge number activated at y)
     - Provides edge(s) activated for each new scanline
     - Constructed by doing a "bin" or "bucket" sort
3. Active Edge List (AEL):
   - Active edge numbers sorted on x
     - A dynamic data structure

Bin Sort for Activation Table



Edge Table

e	ymin	ymax
0	6	11
1	2	11
2	2	5
3	5	10
4	6	10

Activation Table

y	activated edge
2	1 2
3	2
4	
5	3
6	0 4

0

Scanline Polygon Fill Algorithm

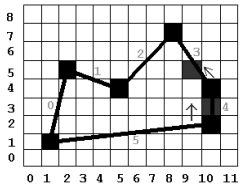
1. Set up edge table from vertex list; determine range of scanlines spanning polygon (miny, maxy)
2. Preprocess edges with nonlocal max/min endpoints
3. Set up activation table (bin sort)
4. For each scanline spanned by polygon:
   - Add new active edges to AEL using activation table
   - Sort active edge list on x
   - Fill between alternate pairs of points (x,y) in order of sorted active edges
   - For each edge e in active edge list:

     If (y != ymax[e]) Compute & store new x (x+=1/m)

     Else Delete edge e from the active edge list

Scanline Polygon Fill Algorithm Example

poly={1,1, 2,5, 5,4, 8,7, 10,4, 10,2, 1,1}



edge	x1	y1	x2	y2	sgn(Dy)	
0	1	1	2	5	+	
1	2	5	5	4	−	
2	5	4	8	7	+	
(3)	8	7	10	4	−	←
(4)	10	4	10	2	−	←
5	10	2	1	1	−	
0	1	1	2	5	+	

Activation Table

y	1	2	3	4	5	6	7
activated edge #s	0 5			4 1 2	3		

Edge Table

edge	1/m	ymin	x	ymax
0	1/4	1	1	5
1	-3	4	5	5
2	1	4	5	7
3	-2/3	4→5	10→9 1/3	7
4	0	2→3	10→10	4
5	9	1	1	2

Scanline Poly Fill Alg. (with example Data)

Edge Table (As Algorithm Executes)

Edge	1/m	ymax	ymin	x
0	1/4	5	1	1, 1.25, 1.5, 1.75, 2
1	-3	5	4	5, 2
2	1	7	4	5, 6, 7, 8
3	-2/3	7	5	9.33, 8.67, 8
4	0	4	3	10, 10
5	9	2	1	1, 10

Active Edge List (As it develops)

y	1	2	3	4	5	6	7
Active Edges	0,5	0,5	0,4	0,1,2,4	0,1,2,3	2,3	2,3
Fill between	1-1	1-10	2-10	2-5,5-10	2-2,6-9	7-9	8-8

Video of BALSA Scanline Poly Fill Algorithm Animator

- Brown University ALgorithm Simulator and Animator
- Mark Brown and Bob Sedgewick
- Scanline Fill Algorithms can be fast if sorting is done efficiently

Demo of Scanline Polygon Fill Algorithm vs. Boundary Fill Algorithm

Adapting Scanline Polygon Fill to other primitives

- Example: a circle or an ellipse
  - Use midpoint algorithm to obtain intersection points with the next scanline
  - Draw horizontal lines between intersection points
  - Only need to traverse part of the circle or ellipse

Scanline Circle Fill Algorithm

```
Modify midpoint circle algorithm
For each step draw 4 horizontal lines

Line4(x,y,h,k)
{
 Line(-x+h,y+k,x+h,y+k);    // 1
 Line(-x+h,-y+k,x+h,-y+k);  // 2
 Line(-y+h,x+k,y+h,x+k);    // 3
 Line(-y+h,-x+k,y+h,-x+k);  // 4
}
```