Chapter 8
Scan Conversion (part 2)– Drawing Polygons on Raster Display

Polygon Rasterization

- Assumptions: well-behaved
  - simple
  - no self intersections
  - simple connected
  - concave
- Solutions
  - Flood Fill
  - Scan Line

Formulation

- Input
  - polygon P with rasterized edges
  - Problem: Fill its interior with specified color on graphics display

Flood Fill Algorithm

- Input
  - polygon P with rasterized edges
  - Find a point inside P

Flood Fill - Drawbacks

- How do we find a point inside?
- Pixels visited up to 4 times to check if already set
- Need per-pixel flag indicating if set already

Scanline Algorithm

- Observation: Each intersection of straight line with boundary moves it forward in polygon
- Detect (6 set) pixels inside polygon boundary (simple closed curve) with set of horizontal lines (pixel apart)

Scannline

- Limit to bounding box to speed up
- Other enhancements...
Scan Conversion - Polygons

Example: draw pixels on left or top edge, but not on

Don't draw - BUT gaps possible between triangles

Note – in literature you can find same

Setup: compute edge equations, bounding

box

Ax1 + By1 + C = 0
Ax2 + By2 + C = 0
" Two equations, three unknowns
" Express A, B in terms of C

Computing A, B, C from (x1, y1), (x2, y2)

A = y2 - y1
B = x1 - x2
C = x1y2 - x2y1

Given P0, P1, P2, what are our three edges?

Half-spaces defined by the edge equations must share the same sign on the interior of the triangle.

Consistency (E1: P0-E0, P1-E1, P2-E2)

How do we make sure that sign is positive?

Test & flip if needed (A+, B+, C-)

Computing Edge Equations

Choose C = x1y2 - x2y1 for convenience

Then A = y1 - y2 and B = x2 - x1

Our original implicit formula

Note – in literature you can find same equation multiplied by -1

Changes sides

Computing Edge Equations

Compute A, B, C from (x, y, z)

A = yz - zy
B = zx - xz
C = xy - yx

Choose C = x1y2 - x2y1 for convenience

Then A = y1 - y2 and B = x2 - x1

Our original implicit formula

Compute edge equations, bounding box

for (y = yMin; y <= yMax; y++)

for (x = xMin; x <= xMax; x++)

evaluate edge equations

Set pixel (x, y)

xL += DxL;
xR += DxR;

Scan Conversion Issues

Exactly which pixels should be lit?

- Pixels inside triangle edges

- What about pixels exactly on the edge?

- Draw - BUT order of triangles matters (it shouldn't)

- Don't draw - BUT gaps possible between triangles

- Need consistent (if arbitrary) rule

- Example: draw pixels on left or top edge, but not on right or bottom edge

Triangle Rasterization Issues

Moving Silvers

Shared Edge Ordering

Interpolate - access triangle interior

Interpolate between vertices:

f = \frac{f_1}{A_1} + \frac{f_2}{A_2}

- g, b - colour components

- x, y - texture coordinates

- N_i - surface normals

- Equivalent:

- Spherical interpolation

- Barycentric coordinates

Barycentric Coordinates

Area

Barycentric coordinates

\alpha_i = A_i / A

\alpha_i = A_i / \sum A_i

P = \alpha_i P_i + \alpha_j P_j + \alpha_k P_k

\alpha_i \leq 1
Barycentric Coordinates

Weighted combination of vertices

\[ P = a_1P_1 + a_2P_2 + a_3P_3 \]
with

\[ a_1 + a_2 + a_3 = 1 \]
\[ 0 \leq a_1, a_2, a_3 \leq 1 \]

\[ P = (a_1, a_2, a_3) \]

Bilinear Interpolation

To verify barycentric properties:

\[ a_1 + a_2 = 1 \]
\[ 0 \leq a_1, a_2 \leq 1 \]

Computing Barycentric Coords

- For point \( P' \) on scanline:

\[ P'_i = P_i + \frac{d_i}{d_{P_i}}(P_{i+1} - P_i) \]
\[ = (+) \frac{d_i}{d_{P_i}} \]
\[ = - \frac{d_i}{d_{P_i}} \]
\[ = + \frac{d_i}{d_{P_i}} \]

Similarly:

\[ P'_i = P_i + \frac{b_i}{b_{P_i}}(P_{i+1} - P_i) \]
\[ = (+) \frac{b_i}{b_{P_i}} \]
\[ = - \frac{b_i}{b_{P_i}} \]
\[ = + \frac{b_i}{b_{P_i}} \]

Combining

\[ P'_i = P_i + \frac{d_i}{d_{P_i}}(P_{i+1} - P_i) \]
\[ = \frac{d_i}{d_{P_i}} \]
\[ = - \frac{d_i}{d_{P_i}} \]
\[ = + \frac{d_i}{d_{P_i}} \]

Gives

\[ P'_i = P_i + \frac{b_i}{b_{P_i}}(P_{i+1} - P_i) \]
\[ = \frac{b_i}{b_{P_i}} \]
\[ = - \frac{b_i}{b_{P_i}} \]
\[ = + \frac{b_i}{b_{P_i}} \]

Interpolate quantity along \( P \) and \( P' \) edges:

\[ P_i = P_i + \frac{1}{L}(P_{i+1} - P_i) \]
\[ = \frac{1}{L} \]
\[ = - \frac{1}{L} \]
\[ = + \frac{1}{L} \]

\[ \mathbf{P}(x,y) \]

then interpolate quantity as a function of \( x \):

\[ \mathbf{P}(x,y) \]

as a function of \( y \):

\[ \mathbf{P}(x,y) \]