Circle Drawing Algorithms

Introduction:

Computer graphics is an art of drawing pictures on computer screens with the help of programming. It involves computations, creation, and manipulation of data. In other words, we can say that computer graphics is a rendering tool for the generation and manipulation of images.

Drawing a circle on the screen is a little complex than drawing a line. Circle have combination of points, which must make a smooth curve. this smooth curve is called as circumference. Thus, circle is the locus of a point which moves at a fixed distance from a fixed point. this fixed point is called center and fixed distance is called radii of the circle.

In computer graphics, we use the symmetricity property of the circle to plot the circle boundary. There are two popular algorithms for generating a circle − Bresenham's Algorithm and Midpoint Circle Algorithm. These algorithms are based on the idea of determining the subsequent points required to draw the circle.

Property Of Circle:

The circle is a symmetrical figure formed by plotting points from a fixed distance taken from a fixed point.

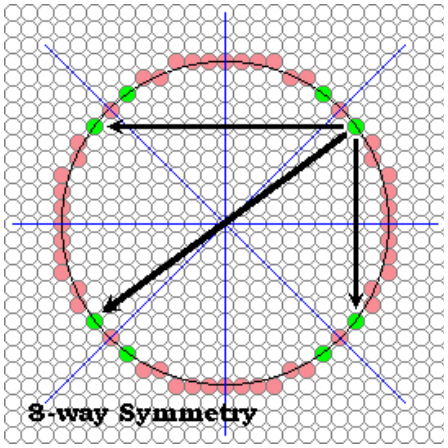
A circle generation algorithm can take advantage of the circle 's symmetry to plot eight points for each value that the algorithm calculates.

Eight-way symmetry is used by reflecting each calculated points around 45^{o} axis.

Circle's Equation: $(x - h)^2 + (y - k)^2 = r^2$,

Where x, y = x & y coordinates with the center being at the point (h, k) and the radius is "r".

Urvi Jain
Varsha Chawla
Vishal Bothra



Introduction to Circle Drawing Algorithms:

Generally, there are two circle generation algorithms that take advantage of the symmetry of the circle but it's still not easy to display a smooth curve on the computer screen because it's just made up of small square pixels. The two algorithms namely:

1. Midpoint circle drawing algorithm
2. Bresenham's circle drawing algorithm

uses the key feature of the circle that it is highly symmetric. So for the whole of 360^{o} of a circle, we will divide it into 8 parts each octant of 45^{o}.

Midpoint Algorithm:

Mid-Point circle algorithm is related to work by **Pitteway** and **Van Aken**.

We need to plot the perimeter to calculate all the perimeter points of the circle in the first octant and then print them along with their mirror points in the other octant.The key feature of the Mid-Point Circle drawing algorithm is that decision parameter depends on the previous parameter and corresponding pixels. The major drawback is that the resulting circle has large gaps and the calculations are not very efficient as it has a square (multiply) and square root operation.

Urvi Jain
Varsha Chawla
Vishal Bothra

Why we use it:

It is based on the function **F** discussed in the next section for testing the spatial relationship between an arbitrary point (x, y) and a circle of radius centered at the origin.

Boundary Condition:

Whether the mid-point lies inside or outside the circle can be decided by using the formula:-

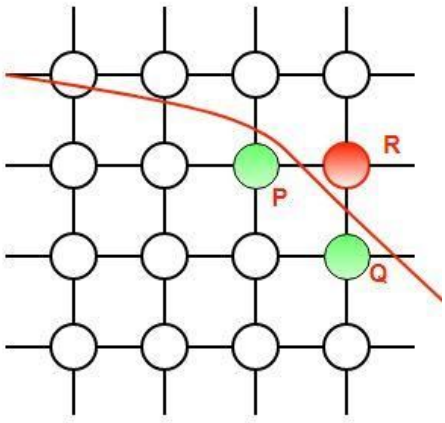
Given a circle centered at (0,0) and radius r and a point p(x,y)

$F(p) = x^2 + y^2 - r^2$

if F(p)<0, the point is inside the circle

F(p)=0, the point is on the perimeter

F(p)>0, the point is outside the circle



Bresenham's Circle Drawing Algorithm:

Bresenham's circle algorithm is named after Jack Elton Bresenham who developed it in 1962 at IBM. The Bresenham's circle drawing algorithm is derived from the mid-point circle drawing algorithm. It determines the points of a raster that have to be selected to form an approximate straight line between two points. The key feature of Bresenham's circle drawing algorithm is that Bresenham decision parameter only depends on previous decision parameter.

