Scan Conversion 2

Scan Converting Circles

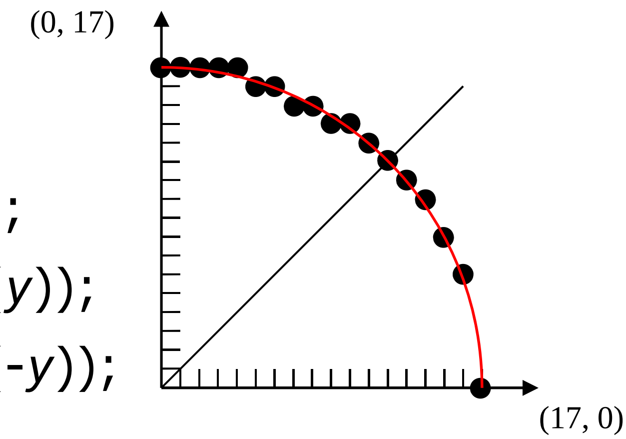
Version 1: <u>really</u> bad

For $x = -R$ to R

$y = \text{sqrt}(R * R - x * x)$;

Pixel (round(x), round(y));

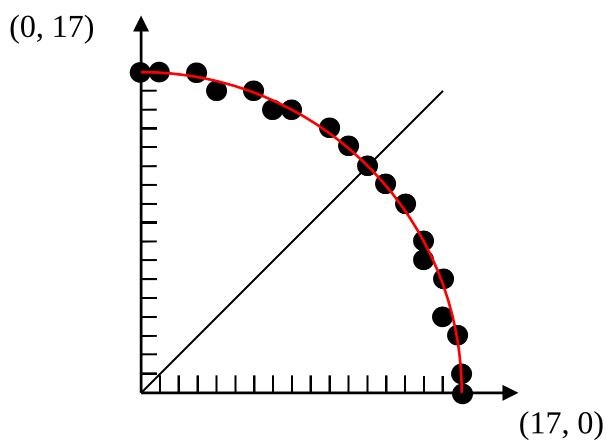
Pixel (round(x), round($-y$));



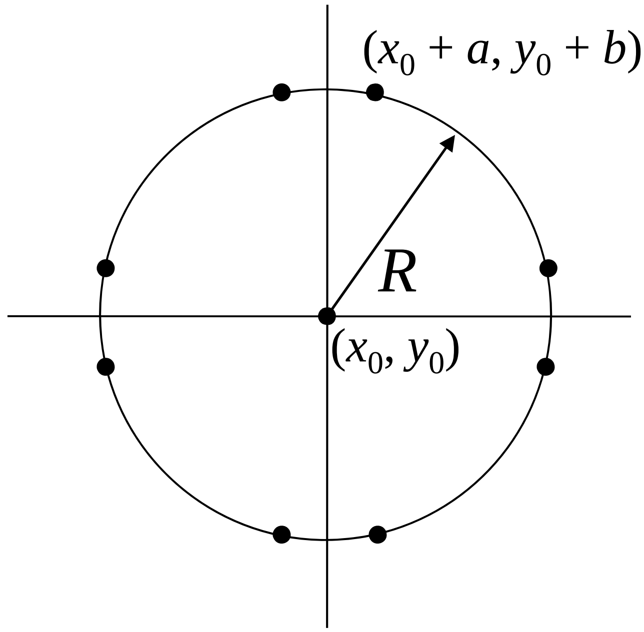
Version 2: slightly less bad

For $x = 0$ to 360

Pixel (round ($R \bullet \cos(x)$), round($R \bullet \sin(x)$));



Version 3 — Use Symmetry



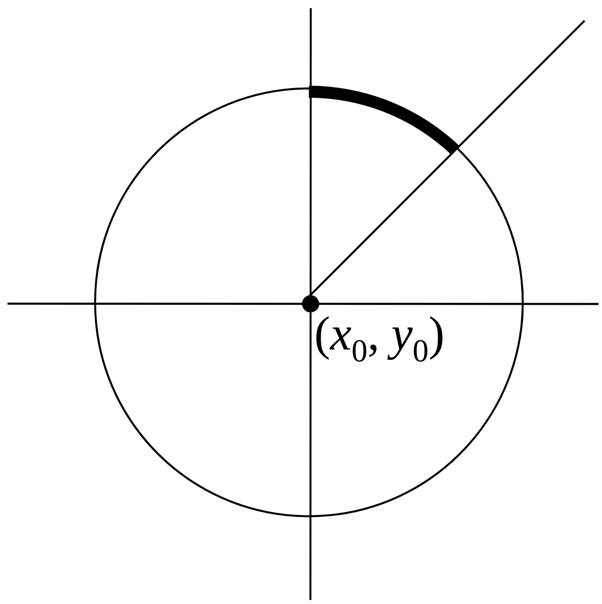
$(x\text{-}x_0)^2 + (y\text{-}y_0)^2 = R^2$

