# Drawing Lines & Anti-Aliasing
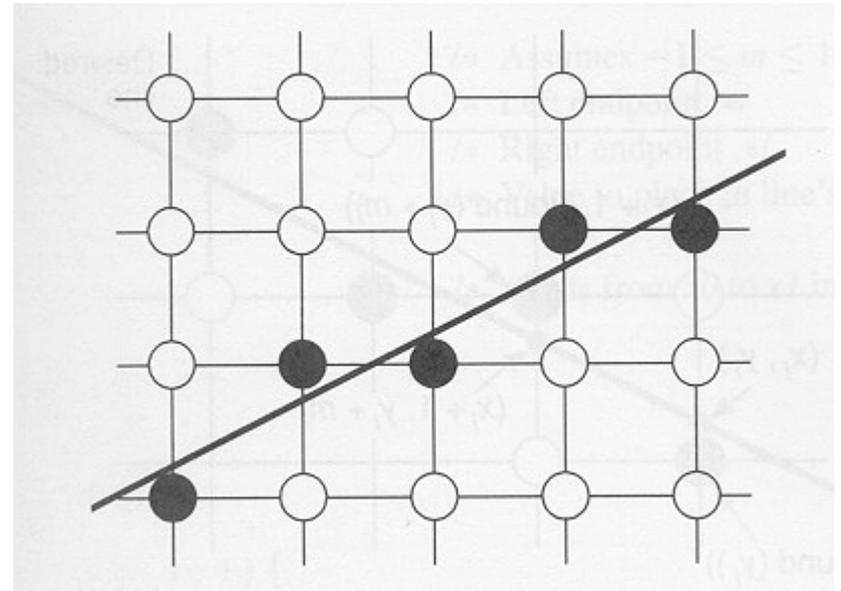
CSCI 4229/5229
Computer Graphics
Summer 2010

# Scan Converting Lines

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

