More on Bresenham's Algorithm

Lecture Set 2

CS5600 ***Introduction to Computer Graphics***

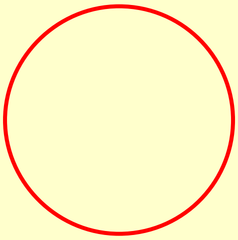
Rich Riesenfeld

January 2003

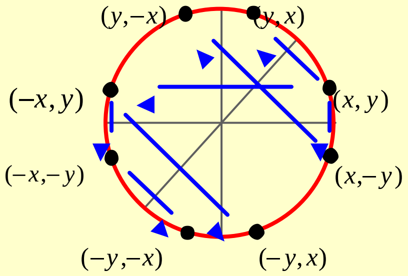
More Raster Line Issues

- Fat lines with multiple pixel width
- Symmetric lines
- End point geometry -- how should it look?
- Generating curves, e.g., circles, etc.
- Jaggies, staircase effect, aliasing...

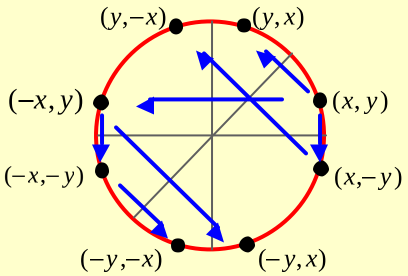
Generating Circles



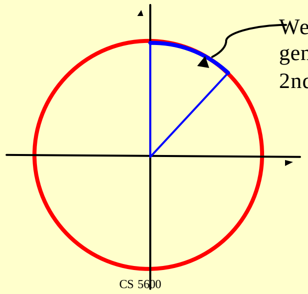
Exploit 8-Point Symmetry



Once More: 8-Point Symmetry



We Only Need to Generate One Octant



We will
generate
2nd Octant

Generating (x, y) gives

The following 8 points:

$$(x, y), (-x, y), (-x,-y), (x,-y),$$

$$(y, x), (-y, x), (-y,-x), (y,-x)$$

2nd Octant Is a Good Arc

- The arc is a *function* in this domain
  - –single-valued
  - –no vertical tangents
- $|slope| < 1$
- Lends itself to Bresenham
  - –only need to consider *E* or *SE*

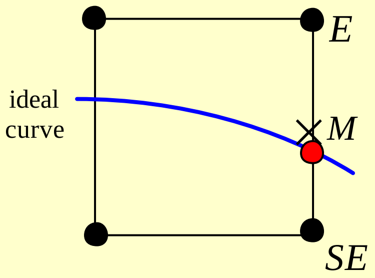
Implicit Circle Equations

- Let $F(x,y) = x^2 + y^2 - r^2$
- For a circle $F(x,y) = 0$
- So $F(x,y) > 0 \quad \Rightarrow \quad Outside$
- And $F(x,y) < 0 \quad \Rightarrow \quad Inside$

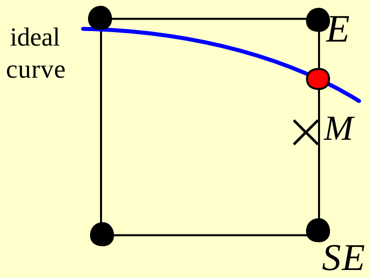
Decide Whether *E* or *SE*

- Function is $x^2 + y^2 - r^2 = 0$
- So $F(M) \geq 0 \quad \Rightarrow \quad SE$
- And $F(M) < 0 \quad \Rightarrow \quad E$

