The Bresenham and Pitteway Algorithms

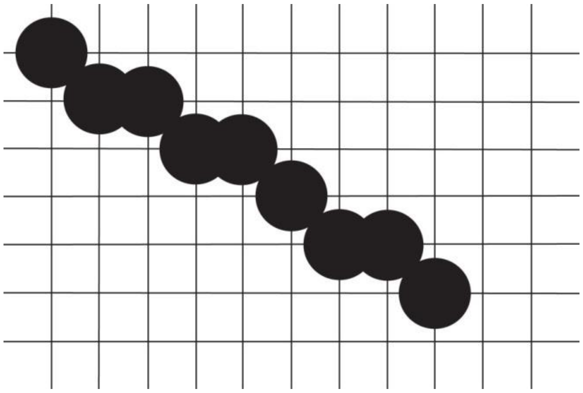
Bresenham's algorithm finds the pixels covered by a line, using only integer arithmetic. Pitteway finds the same pixels and also the intensities needed to anti-alias the line when it is used as the boundary of a polygon. Pitteway uses floating point arithmetic, which is slower.

For both algorithms we assume the line is y=mx+b, where 0 <= m <= 1, we assume m = v/h, and the left boundary of the line is pixel (x_0, y_0).

Bresenham: Start with d_0 = 2v-h. At each step, if d_i <= 0, choose $y_{i+1} = y_i$, $d_{i+1} = d_i+2v$. On the other hand, if di>0 choose $y_{i+1} = y_i+1$, $d_{i+1} = d_i+2v-2h$

Bresenham Example: To find the pixels covered by the line from (100, 325) to (108, 330). Here h=8, v=5, m=5/8, d_0=2.

x	100	101	102	103	104	105	106	107	108
y	325	326	326	327	327	328	329	329	330
d	2	-4	6	0	10	4	-2	8	2



Pitteway: Start with d_0 = ½. At each step, if d_i >= m, choose $y_{i+1} = y_i$, $d_{i+1} = d_i$-m. On the other hand, if d_i<m choose $y_{i+1} = y_i+1$, $d_{i+1} = d_i+1$-m. You can take the value of d as the intensity for a white polygon drawn on a black background; use 1-d for a black polygon on a white background.

Pitteway Example: Same example as with Bresenham.

x	100	101	102	103	104	105	106	107	108
y	325	326	326	327	327	328	329	329	330
d	1/2	7/8	2/8	5/8	0	3/8	6/8	1/8	4/8