COMP 557 Bresenham's Algorithm Sept. 4, 2007

These notes provide further details on the topic of how to draw a line, which was discussed in lecture 1. The notes are provided for your interest only. (I also throw in some notes on how to draw a circle – which I did not mention in lecture 1.)

Bresenham's key insight was how to avoid using a floating point variable m. Very roughly, the idea is to take advantage of the fact that m is a ratio of two integers. The mechanism is a bit tricky though. Here's how it works.

We assume the case $|m| < 1$ and again assume $x_1 > x_0$. (The extension to $|m \geq 1|$ is straightforward, similar to what I described in lecture 1.) For any value of x, suppose we know that the line passes between (x, i) and $(x, i + 1)$. When we say that we want to know $(x, round(y))$, we are saying that we want to assign $y := i+1$ if $y > i+\frac{1}{2}$, and $y := i$ if $y < i+\frac{1}{2}$. Thus, to know $round(y)$, it is sufficient that we know which of the two cases holds.

Define a function $F(x, y)$ that takes the value 0 along the line, and that is positive above the line and negative below the line, where "positive" and "negative" refer to the y value,

$$F(x, y) \equiv 2(-(y_1 - y_0), x_1 - x_0) \cdot (x - x_0, y - y_0)$$

Note that the vector $(-(y_1 - y_0), x_1 - x_0)$ is perpendicular to the direction of the line $(x_1 - x_0, y_1 - y_0)$. Notice that this function $F(x, y)$ has integer values on any of the pixels $F(x, y)$. It also has integer values on any of the midpoints $(x, y + \frac{1}{2})$.

Bresenham's algorithm for drawing the line is this:

```
y = y0;
writepixel(x0,y0)
for x = x0 to x1-1 {
    if  F(x + 1,y + 1/2)  < 0  \\  if the line is above the midpoint
        y = y + 1;
   writepixel(x + 1, y)
}
```

How do we evaluate whether $F(x + 1, y + 1/2) < 0$? This is easy. You can also verify that $F(x_0, y_0) = 0$, i.e. this point is on the line. You can also verify that

$$F(x + 1, y + \frac{1}{2}) \;=\; F(x, y) - 2(y_1 - y_0) + (x_1 - x_0).$$

Thus, if we know $F(x, y)$, then we can evaluate whether $F(x+1, y+1/2) < 0$ using integer addition and subtraction. (We also need to multiply by 2, but this can be done quickly as well, by "shifting left" the binary representation of $y_1 - y_0$. Note that this only needs to be done once per line anyhow, since the endpoints of the line are constants.)

As we draw the line, we update the (x, y) values by incrementing x and possibly y. Thus, we will either compute

$$F(x + 1, y) = F(x, y) - 2(y_1 - y_0)$$

or

$$F(x + 1, y + 1) = F(x, y) - 2(y_1 - y_0) + 2(x_1 - x_0)$$

depending on whether we increment y or not.

Example

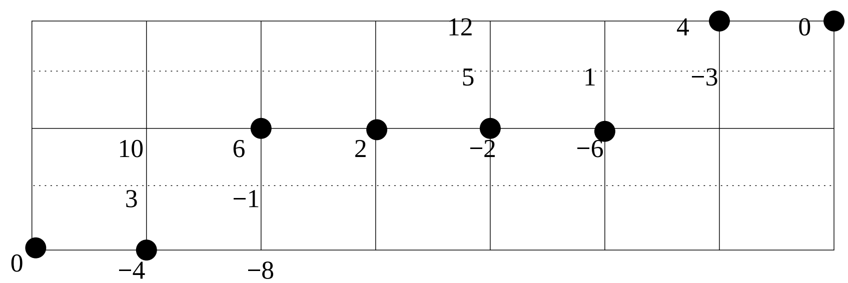
Consider $(x_0, y_0) = (0, 0)$ and $(x_1, y_1) = (7, 2)$. Then,

$$F(x+1, y) = F(x, y) - 4$$

$$F(x+1, y+1) = F(x, y) + 10$$

$$F(x+1, y+\frac{1}{2}) = F(x, y) + 3$$

The values of $F(x, y)$ at various points on the line segment between (x_0, y_0) and (x_1, y_1) are shown in the sketch on the next page. The dotted line shows the "midpoint" locations of y, i.e. 1/2, 3/2, etc.



Bresenham's algorithm for a circle

There is also a version of Bresenham's algorithm for drawing (scan converting) a circle. The brute force way to scan convert a circle of integer radius R is as follows. Consider the arc of the circle from $x = x_0$ to the line $y - y_0 = x - x_0$ which intersects the circle at $(x_0 + R/\sqrt{2}, y_0 + R/\sqrt{2})$. (The rest of the circle can be handled with appropriate symmetry.)

```
for x = x0 to x0+round( R / sqrt(2) ){
    y :=  y0 + Round(sqrt( R^2 - (x - x0)^2));
    writepixel( x, y);
}
```

This algorithm is very inefficient though, since it requires computing square roots. Bresenham came up with a clever way of scan converting a circle which uses only integer arithmetic. Define

$$F(x, y) = 4((x - x_0)^2 + (y - y_0)^2 - R^2).$$

Observe that $F(x, y)$ is greater than zero outside the circle, less than zero inside the circle, and zero on the circle.

Suppose we know $F(x, y)$ and we want to evaluate $F(x + 1, y - 1/2)$, i.e. the y values are decreasing since we start from the top of the circle. Verify for yourself that

$$F(x+1, y-\frac{1}{2}) = F(x, y) + 4(2(x - x_0) + 1 - (y - y_0) + 1/4).$$

Since x, y, x_0, y_0 are integers, this can obviously be computed using only integer addition and subtraction (and "shift lefts", in order to multiply by 2 or 4).

```
x = x0;
y = y0 + R;
Repeat until x > y {
   writepixel (x, y);
   if  F(x + 1, y - 1/2) > 0     /*  the midpoint is outside the circle  */
       y := y - 1;
   x := x + 1;
}
```

Again, to implement this algorithm, we need to compute the new value of $F(x, y)$ after x and y have been updated. This can be done easily using, e.g.

$$F(x+1, y) = F(x, y) + 4(2(x - x_0) + 1)$$

$$F(x+1, y-1) = F(x, y) + 4(2(x - x_0) + 1 - 2(y - y_0) + 1)$$