$F(M) \geq 0 \quad \Rightarrow \quad SE$



$F(M) < 0 \quad \Rightarrow \quad E$



The Decision Variable d

Again we let,

$$d = F(M)$$

Look at Case 1: E

$d_{old} < 0 \quad \Rightarrow \quad E$

$$d_{old} = F(x_p + 1, y_p - \frac{1}{2})$$

$$= (x_p + 1)^2 + (y_p - \frac{1}{2})^2 - r^2$$

$d_{old} < 0 \quad \Rightarrow \quad E$

$$d_{new} = F(x_p + 2, y_p - \frac{1}{2})$$

$$= (x_p + 2)^2 + (y_p - \frac{1}{2})^2 - r^2$$

$d_{old} < 0 \quad \Rightarrow \quad E$

$$d_{new} = d_{old} + (2x_p + 3)$$

Since,

$$(x_p + 2)^2 - (x_p + 1)^2 = (4x_p + 4) - (2x_p + 1)$$

$$= 2x_p + 3$$

$d_{old} < 0 \quad \Rightarrow \quad E$

$$d_{new} = d_{old} + \Delta_E,$$

$$\boxed{\Delta_E = 2x_p + 3}$$

Look at Case 2: *SE*

$d_{old} \geq 0 \quad \Rightarrow \quad SE$

$$d_{new} = F(x_p + 2, y_p - \frac{3}{2})$$

$$= (x_p + 2)^2 + (y_p - \frac{3}{2})^2 - r^2$$

$d_{old} \geq 0 \quad \Rightarrow \quad SE$

$$d_{new} = d_{old} + (2x_p - 2y_p + 5)$$

Because,…

$d_{old} \geq 0 \quad \Rightarrow \quad SE$

$$d_{new} - d_{old} =$$

$$\underbrace{(2x_p + 3)}_{\text{From } \Delta_E \text{ calculation}} + \underbrace{(-3y_p + \frac{9}{4})}_{\text{From } \textit{new} \ y\text{-coordinate}} - \underbrace{(-y_p + \frac{1}{4})}_{\text{From } \textit{old} \ y\text{-coordinate}}$$

$d_{old} \geq 0 \quad \Rightarrow \quad SE$

That is,

$$d_{new} = d_{old} + (2x_p - 2y_p + 5)$$

$d_{old} \geq 0 \quad \Rightarrow \quad SE$

$$d_{new} = d_{old} + \Delta_{SE},$$

$$\boxed{\Delta_{SE} = 2x_p - 2y_p + 5}$$

Nonconstant $\Delta' s$

There are dependencies on x_p *and* y_p

in computing Δ_E *and* Δ_{SE}

Summary

- The only difference from the line algorithm is that point evaluations are needed for $\Delta' s$
- Algorithm structure is exactly the same

Initial Condition

- Let r be an integer. Start at $(0, r)$
- Next midpoint lies at $(1, r - \frac{1}{2})$
- So,

$$F(1, r - \frac{1}{2}) = 1 + (r^2 - r - \frac{1}{4}) - r^2$$

$$\boxed{= \frac{5}{4} - r}$$

Ellipses

- Evaluation is analogous
- Structure is same
- Have to work out the $\Delta' s$

Getting to Integers

- Note the previous algorithm involves *real* arithmetic
- Can we modify the algorithm to use integer arithmetic?

Integer Circle Algorithm

- Define a shift decision variable

$$h = d - \frac{1}{4}$$

- In the code, plug in $d = h + \frac{1}{4}$

Integer Circle Algorithm

- Now, the initialization is $h = 1 - r$
- So the initial value becomes

$$F(1, r - \frac{1}{2}) - \frac{1}{4} = (\frac{5}{4} - r) - \frac{1}{4}$$

$$= 1 - r$$

Integer Circle Algorithm

- Then, $d < 0$ becomes $h < \frac{1}{4}$
- Since *h* an integer

$$\boxed{h < \frac{1}{4} \quad \Leftrightarrow \quad h < 0}$$

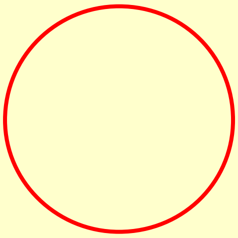
Integer Circle Algorithm

- But h begins as an integer
- And h gets incremented by integers
- Hence, we have an integer circle algorithm
- Sufficient to test for $h < 0$

Integer Circle Algorithm

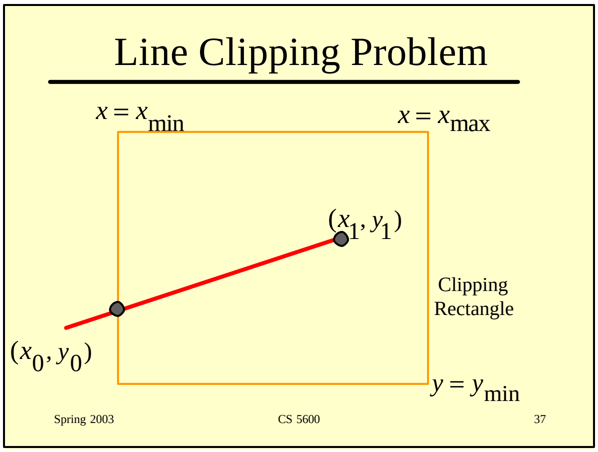
- But h begins as an integer
- And h gets incremented by integers
- Hence, we have an integer circle algorithm
- Sufficient to test for $h < 0$

