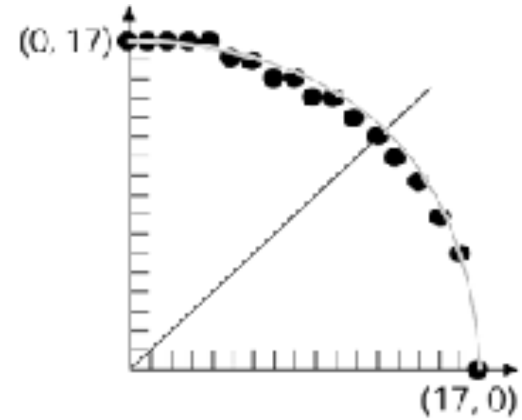
Outline

- Scan conversion of circles

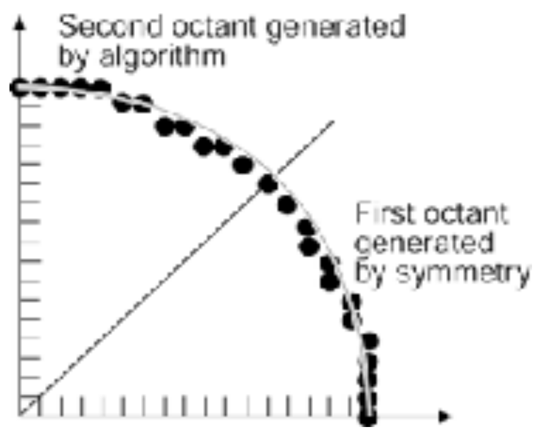
Scan Conversion of Circles

- Generalization of the line algorithm
- Assumptions:
  – circle at (0,0)
  – Fill 1/8 of the circle, then use symmetry



Not using the 8-way symmetry of a circle

Using the 8-way symmetry of a circle:



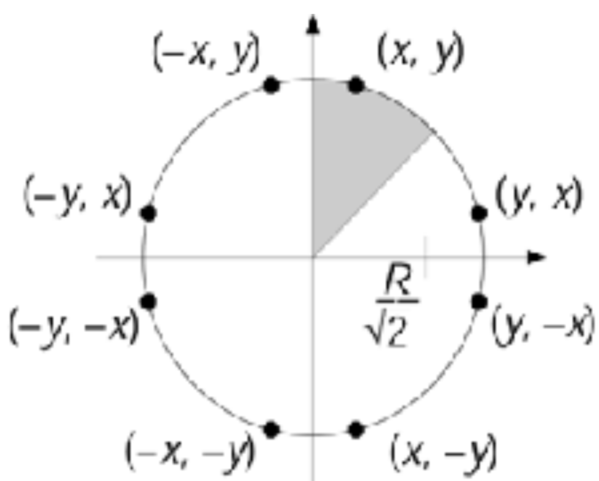
1994 Foley/VanDam/Finer/Huges/Phillips ICG

Scan Conversion of Circles

- Implicit representation of the circle function:

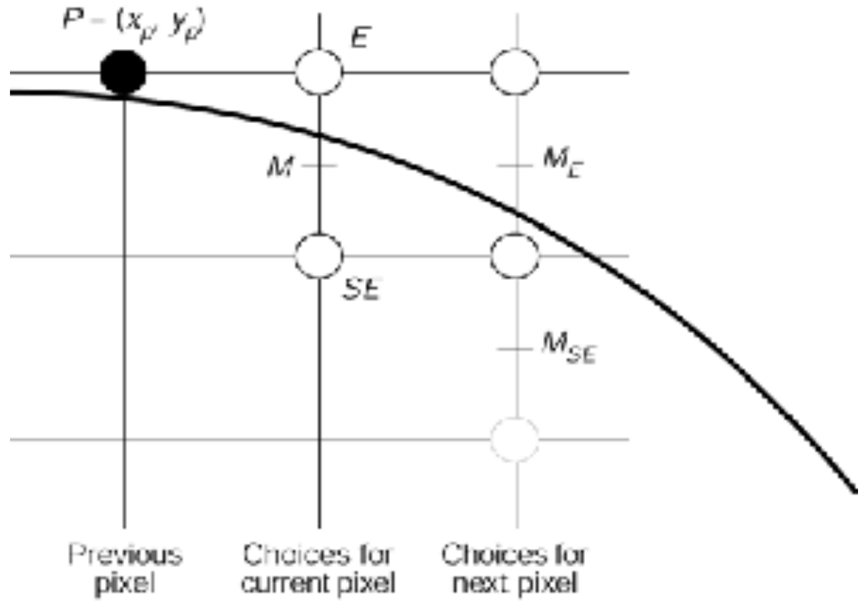
$$F(x,y) = x^2 + y^2 - R^2 = 0.$$

- Note: $F(x,y) < 0$ for points *inside* the circle, and $F(x,y) > 0$ for points *outside* the circle



