Circle Drawing Algorithm

Bresenhams circle drawing algorithm

Mid point circle drawing algorithm

- Circle is eight way symmetrical figure.
- If one point is calculated with circle algorithm seven more points could be found by reflection.

plot (y, x)
plot (y, – x)
plot (x, – y)
plot (– x, – y)
plot (– y, – x)
plot (– y, x) and
plot (– x, y)

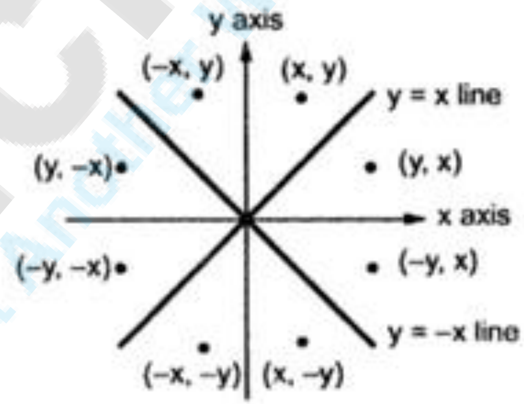
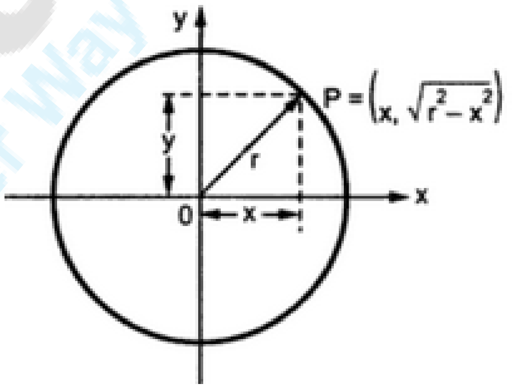


Fig. 1.29 Eight-way symmetry of the circle

- Circle can be define using two methods.
- Polynomial method:
- Equation of circle

$$x^2+y^2=r^2$$

where x : The x-co-ordinate
y : The y-co-ordinate
r : Radius of the circle



Using this method we find y for the value of x. this will generate 1/8th portion of the circle.

Disadvantage: for each point both a and r must be squared, x2 subtracted from r2 and square root of result

- **Trigonometric method:**
- It uses trigonometric function,

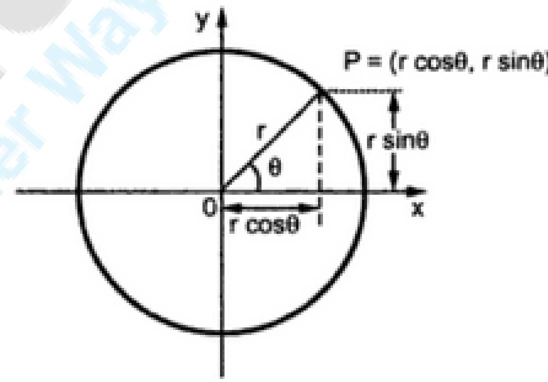
$$x = r \cos \theta \quad \text{and} \quad y = r \sin \theta$$

where θ : Current angle

r : Radius of the circle

x : The x coordinate

y : The y coordinate



- In this method θ is stepped from 0 to ∏/4 and each x and y is calculated.
- **Disadvantage**: It is more inefficient than polynomial method because the computation of cos and sin values is more time consuming

Bresenham's Circle Algorithm

- We have to select those pixel in raster that fall the least distance from the true circle.
- If points are generated from 90° to 45°,each new point closest to the true circle can be found by taking either of two actions:
  - Move in positive x direction by one unit.
  - Move in positive x direction by one unit and move in the negative y direction by one unit

