AIF 313 - Computer Graphics

# Raster Graphics

Luciana Abednego

Department of Informatics
Faculty of Information Technology and Sciences
Parahyangan Catholic University

# Contents

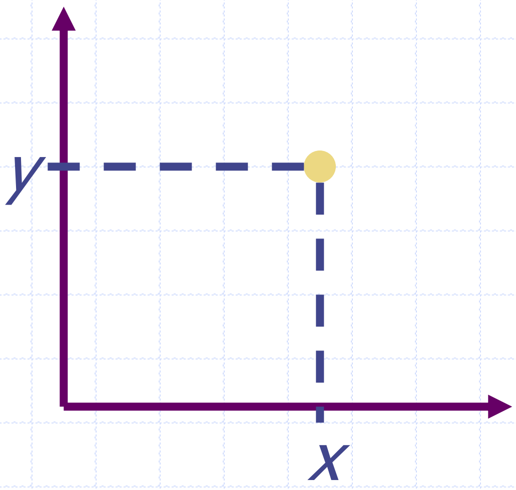
- Output Primitives
  - How can we describe shapes with primitives?
- Color Models
  - How can we describe and represent colors?

# Output Primitives

- **Points**
- **Lines**
  - DDA Algorithm
  - Bresenham’s Algorithm
- **Polygons**
  - Scan-Line Polygon Fill
  - Inside-Outside Tests
  - Boundary-Fill Algorithm
  - Antialiasing

# Points

- **Single Coordinate Position**
  - Set the bit value(color code) corresponding to a specified screen position within the frame buffer




```
setPixel (x, y)
```

# Lines

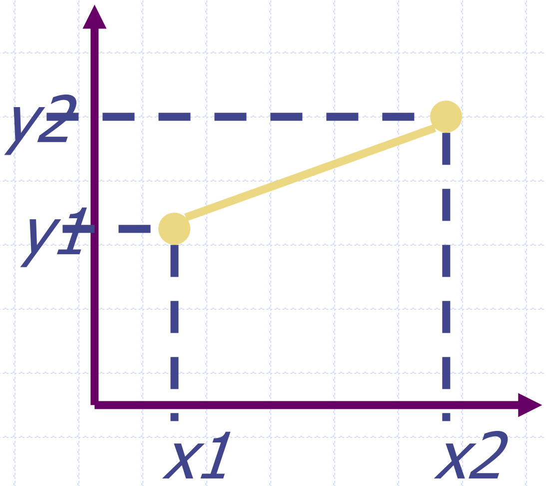
- Intermediate Positions between Two Endpoints
- Line Equations: $y = m.x + c$
- Given 2 endpoints $(x_0, y_0)$ and $(x_{end}, y_{end})$:

$$m = \frac{y_{end} - y_0}{x_{end} - x_0}$$

$$c = y_0 - m.x_0$$

# DDA Algorithm

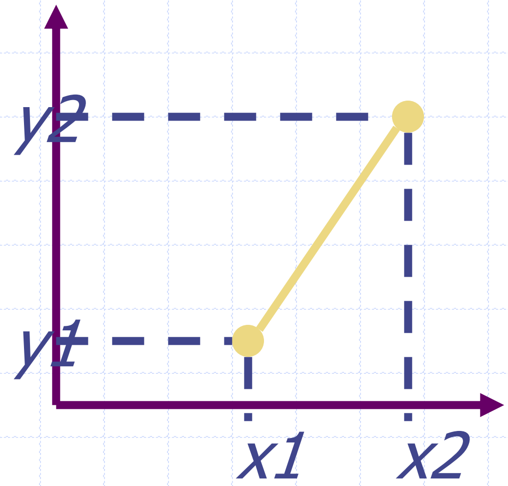
- **Digital Differential Analyzer**
  - 0 < Slope <= 1
    - Unit x interval = 1