- Symmetry: If $(x_0 + a, y_0 + b)$ is on circle
  - also $(x_0 \pm a, y_0 \pm b)$ and $(x_0 \pm b, y_0 \pm a)$; hence 8-way symmetry.

- Reduce the problem to finding the pixels for 1/8 of the circle

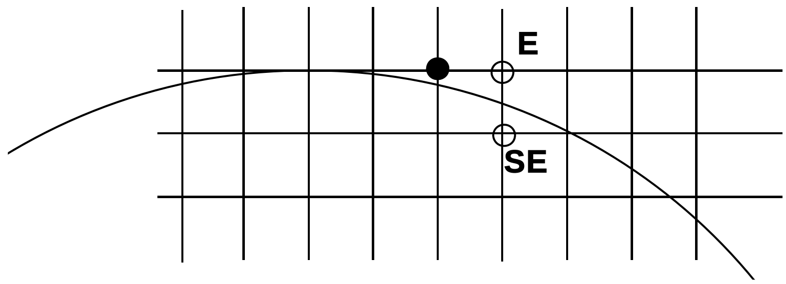
Using the Symmetry

- Scan top right 1/8 of circle of radius *R*



- Circle starts at (x_0, $y_0 + R$)
- Let's use another incremental algorithm with decision variable evaluated at midpoint

Sketch of Incremental Algorithm




```
x = x0; y = y0 + R; Pixel(x, y);
for (x = x0+1; (x – x0) > (y – y0); x++) {
   if (decision_var < 0) {
    /* move east */
    update decision_var;
   }
   else {
    /* move south east */
    update decision_var;
    y--;
   }
   Pixel(x, y);
}
```

- Note: can replace all occurrences of x_0, y_0 with 0, 0 and Pixel ($x_0 + x$, $y_0 + y$) with Pixel (x, y)
- Shift coordinates by ($-x_0$, $-y_o$)

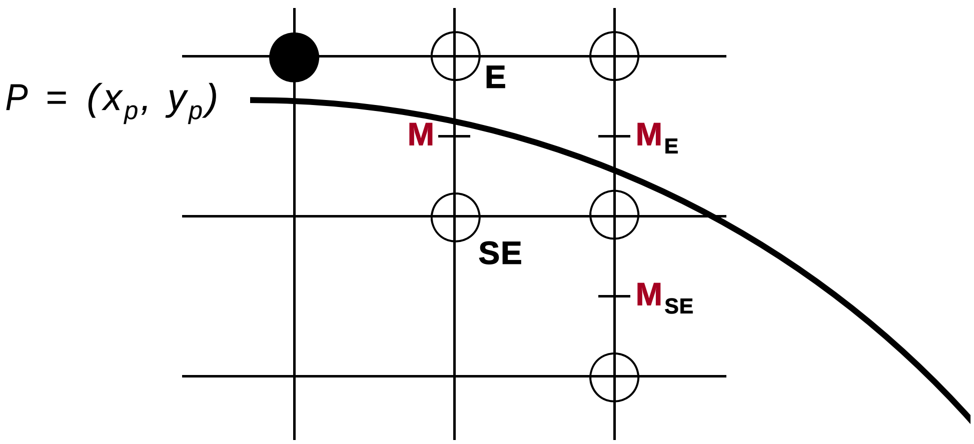
What we need for Incremental Algorithm

- Decision variable
  - negative if we move E, positive if we move SE (or vice versa).
- Follow line strategy: Use implicit equation of circle

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

  $f(x,y)$ is zero on circle, negative inside, positive outside
- If we are at pixel (x, y)
  - examine ($x + 1$, y) and ($x + 1$, $y - 1$)
- Compute f at the midpoint

Decision Variable



- Evaluate $f(x,y) = x^2 + y^2 - R^2$ at the point

$$\left(x+1, y-\frac{1}{2}\right)$$

- We are asking: "Is

$$f\left(x+1, y-\frac{1}{2}\right) = (x+1)^2 + \left(y-\frac{1}{2}\right)^2 - R^2$$

positive or negative?" (it is zero on circle)

- If **negative**, midpoint inside circle, **choose E**
  - *vertical* distance to the circle is less at $(x + 1, y)$ than at $(x + 1, y-1)$.

