075BCT092

Date ____ Page ____

1

LAB 3: Midpoint Circle Algorithm

THEORY

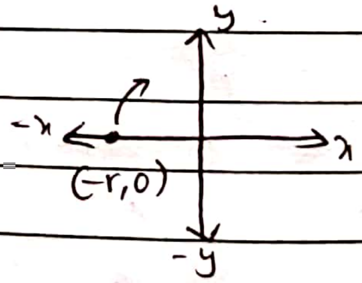
The equation of circle in Cartesian form is

$$(x-x_c)^2 + (y-y_c)^2 = r^2.$$

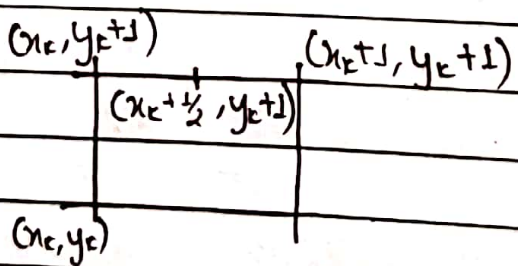
We sample at unit intervals and determine the closest pixel position to the specified circle at each steps. Interchange the rate to x and y wherever the absolute value of slope of the circle tangent greater than 1. To solve completely we use symmetry of circle i.e. calculate for one octant and use symmetry for others.

Starting at -ve x-axis and clockwise direction.

Here, we sample in y direction, as y is increasing and also $|x|$.



We have,



$(x_k, y_k) \rightarrow$ starting pixel

(x_{k+1}, y_{k+1}) ~~next~~ next pixel $= \left(\begin{matrix} x_k+1 \\ \text{or} \\ x_k \end{matrix}, y_k+1\right)$

Decision parameter $P_k = f_c(x_k + \frac{1}{2}, y_k+1)$

next successive

$$P_{k+1} = f_c(x_{k+1} + 1/2, y_{k+1}+1)$$

Subtracting P_k from P_{k+1}

$$P_{k+1} - P_k = f_c(x_{k+1} + 1/2,\ y_{k+1} + 1) - f_c(x_k + 1/2,\ y_k + 1)$$

$$= (x_{k+1} + 1/2)^2 + (y_{k+1} + 1)^2 - r^2 - (x_k + 1/2)^2 - (y_k + 1)^2 + r^2.$$

$$= x_{k+1}^2 + x_{k+1} + \cancel{\tfrac{1}{4}} + \cancel{y_{k+1}^2} + 2y_{k+1} + \cancel{1} - \left[x_k^2 + x_k + \cancel{\tfrac{1}{4}}\right]$$

$$\cancel{-\left[y_k^2 + 2y_k + 1\right]}$$

$$= x_{k+1}^2 + x_{k+1} + 2y_{k+1} - x_k^2 - x_k + 1.$$

$$= (x_{k+1}^2 - x_k)^2 + (x_{k+1} - x_k) + 2(y_{k+1}) + 1.$$

$$\therefore P_{k+1} = P_k + (x_{k+1}^2 - x_k^2) + (x_{k+1} - x_k) + 2(y_{k+1}) + 1$$

If P_k is '-ve'

$$x_{k+1} = x_k, \quad y_{k+1} = y_k + 1.$$

$$P_{k+1} = P_k + (x_k^2 - x_k^2) + (x_k - x_k) + 2(y_{k+1}) + 1.$$

$$P_{k+1} = P_k + 2y_{k+1} + 1.$$

If P_k is '+'ve.

$$\& \ x_{k+1} = x_k + 1, \quad y_{k+1} = y_k + 1$$

$$P_{k+1} = P_k + (x_k^2 + 2x_k + 1 - x_k^2) + (x_k + 1 - x_k) + 1 + 2(y_{k+1})$$

$$= P_k + 2y_{k+1} + 1 + 2x_k + 2.$$

$$= P_k + 2y_{k+1} + 2x_{k+1} + 1.$$

Now, For initial decision parameter,
we have, $(x_0, y_0) = (-r, 0)$

$$p_0 = f_c(-r + 1/2, 0 + 1)$$

$$= \left(\frac{1}{2} - r\right)^2 + 1^2 - r^2.$$

$$= \frac{1}{4} - r + r^2 - r^2 + 1.$$

$$= \frac{5}{4} - r.$$

$$\simeq 1 - r.$$

$$\therefore p_0 = 1 - r.$$

Algorithm.

