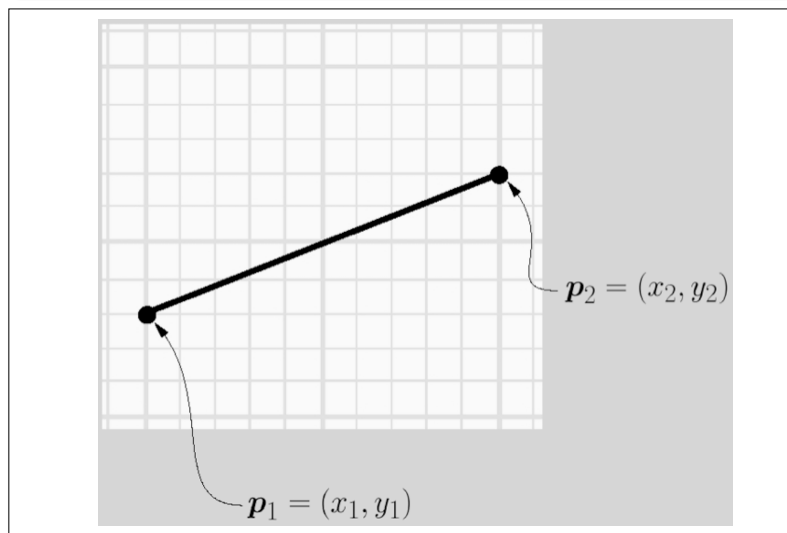


Line Drawing Algorithm

- Modified from Prof. James O' Brien's lecture note on Scan Conversion for CS184 SP06

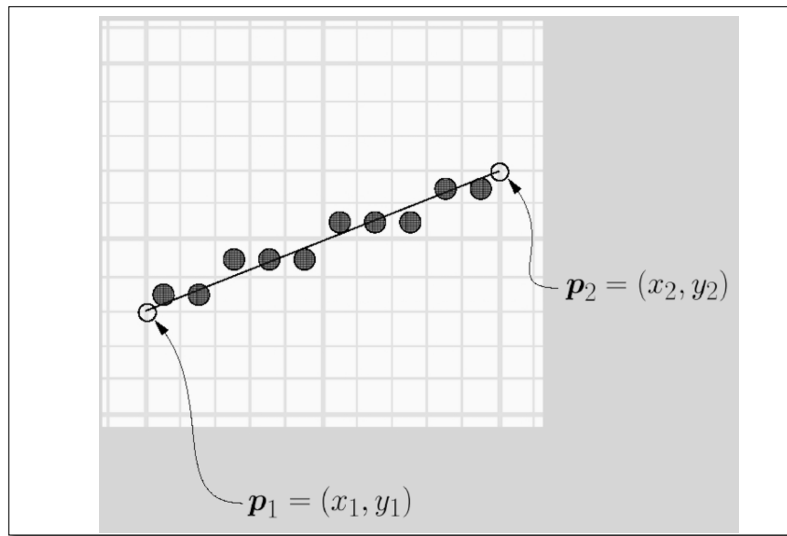
1

Drawing a Line



2

Drawing a Line

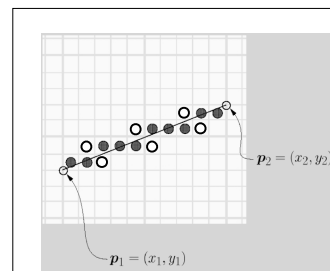


3

Drawing a Line

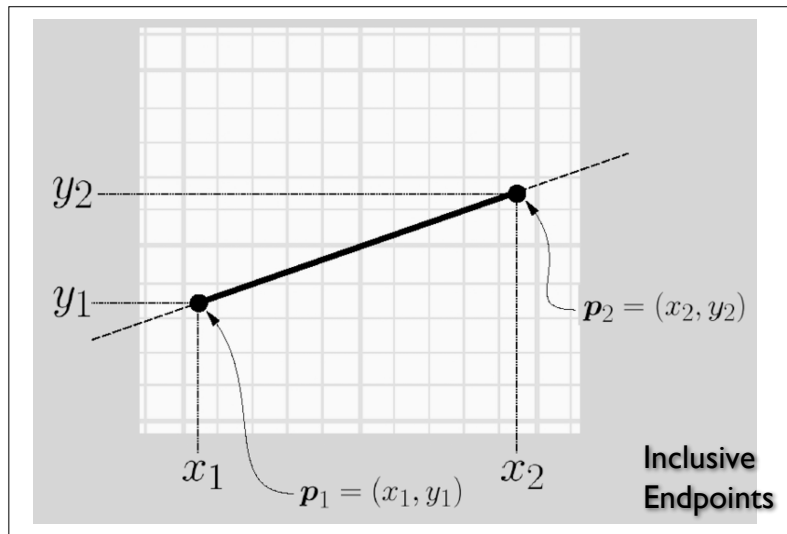
- Some things to consider
 - How thick are lines?
 - How should they join up?
 - Which pixels are the right ones?