Urvi Jain
Varsha Chawla
Vishal Bothra

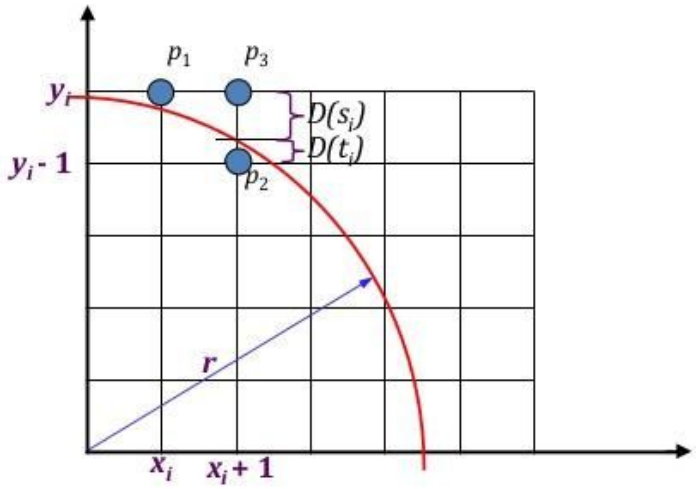
Why it is used?

It uses integer addition, bit shifting, and subtraction which are cheap operations in computer architecture. It is one of the easiest algorithms and is important even at present because of its speed and simplicity. It overcomes the shortcomings of the Midpoint circle drawing algorithm by using cheaper mathematical operations and using less memory resulting in a much more smoother circle as compared to the Midpoint circle drawing algorithm.

It is very less time-consuming.

Working:

In Bresenham's algorithm at any point (x, y) we have two option either to choose the next pixel in the east i.e. (x+1, y) or in the south-east i.e. (x+1, y-1).



Midpoint circle drawing algorithm:

In computer graphics, the midpoint circle algorithm is an algorithm used to determine the points needed for rasterizing a circle.

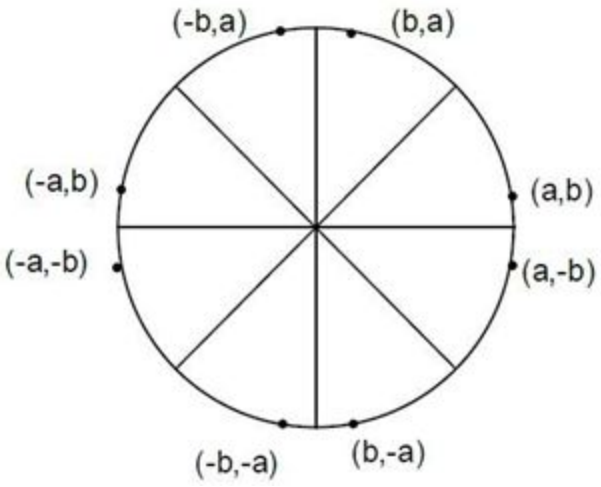
Urvi Jain
Varsha Chawla
Vishal Bothra

The algorithm can be generalized to conic sections. The algorithm is related to work by Pitteway and Van Aken.

To form a brief introduction of the algorithm we are about to study, we perform unit increments along one axis while calculating the respective increment for the other axis, in accordance with the closest pixel position to the specified circle path at each step. Initially, we assume the center of the Cartesian plane, the origin, as the center of our circle. Hence for a given radius r and screen center position (x, y,), we can first set up our algorithm to calculate pixel positions around a circular path centered at the coordinate origin (0,0).

We need to plot the perimeter points of a circle whose center coordinates and radius are given using the Midpoint Circle Drawing Algorithm.

We use the above algorithm to calculate all the perimeter points of the circle in the first octant and then print them along with their mirror points in the other octants. This will work only because a circle is symmetric about its center.



Derivation :

To apply the midpoint method, we define a circle function:

Urvi Jain
Varsha Chawla
Vishal Bothra

In Midpoint Circle Algorithm, the decision parameter at the kth step is the circle function evaluated using the coordinates of the midpoint of the two-pixel centers which are the next possible pixel position to be plotted. Let us assume that we are giving unit increments to x in the plotting process and determining the y position using this algorithm. Assuming we have just plotted the kth pixel at (Xk, Yk), we next need to determine whether the pixel at the position (Xk+1, Yk) or the one at (Xk+1 , Yk-1) is closer to the circle.

