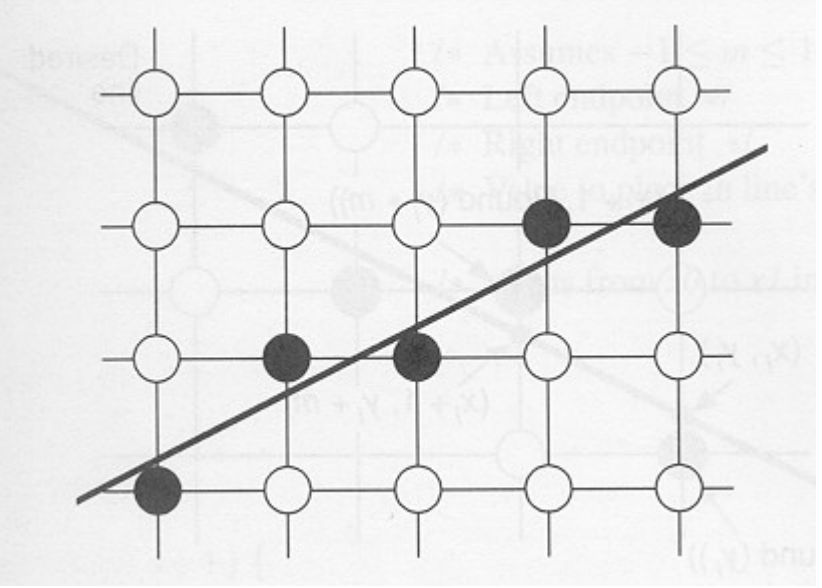
Drawing Lines & Anti-Aliasing

CSCI 4229/5229
Computer Graphics
Fall 2018

Scan Converting Lines

- Which pixels to turn on?
  - Floating point
  - Bresenham algorithm



Floating Point Algorithm

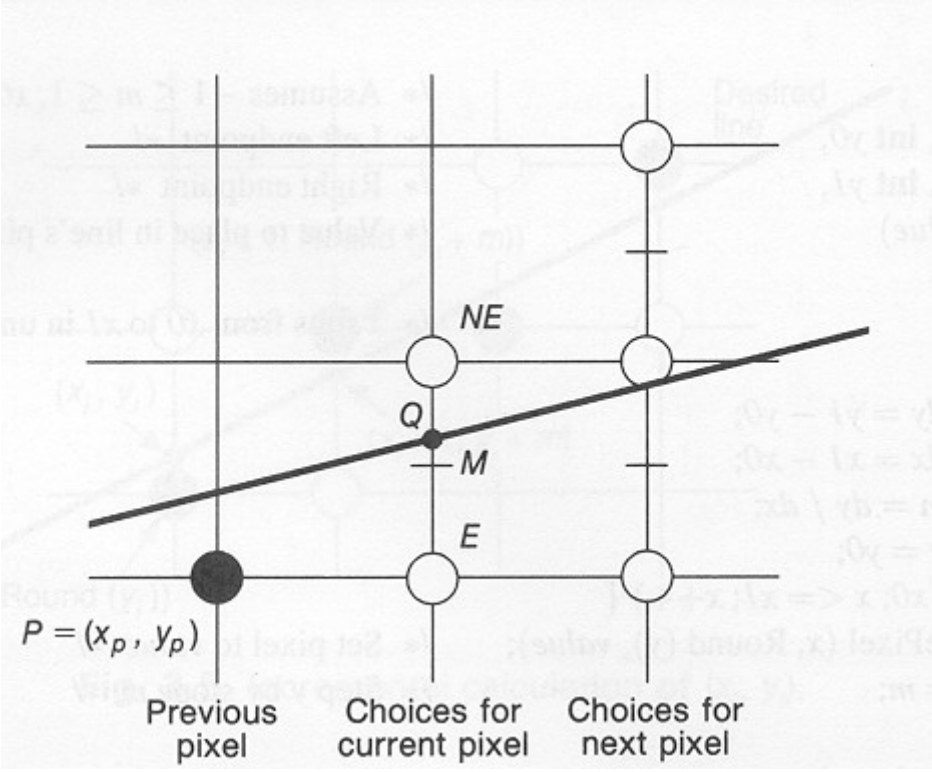
- Functional form

  $y = (x-x_0)(y_1-y_0)/(x_1-x_0)+y_0$ (use when $|y_1-y_0|<|x_1-x_0|$)