- If **positive**, opposite is true, **choose SE**

The right decision variable?

- Decision based on vertical distance
- Ok for lines, since *d* and d_{vert} are proportional
- For circles, not true:

$$d((x+1, y), Circ) = \sqrt{(x+1)^2 + y^2} - R$$

$$d((x+1, y-1), Circ) = \sqrt{(x+1)^2 + (y-1)^2} - R$$

- Which *d* is closer to zero? (i.e. which of the two values below is closer to *R*):

$$\sqrt{(x+1)^2 + y^2} \quad or \quad \sqrt{(x+1)^2 + (y-1)^2}$$

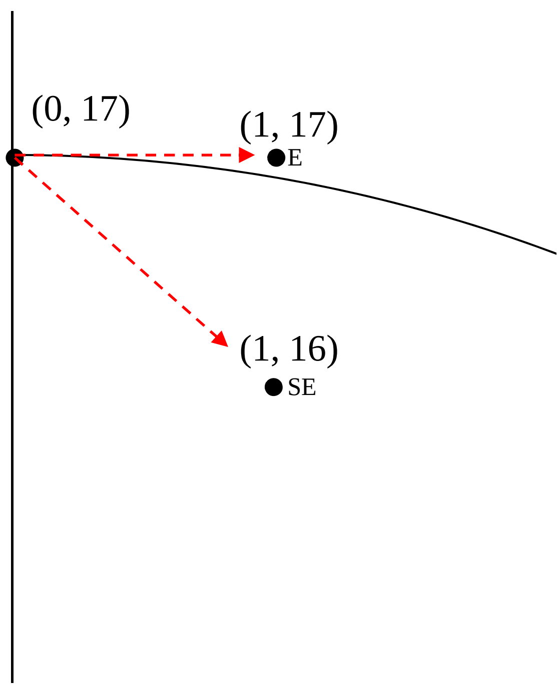
Alternate Phrasing (1/3)

- We could ask instead:

  "Is $(x + 1)^2 + y^2$ or $(x + 1)^2 + (y - 1)^2$ closer to R^2?"

- The two values in equation above differ by

$$[(x+1)^2 + y^2] - [(x+1)^2 + (y-1)^2] = 2y - 1$$



$f_E = 1^2 + 17^2 = 290$

$f_{SE} = 1^2 + 16^2 = 257$

$f_E - f_{SE} = 290 - 257 = 33$

$2y - 1 = 2(17) - 1 = 33$

Alternate Phrasing (2/3)

- The second value, which is always less, is *closer* if its difference from R^2 is less than

$$\left(\frac{1}{2}\right)(2y-1)$$

i.e., if

$$R^2 - [(x+1)^2 + (y-1)^2] < \tfrac{1}{2}(2y-1)$$

then

$$0 < y - \tfrac{1}{2} + (x+1)^2 + (y-1)^2 - R^2$$

so

$$0 < (x+1)^2 + y^2 - 2y + 1 + y - \tfrac{1}{2} - R^2$$

so

$$0 < (x+1)^2 + y^2 - y + \tfrac{1}{2} - R^2$$

so

$$0 < (x+1)^2 + \left(y - \frac{1}{2}\right)^2 + \frac{1}{4} - R^2$$

Alternate Phrasing (3/3)

- The *radial* distance decision is whether

$$d1 = (x+1)^2 + \left(y - \frac{1}{2}\right)^2 + \frac{1}{4} - R^2$$

is positive or negative

- And the *vertical* distance decision is whether

$$d2 = (x+1)^2 + \left(y - \frac{1}{2}\right)^2 - R^2$$

is positive or negative; $d1$ and $d2$ are $\frac{1}{4}$ apart.

- The integer $d1$ is positive only if $d2 + \frac{1}{4}$ is positive (except special case where $d2 = 0$).

Incremental Computation, Again (1/2)

- How to compute the value of

$$f(x,y) = (x+1)^2 + \left(y - \frac{1}{2}\right)^2 - R^2$$

at successive points?