$$y_{k+1} = y_k + m$$

# DDA Algorithm

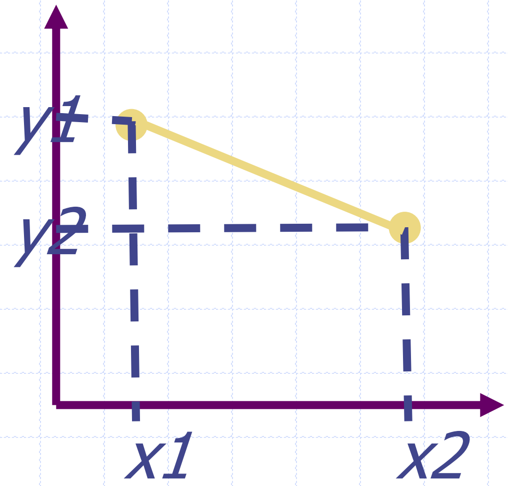
- **Digital Differential Analyzer**
  - 0 < Slope <= 1
    - Unit x interval = 1
  - Slope > 1
    - Unit y interval = 1




$$x_{k+1} = x_k + \frac{1}{m}$$

# DDA Algorithm

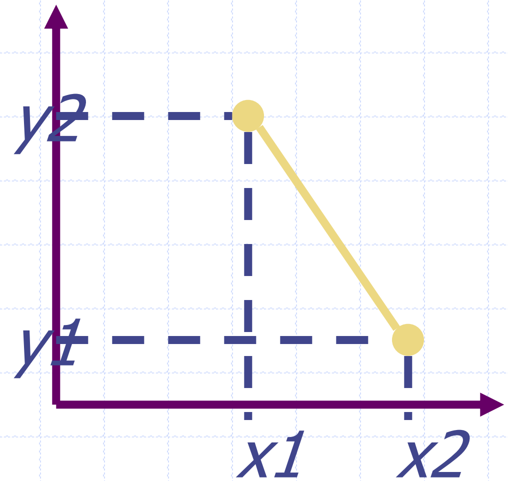
- **Digital Differential Analyzer**
  - 0 < Slope <= 1
    - Unit x interval = 1
  - Slope > 1
    - Unit y interval = 1
  - -1 <= Slope < 0
    - Unit x interval = -1




$$y_{k+1} = y_k - m$$

# DDA Algorithm

- **Digital Differential Analyzer**
  - Slope >= 1
    - Unit x interval = 1
  - 0 < Slope < 1
    - Unit y interval = 1
  - -1 <= Slope < 0
    - Unit x interval = -1
  - Slope < -1
    - Unit y interval = -1




$$x_{k+1} = x_k - \frac{1}{m}$$

# DDA Algorithm

- **Line Equations:**

$$y = m.x + c$$

- Given 2 endpoints $(x_0, y_0)$ and $(x_{end}, y_{end})$:

$$m = \frac{y_{end} - y_0}{x_{end} - x_0}$$

$$c = y_0 - m.x_0$$

$$|m| < 1 \rightarrow x_{k+1} = x_k + 1$$
$$y_{k+1} = y_k + \mathrm{m}$$

$$|m| > 1 \rightarrow y_{k+1} = y_k + 1$$
$$x_{k+1} = x_k + \frac{1}{m}$$

# DDA Algorithm

```
void lineDDA(int x_0, int y_0, int x_end, int y_end)
    dx = x_end - x_0
    dy = y_end - y_0
    x = x_0, y = y_0
    if(abs(dx) > abs(dy))
        steps = abs(dx)
    else  steps = abs(dy)
    xIncrement = dx/steps
    yIncrement = dy/steps
    setPixel(round(x),round(y))
    for k = 0 to steps-1
        x = x + xIncrement
        y = y + yIncrement
        setPixel(round(x),round(y))
```

# DDA Algorithm

**(+)** A **faster method for calculating pixel positions** than directly implements line equation. It eliminates the multiplication by making use of raster characteristics.