Our decision parameter pk at the kth step is the circle function evaluated at the midpoint of these two pixels. The coordinates of the midpoint of these two pixels are (Xk+1, Yk-1/2).

Thus pk x Successive decision parameters are obtained using incremental calculations, thus avoiding a lot of computation at each step. We obtain a recursive expression for the next decision parameter i.e. at the k+1th step, in the following manner. Using Equ. (4), at the k+1th step, we have:

Now if Pk <=0, then the midpoint of the two possible pixels lies within the circle, thus north pixel is nearer to the theoretical circle. Hence, Yk+1= Yk . Substituting this value of in Equ. (6), we have

If pk > 0 then the midpoint of the two possible pixels lies outside the circle, thus south pixel is nearer to the theoretical circle. Hence, Yk+1= Yk-1. Substituting this value of in Equ. (6), we have

For the boundary condition, we have x=0, y=r. Substituting these values in (4), we have

For integer values of pixel coordinates, we can approximate P0=1-r,

Thus we have:

If $p_k \leq 0$: $y_{k+1} = y_k$ and $p_{k+1} = p_k + (2x_k + 3)$

If $p_k > 0$: $y_{k+1} = y_k - 1$ and $p_{k+1} = p_k + 2(x_k - y_k) + 5$

Also, $p_0 = 1 - r$

Algorithm:

Urvi Jain
Varsha Chawla
Vishal Bothra

1. Input radius r and circle center (xc,yc) and obtain the first point on the circumference of the circle centered on the origin as

   (xo,yo)=(0,r)

2. Calculate the initial value of decision parameter as

   P0 = 1-r

3. At each xk position starting at a=0 perform the following test.

   3.1 pk<0 the next point along the circle centered on origin is (xk+1,yk) and =pk+2xk+3

   3.2 If pk>=0 the next point along the circle is (xk+1,yk-1)

   pk+1 =pk+2xk-2yk+5

   Or =pk+2xk+1-2yk+1+1

4. Determine symmetry points in the other 7 octants.

5. Transfer each calculated pixel position (x,y) onto a circular path centered on (xc,yc) and plot the coordinate values as x=x+xc and y=y+yc.

6. Repeat step 3 to 5 until x<=y.

Numerical:

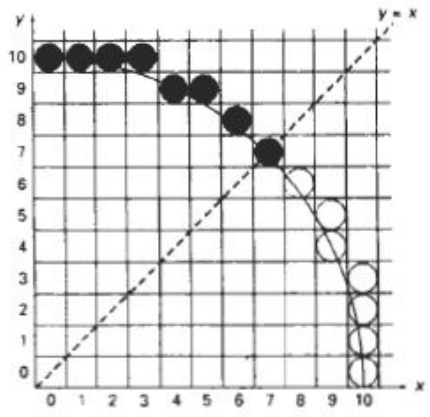
r=10

po=1-r=-9

(xo,yo)=(0,10)

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

Urvi Jain
Varsha Chawla
Vishal Bothra



Advantages:

The midpoint method is used for deriving efficient scan-conversion algorithms to draw geometric curves on raster displays.

The method is general and is used to transform the nonparametric equation f(x,y) = 0, which describes the curve, into an algorithm that draws the curve.

Disadvantages:

time consumption is high

the distance between the pixels is not equal so we won't get a smooth circle.

Bresenham's circle drawing algorithm:

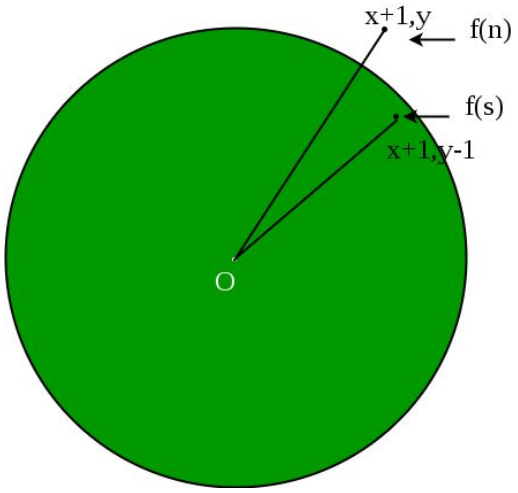
It's a difficult task to make a smooth arc point by point on a computer screen, i.e viewport. This is because our computer screen is organized in forms of the matrix of pixels. Hence, to make a smooth continuous arc, we plot a point on a pixel and then extend the arc by choosing the point as near as possible on the circumference.