- Answer: Note that

$$f(x+1, y) - f(x,y)$$

is just

$$\Delta_E(x,y) = 2x + 3$$

and that

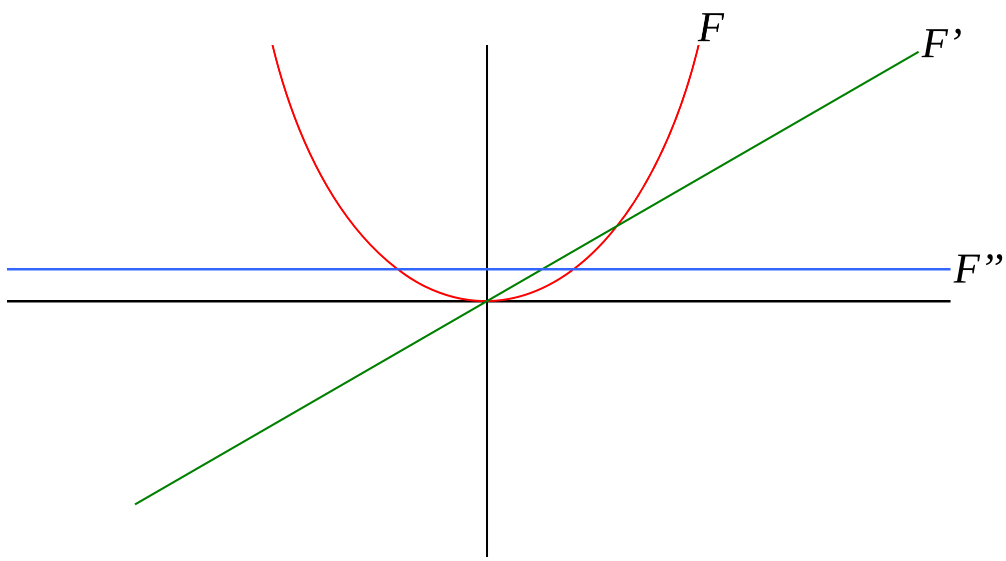
$$f(x+1, y-1) - f(x,y)$$

is just

$$\Delta_{SE}(x,y) = 2x + 3 - 2y + 2$$

Incremental Computation (2/2)

- If we move E, update by adding $2x + 3$
- If we move SE, update by adding $2x + 3 - 2y + 2$.
- Forward differences of a 1st degree polynomial are constants and those of a 2nd degree polynomial are 1st degree polynomials
  - this "first order forward difference," like a partial derivative, is one degree lower



Second Differences (1/2)

- The function $\Delta_{\mathrm{E}}(x, y) = 2x + 3$ is linear, hence amenable to incremental computation:

$$\Delta_{\mathrm{E}}(x+1, y) - \Delta_{\mathrm{E}}(x, y) = 2$$

$$\Delta_{\mathrm{E}}(x+1, y-1) - \Delta_{\mathrm{E}}(x, y) = 2$$

- Similarly

$$\Delta_{\mathrm{SE}}(x+1, y) - \Delta_{\mathrm{SE}}(x, y) = 2$$

$$\Delta_{\mathrm{SE}}(x+1, y-1) - \Delta_{\mathrm{SE}}(x, y) = 4$$

Second Differences (2/2)

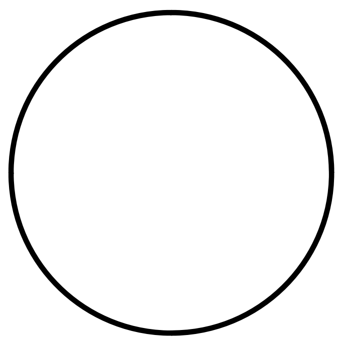
- For any step, can compute new $\Delta_E(x, y)$ from old $\Delta_E(x, y)$ by adding appropriate second constant increment – update delta terms as we move.
  - This is also true of $\Delta_{SE}(x, y)$
- Having drawn pixel (a,b), decide location of new pixel at $(a + 1, b)$ or $(a + 1, b - 1)$, using previously computed $\Delta(a, b)$.
- Having drawn new pixel, must update $\Delta(a, b)$ for next iteration; need to find either $\Delta(a + 1, b)$ or $\Delta(a + 1, b - 1)$ depending on pixel choice
- Must add $\Delta_E(a, b)$ or $\Delta_{SE}(a, b)$ to $\Delta(a, b)$
- So we…
  - Look at $\Delta(i)$ to decide which to draw next, update x and y
  - Update d using $\Delta_E(a,b)$ or $\Delta_{SE}(a,b)$
  - Update each of $\Delta_E(a,b)$ and $\Delta_{SE}(a,b)$ for future use
  - Draw pixel

