

# Midpoint Distance Circle Generation Algorithm based on Midpoint Circle Algorithm and Bresenham Circle Algorithm

Minghua Cao<sup>1,a</sup>, Siyuan Liu<sup>2,b</sup> and Fanghua Cao<sup>3\*,c</sup>

<sup>1</sup>Department of Computer Science and Technology, University of Electronic Science and Technology of China, Faculty of Computer Science and Engineering, No. 2006 Xiyuan Avenue, Chengdu, China.

<sup>2</sup>University of Electronic Science and Technology of China, Faculty of Public Administration, Chengdu, China.

Email:

<sup>3\*</sup>Hubei Yikatong Technology Co., Ltd. Shanghai City, China.

Email: <sup>a</sup>minghuacao@std.uestc.edu.cn; <sup>b</sup>siyuanliu@std.uestc.edu.cn;

<sup>c</sup>fanghua\_cao@163.com

**Abstract.** In the process of generating a circle, the existing midpoint circle algorithm and Bresenham circle algorithm have the problems of slow drawing speed, high resource consumption, complicated calculation steps and large pixel spacing in some areas, and a combination of midpoint painting is proposed. A new algorithm for the circular algorithm and the Bresenham circular algorithm. The algorithm first calculates the positional relationship between the midpoint and the circumference of two adjacent pixel points, and then calculates the distance difference between two adjacent pixel points and the circle by two approximate distances, and then obtains the decision parameters of the algorithm. A complete circle is obtained by the symmetry of the octant. Experiments show that the proposed algorithm is an effective midpoint distance circle generation algorithm based on the midpoint circle algorithm and the Bresenham algorithm. Compared with similar algorithms, it has higher stability and precision, and has less resource consumption and more flexible calculation methods.

## 1. Introduction

Graphics are usually composed of basic elements such as points, lines, faces, and bodies. Lines are divided into lines and curves. The algorithm for generating curves can be divided into line generation algorithm [1] and point generation algorithm [2]. Circle is a commonly used perfect figure. The circle generation algorithm is a basic algorithm in computer graphics [3]. Common circle generation algorithms mainly include midpoint circle algorithm [4], Bresenham circle algorithm [5] and positive and negative The circle algorithm [6] is classified into three categories, but all three algorithms have their own problems [6,7]. In order to solve the above problems, this paper proposes a new algorithm combining the advantages of the midpoint circle algorithm and the Bresenham circle algorithm.

## 2. Basic Principles

Introduce the basic principles that all algorithms need to use and the theory to be used in algorithm optimization.



### 2.1. Definition of circle

Define the center of the circle  $(x_c, y_c)$ , Radius is  $r$ , the coordinates of any point on the circle are  $(x, y)$ . Since the circle at any point can be obtained by the translation transformation of the coordinates [3], in order to simplify the complexity of the algorithm, Let the coordinates of the center of the circle be  $(0, 0)$ , then there is a point coordinate equation for the circle.

$$x^2 + y^2 = r^2 \quad (1)$$

### 2.2. Octagon circle

For any kind of circle generation algorithm, the symmetry of the circle can be used to reduce the amount of calculation. The circle with the center of the origin has four axes of symmetry, which are  $x = 0, y = 0, y = x, y = -x$ , if any point  $P(x, y)$  on the circumference is known, then the other seven symmetry points on the circumference can be obtained by four symmetry axes, which are  $(y, x), (-x, y), (x, -y), (y, -x), (-y, x), (-x, -y), (-y, -x)$ . This property is called the octant symmetry of the circle [3, 8, 9]. Therefore, with the octave symmetry of the circle, in the case of knowing the four symmetry axes of the circle, the entire circle can be drawn by drawing an arbitrary arc of one eighth.

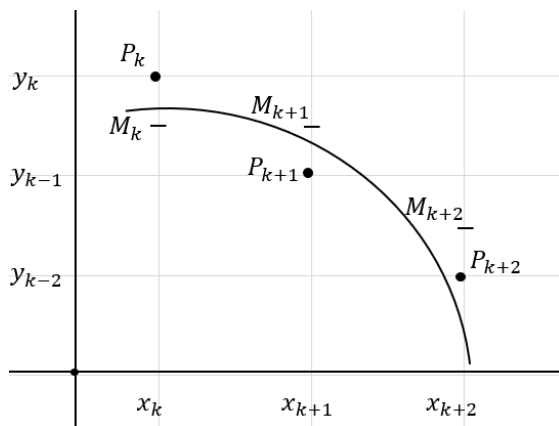


Figure 1. Midpoint circle algorithm.

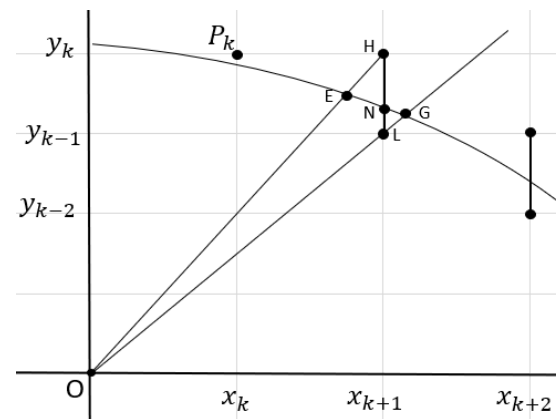


Figure 2. Bresenham circle algorithm.

