Computer Graphics

Lecture 8

Bresenham's Line Algorithm:

An accurate and efficient raster line-generating algorithm, developed by Bresenham, scan converts lines using only incremental integer calculations that can be adapted to display circles and other curves. Figures 3-5 and 3-6 illustrate sections of a display screen where straight line segments are to be drawn. The vertical axes show-scan-line positions, and the horizontal axes identify pixel columns. Sampling at unit x intervals in these examples, we need to decide which of two possible pixel positions is closer to the line path at each sample step. Starting from the left endpoint shown in Fig. 3-5, we need to determine at the next sample position whether to plot the pixel at position (11, 11) or the one at (11, 12). Similarly, Fig. 3-6 shows-a negative slope-line path starting from the left endpoint at pixel position (50, 50). In this one, do we select the next pixel position as (51,50) or as (51,49)? These questions are answered with Bresenham's line algorithm by testing the sign of an integer parameter, whose value is proportional to the difference between the separations of the two pixel positions from the actual line path.

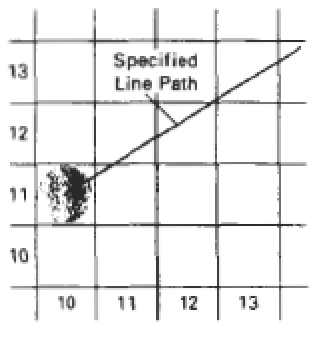

Figure 3-5

Section of a display screen where a straight line segment is to be plotted, starting from the pixel at column 10 on scan Line 11

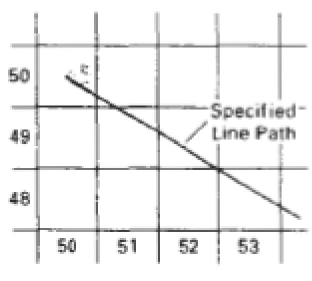

Figure 3-6

Section of a display screen where a negative slope line segment 1s to be plotted, starting from the pixel at column 50 on scan line 50

To illustrate Bresenharn's approach, we first consider the scan-conversion process for lines with positive slope less than 1. Pixel positions along a line path are then determined by sampling at unit x intervals. Starting from the left endpoint (x_0, y_0) of a given line, we step to each successive column (x position) and plot the pixel whose scan-line y value is closest to the line path. Figure 3-7 demonstrates the *k*th step in this process. Assuming we have determined that the pixel at (x_k, y_k) is to be displayed, we next need to decide which pixel to plot in column x_{k+1},. Our choices are the pixels at positions $(x_k +1, y_k)$ and $(x_k +1, y_k + 1)$.

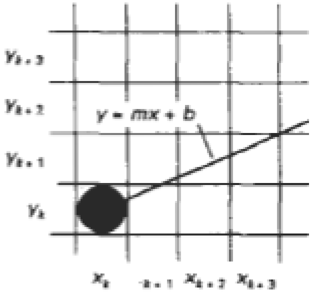

Figure 3-7
Section of the screen grid showing a pixel in column x_k on scan line y_k that is to be plotted along the path of a line segment with slope $0<m<1$.

At sampling position x_k+1, we label vertical pixel separations from the mathematical line path as d_1, and d_2 (Fig. 3-8).

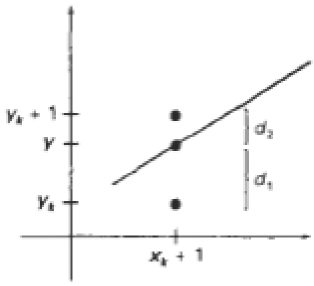

Figure 3-8
Distances between pixel positions and the line y coordinate at sampling position $x_k+ 1$.

The y coordinate on the mathematical line at pixel column position x_k+1 is calculated as

$$y = m(x_k + 1) + b \qquad (2.10)$$

Then

$$d_1 = y - y_k$$
$$= m(x_k + 1) + b - y_k \qquad \text{(Using equation 2.10)}$$

And

$$d_2 = (y_k + 1) - y$$
$$= y_k + 1 - m(x_k + 1) - b \qquad \text{(Using equation 2.10)}$$

The difference between these two separations is

$$d_1 - d_2 = 2m(x_k + 1) - 2y_k + 2b - 1 \qquad (2.11)$$

A decision parameter p_k for the *k*th step in the line algorithm can be obtained by rearranging Eq. 2.11 so that it involves only integer calculations. We accomplish this by substituting m = Δy/Δx, where Δy and Δx are the vertical and horizontal separations of the endpoint positions, and defining

$$p_k = \Delta x(d_1 - d_2)$$
$$= 2\Delta y . x_k - 2\Delta x . y_k + c \qquad (2.12)$$

The sign of p_k is the same as the sign of $d_1 - d_2$, since Δx > 0 for our example. Parameter c is constant and has the value 2Δy + Δx(2b - 1), which is independent of pixel position and will be eliminated in the recursive calculations for p_k. If the pixel at y_k is closer to the line path than the pixel at y_k+l (that is, d_1 < d,), then decision parameter p_k is negative. In that case, we plot the lower pixel; otherwise, we plot the upper pixel.

Coordinate changes along the line occur in unit steps in either the x or y directions. Therefore, we can obtain the values of successive decision parameters using incremental integer calculations. At step k + 1, the decision parameter is evaluated from Eq. 2.12 as

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

Subtracting Eq. 2.12 from the preceding equation, we have

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

But $x_{k+1} = x_k + 1$, so that

$$p_{k+1} = p_k + 2\Delta y - 2\Delta x(y_{k+1} - y_k) \qquad (2.13)$$

where the term yk+1 - yk is either 0 or 1, depending on the sign of parameter p_k

This recursive calculation of decision parameters is performed at each integer x position, starting at the left coordinate endpoint of the line. The first parameter, p_0, is evaluated from Eq. 2.12 at the starting pixel position (x_0, y_0) and with m evaluated as Δy/Δx:

$$p_0 = 2\Delta y - \Delta x \tag{2.14}$$

We can summarize Bresenham line drawing for a line with a positive slope less than 1 in the following listed steps. The constants 2Δy and 2Δy - 2Δx are calculated once for each line to be scan converted, so the arithmetic involves only integer addition and subtraction of these two constants.

1. Input the two line endpoints and store the left endpoint in (x_0,y_0)

2. Load (x_0,y_0) into the frame buffer; that is, plot the first point.

3. Calculate constants Δx, Δy, 2Δy, and 2Δy - 2Δx, and obtain the starting value for the decision parameter as

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

4. At each x_k along the line, starting at k = 0, perform the following test:

If $p_k < 0$, the next point to plot is ($x_k + 1$, y_k) and

$$p_{k+1} = p_k + 2\Delta y$$

Otherwise, the next point to plot is ($x_k + 1$, $y_k + 1$) and

$$p_{k+1}, = p_k + 2\Delta y - 2\Delta x$$

5. Repeat step 4 Δx times.