Output Primitives

Output Primitives

- **Output Primitives:** Basic geometric structures (points, straight line segment, circles and other conic sections, quadric surfaces, spline curve and surfaces, polygon color areas, and character strings)
- These picture components are often defined in a continuous space.

Output Primitives

- In order to draw the primitive objects, one has to first **scan convert** the object.

- **Scan convert:** Refers to be operation of finding out the location of pixels to the intensified and then setting the values of corresponding bits, in the graphic memory, to the desired intensity code.

Output Primitives

- Each pixel on the display surface has a finite size depending on the screen resolution and hence a pixel ***cannot represent*** a single mathematical point.

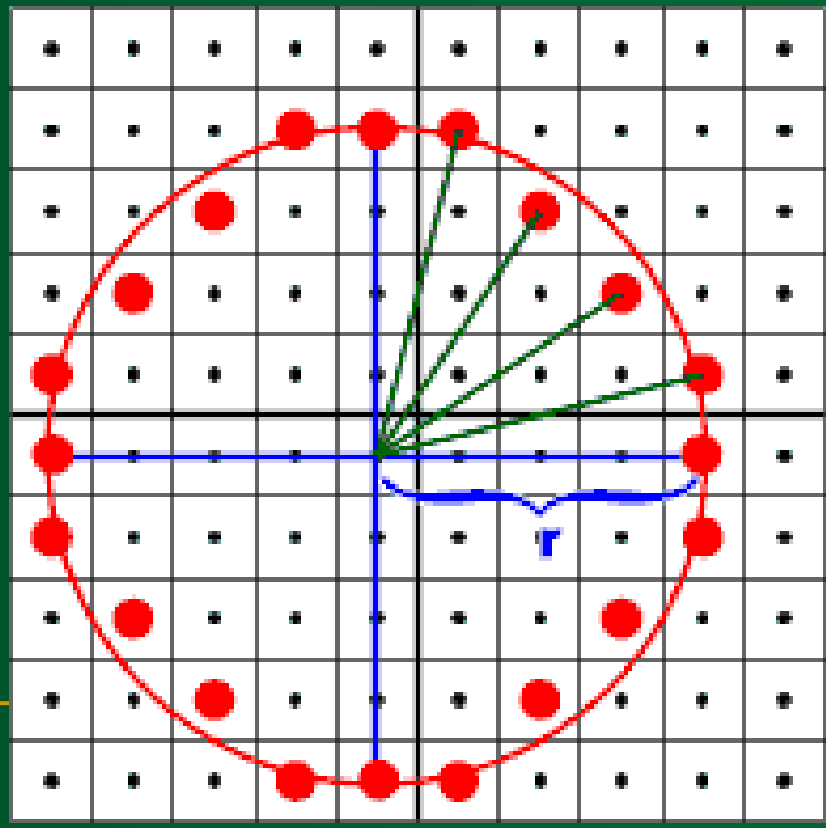
Circle Generation Algorithms

- The equation of a circle:

$$(x - x_0)^2 + (y - y_0)^2 = r^2$$

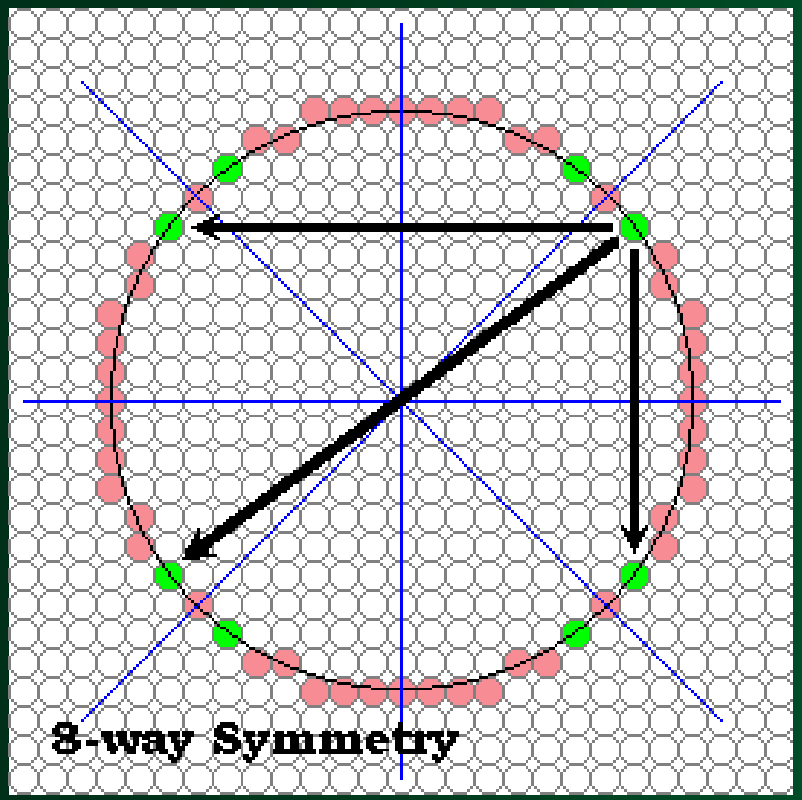
We could solve for y in terms of x

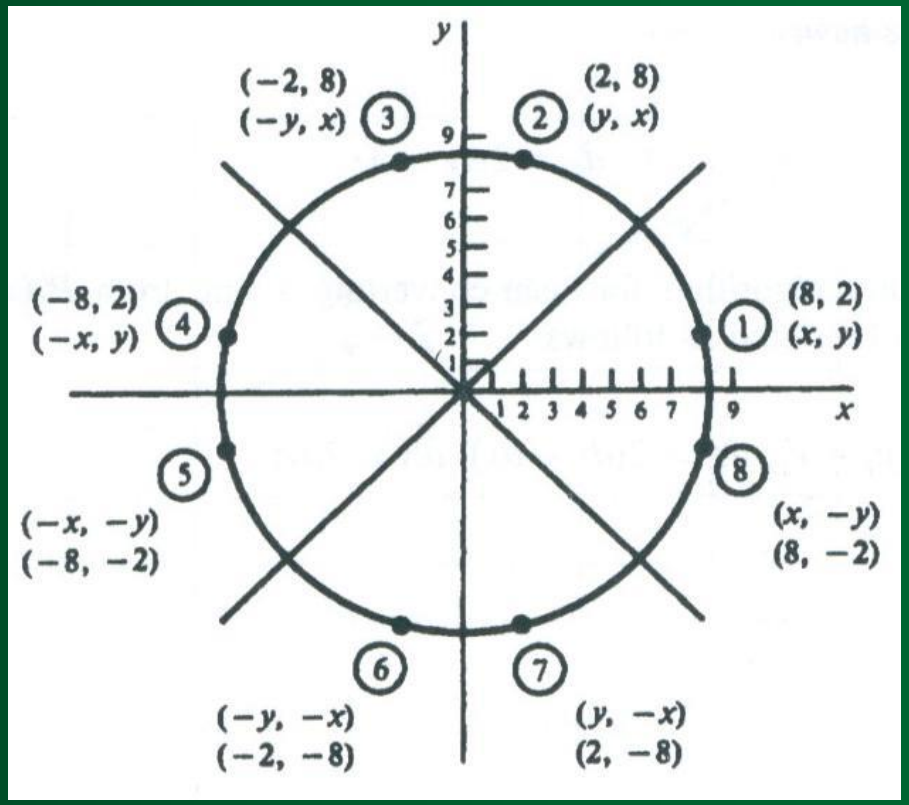
$$y = y_0 \pm \sqrt{r^2 - (x - x_0)^2}$$



