Scan-line and Flood-fill Algorithms

CS116A
Chris Pollett
Introduction

- Scan-line Algorithm
- Scan-line for Convex Polygons
- Scan-line for Curved Boundaries
- Methods for Irregular Boundaries
  - Boundary-fill
  - Flood-fill
- OpenGL Fill-area Attributes
Scan-line Algorithm

- Basic idea: Figure out where the scan line crosses a line of the figure and use even odd rule to determine if inside-outside figure. If inside fill.
- Determining where a scan line intersects an edge involves solving a pair of simultaneous linear equations where one of the two lines of the form \( y = \text{constant} \).
Scan-line Fill Diagrams

One would continue to fill in line for rest of the interiors.
Potential Headaches

- Where two edges meet:

  ![Diagram of two cases]

- Left case can count edges crossed at vertex as 2, but in right case should count as 1
Solution

• For right case shorten by one pixel one of the two edges.

becomes

Scan lines
Making the Algorithm Incremental

• We don’t want to solve linear equations for each scan line.
• To incrementally calculate intersection use slope:
  \[ m = \frac{y_{k+1} - y_k}{x_{k+1} - x_k} \]
  – In going from one scan line to the next \( y_k \) changes by one, \( x_{k+1} = x_k + 1/m \).
  – If do things in parallel, and start position of a processor was \( x_k \) could calculate as \( x_k = x_0 + k/m \).
  – Assuming slope is rational \( 1/m = \Delta x/\Delta y \). Have a counter for change in \( x \) -- step by \( \Delta x \) when this gets bigger than \( \Delta y \) decrement it and move are \( x \) loc over by 1.
Using a Sorted-edge Table

- Still need to deal with start and end positions of edges. Can keep an active edge list sorted by starting scan-line

![Diagram with points A and B and an active edge list sorted by scan-line.](image-url)
Scan-line Fill of Convex Polygons

• Only need to find two boundary for each scan-line

• If a scan line intersects a single vertex only plot that point
Scan-line Fill for Regions with Curved Boundaries

• Boundary intersections now must be calculated using the curves equations
• For some curves this isn’t too hard. Ex. Conics.
• In which case can adapt the previously described algorithm and come up with incremental variants.
• Can also fill sections of curves using this technique (A section is area between a curve and a line) 

• For complicated curves hard to compute intersection and other methods must be used
Fill Methods for Areas With Irregular Boundaries

- Basic idea for these techniques is to start at some position interior to fill
- Fill outward to boundary
Boundary Fill

- Boundary is assumed to be one fixed color
- Fill towards this boundary using either:
  - 4-connected
  - 8-connected fill pattern
- Algorithm can be implemented either recursively or using a stack of places to be filled
Flood Fill

• Similar to boundary fill but now boundary is given by any colors other than the color of pixel started fill on.
• Implementation is similar.
OpenGL Fill area attribute Functions

Four steps to filling a convex polygon in OpenGL:

1. Define a fill pattern
2. Invoke the polygon-fill routine
3. Activate the polygon-fill feature of OpenGL
4. Describe the polygons to be filled.
Define a Fill Pattern

Glubyte fillPattern[] = {0xff, 0x00, …};
/* masks are 32 by 32 bits starting at bottom row */

//To use
glPolygonStipple(fillPattern);

//To enable
glEnable(GL_POLYGON_STIPPLE);

//To disable
glDisable(GL_POLYGON_STIPPLE);

//Start position of tiling bottom left of screen
OpenGL Texture and Interpolation Patterns

Another method to specify the kind of fill is to use a texture or interpolation patterns. We’ll talk about the former much later. For the latter can do:

```c
glShadeModel(GL_SMOOTH);

glBegin(GL_TRIANGLES);
  glColor3f(0.0, 0.0, 1.0);
  glVertex2i(50, 50);
  glColor3f(1.0, 0.0, 0.0);
  glVertex2i(150, 50);
  glColor3f(0.0, 1.0, 0.0);
  glVertex2i(75, 150);

glEnd();
```