End of Bresenham Circles

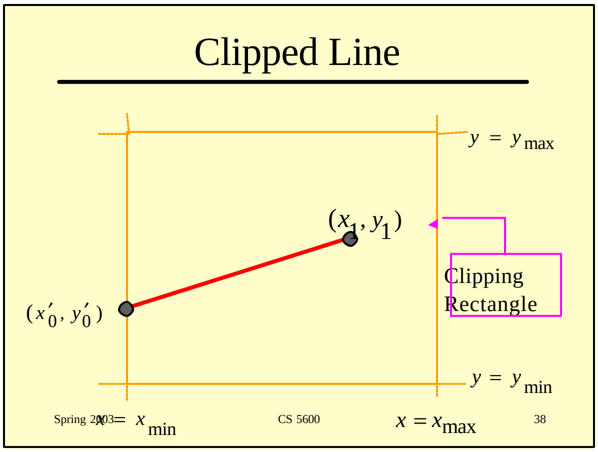


Another Digital Line Issue

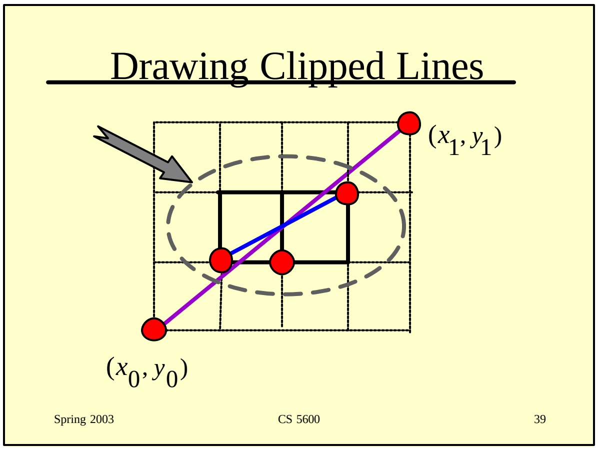
- Clipping Bresenham lines
- The integer slope is not the true slope
- Have to be careful
- More issues to follow



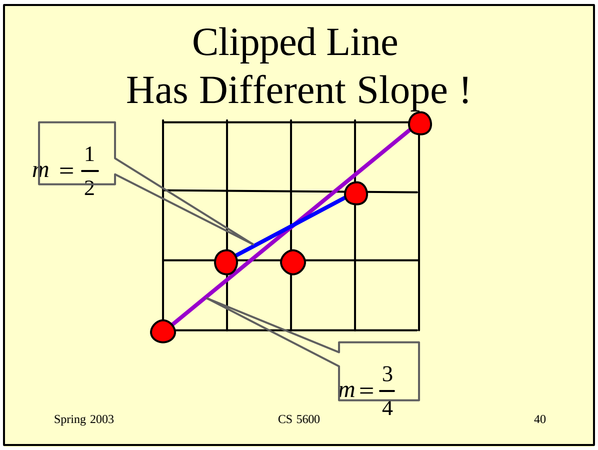
Line Clipping Problem



Clipped Line

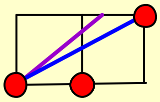


Drawing Clipped Lines



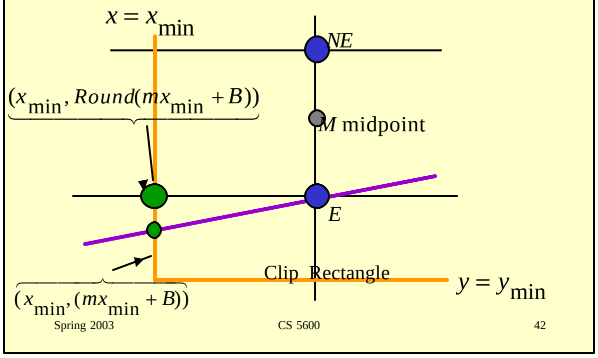
Clipped Line Has Different Slope !