Circle Generation Algorithms

- Computation can be reduced by considering the symmetry of circles



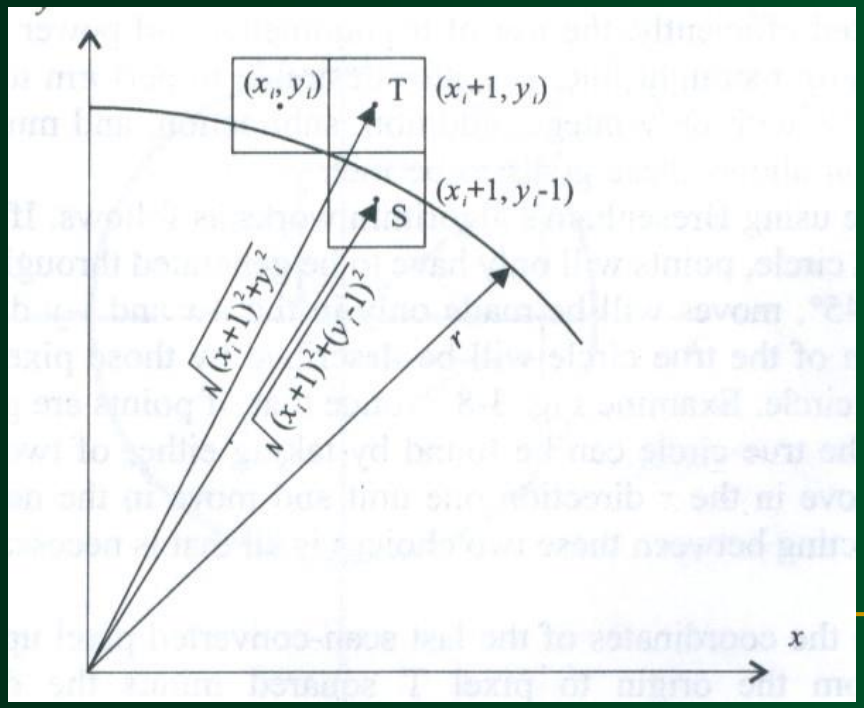


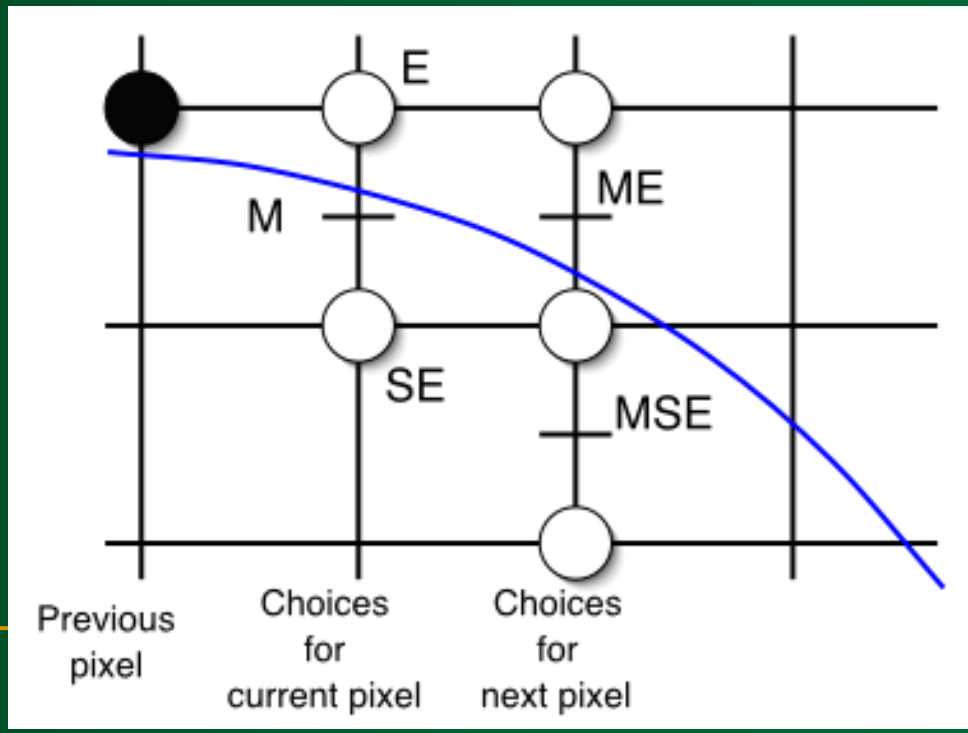
8 -Way symmetry

Midpoint Circle Algorithm

Circle Generation Algorithms

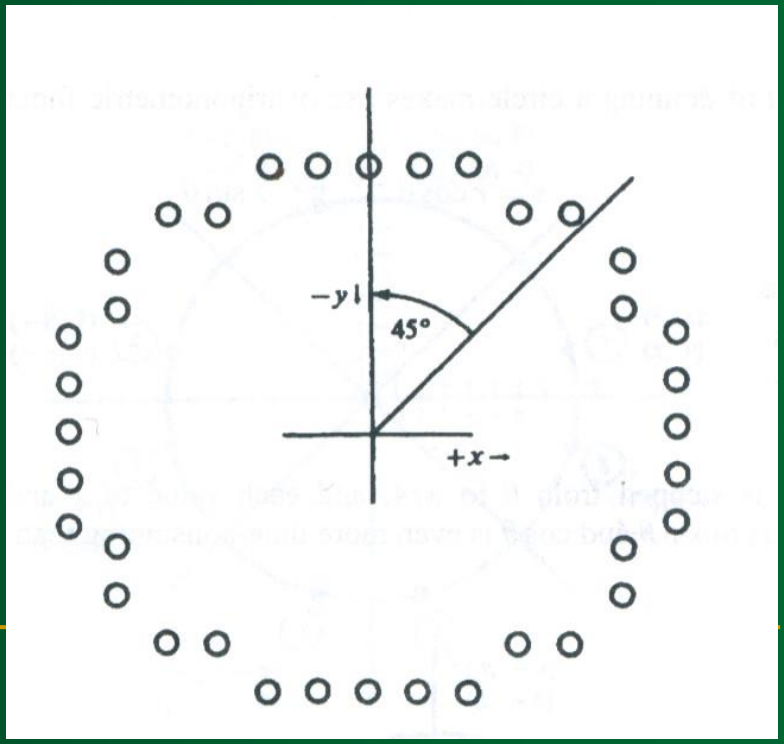
