Lecture-9
MDU

Anti-aliasing

- Set the sampling frequency to at least twice that of the highest frequency occurring in the object, referred to as the Nyquist sampling frequency : $f_s = 2f_{max}$

- The sampling interval should be no larger than one half of the cycle interval (called the Nyquist sampling interval)

- For x-interval sampling, the Nyquist sampling interval is $\Delta x_s = \frac{\Delta x_{cycle}}{2}$ where $\Delta x_{cycle} = 1/f_{max}$

- Unless hardware technology is developed to handle arbitrarily large frame buffers, increased screen resolution is not a complete solution to the aliasing problem.

- The technique of sampling object characteristics at a higher resolution and displaying the results at a lower resolution is called supersampling (post-filtering).

– Area-sampling or prefiltering

–

– ~~Phase~~ Pixel phasing

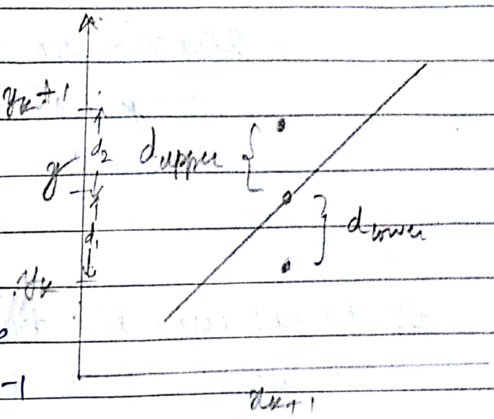
Bresenham's algorithm

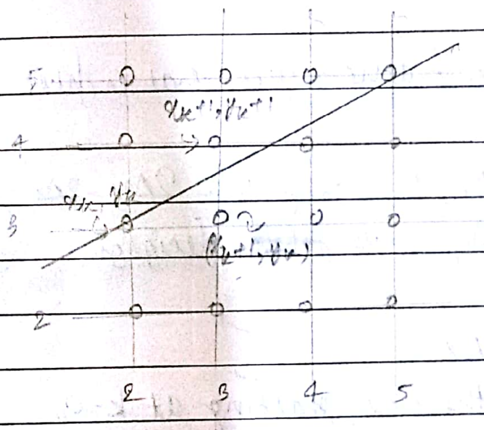
- Accurate and efficient raster line generating.
- Scan converts lines using fully incremental integer calculations.
- $m <= 1$ and +ve.
- sampling at unit x intervals.
- Starting from the left end point (x_0, y_0) of a given line, we step to each successive column (x position) and plot the pixel whose scan line y value is closest the line path.

- The pixel (x_k, y_k) is displayed.
  - Next pixel to determine → (x_k+1, y_k) or (x_k+1, y_{k+1})
- At sampling pos^n x_k+1, we label vertical pixel separations from mathematical line path as d_1 and d_2

- The y-coordinate on the mathematical line at pixel column position x_k+1 is calculated as: $y = m(x_k+1)+b$
  - $\rightarrow d_1 = y - y_k = m(x_k+1)+b-y_k$
  - $\rightarrow d_2 = (y_k+1) - y = (y_k+1) - m(x_k+1) - b$
  - $\rightarrow d_1 - d_2 = 2m(x_k+1) - 2y_k + 2b - 1$

$m = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1}$





$y_{k+1} = \{y_k, y_k+1\}$

$d_1 - d_2$

$p_k = \Delta x(d_1 - d_2)$

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

$= 2\Delta y\, x_k + 2\Delta y - 2\Delta x\, y_k + \Delta x(2b-1)$

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

$\Delta x, \Delta y$ diff b/w endpoint coordinate

- Putting the value of m, we get the following decision parameter
  $p_k = \Delta x(d_1 - d_2)$
  $= 2\Delta y \cdot x_k - 2\Delta x \cdot y_k + c$
- Constant c is $2\Delta y + \Delta x(2b-1)$
- Δx is +ve → p_k and $d_1 - d_2$ are of same sign.
- y_k is closer to the line path $(d_1 < d_2)$ is $p_k < 0$, plot lower pixel.
  - otherwise plot upper pixel
- Use incremental integer calculations
- The decision parameter in step k+1, $p_{k+1} = 2\Delta y \cdot x_{k+1} - 2\Delta x \cdot y_{k+1} + c$
  p_k
- As $x_{k+1} = x_k + 1$, $p_{k+1} - p_k = 2\Delta y(x_{k+1} - x_k) - 2\Delta x(y_{k+1} - y_k)$
- $y_{k+1} - y_k = 0$ depending on the sign of p_k
- Initial parameter $p_0 = 2\Delta y - \Delta x$

The following constants are calculated once for each line to be scan converted $2\Delta y$ and $2\Delta y - 2\Delta x$.

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

$= 2\Delta y . x_0 - 2\Delta x . y_0 + 2\Delta y - \Delta x + 2y_0\Delta x - 2\Delta y x_0$

$= 2\Delta y - \Delta x$

$y_0 = mx_0 + b$

$b = y_0 - mx_0$

$= y_0 - \frac{\Delta y}{\Delta x} x_0$

Bresenham's line drawing algorithm

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 your first point.
3. Calculate constants Δx, Δy, $2\Delta y$ and $2\Delta y - 2\Delta x$, 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.