2D Circle Drawing

Circle Drawing

- Symmetry: Circle sections in adjacent octants within one quadrant are symmetric with respect to the 45^0 line dividing the two octants.
- Circle equation: $(x - x_c)^2 + (y - y_c)^2 = r^2$

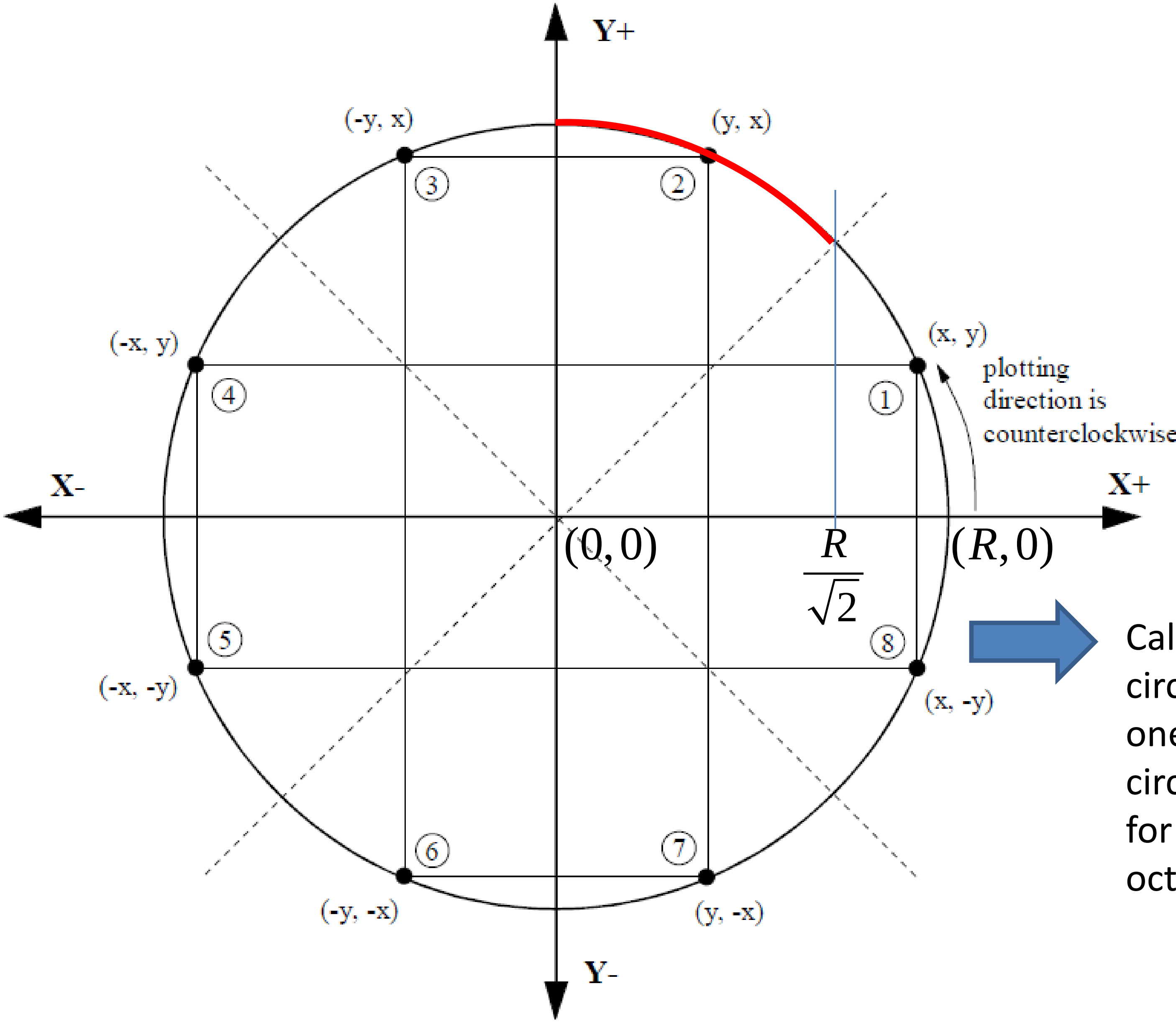
$$x^2 + y^2 = r^2, \text{ with } (x_c, y_c) = (0,0)$$