- As in the line algorithm, we sample at unit intervals and determine the closet pixel position to the circle path at each step.





Circle Generation Algorithms

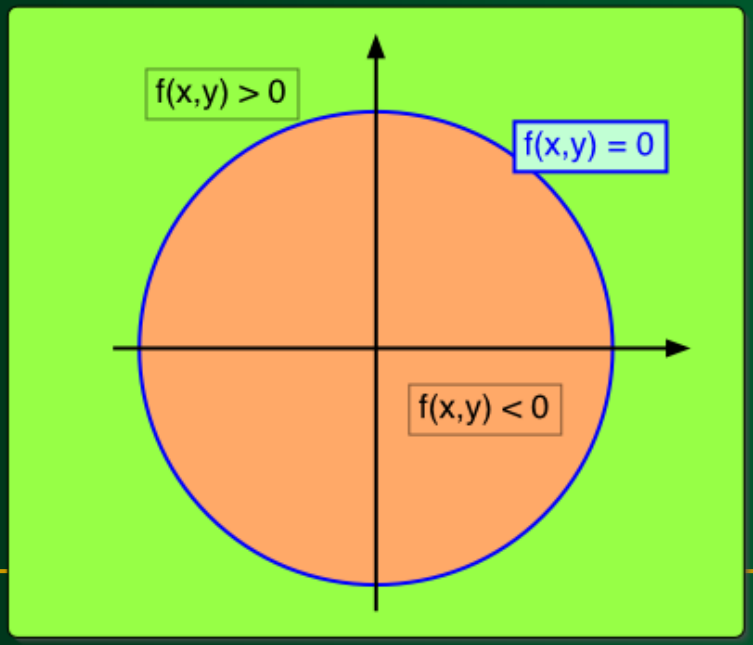
- Points are generated from 90º to 45º, moves will be made only in the +x and –y direction.
- positive x direction over this octant and use a **decision parameter**