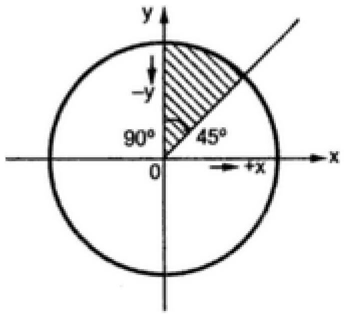
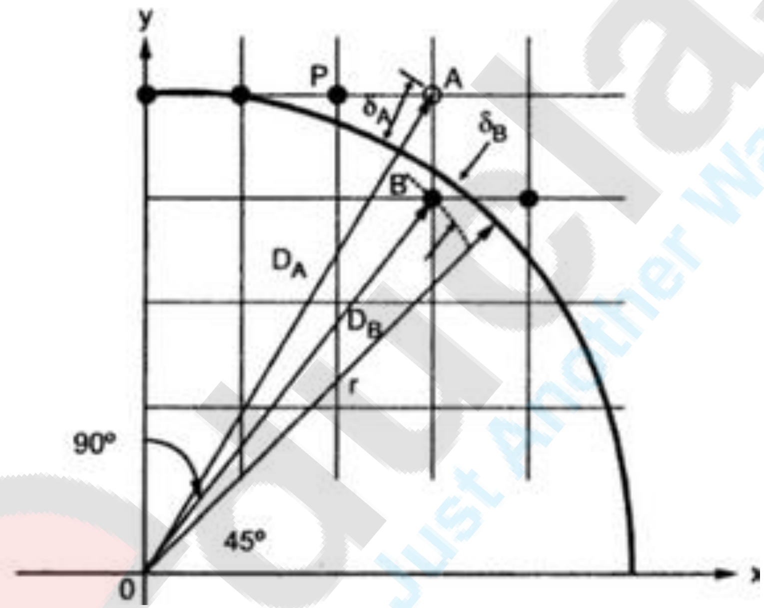


Fig. 1.27 1/8 part of circle



The distances of pixels A and B from the origin are given as

$$D_A = \sqrt{(x_{i+1})^2 + (y_i)^2} \quad \text{and}$$

$$D_B = \sqrt{(x_{i+1})^2 + (y_i - 1)^2}$$

- Distance of pixel A and B from true circle are given by ,

$$\delta_A = D_A - r \text{ and } \delta_B = D_B - r$$

- To avoid the square root , $\delta_A = D_A^2 - r^2$ and

$$\delta_B = D_B^2 - r^2$$

δ_A is always positive and δ_B always negative.

define decision variable d_i as

$$d_i = \delta_A + \delta_B$$

$$
\begin{aligned}
d_i &= \delta_A + \delta_B \\
&= (x_i + 1)^2 + (y_i)^2 - r^2 + (x_i + 1)^2 + (y_i - 1)^2 - r^2 \\
&= (0 + 1)^2 + (r)^2 - r^2 + (0 + 1)^2 + (r - 1)^2 - r^2 \\
&= 1 + r^2 - r^2 + 1 + r^2 - 2r + 1 - r^2 \\
&= 3 - 2r
\end{aligned}
$$

if $d_i < 0$, i.e., $\delta_A < \delta_B$ then only x is incremented; otherwise x is incremented in positive direction and y is incremented in negative direction. In other words we can write,

For $d_i < 0$, $x_{i+1} = x_i + 1$ and

For $d_i \geq 0$, $x_{i+1} = x_i + 1$ and $y_{i+1} = y_i - 1$

For $d_i < 0$, $d_{i+1} = d_i + 4x_i + 6$ and

For $d_i \leq 0$, $d_{i+1} = d_i + 4(x_i - y_i) + 10$

Algorithm to plot 1/8 of the circle

1. Read the radius (r) of the circle.
2. d = 3 – 2 r

   [Initialize the decision variable
3. x = 0, y = r

   [Initialize starting point]
4. do

```
{
    plot (x, y)
    if (d < 0) then
    {
        d = d + 4x + 6
    }
    else
    { d = d + 4 (x – y) + 10
      y = y – 1
    }
    x = x + 1
} while (x < y)
```

5. Stop

Questions

- Discuss the logic of Bresenhams circle drawing algorithm .Give the algorithm for a circle with center at the origin and radius R unit.
- Indicate which location would be chosen by Bresenham’s algorithm when scan-converting a circle of radius 10.