Midpoint Eighth Circle Algorithm

```
MEC (R) /* 1/8th of a circle w/ radius R */
{
    int x = 0, y = R;
    int delta_E, delta_SE;
    float decision;
    delta_E = 2*x + 3;
    delta_SE = 2(x-y) + 5;
    decision = (x+1)*(x+1) + (y + 0.5)*(y + 0.5) –R*R;
    Pixel(x, y);
    while( y > x ) {
        if (decision > 0) {/* Move east */
            decision += delta_E;
            delta_E += 2; delta_SE += 2; /*Update delta*/
        }
        else {/* Move SE */
            y--;
            decision += delta_SE;
            delta_E += 2; delta_SE += 4; /*Update delta*/
        }
        x++;
        Pixel(x, y);
    }
}
```

Analysis

- Uses floats!
- 1 test, 3 or 4 additions per pixel
- Initialization can be improved
- Multiply everything by 4 → No Floats!
  - Makes the components even, but sign of decision variable remains same

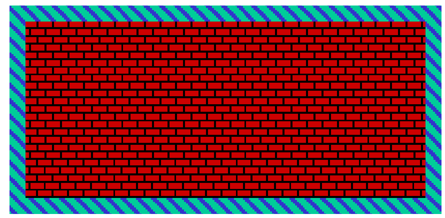


Questions

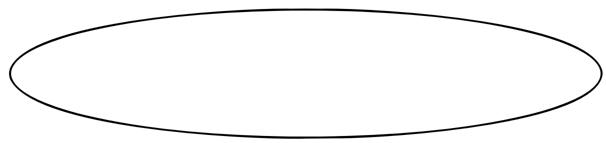
- Are we getting <u>all</u> pixels whose distance from the circle is less than ½?
- Why is ***y > x*** the right stopping criterion?
- What if it were an ellipse?

Other Scan Conversion Problems

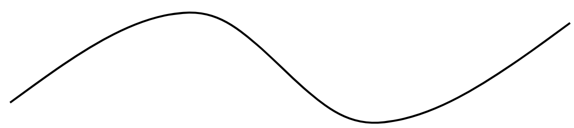
- Patterned primitives



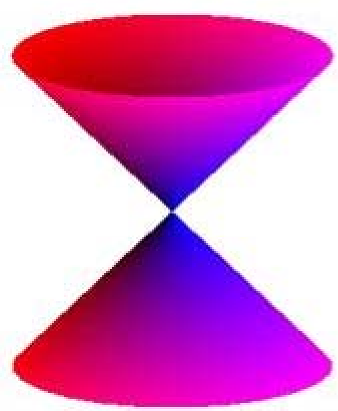
- Aligned Ellipses



- Non-integer primitives

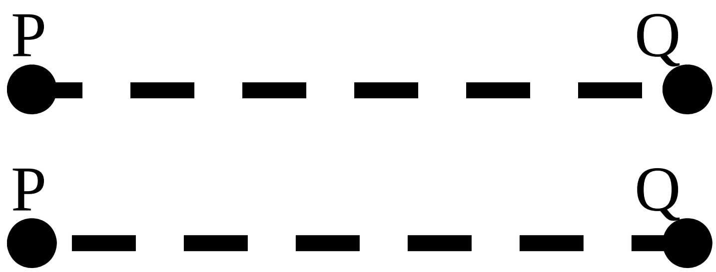


- General conics



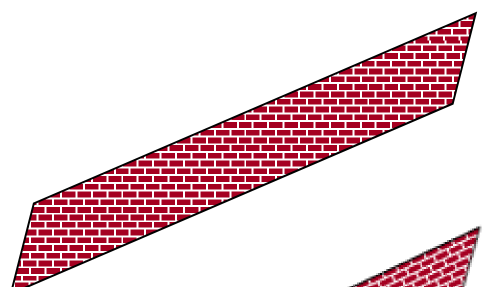
Patterned Lines

- Patterned line from *P* to *Q* is not same as patterned line from *Q* to *P*.