Pick Right Slope to Reproduce Original Line Segment

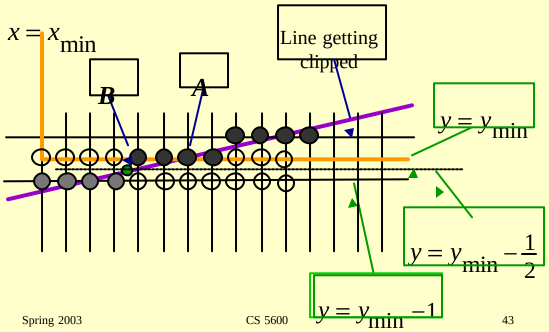


Zoom of previous situation

Clipping Against $x = x_{\min}$

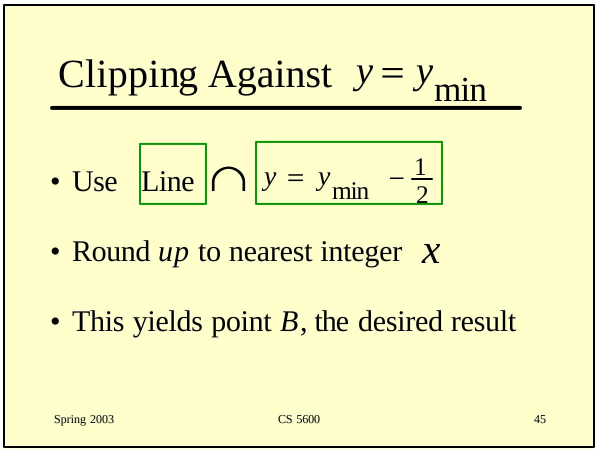


Clipping Against $y = y_{\min}$



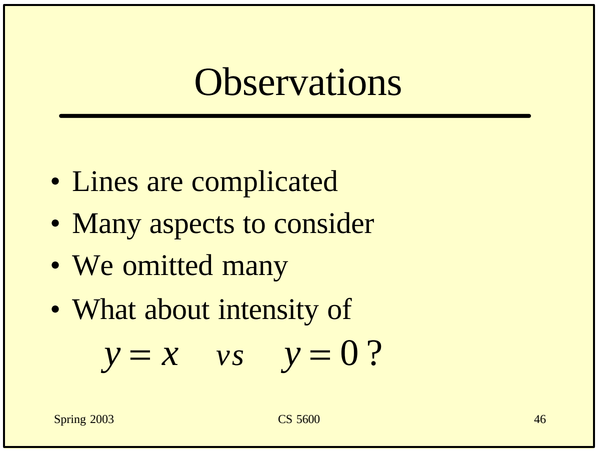
Clipping Against $y = y_{\min}$

- Situation is complicated
- Multiple pixels involved at $(y = y_{\min})$
- Want all of those pixels as "*in*"
- Analytic $\cap$, rounding x gives A
- We want point B



Clipping Against $y = y_{\min}$

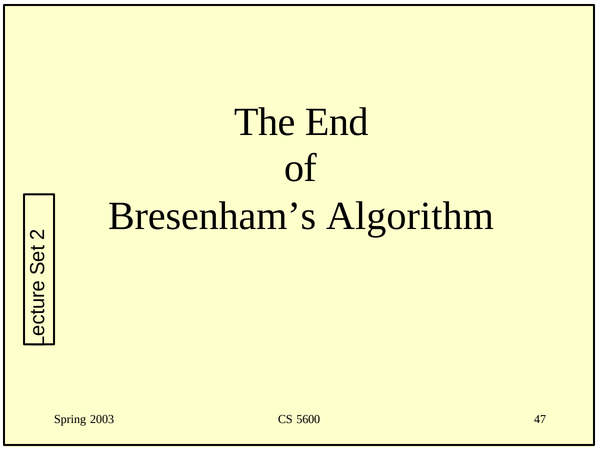
- Use $\boxed{\text{Line}} \cap \boxed{y = y_{\min} - \frac{1}{2}}$
- Round *up* to nearest integer x
- This yields point *B*, the desired result



Observations

- Lines are complicated
- Many aspects to consider
- We omitted many
- What about intensity of

$$y = x \quad vs \quad y = 0\,?$$



The End
of
Bresenham's Algorithm

Lecture Set 2