### 3. Midpoint circle algorithm

In the computer graphics circle generation algorithm, because of the need to display or output graphics on the dot matrix output device, a raster scan conversion algorithm [3,6] is also needed to perform pixel point conversion. Directly using the point coordinate equation to calculate the position of the point on the circle requires multiplication and square root operations, which will increase the computational cost and algorithm complexity. Therefore, the method of drawing the midpoint circle is introduced.

Set the circular function to

$$f(x, y) = x^2 + y^2 - r^2 \quad (2)$$

Using the octave symmetry of the circle, only the arc between point A and point B in the Figure 1 is studied, and the coordinates of any point between the drawn arcs AB are  $P_k(x_k, y_k)$ , so on and so forth,  $P_{k+1}(x_{k+1}, y_{k+1})$ ,  $M_{k+1}(x_{k+1}, y_k - \frac{1}{2})$  is the midpoint of  $(x_{k+1}, y_k)$  and  $(x_{k+1}, y_{k-1})$ , among them,  $x_{k+1} = x_k + 1, y_{k-1} = y_k - 1, y_{k+1} \neq y_k + 1$ .

Set the decision parameters to

$$dm_k = f(x_{k+1}, y_k - \frac{1}{2}) = (x_k + 1)^2 + (y_k - \frac{1}{2})^2 - r^2 \quad (3)$$

When  $dm_k < 0$ , the midpoint  $M_{k+1}(x_{k+1}, y_k - \frac{1}{2})$  is in the circle, the point on the arc specified at this time is closer to point  $(x_{k+1}, y_k)$ ; when  $dm_k \geq 0$ , the midpoint  $M_{k+1}(x_{k+1}, y_k - \frac{1}{2})$  is on the circle or outside the circle, the point on the arc specified at this time is closer to point  $(x_{k+1}, y_{k-1})$ .

Therefore,

$$P_{k+1} = \begin{cases} (x_{k+1}, y_k), & dm_k < 0 \\ (x_{k+1}, y_{k-1}), & dm_k \geq 0 \end{cases} \quad (4)$$

That is, the value of  $y_{k+1}$  is determined by the positive or negative of the decision parameter  $dm_k$ .

Set the increment of the decision parameter to

$$\begin{aligned} \Delta dm_k &= dm_{k+1} - dm_k = f(x_{k+1} + 1, y_{k+1} - \frac{1}{2}) - f(x_{k+1}, y_k - \frac{1}{2}) \\ &= (x_k + 2)^2 + (y_{k+1} - \frac{1}{2})^2 - (x_k + 1)^2 - (y_k - \frac{1}{2})^2 \\ &= (2x_k + 3) + (y_{k+1}^2 - y_k^2) + (y_k - y_{k+1}) \end{aligned} \quad (5)$$

When  $dm_k < 0$ ,  $y_{k+1} = y_k$ , then  $\Delta dm_k = 2x_k + 3 = 2(x_k + 1) + 1 = 2x_{k+1} + 1$ ; when  $dm_k \geq 0$ ,  $y_{k+1} = y_{k-1} = y_k - 1$ , then  $\Delta dm_k = (2x_k + 3) - 2(y_k - 1) = 2x_{k+1} - 2y_{k-1} + 1$ . which is

$$dm_{k+1} = dm_k + \Delta dm_k = \begin{cases} dm_k + 2x_{k+1} + 1, & dm_k < 0 \\ dm_k + 2x_{k+1} - 2y_{k-1} + 1, & dm_k \geq 0 \end{cases} \quad (6)$$

Substituting the circular function at the starting position coordinate  $(x_k, y_k) = (0, r)$  into the above circular function to obtain the initial value of the decision parameter.

$$dm_0 = \frac{5}{4} - r \quad (7)$$

The coordinates of the next point can be obtained from equation (4), (6) and (7).

#### 4. Bresenham drawing circle algorithm

The principle of Bresenham's circle algorithm [5,10] is similar to that of the midpoint circle algorithm. The difference is that the Bresenham circle algorithm simplifies the calculation and makes full use of the principle of similarity to simplify the problem.