Circle Generation Algorithms

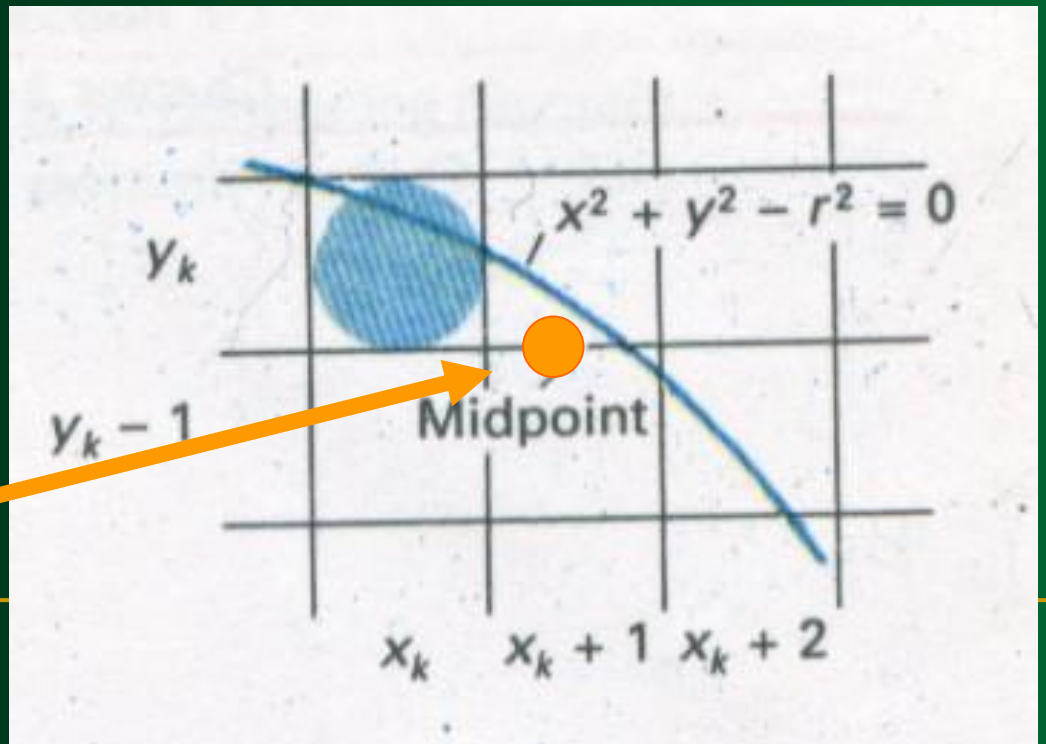
- We define a circle function:

$$f_{circle}(x,y) = x^2 + y^2 - r^2 \begin{cases} < 0 \\ = 0 \\ > o \end{cases}$$



Circle Generation Algorithms

- Midpoint $\left(x_i + 1, y_i - \frac{1}{2}\right)$
- Consider the coordinates of the point halfway between pixel T and pixel S



Midpoint

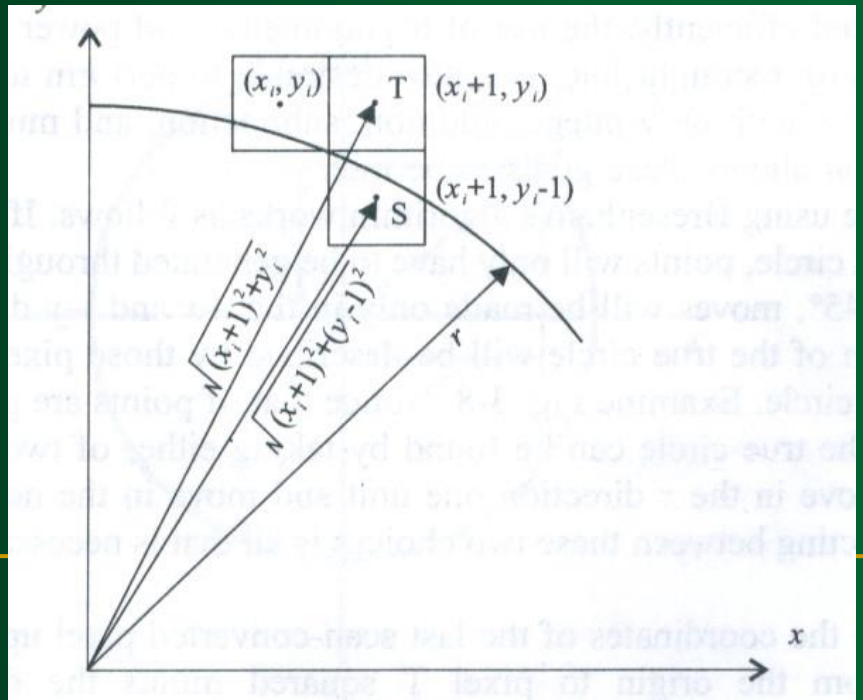
Circle Generation Algorithms

- We use it to define a decision parameter

$$p_i = f(x_i + 1, y - \frac{1}{2}) = (x_i + 1)^2 + (y_i - \frac{1}{2})^2 - r^2$$

Circle Generation Algorithms

- If $p_i < 0$ is negative, the midpoint is inside the pixel, and we choose pixel T.
- If $p_i \geq 0$ we choose pixel S.