  $x = (y-y_0)(x_1-x_0)/(y_1-y_0)+x_0$ (use when $|x_1-x_0|<|y_1-y_0|$)
- Evaluate y or x at integral values of x or y
- Round result to nearest integer to decide pixel
- Slow
  - integer -> float
  - float multiply and two float additions
  - float -> integer

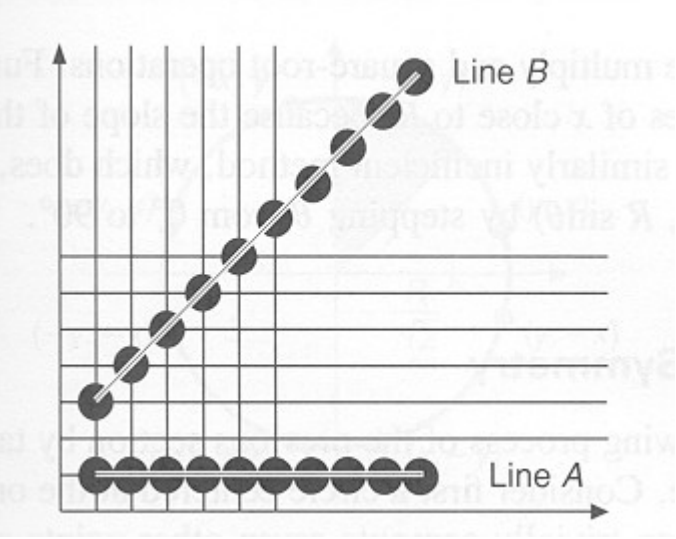
Bresenham Algorithm

- Select next pixel from two choices: E or NE
  - Only works when slope is <=1
  - Is midpoint above or below the line?
- All integer operations
  - One or two adds