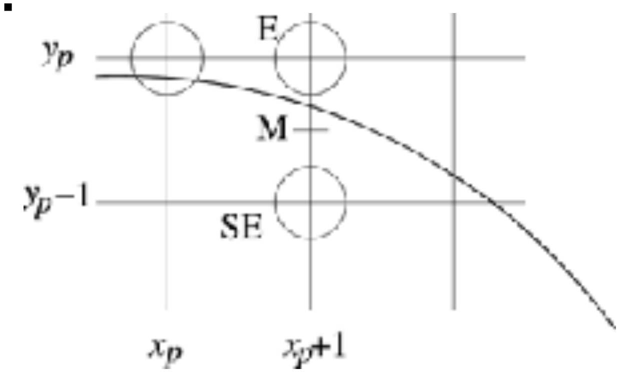
Scan Conversion of Circles

- Assume we finished pixel (x_p, y_p)
- What pixel to draw next? (going clockwise)
- Note: the slope of the circular arc is between 0 and –1
  - Hence, choice is between: *E* and *SE*
- Idea:
  If the circle passes above the midpoint *M*, then we go to *E* next, otherwise we go to *SE*



Scan Conversion of Circles

- We need a decision variable D:

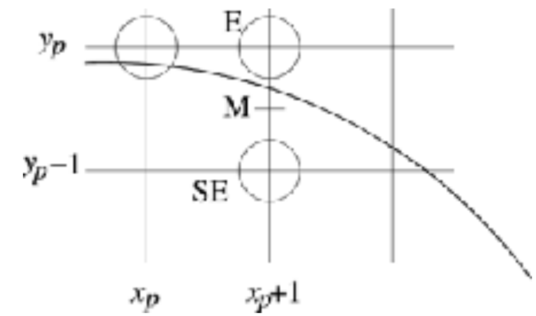


$$\begin{aligned} D &= F(M) = F(x_p + 1, y_p - \frac{1}{2}) \\ &= (x_p + 1)^2 + (y_p - \frac{1}{2})^2 - R^2. \end{aligned}$$

- If $D < 0$ then *M* is *below* the arc, hence the *E* pixel is closer to the line.
- If $D \geq 0$ then *M* is *above* the arc, hence the *SE* pixel is closer to the line.

Case I: When *E* is next

- What increment for computing a new *D*?
- Next midpoint is: $(x_p+2, y_p-(1/2))$



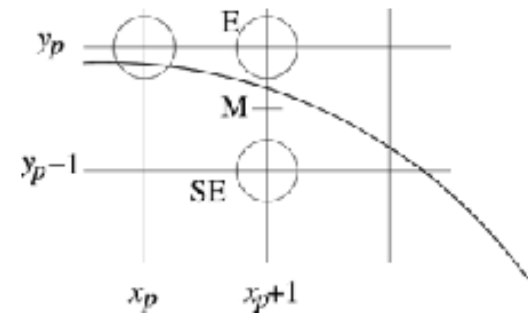
$$
\begin{aligned}
D_{new} &= F(x_p+2, y_p-\frac{1}{2}) \\
&= (x_p+2)^2+(y_p-\frac{1}{2})^2-R^2 \\
&= (x_p^2+4x_p+4)+(y_p-\frac{1}{2})^2-R^2 \\
&= (x_p^2+2x_p+1)+(2x_p+3)+(y_p-\frac{1}{2})^2-R^2 \\
&= (x_p+1)^2+(2x_p+3)+(y_p-\frac{1}{2})^2-R^2 \\
&= D+(2x_p+3).
\end{aligned}
$$

- Hence, increment by: $(2x_p+3)$

Pics/Math courtesy of Dave Mount @ UMD-CP

Case II: When *SE* is next

- What increment for computing a new *D*?
- Next midpoint is: $(x_p + 2,\ y_p - 1 - (1/2))$



$$
\begin{aligned}
D_{new} &= F(x_p + 2, y_p - \frac{3}{2}) \\
&= (x_p + 2)^2 + (y_p - \frac{3}{2})^2 - R^2 \\
&= (x_p^2 + 4x_p + 4) + (y_p^2 - 3y_p + \frac{9}{4}) - R^2 \\
&= (x_p^2 + 2x_p + 1) + (2x_p + 3) + (y_p^2 - y_p + \frac{1}{4}) + (-2y_p + \frac{8}{4}) - R^2 \\
&= (x_p + 1)^2 + (y_p - \frac{1}{2})^2 - R^2 + (2x_p + 3) + (-2y_p + 2) \\
&= D + (2x_p - 2y_p + 5)
\end{aligned}
$$

- Hence, increment by: $(2x_p - 2y_p + 5)$

Pics/Math courtesy of Dave Mount @ UMD-CP

Scan Conversion of Circles

- How to compute the *initial* value of D:
- We start with $x = 0$ and $y = R$, so the first midpoint is at $x = 1$, $y = R\text{-}1/2$:

$$\begin{aligned} D_{init} &= F(1, R - \frac{1}{2}) \\ &= 1 + (R - \frac{1}{2})^2 - R^2 \\ &= 1 + R^2 - R + \frac{1}{4} - R^2 \\ &= \frac{5}{4} - R. \end{aligned}$$

Scan Conversion of Circles

- Converting this to an integer algorithm:
  - Need only know if D is positive or negative
  - D & R are integers
  - Note D is incremented by an integer value
  - Therefore $D + 1/4$ is positive only when D is positive; it is safe to drop the $1/4$
- Hence: set the initial D to $1 - R$ (subtracting $1/4$)

Circle Scan Conversion Algorithm

- Given radius R and center (0, 0)
  - First point ⍰ (0, R)
- Initial decision parameter D = 1- R
- While x ≤ y
  - If (D < 0)
    - x++; D += 2x + 3;
  - else
    - x++; y--; D += 2(x - y) + 5
  - WritePoints(x,y)

WritePoints(x,y)

- Writes pixels to the seven other octants