

Line-Drawing Algorithms

- Simple algorithm
 - Evaluate $y = f(x)$ at each x position
 - Floating multiplication
- DDA algorithm
 - Floating addition only
- Can we do better?
 - Bresenham's algorithm

1

Who Is Bresenham?

Bresenham, J. E. Algorithm for computer control of a digital plotter, *IBM Systems Journal*, 4(1), 1965, pp. 25-30.

Bresenham, J. E. A linear algorithm for incremental digital display of circular arcs. *Communications of the ACM*, 20(2), 1977, pp. 100-106.

Note the dates!

2

Analyzing Errors at Each Point

3

Error Analysis Detail

Represent error for each pixel choice

The diagram shows a 2x5 grid of green circles representing pixels. A yellow line passes through the grid from the bottom-left to the top-right. Vertical red lines are drawn from the center of each pixel to the yellow line. The distance from the line to the center of the second pixel in the top row is labeled d_2 . The distance from the line to the center of the second pixel in the bottom row is labeled d_1 . A small number '4' is in the bottom right corner of the slide.

Deciding on the Next Point

The diagram shows a single green circle representing a pixel choice. A yellow line passes through it. The current pixel is at (x_k, y_k) . The next pixel is at (x_{k+1}, y_{k+1}) . The error distance from the line to the current pixel is d_1 . The error distance from the line to the next pixel is d_2 . The decision value is the difference between these two error distances.

$$d_1 = y - y_k$$

$$d_2 = (y_k + 1) - y$$

$$d_1 - d_2 = 2m(x_k + 1) - 2y_k + 2b - 1$$

Decision value

Status Check

- What have we accomplished?
 - One value to test (+/-)
- But ...
 - Still have floating-point calculation
 - "m" and "b" are non-integer

$$d_1 - d_2 = 2m(x_k + 1) - 2y_k + 2b - 1$$

6

Removing Non-Integral Values

$$d_1 - d_2 = 2m(x_k + 1) - 2y_k + 2b - 1$$

$$m = \frac{\Delta y}{\Delta x}$$

$$d_1 - d_2 = 2 \frac{\Delta y}{\Delta x} (x_k + 1) - 2y_k + 2b - 1$$

$$p_k = \Delta x(d_1 - d_2) = 2\Delta y \cdot x_k + 2\Delta y - 2\Delta x \cdot y_k + \Delta x(2b - 1)$$

$$p_k = 2\Delta y \cdot x_k - 2\Delta x \cdot y_k + c$$

$$c = 2\Delta y + \Delta x(2b - 1)$$

Decision value

Recurrence Relation for p_k

$$p_k = 2\Delta y \cdot x_k - 2\Delta x \cdot y_k + c$$

$$p_{k+1} = 2\Delta y \cdot x_{k+1} - 2\Delta x \cdot y_{k+1} + c$$

$$p_{k+1} - p_k = 2\Delta y(x_{k+1} - x_k) - 2\Delta x(y_{k+1} - y_k)$$

$$p_{k+1} = p_k + 2\Delta y - 2\Delta x(y_{k+1} - y_k)$$

0 or +1

Initializing the Algorithm

Initial value of decision parameter

$$p_0 = 2\Delta y \cdot x_0 - 2\Delta x \cdot y_0 + c$$

$$p_0 = 2\Delta y \cdot x_0 - 2\Delta x \cdot y_0 + 2\Delta y + \Delta x(2b - 1)$$

But normalize: $x_0 = y_0 = b = 0$

$$p_0 = 2\Delta y - \Delta x$$

Bresenham Algorithm Summary

Calculate: $\Delta x, \Delta y, 2\Delta y, 2\Delta y - 2\Delta x, p_0$

At each point:

If $p_k < 0$: Plot $(x_k + 1, y_k)$
 $p_{k+1} = p_k + 2\Delta y$

If $p_k \geq 0$: Plot $(x_k + 1, y_k + 1)$
 $p_{k+1} = p_k + 2\Delta y - 2\Delta x$

10

Example 1 Parameters


$\Delta x = 10$
 $2\Delta x = 20$
 $\Delta y = 3$
 $2\Delta y = 6$
 $2\Delta y - 2\Delta x = -14$
 $p_0 = 2\Delta y - \Delta x = -4$

11

Example 1 Path

Same as simple line algorithm


12



Bresenham Summary

- Only integer arithmetic
- Sign detection
 - Magnitude not tested
- Only multiply is by two
 - Left shift (<<) operator


13



Bresenham Modifications

- So far, $0 < m < 1, \Delta x > 0$
- How handle other cases?
 - Horizontal
 - Vertical
 - Other octants
 - $\text{abs}(\text{Slope}) \in (1, \infty]$
 - Transpose x with y, Δx with Δy , etc.
 - Plot(y,x)
 - Slope < 0
 - Revise: $y_{k+1} - y_k = 0$ or -1

14



Alternative Viewpoint

- Assume $0 \leq \text{slope} \leq 1$
- Each step in x moves y up a fraction ($\Delta y/\Delta x$)
- As soon as the fraction exceeds 1/2, move up rather than over (subtract 1 [$\Delta x/\Delta x$] to get back to fraction between -1/2 and 1/2)
- Clear the floats by multiplying by $2\Delta x$ in all terms

15