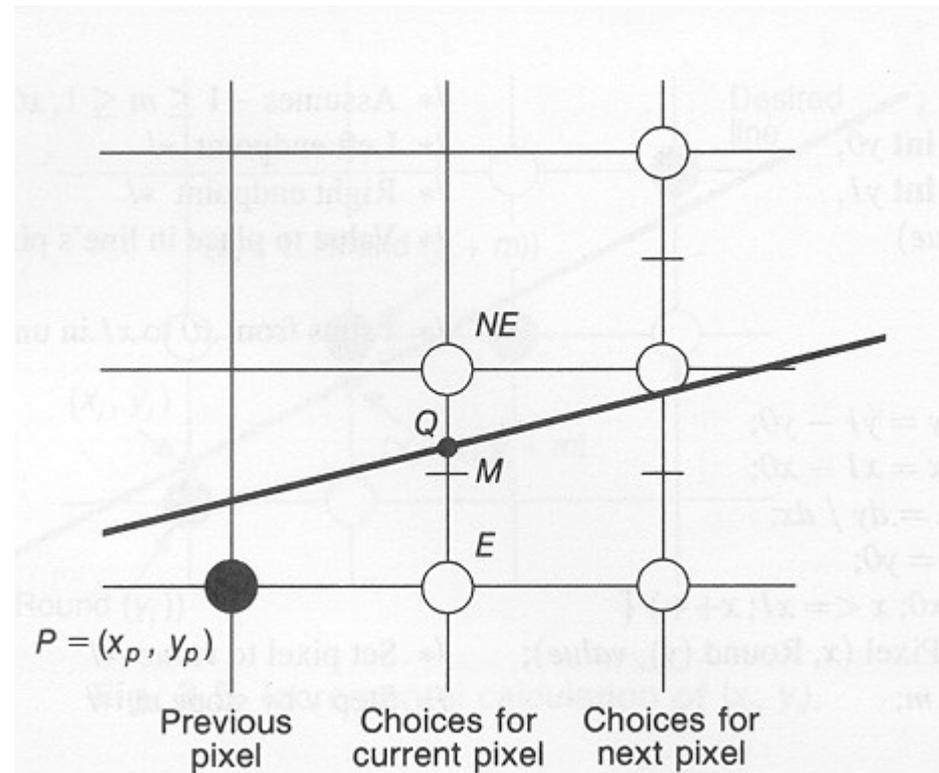# Floating Point Algorithm

- 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?
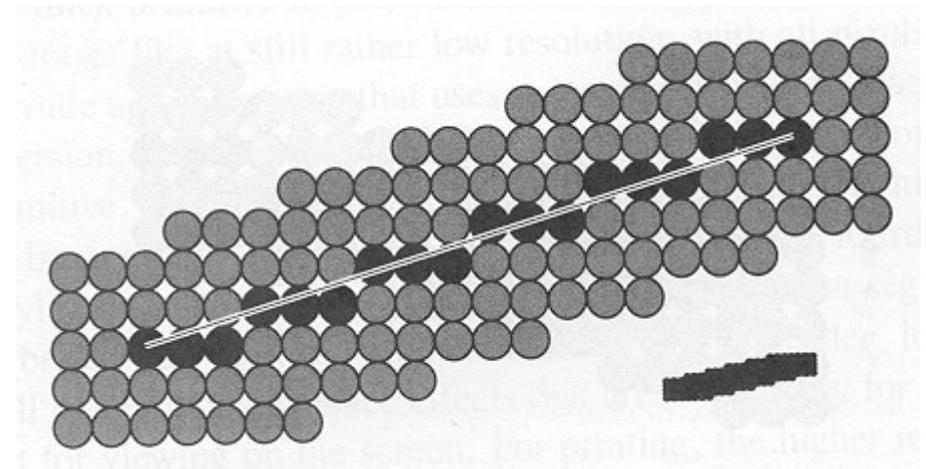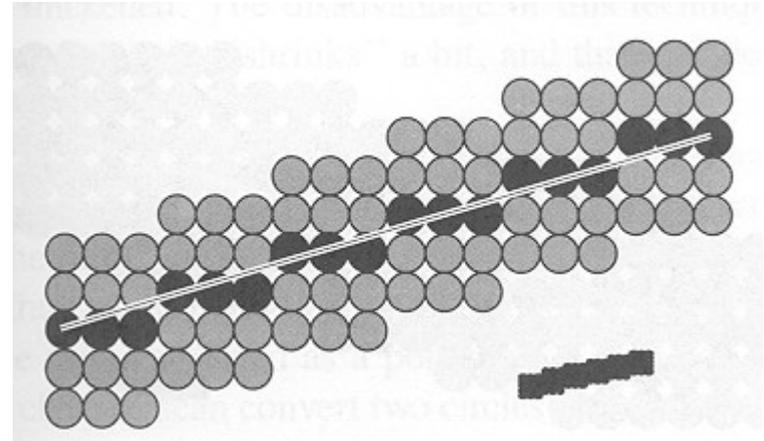
- All integer operations
  - One or two adds