1. Start.
2. Input (x_c, y_c) circle center, and radius r and set the co-ordinates for first point on the circumference of the circle centered at origin as.

   $(x_0, y_0) = (-r, 0)$
3. Calculate initial value of decision parameter

   $p_0 = 5/4 - r$
4. At each x_k position, starting from $k=0$

   If $p_k < 0$.

   $x_{k+1} = x_k$

   $y_{k+1} = y_k + 1$

   $p_{k+1} = p_k + 2y_{k+1} + 1$

   Else,

   $x_{k+1} = x_k + 1$.

   $y_{k+1} = y_k + 1$.

   $p_{k+1} = p_k + 2y_{k+1} + 2x_{k+1} + 1$.

5 Determine the symmetry in other seven octants

6. Move each calculated pixel position (x, y) onto the circular path centered on (x_c, y_c).

Plot the co-ordinates values.

$$x = x + x_c$$

$$y = y + y_c.$$

7. Repeat steps 4 to 6 until $y \geq -x$. $\rightarrow$ $|y| \geq |x|$

Example: $(x_c, y_c) = (10, 20)$

Sol^n, $r = 10$

Initial point $(0, r) = (0, 10)$

$p_0 = 1 - r = -9$.

k	x_{k+1}	y_{k+1}	p_{k+1}
0	1	10	-6
1	2	10	-1
2	3	10	6
3	4	9	-3
4	5	9	8
5	6	8	5
6	7	7	–