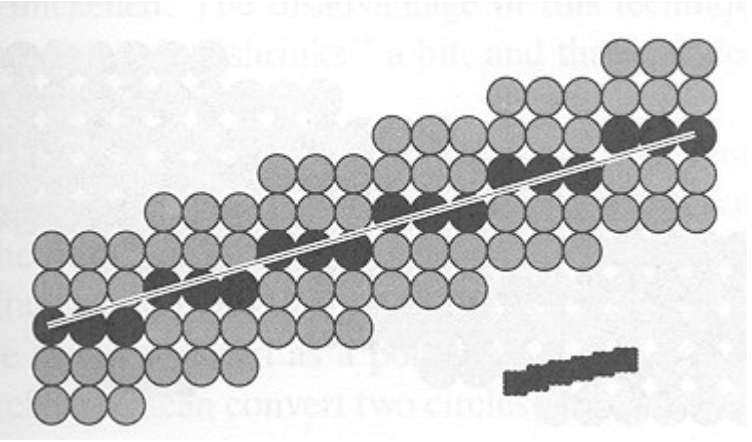
Line intensity

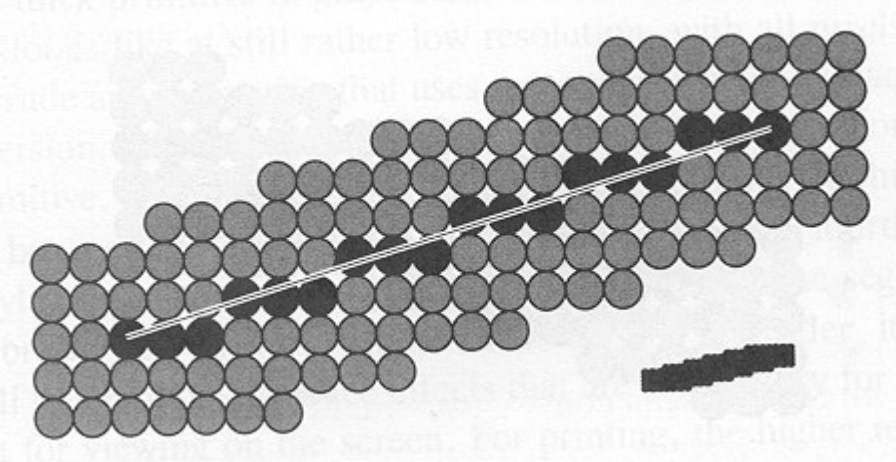
- Lines parallel to axes appear more dense than lines at 45 degree angles by $\sqrt{2}$
- If this is an issue you can adjust the pixel intensity inversely proportional to the cosine



Thick Lines

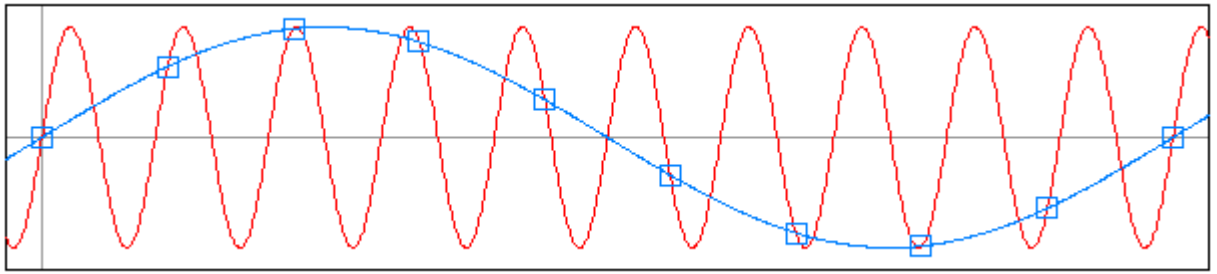
- Column replication
- Rectangular pen
- Polygon fill





Anti-aliasing in signal processing

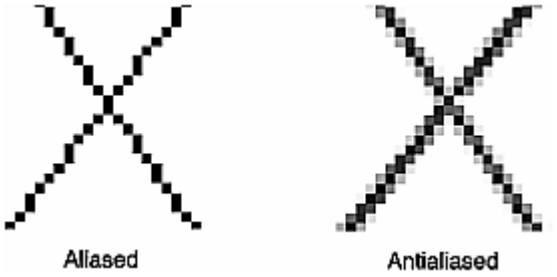
- Discrete samples of a signal



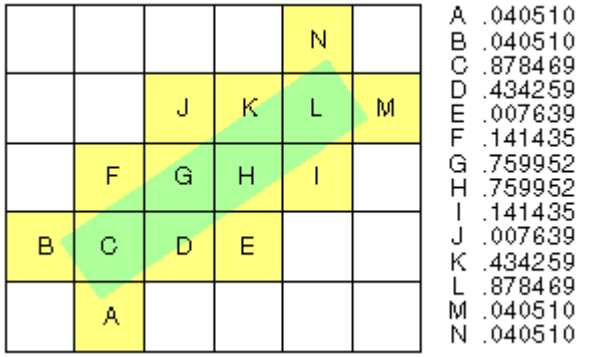
- Low and high frequency samples are the same

Anti-aliasing in Computer Graphics

- Aliased lines
  - Discrete pixels are turned on
  - Nearest pixel selected
  - Leads to “jaggies”



- Anti-aliased lines
  - Pixels are partially turned on
  - Level selected by line overla
  - Leads to smoother lines



OpenGL Anti-aliased Lines

- glEnable(GL_LINE_SMOOTH);
- glEnable(GL_BLEND);
- glBlendFunc (GL_SRC_ALPHA,
  GL_ONE_MINUS_SRC_ALPHA);
- glLineWidth(1.5);