- Parametric equation

$$x = x_c + r\cos\theta$$

$$y = y_c + r\sin\theta$$

Circle Drawing

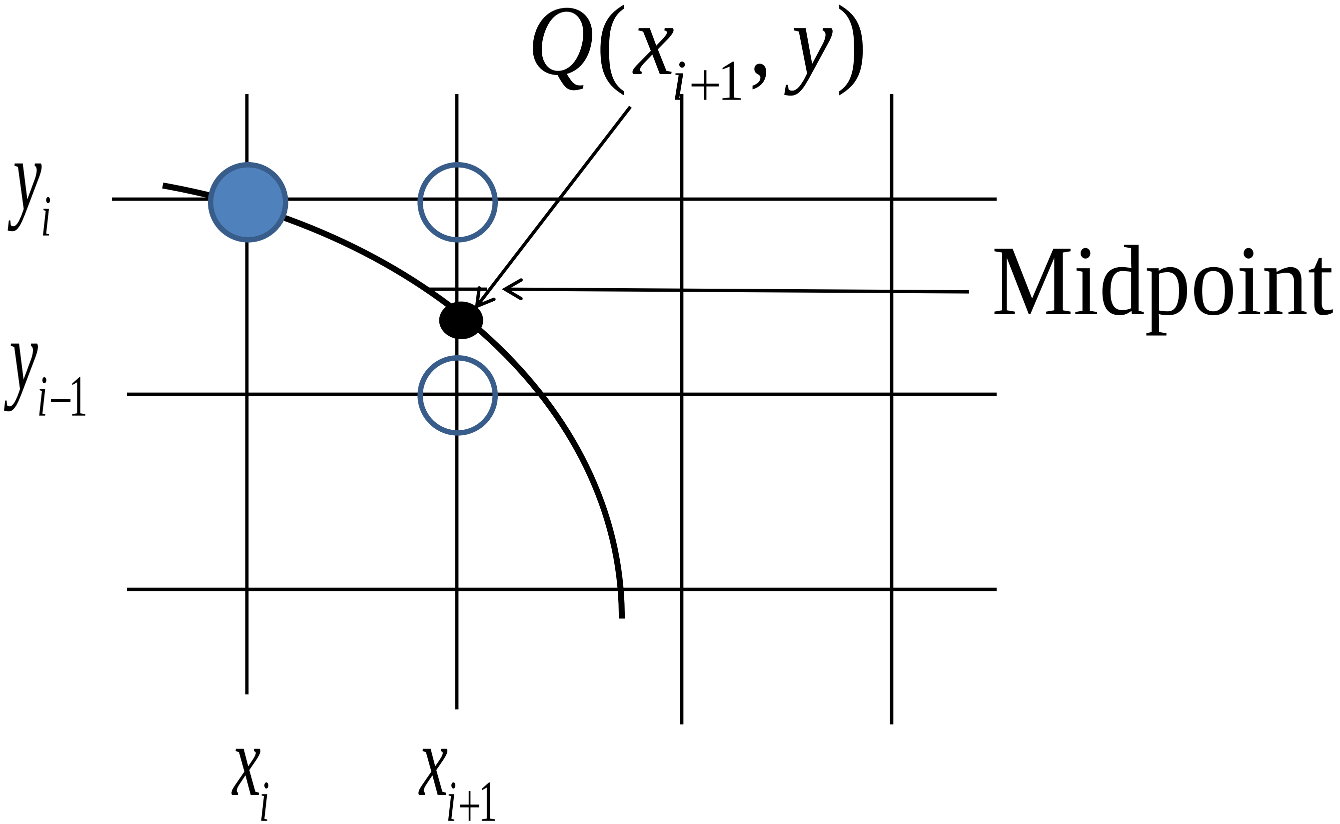


Calculation of a circle point (x,y) in one octant yields the circle points shown for the other seven octants.