For example:



4

Drawing a Line

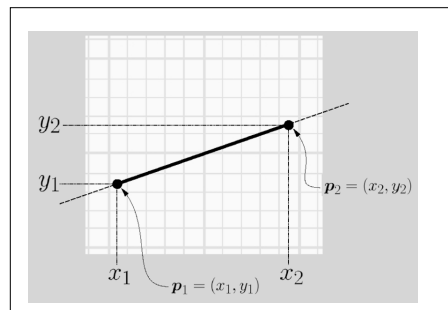


Drawing a Line

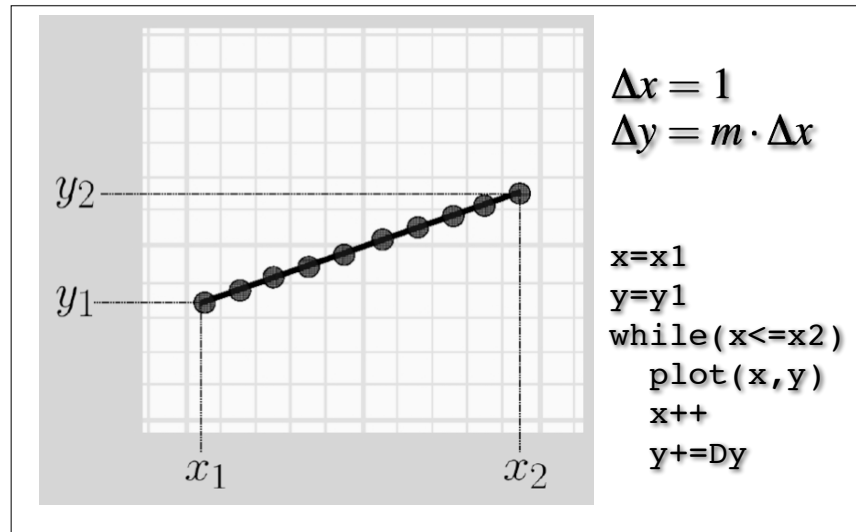
$$y = m \cdot x + b, x \in [x_1, x_2]$$

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

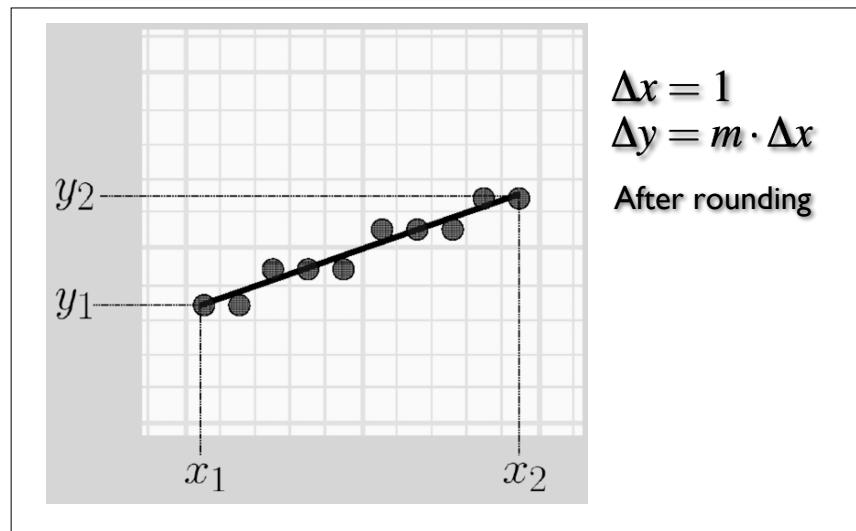
$$b = y_1 - m \cdot x_1$$



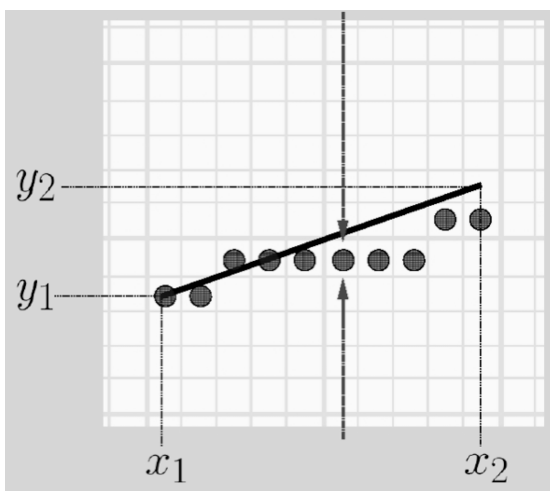
Drawing a Line



Drawing a Line



Drawing a Line



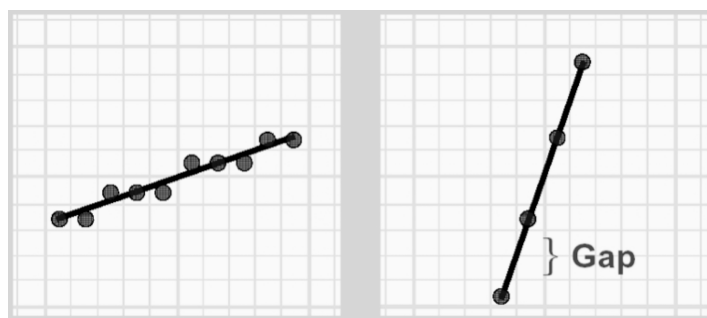
$$\Delta x = 1$$
$$\Delta y = m \cdot \Delta x$$
$$y += \Delta y$$

Accumulation of
roundoff errors

How slow is float-
to-int conversion?

9

Drawing a Line



$$|m| \leq 1$$

$$|m| > 1$$

10

Drawing a Line

```
void drawLine-Error1(int x1,x2, int y1,y2)

float m = float(y2-y1)/(x2-x1)
int x = x1
float y = y1

while (x <= x2)

    setPixel(x,round(y),PIXEL_ON)

    x += 1
    y += m
```

Not exact math

Accumulates errors

11

Drawing a Line

```
void drawLine-Error2(int x1,x2, int y1,y2)

float m = float(y2-y1)/(x2-x1)
int x = x1
int y = y1
float e = 0.0

while (x <= x2)

    setPixel(x,y,PIXEL_ON)

    x += 1
    e += m
    if (e >= 0.5)
        y+=1
        e-=1.0
```

No more rounding

12

Drawing a Line

```
void drawLine-Error3(int x1,x2, int y1,y2)

    int x = x1
    int y = y1
    float e = -0.5

    while (x <= x2)

        setPixel(x,y,PIXEL_ON)