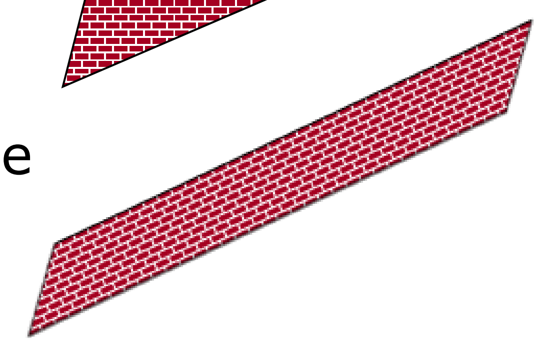


- Patterns can be geometric or cosmetic
  - Cosmetic: Texture applied after transformations
  - Geometric: Pattern subject to transformations

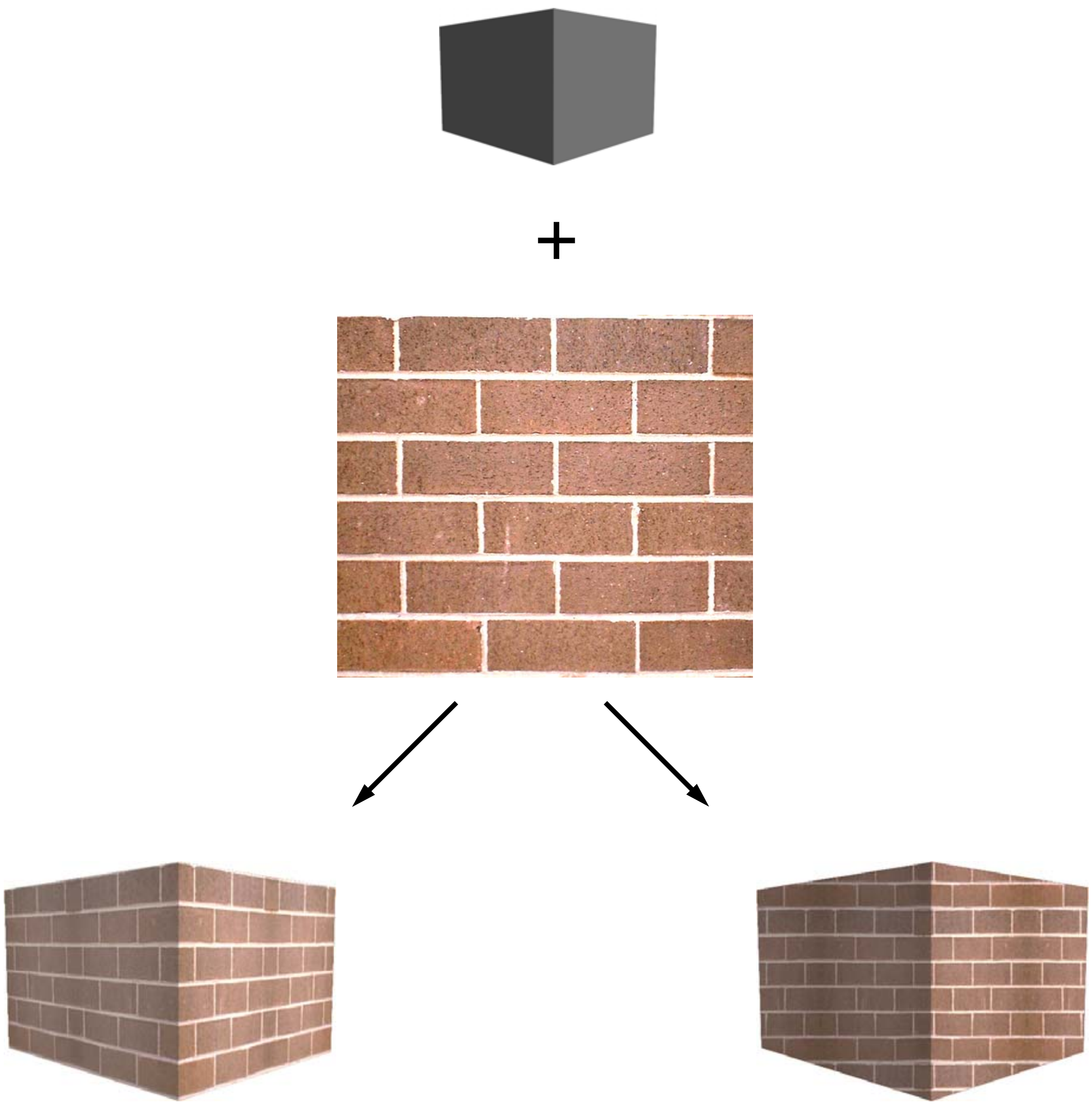
Cosmetic patterned line



Geometric patterned line



Geometric Pattern vs. Cosmetic Pattern



Geometric
(Perspectivized/Filtered)

