Chapter 8

Scan Conversion – Drawing on Raster Display (part 1 - Lines)

Midterm 1: Grade Distribution

Average: 84

Scan Conversion - Rasterization

Convert continuous rendering primitives into discrete fragments/pixels
- Lines
  - Bresenham (Midpoint)
- Triangles
  - Implicit formulation
  - Scanline
  - Interpolation

Scan Conversion - Lines

Rendering Pipeline

- Discard geometry outside viewport window

Copyright 2012, Alla Sheffer
Computer Graphics

Scan Conversion

Idea: Use Explicit Line Formula

Explicit - one coordinate as function of the others

\[ y = f(x) \]

\[ z = f(x, y) \]

line

\[ y = mx + b \]

\[ y = \frac{(y_2 - y_1)}{(x_2 - x_1)}(x - x_1) + y_1 \]

Typically separate into 4 (or 8) cases (why?)

Midpoint (Bresenham) Algorithm

Key Observation 2:
- Can decide based on whether midpoint is above/below line
- How?
  - Evaluate implicit line equation at \((x+1, y+1/2)\)

Bresenham Algorithm

Implicit formulation = distance (up to scale)

\[ \tau = \{ (x, y) | ax + by + c = sxy - ydx + c = 0 \} \]

\[ d(x, y) = 2(sydx + ydxc) \]

- Given point \( P = (x, y) \) is signed distance of \( P \)
  to \( \tau \) (up to scale)
- \( d \) is zero for \( P \in \tau \)

Midpoint (Bresenham) Algorithm

Key Observation 1:
- At each step have ONLY 2 choices
  - East/North-East

Bresenham (Midpoint) Algorithm

- Starting point satisfies \( d(x_1, y_1) = 0 \)
- Each step moves right (east) or upper right (northeast)
- Sign of \( d(x + 1; y + 1/2) \)
  indicates if to move east or northeast

Questions: Can this algorithm use integer arithmetic?
Bresenham (Midpoint) Algorithm

Insanely efficient version (less computations inside the loop)
compute \( d \) incrementally

At \((x_0, y_0)\)
\[ d_{\text{initial}} = \left| \frac{1}{2} (y_0 + 1) - 2x_0 \right| \]

Increment in \( d \) (after each step)
- If move east \[ \Delta_d = 2(2x_0 + 1) - 2(y_0 + 1) = 2(x_0 + 2y_0 + 2 - (y_0 + 1)) \]
- If move northeast \[ \Delta_d = 2(x_0 + 1) - 2(y_0 + 1) = 2(x_0 + 1 + 2) \]

Comparison: float/integer
Assume \( s_i < s_f \) & line slope is \( \geq 1 \)

Implicit test
- Instead of clipping line in continuous space
  - For each integer value of \((x,y)\) test if inside window just before drawing
  - Inefficient on CPU
  - On a parallel (GPU) processor can be surprisingly fast
Scan Conversion of Lines

Discussion
- Integer: Bresenham
  - Good for hardware implementations (integer!)
- Floating Point
  - May be faster for software (depends on system)!  
  - Easier to parallelize