CS 430
Computer Graphics

Circle Drawing and Clipping

Week 3, Lecture 6

David Breen, William Regli and Maxim Peysakhov
Department of Computer Science
Drexel University

1

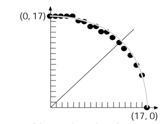
Outline

- Scan conversion of circles
- Clipping circles
- Scan conversion of ellipses

2

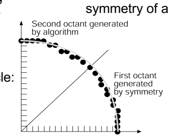
Scan Conversion of Circles

- 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:



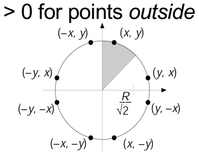
3

Scan Conversion of Circles

- Implicit representation of the circle function:

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

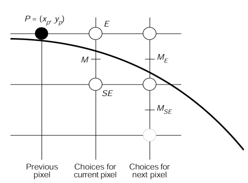
- Note: $F(x,y) < 0$ for points *inside* the circle, and $F(x,y) > 0$ for points *outside* the circle



4

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*

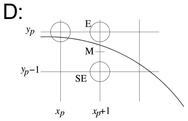


5

Scan Conversion of Circles

- We need a decision variable D:

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

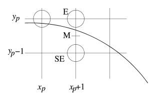


- If $D < 0$ then *M* is *below* the arc, hence the *E* pixel is closer to the line.
- If $D \geq 0$ then *M* is *above* the arc, hence the *SE* pixel is closer to the line.

6

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: $(2x_p + 3)$

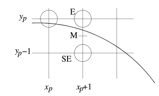
7

Pics/Math courtesy of Dave Mount @ UMD-CP

7

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: $(2x_p - 2y_p + 5)$

8

Pics/Math courtesy of Dave Mount @ UMD-CP

8

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-1/2*:

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

9

9

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*)

10

10

Circle Scan Conversion Algorithm

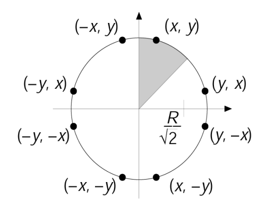
- Given radius R and center (0, 0)
  - First point ➜ (0, R)
- Initial decision parameter D = 1- R
- While x ≤ y
  - If (D < 0)
    - x++; D += 2x + 3;
  - else
    - x++; y--; D += 2(x - y) + 5
  - WritePoints(x,y)

11

11

WritePoints(x,y)

- Writes pixels to the seven other octants

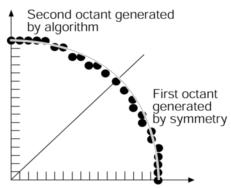


12

12

Clipping Circles

- Accept/Reject test
  - Does bounding box of the circle intersect with clipping box?
- If yes, condition pixel write on clipping box inside/outside test



13

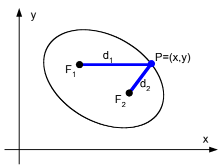
13

Properties of Ellipses

- "Elongated circle"
- For all points on ellipse, sum of distances to foci is constant

  $d_1 + d_2 = const$

- If $F_1=(x_1, y_1)$ and $F_2=(x_2, y_2)$ then $\sqrt{(x-x_1)^2+(y-y_1)^2}+\sqrt{(x-x_2)^2+(y-y_2)^2}=const$



14

14

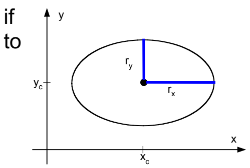
Properties of Ellipses

- Equation simplified if ellipse axis parallel to coordinate axis

  $$\left(\frac{x-x_c}{r_x}\right)^2+\left(\frac{y-y_c}{r_y}\right)^2=1$$

- Parametric form

  $$x = x_c + r_x \cos\theta$$
  $$y = y_c + r_y \sin\theta$$



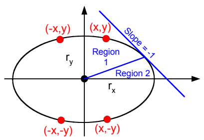
15

15

Symmetry Considerations

- 4-way symmetry
- Unit steps in x until reach region boundary
- Switch to unit steps in y

$$f(x,y)=r_y^2x^2+r_x^2y^2-r_x^2r_y^2$$
$$\frac{dy}{dx}=-\frac{r_y^2x}{r_x^2y}$$
$$\frac{dy}{dx}=-1$$
$$r_y^2x=r_x^2y$$



- Step in x while

  $r_y^2x < r_x^2y$

- Switch to steps in y when

  $r_y^2x \ge r_x^2y$

16

16

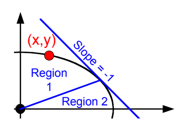
Midpoint Algorithm (initializing)

- Similar to circles
- The initial value for region 1

$$D_{init1} = f(1, r_y - \tfrac{1}{2})$$
$$= r_y^2 - r_x^2r_y + \tfrac{1}{4}r_x^2$$