Circle Generation Algorithms

- Parameter for the next step is:

$$p_{i+1} = (x_{i+1} + 1)^2 + (y_{i+1} - \frac{1}{2}) - r^2$$

- since

$$x_{i+1} = x_i + 1$$

$$p_{i+1} - p_i = [(x_i + 1) + 1]^2 - (x_i + 1)^2 + (y_{i+1} - \tfrac{1}{2})^2 - (y_i - \tfrac{1}{2})^2$$

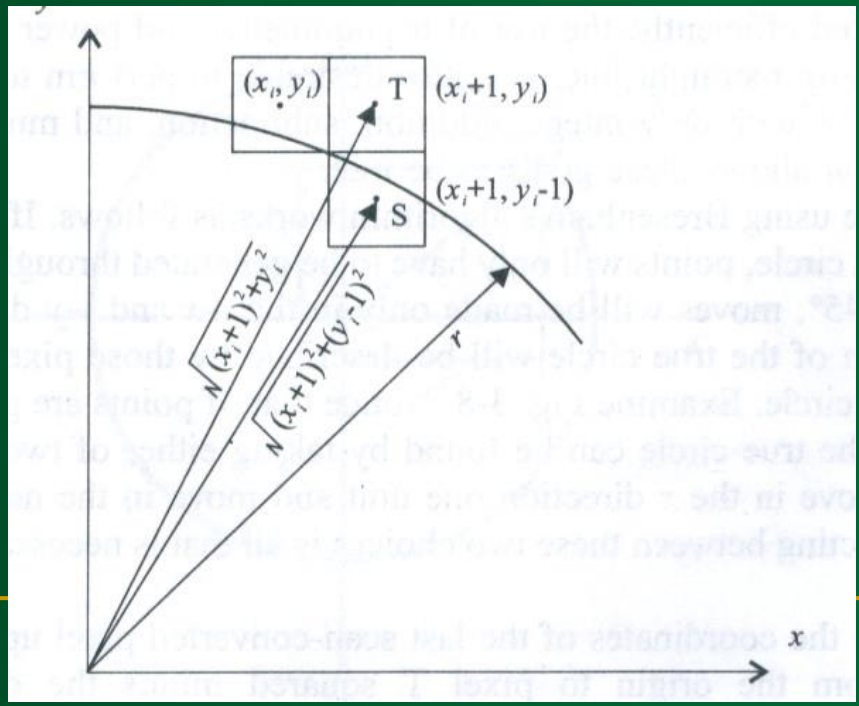
$$p_{i+1} = p_i + 2(x_i + 1) + 1 + (y_{i+1}^2 - y_i^2) - (y_{i+1} - y_i)$$

Circle Generation Algorithms

- If T is chosen ($p_i < 0$) we have:

$$y_{i+1} = y_i$$

- If pixel S is chosen ($p_i \geq 0$) we have $y_{i+1} = y_i - 1$



Circle Generation Algorithms

In terms of (x_{i+1}, y_{i+1})

$$p_{i+1} = \begin{cases} p_i + 2x_{i+1} + 1 & \text{if } p_i < 0 \\ p_i + 2(x_{i+1} - y_{i+1}) + 1 & \text{if } p_i \geq 0 \end{cases}$$

Circle Generation Algorithms

Initial value for the decision parameter using the original function of (0,r(

$$p_{i+1} = (x_{i+1} + 1)^2 + (y_{i+1} - \tfrac{1}{2})^2 - r^2$$

$$p_0 = (0 + 1)^2 + (r - \tfrac{1}{2})^2 - r^2 = \tfrac{5}{4} - r$$

When r is an integer we can simply set

$$p_0 = 1 - r$$

Circle Generation Algorithms

- **Example:** A circle radius r=10
- x=0 to x=y
- P_0=1-r = -9

$$p_{i+1} = \begin{cases} p_i + 2x_{i+1} + 1 & \text{if } p_i < 0 \\ p_i + 2(x_{i+1} - y_{i+1}) + 1 & \text{if } p_i \geq 0 \end{cases}$$

$$p_0 = 1 - r$$

Exercises?