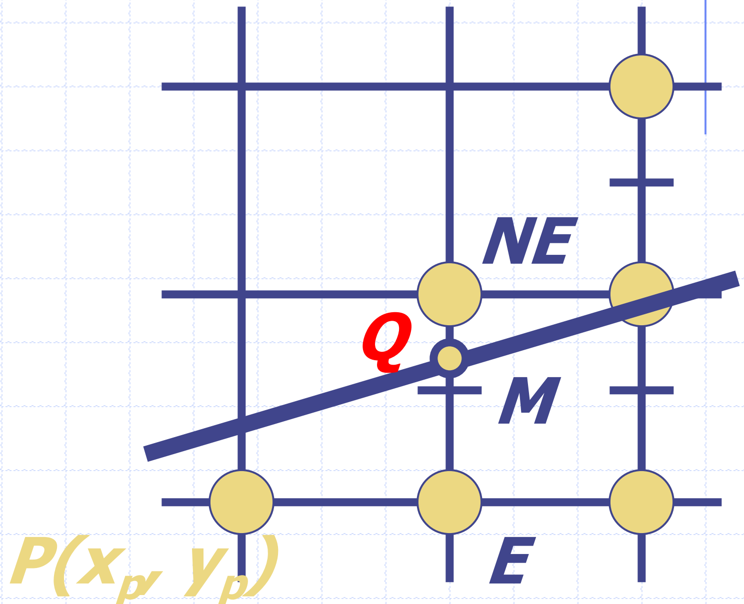
**(-)** The accumulation of round-off error in successive additions of the floating-point increment can cause the calculated **pixel positions** to **drift away** from the true line path for long line segments.

**(-)** Rounding operations are still **time consuming**.

# Bresenham's Line Algorithm

- **Midpoint Line Algorithm**
  - Decision variable

$$d = F(M)$$
$$= F\left(x_p + 1, y_p + \frac{1}{2}\right)$$
$$= a(x_p + 1) + b\left(y_p + \frac{1}{2}\right) + c$$




  - d > 0 : choose NE
    - $d_{new} = F\left(x_p + 2, y_p + \frac{3}{2}\right)$ : $d_{new} = d_{old} + a + b$
  - d <= 0 : choose E
    - $d_{new} = F\left(x_p + 2, y_p + \frac{1}{2}\right)$ : $d_{new} = d_{old} + a$

# Bresenham's Algorithm(cont.)

- **Initial Value of d**

$$F\left(x_0+1, y_0+\frac{1}{2}\right)=a(x_0+1)+b\left(y_0+\frac{1}{2}\right)+c$$

$$=F(x_0, y_0)+a+\frac{1}{2}b$$

$$\rightarrow F(x,y)=2(ax+by+c)$$

$$\rightarrow d=2a+b$$

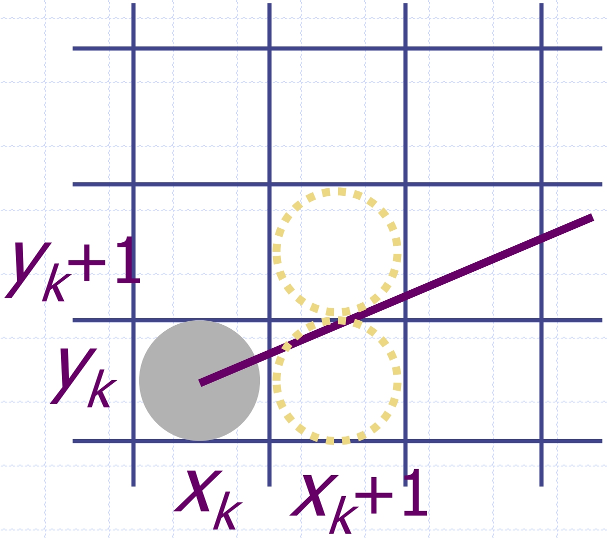
- **Update d**

$$\text{if } d>0, \text{ then } \begin{cases} x++, \\ y++, \\ d+=2(a+b) \end{cases} \qquad \text{if } d<=0, \text{ then } \begin{cases} x++, \\ d+=2a \end{cases}$$

# Bresenham's Line Algorithm

- Accurate and Efficient
  - Use only incremental integer calculations
  - Test the sign of an integer parameter
- Case) Positive Slope Less Than 1
  - After the pixel $(x_k, y_k)$ is displayed, next which pixel is decided to plot in column $x_{k+1}$?