- The initial value for region 2

$$D_{init2} = f(x_p + \tfrac{1}{2}, y_p - 1)$$
$$= r_y^2(x_p+\tfrac{1}{2})^2 + r_x^2(y_p-1)^2 - r_x^2r_y^2$$



- We have initial values, now we need the increments

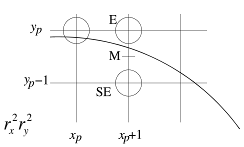
17

17

Making a Decision

- Computing the decision variable

$$D = f(x_p+1, y_p-\tfrac{1}{2})$$
$$= r_y^2(x_p+1)^2 + r_x^2(y_p-\tfrac{1}{2})^2 - r_x^2r_y^2$$



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

18

18

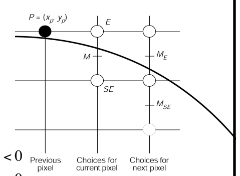
Computing the Increment

E case

$$D_{new} = f(x_p + 2, y_p - \tfrac{1}{2})$$
$$= r_y^2 (x_p + 2)^2 + r_x^2 (y_p - \tfrac{1}{2})^2 - r_x^2 r_y^2$$
$$= D_{old} + r_y^2 (2x_p + 3)$$

SE case

$$D_{new} = f(x_p + 2, y_p - \tfrac{3}{2})$$
$$= r_y^2 (x_p + 2)^2 + r_x^2 (y_p - \tfrac{3}{2})^2 - r_x^2 r_y^2$$
$$= D_{old} + r_y^2 (2x_p + 3) + r_x^2 (-2y_p + 2)$$



$$increment = \begin{cases} r_y^2 (2x_p + 3) & D_{old} < 0 \\ r_y^2 (2x_p + 3) + r_x^2 (-2y_p + 2) & D_{old} \geq 0 \end{cases}$$

19

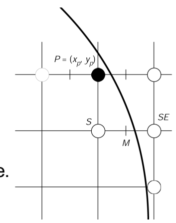
19

Computing the Increment in 2nd Region

- Decision variable in 2nd region

$$D = f(x_p + \tfrac{1}{2}, y_p - 1)$$
$$= r_y^2 (x_p + \tfrac{1}{2})^2 + r_x^2 (y_p - 1)^2 - r_x^2 r_y^2$$

- If $D < 0$ then *M* is *left of* the arc, hence the *SE* pixel is closer to the line.
- If $D \geq 0$ then *M* is *right of* the arc, hence the *S* pixel is closer to the line.



21

21

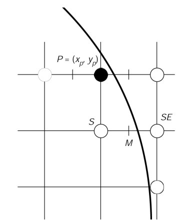
Computing the Increment in 2nd Region

SE case

$$D_{new} = f(x_p + \tfrac{3}{2}, y_p - 2)$$
$$= r_y^2 (x_p + \tfrac{3}{2})^2 + r_x^2 (y_p - 2)^2 - r_x^2 r_y^2$$
$$= D_{old} + r_x^2 (-2y_p + 3) + r_y^2 (2x_p + 2)$$

S case

$$D_{new} = f(x_p + \tfrac{1}{2}, y_p - 2)$$
$$= r_y^2 (x_p + \tfrac{1}{2})^2 + r_x^2 (y_p - 2)^2 - r_x^2 r_y^2$$
$$= D_{old} + r_x^2 (-2y_p + 3)$$



$$increment = \begin{cases} r_x^2 (-2y_p + 3) + r_y^2 (2x_p + 2) & D_{old} < 0 \\ r_x^2 (-2y_p + 3) & D_{old} \geq 0 \end{cases}$$

22

22

Midpoint Algorithm for Ellipses

Region 1

Set first point to (0,ry)
Set the Decision variable to
$D_{init1} = r_y^2 - r_x^2 r_y + \tfrac{1}{4} r_x^2$
Loop $(x = x+1)$
If $D < 0$ then pick *E* and
$D \mathrel{+}= r_y^2 (2x_p + 3)$
If $D \geq 0$ then pick *SE* and
$D \mathrel{+}= r_y^2 (2x_p + 3) + r_x^2 (-2y_p + 2)$

Until $2r_y^2 x_k \geq 2r_x^2 y_k$

Region 2

Set first point to the last computed
Set the Decision variable to
$D_{init2} = r_y^2 (x_p + \tfrac{1}{2})^2 + r_x^2 (y_p - 1)^2 - r_x^2 r_y$
Loop $(y = y - 1)$
If $D < 0$ then pick *SE* and
$D \mathrel{+}= r_y^2 (2x_p + 2) + r_x^2 (-2y_p + 3)$
If $D \geq 0$ then pick *S* and
$D \mathrel{+}= r_x^2 (-2y_p + 3)$

Until $y < 0$
Use symmetry to complete the ellipse

23

23