The circle is known as symmetrical figure, thus, we can use its property by plotting points of only one octant and reflecting it in other seven octants, making our methods efficient.

Urvi Jain
Varsha Chawla
Vishal Bothra

Circle drawing algorithm mainly deals with the decision, which method to choose to plot the next pixel position in order to generate a smooth arc. Bresenham's circle drawing algorithm:

For any point (x.y), we have two options either to choose the next pixel in east i.e. (x+1, y) or in the south east i.e. (x+1, y-1).



Derivation:

Let us consider the circle at origin (0,0) i.e. $x^2 + y^2 = r^2$ and (xk , yk) be initial points.

$F(x,y) = x^2 + y^2 - r^2$

$F(N) = (x_{k+1})^2 + y^2_k - r^2$

$F(S) = (x_{k+1})^2 + (y_{k-1})^2 - r^2$

Decision parameter (Pk) = F(N) + F(S)

$P_k = 2(x_{k+1})^2 + y^2_k + (y_{k-1})^2 - r^2$

Urvi Jain
Varsha Chawla
Vishal Bothra

Initial decision parameter (Po) = $2(1+0)^2 + r^2 + (r-1)^2 - 2r^2$

Po = 3-2r

$P_{k+1} = 2(x_k + 2)^2 + y^2_{k+1} + (y_{k+1} -1)^2 - 2r^2$

$P_{k+1} - P_k = 2[(x_k+2)^2 - (x_{k+1})^2] + (y^2_{k+1} - y^2_k) + (y_{k+1} -1)^2 - (y_{k-1})^2$

$P_{k+1} = p_k + 2(2x_k+3) + [(y_{k+1} + y_k)(y_{k+1} - y_k)] + [(y_{k+1} + y_k - 2)(y_{k+1} - y_k)]$----------(A)

Case 1 :

If P is negative

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

$y_{k+1} - y_k = 0$

Put in (A)

$P_{k+1} = P_k + 2(2x_k + 3)$

$P_{k+1} = Pk + 4x_k + 6$

Case 2 :

If P is positive

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

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

Put in (A)

$P_{k+1} = P_k + 4x_k+6 + (y_{k+1}+y_k)(-1) + (y_{k+1}+y_k -2)(-1)$

Urvi Jain
Varsha Chawla
Vishal Bothra

$$P_{k+1} = P_k + 4x_k - 4y_k + 10$$

The decision of choosing a point is based on decision parameter 'p'.

If d > 0, then (x+1, y-1) is to be chosen as the next pixel as it will be closer to the arc

Else (x+1, y) is to be chosen as next pixel.

And as derived, initial value of decision parameter is :

d= 3- 2*r

To draw the circle for a given radius 'r' and center (xc, yc) We will start from (0, r) and move in the first quadrant till x=y (i.e. 45 degrees).

Algorithm:

1. Input radius r, center (xc,yc) and obtain the first point on the circumference at P(0,r).
2. Calculate the initial value of decision parameter p=3 – 2*r.
3. At each xk, starting at k=0, perform following tests:
   3.1 if pk<0, then the next point of the circle is $(x_k + 1, y_k)$
   and $p_{k+1} = p_k + 4x_k + 6$.
   3.2 if pk>=0, then the next point of the circle is $(x_k + 1, y_k - 1)$
   and $p_{k+1} = p_k + 4x_k - 4y_k + 10$.
4. Determine the symmetry points in other 7 octants by circle symmetry property.
5. Transfer each point (x,y) according to the original center of the circle by substituting x=x+xc and y=y+yc.
6. Repeat steps (3) to (5) for x<=y times.