NE

Q

M

E

$P = (x_p, y_p)$

Previous pixel    Choices for current pixel    Choices for next pixel

# Line intensity

- Lines parallel to axes appear more dense than lines at 45 degree angles by √2

- If this is an issue you can adjust the pixel intensity inversely proportional to the cosine
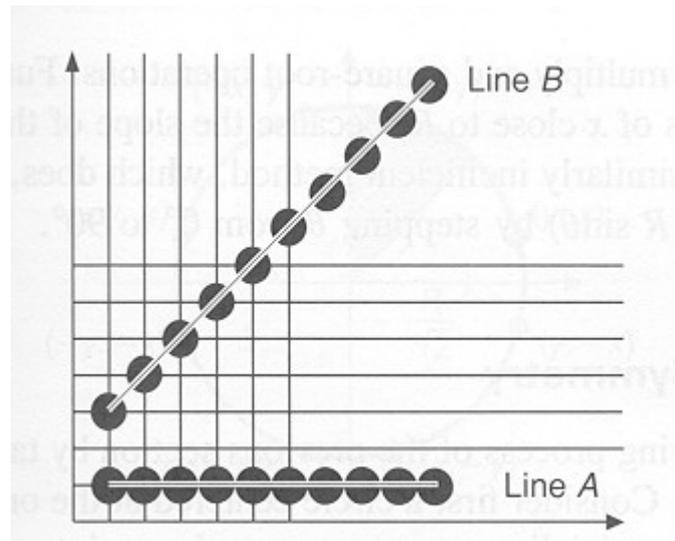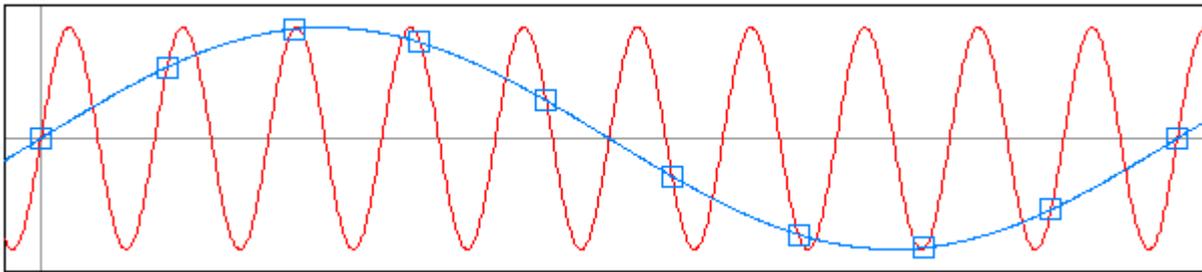
# Thick Lines

- Column replication

- Rectangular pen

- Polygon fill

# Anti-aliasing in signal processing
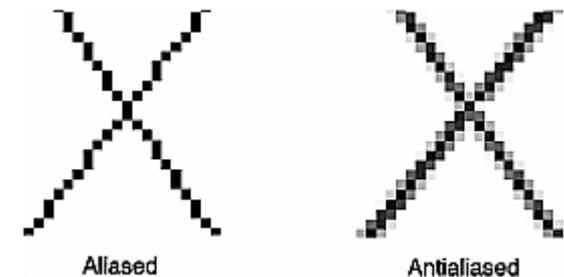
- Discrete samples of a signal



- Low and high frequency samples are the same
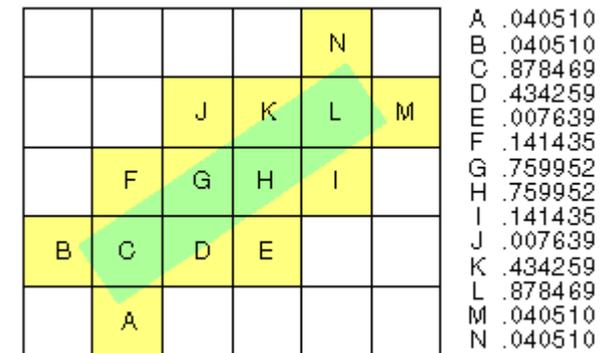
# Anti-aliasing in Computer Graphics

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



Aliased            Antialiased

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



| | |
|---|---|
| A | .040510 |
| B | .040510 |
| C | .878469 |
| D | .434259 |
| E | .007639 |
| F | .141435 |
| G | .759952 |
| H | .759952 |
| I | .141435 |
| J | .007639 |
| K | .434259 |
| L | .878469 |
| M | .040510 |
| N | .040510 |

# OpenGL Anti-aliased Lines

- glEnable(GL_LINE_SMOOTH);

- glEnable(GL_BLEND);

- glBlendFunc (GL_SRC_ALPHA,
    GL_ONE_MINUS_SRC_ALPHA);

- glLineWidth(1.5);