8-way symmetry

```
void circleSym8(int R)
{
      r2 = R * R;
      putPixel(0, R); putPixel(0, -R);
      putPixel(R, 0); putPixel(-R, 0);
      x = 1; y = round(sqrt(r2 - x*x));
      while (x < y) {
            putPixel(x, y);   putPixel(x, -y);
            putPixel(-x, y);  putPixel(-x, -y);
            putPixel(y, x);   putPixel(y, -x);
            putPixel(-y, x);  putPixel(-y, -x);
            x++
            y = round(sqrt(r2 - x*x));
      }
}
```

Midpoint Algorithm



$$f_{circle}(x, y) = x^2 + y^2 - R^2$$

Midpoint Algorithm

$$f_{circle}(x,y)\begin{cases} <0, (x,y) \text{ is inside the circle boundary} \\ =0, (x,y) \text{ is on the circle boundary} \\ >0, (x,y) \text{ is outside the circle boundary} \end{cases}$$

Assuming (x_k,y_k) has been plotted, we need to determine whether the pixel at positions (x_k+1,y_k) or (x_k+1,y_k-1) is closer to the circle. We calculate p_k by

$$p_k = f_{circle}(x_k+1, y_k-\frac{1}{2}) = (x_k+1)^2 + (y_k-\frac{1}{2})^2 - R^2 \quad \textbf{(***)}$$

- $p_k < 0,$ the midpoint is inside the circle

  $\Rightarrow y_k$ is closer to the circle boundary

- $p_k \geq 0,$ the midpoint is on or outside the circle

  $\Rightarrow y_{k-1}$ is selected

Midpoint Algorithm

- Reduce calculation of $p_k(x_k,y_k)$ in **(***)** by

$$p_{k+1} = f_{circle}(x_{k+1}+1, y_{k+1}-\frac{1}{2}) = \left[\left(x_k+1\right)+1\right]^2 + \left(y_{k+1}-\frac{1}{2}\right)^2 - R^2$$

$$\Rightarrow p_{k+1} = p_k + 2(x_k+1) + (y_{k+1}^2 - y_k^2) - (y_{k+1} - y_k) + 1$$

and define an adaptive increment for each step

$$p_k < 0, p_{k+1} = p_k + 2(x_k+1) + 1 \text{ by } y_{k+1} = y_k$$

$$p_k \geq 0, p_{k+1} = p_k + 2(x_k+1) + 1 - \ 2y_{k+1} \text{ by y}_{k+1} = y_k - 1$$

where the initial value of p_i is defined by

$$p_0 = f_{circle}(x_0+1, y_0-\frac{1}{2}) = f_{circle}(1, R-\frac{1}{2}) = 1 + \left(R-\frac{1}{2}\right)^2 - R^2$$

$$\Rightarrow p_0 = \frac{5}{4} - R \approx 1 - R$$

Example

Given radius of circle R=10 and the starting point (0,R) of the first octant, we will draw the first octant (until x=y).

The initial value p_0 = 1-R = -9.

k	p_k	(x_{k+1},y_{k+1})	$2x_{k+1}$	$2y_{k+1}$
0	-9	(1,10)	2	20
1	-6	(2,10)	4	20
2	-1	(3,10)	6	20
3	6	(4,9)	8	18
4	-3	(5,9)	10	18
5	8	(6,8)	12	16
6	5	(7,7)	14	14

(Computer Graphic: C Version, 2nd Ed., D.Hearn, et.al., p. 101)

Questions

- How can we generalize the Midpoint concept to draw conics such as ellipse, parabola, and hyperbole?
- Can we apply the Midpoint concept to draw trigonometric functions such as sin(x) and cos(x) ?

References

[1] D. Hearn, M.P. Baker, Computer Graphic: C Version in 2nd Ed., Prentice Hall, 1996.

[2] D.N.D.Tien, V.Q. Hoang, L. Phong, CG-Course Slide, HCM-University of Science.