Cosmetic
(Contact Paper)

Aligned Ellipses

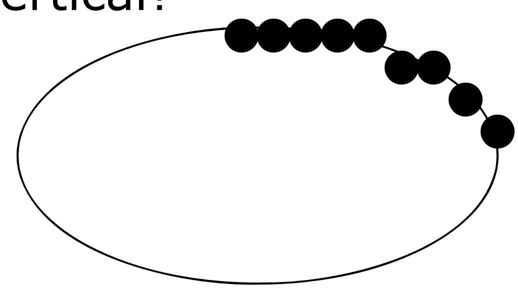
- Equation is

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

i.e,

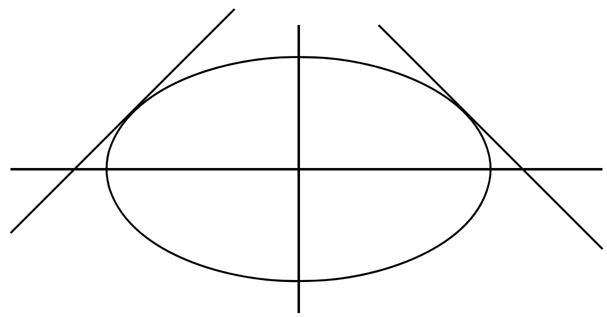
$$b^2x^2 + a^2y^2 = a^2b^2$$

- Computation of Δ_E and Δ_{SE} is similar
- Only 4-fold symmetry
- When do we stop stepping horizontally and switch to vertical?



Direction Changing Criterion (1/2)

- When absolute value of slope of ellipse is more than 1:



- How do you check this? At a point (x,y) for which $f(x,y) = 0$, a vector <u>perpendicular</u> to the level set is $\nabla f(x,y)$ which is

$$\left[\frac{\partial f}{\partial x}(x,y), \frac{\partial f}{\partial y}(x,y)\right]$$

- This vector points more <u>right</u> than <u>up</u> when

$$\frac{\partial f}{\partial x}(x,y) - \frac{\partial f}{\partial y}(x,y) > 0$$

Direction Changing Criterion (2/2)

- In our case,

$$\frac{\partial f}{\partial x}(x, y) = 2a^2 x$$

and

$$\frac{\partial f}{\partial y}(x, y) = 2b^2 y$$

so we check for

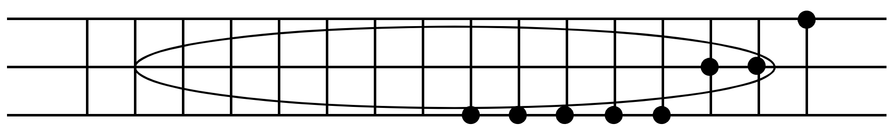
$$2a^2 x - 2b^2 y > 0$$

i.e.