        x += 1
        e += float(y2-y1)/(x2-x1)
        if (e >= 0.0)
            y+=1
            e-=1.0
```

13

Drawing a Line

```
void drawLine-Error4(int x1,x2, int y1,y2)

    int x = x1
    int y = y1
    float e = -0.5*(x2-x1)           // was -0.5

    while (x <= x2)

        setPixel(x,y,PIXEL_ON)

        x += 1
        e += y2-y1                   // was /(x2-x1)
        if (e >= 0.0)                // no change
            y+=1
            e-=(x2-x1)               // was 1.0
```

14

Drawing a Line

```
void drawLine-Error5(int x1,x2, int y1,y2)

int x = x1
int y = y1
int e = -(x2-x1)           // removed *0.5

while (x <= x2)

    setPixel(x,y,PIXEL_ON)

    x += 1
    e += 2*(y2-y1)         // added 2*
    if (e >= 0.0)         // no change
        y+=1
        e-=2*(x2-x1)      // added 2*
```

15

Drawing a Line

```
void drawLine-Bresenham(int x1,x2, int y1,y2)

int x = x1
int y = y1
int e = -(x2-x1)

while (x <= x2)

    setPixel(x,y,PIXEL_ON)

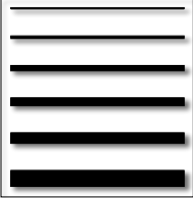
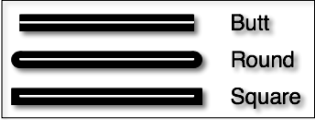
    x += 1
    e += 2*(y2-y1)
    if (e >= 0.0)
        y+=1
        e-=2*(x2-x1)
```

Faster
Not wrong

$$|m| \leq 1$$
$$x_1 \leq x_2$$


16

Drawing a Line

- How thick?

- Ends?


17

Drawing a Line

- Joining?

Ugly Bevel Round Miter

18