if $p_k < 0$

$y_{k+1} = y_k$, $x_{k+1} = x_k + 1$

$p_{k+1} = p_k + 2x_{k+1} + 1$

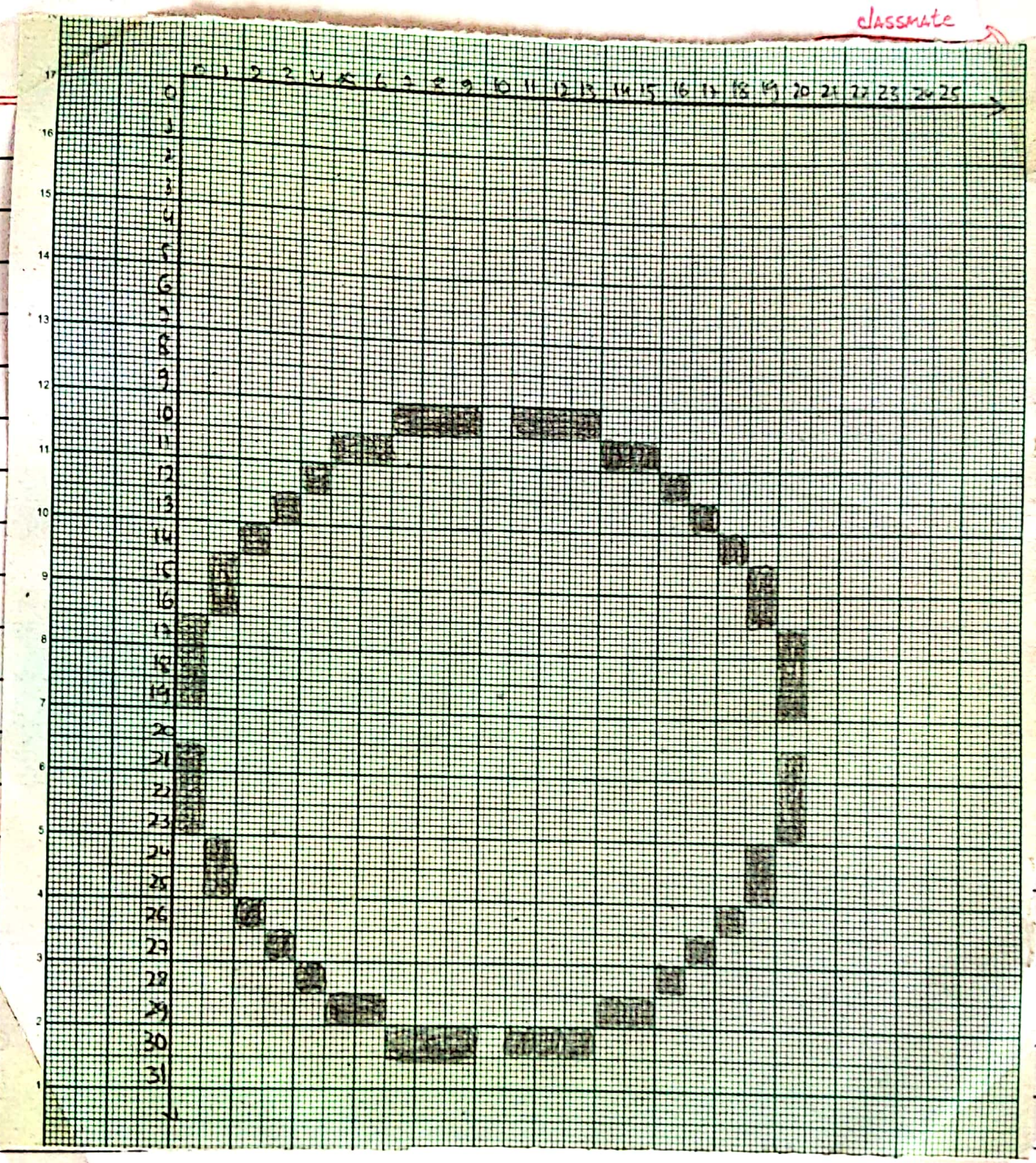
else.

$y_{k+1} = y_k - 1$, $x_{k+1} = x_k + 1$

$p_{k+1} = p_k + 2x_{k+1} + 1 - 2y_{k+1}$

until $x \leq y$.

(x_{k+1}, y_{k+1})	(x, y)	(y, x)	$(y, -x)$	$(x, -y)$	$(-x, -y)$	$(-y, -x)$	$(-y, x)$	$(-x, y)$
(1,10)	(11,30)	(20,21)	(20,19)	(11, 10	(9, 10)	(0, 19)	(0, 21)	(9, 30)
(2,10)	(12,30)	(20,22)	(20, 18)	(12, 10	(8,10)	(0, 18)	(0, 22)	(8, 30)
(3,10)	(13,30)	(20,23)	(20, 17	(13, 10	(7, 10)	(0, 17)	(0, 23)	(7, 30)
(4,9)	(14,29)	(19,24)	(19, 16	(14, 11	(6, 11)	(1, 16)	(1, 24)	(6, 29)
(5,9)	(15,29)	(19,25)	(19, 15	(15, 11	(5,11)	(1, 15)	(1, 25)	(5, 29)
(6,8)	(16,28)	(18,26)	(18, 14	(16, 12	(4,12)	(2, 14)	(2, 26)	(4, 28)
(7,7)	(17,27)	(17,27)	(17, 13)	(17, 13)	(3,13)	(3, 13)	(3, 27)	(3, 27)



Discussion and conclusion.

In this lab we analyse the midpoint circle algorithm. Midpoint circle algorithm is a very efficient way to obtain circles with the least amount of computation. We derived decision parameter, obtained the algorithm and solved one example.