$$a^2 x - b^2 y > 0$$

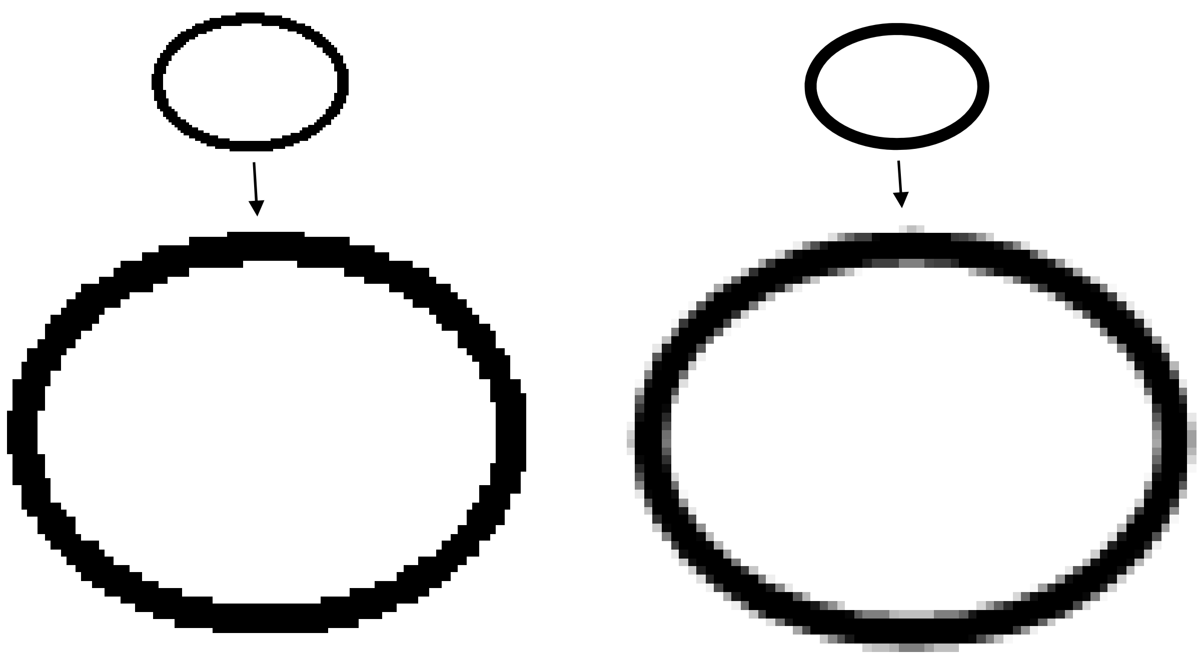
- This, too, can be computed incrementally

Problems with Aligned Ellipses



- Now in ENE octant, not ESE octant

- This problem is artifact of *aliasing* – much more on this later



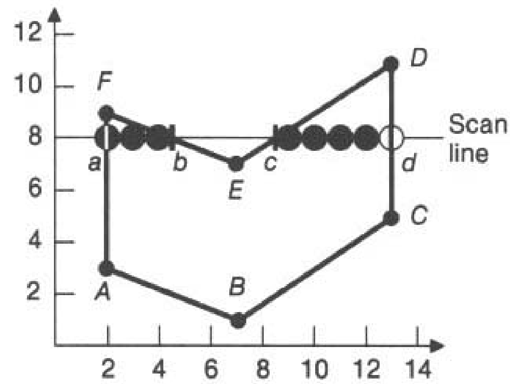
Non – Integer Primitives and General Conics

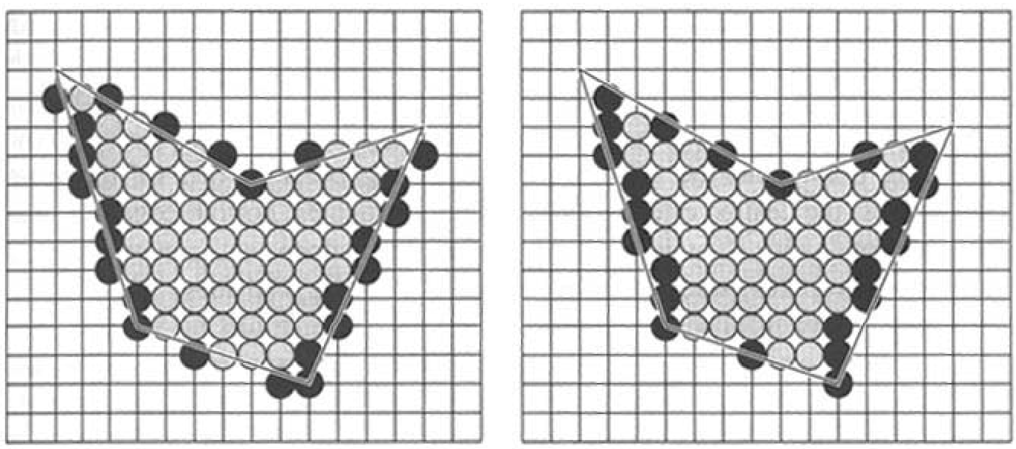
- **Non-Integer Primitives**
  - Initialization is harder
  - Endpoints are hard, too
    - making Line (*P*,*Q*) and Line (*Q*,*R*) join properly is a good test
  - Symmetry is lost

- **General Conics**
  - Very hard--the octant-changing test is tougher, the difference computations are tougher, etc.
    - do it only if you have to.

  - Note that drawing gray-scale conics is easier than drawing B/W conics

Generic Polygons





(More information and these pictures on page 92-93 of textbook)