➔ $(x_k+1, y_k)$ or $(x_k+1, y_k+1)$



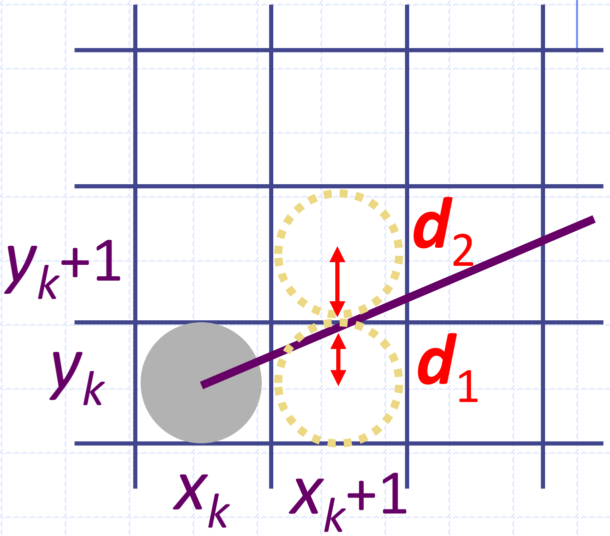
# Bresenham's Algorithm(cont.)

- Case) Positive Slope Less Than 1
  - *y* at sampling position $x_k$

    $$y = m(x_k + 1) + b$$
  - Difference

    $$d_1 = y - y_k = m(x_k + 1) + b - y_k$$
    $$d_2 = y_k + 1 - y = y_k + 1 - m(x_k + 1) - b$$

    **$d_1 - d_2 < 0 \rightarrow (x_k+1, y_k)$**
    **$d_1 - d_2 > 0 \rightarrow (x_k+1, y_k+1)$**




  - Decision parameter

$$p_k = \Delta x(d_1 - d_2)$$
$$= 2\Delta y \cdot x_k - 2\Delta x \cdot y_k + 2\Delta y + \Delta x(2b - 1)$$
$$= 2\Delta y \cdot x_k - 2\Delta x \cdot y_k + c$$

# Bresenham's Algorithm(cont.)

- Case) Positive Slope Less Than 1
  - Decision parameter

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

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

  - Decision parameter of a starting pixel ($x_0$, $y_0$)

$$p_0 = 2\Delta y \cdot x_0 - 2\Delta x \cdot y_0 + 2\Delta y + \Delta x(2b - 1)$$
$$= 2\Delta y \cdot x_0 - 2\Delta x \cdot (mx_0 + b) + 2\Delta y + \Delta x(2b - 1)$$
$$= 2\Delta y \cdot x_0 - 2\Delta y \cdot x_0 - 2b\Delta x + 2\Delta y + 2b\Delta x - \Delta x$$

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

# Bresenham's Algorithm(cont.)

- Algorithm for $0<m<1$
  - Input the two line endpoints and store the left end point in $(x_0, y_0)$
  - Load $(x_0, y_0)$ into the frame buffer; that is, plot the first point
  - Calculate constants $\Delta x$, $\Delta y$, $2\Delta y$, and $2\Delta y - 2\Delta x$, and obtain the starting value for the decision parameter as

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

  - At each $x_k$ along the line, start 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

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

  - Repeat step 4 $\Delta x$ times

# Bresenham Line Drawing

- Two endpoints: (20,10) and (30,18)
- Slope = 0.8
- $\Delta x = 10 \qquad \Delta y = 8$
- $p_0 = 2\Delta y - \Delta x = 6$
- $2\Delta y = 16 \quad 2\Delta y - 2\Delta x = -4$

| k | pk | (xk+1,yk+1) |
|---|---|---|
| 0 | 6 | (21,11) |
| 1 | 2 | (22,12) |
| 2 | -2 | (23,12) |
| 3 | 14 | (24,13) |
| 4 | 10 | (25,14) |

| k | pk | (xk+1,yk+1) |
|---|---|---|
| 5 | 6 | (26,15) |
| 6 | 2 | (27,16) |
| 7 | -2 | (28,16) |
| 8 | 14 | (29,17) |
| 9 | 10 | (30,18) |