Numerical:

Plot a circle with radius 6 and center (3,3):

Urvi Jain
Varsha Chawla
Vishal Bothra

x0 = 0

y0 = r

p0 = 3 – 2r

if pi < 0 ,

  then yi+1 = yi

  and pi+1 = pi + 4xi + 6

else if pi ≥ 0,

  then yi+1 = yi – 1

  and pi+1 = pi + 4(xi – yi) + 10

Stop when xi >= yi and determine symmetry points in the other octants

Given radius=6

So, first point is (0, r) = (0, 6)

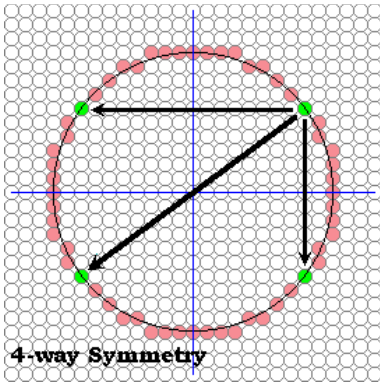
K	PI	XK+1	YK+1	PLOTTED POINTS
0	-9	1	6	X=xc+x y=yc+y X=3+1=4,y=3+6=9
1	1	2	5	X=xc+x y=yc+y X=3+2=5,y=3+5=8
2	-1	3	5	X=xc+x y=yc+y X=3+3=6,y=3+5=8
3	17	4	4	X=xc+x y=yc+y X=3+4=7,y=3+4=7

Urvi Jain
Varsha Chawla
Vishal Bothra

Now, reflecting each point as:

XC+X, YC-Y
XC-X, YC+Y
XC-X, YC-Y
XC+Y, YC+X
XC+Y, YC-X
XC-Y, YC+X
,XC-Y, YC-X

Figure:



Advantages and disadvantages:

It uses integer value calculations, thus eliminating the time needed for floating point calculations. Apart from this, using the symmetry property of a circle makes this circle drawing algorithm much efficient. It works by analyzing and manipulating the linear equations for only integer calculations. It has an advantage of speed and precision because working with floating point values requires more time and memory and such values would need to be rounded to integers anyway.

Urvi Jain
Varsha Chawla
Vishal Bothra

Apart from this, There is no loss of generality here since once the points are determined they may be translated relative to any center that is not the origin (0,0) and we can easily determine the points when the origin is shifted.

Difference Between two Algorithms:

Bresenham's line algorithm determines the points of a raster that have to be selected to form an approximate straight line between two points. It is used in a bitmap image to draw line primitives. It uses integer addition, bit shifting and subtraction which are cheap operations in computer architectures. It is one of the earliest algorithms and is important even in the present because of its speed and simplicity. Its extension can be used for drawing circles.

Midpoint circle algorithm is used in computer graphics to determine the points required for rasterizing a circle. The Bresenham's circle algorithm is derived from it. The midpoint circle algorithm may be generalized to conic sections.

Applications:

The circle which is symmetric yet complex and difficult to draw shape can be plotted to least error and smoothly on a computer screen without complex calculations which are quite time-consuming.
One needs to have an efficient algorithm for drawing circle, especially for animated display.
It is used in bitmap image to draw line primitives.

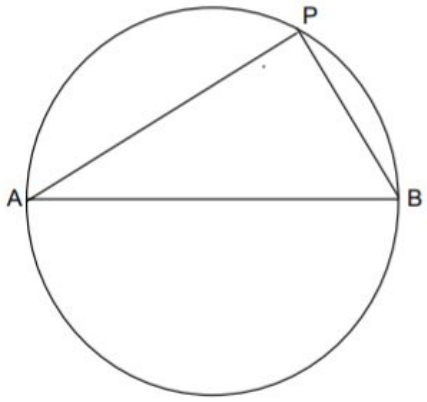
It is also possible to use the same concept to rasterize a parabola, ellipse or any other two-dimensional curve.

Future Aspects and Other Methods :

Urvi Jain
Varsha Chawla
Vishal Bothra

Many alternative approaches are suggested by many authors to draw circle much efficiently and much smoothly. One such algorithm was suggested which made use of the fact that the semicircular angle is 90^o. The algorithm involves only integer arithmetic only. The computational load, as well as the quality of circle drawn by this algorithm, is comparable to that of the mid-point algorithm.

Let point P has coordinates (x, y). If we take m as the slope of the segment AP then the slope of the segment BP which is perpendicular to the segment AP is –1/m.



The equation of the segment AP is $y = mx + mr$ and that of the segment BP is $x + my = r$.
Solving these equations we get the parametric equations of a circle with center at the origin and radius r as
$x = r\{(1 – m^2)/(1 + m^2)\}$ and $y = 2mr/(1 + m^2)$.
We can use these equations to draw a circle using an appropriate step size of m.

Urvi Jain
Varsha Chawla
Vishal Bothra