The Bresenham circle algorithm uses the line segment EH shown in Figure 2 instead of the line segment HN. The line segment LG is used instead of the line segment NL. The magnitude relationship between HN and NL is reflected by the positive and negative of EH-LG. For the center of the circle, the radius is r. The drawing process of one arc, the derivation process of the decision parameters of the Bresenham circle algorithm is as follows:

$$\begin{aligned}
 db_k &= db_{HN} - db_{NL} = db_{EH} - db_{LG} = (db_{OH} - db_{OE}) - (db_{OG} - db_{OL}) \\
 &= (\sqrt{x_{k+1}^2 + y_k^2} - r) - (r - \sqrt{x_{k+1}^2 + y_{k-1}^2})
 \end{aligned} \quad (8)$$

Since the above operation is still designed to be squared, the above equation is approximated as

$$\begin{aligned}
 db_k &= [(x_k + 1)^2 + y_k^2 - r^2] - [r^2 - (x_k + 1)^2 - (y_k - 1)^2] \\
 &= 2(x_k + 1)^2 + 2y_k^2 - 2y_{k-1} + 1 - 2r^2
 \end{aligned} \quad (9)$$

The increment of the decision parameter is

$$\begin{aligned}
 \Delta db_k &= db_{k+1} - db_k \\
 &= [2(x_{k+1} + 1)^2 + 2y_{k+1}^2 - 2y_{k+1} + 1 - 2r^2] - [2(x_k + 1)^2 + 2y_k^2 - 2y_k + 1 - 2r^2] \\
 &= 4x_k + 6 + 2(y_{k+1}^2 - y_k^2 - y_{k+1} + y_k)
 \end{aligned} \quad (10)$$

When  $db_k < 0$ ,  $y_{k+1} = y_k$ , then  $\Delta db_k = 4x_k + 6$ ; when  $db_k \geq 0$ ,  $y_{k+1} = y_k - 1$ , then  $\Delta db_k = 4(x_k - y_k) + 10$ .

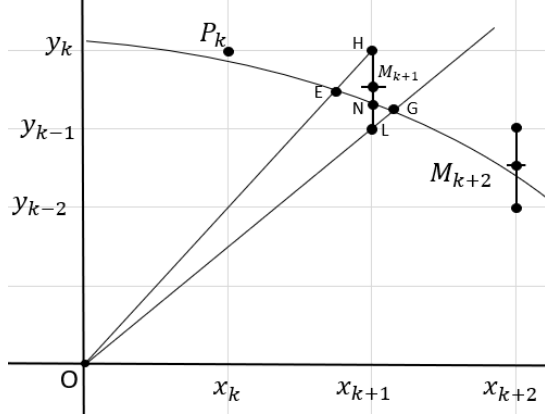
Which is

$$db_{k+1} = db_k + \Delta db_k = \begin{cases} db_k + 4x_{k+1} + 6, & db_k < 0 \\ db_k + 4(x_k - y_k) + 10, & db_k \geq 0 \end{cases} \quad (11)$$

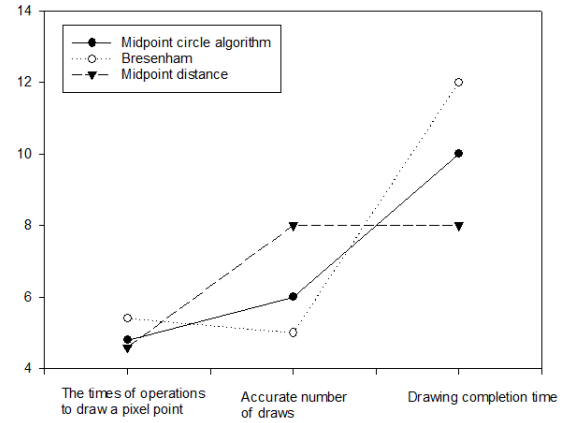
Substituting the circular function at the starting position coordinate  $(x_k, y_k) = (0, r)$  into the above circular function to obtain the initial value of the decision parameter.

$$db_0 = 3 - 2r \quad (12)$$

The coordinates of the next point can be obtained from equation (4), (11) and (12).



**Figure 3.** Midpoint distance circle generation algorithm.



**Figure 4.** Comparison of three algorithms.

## 5. Midpoint distance circle generation algorithm

As shown in Figure 3, the midpoint distance circle generation algorithm combines the advantages of the midpoint circle algorithm and the Bresenham circle algorithm, using both the midpoint as the

judgment basis and the distance approximation as the decision factor. Let the decision parameter of the midpoint distance circle algorithm be

$$d_k = \alpha dm_k + \beta db_k \quad (13)$$

among them,  $dm_k$  is the decision parameter of the midpoint circle algorithm, which can be obtained by equations (6) and (7),  $db_k$  is the decision parameter of the Bresenham circle algorithm, which can be obtained by equations (11) and (12),  $\alpha$  and  $\beta$  are the coefficients of the above two decision parameters, respectively, satisfying  $\alpha + \beta = 1, 0 \leq \alpha \leq 1, 0 \leq \beta \leq 1$ .

The recursive expression of the decision parameter of the available midpoint distance circle generation algorithm is

$$d_{k+1} = \begin{cases} \alpha(dm_k + 2x_{k+1} + 1) + \beta(db_k + 4x_{k+1} + 6), & dm_k < 0 \text{ and } db_k < 0 \\ \alpha(dm_k + 2x_{k+1} - 2y_{k-1} + 1) + \beta(db_k + 4(x_k - y_k) + 10), & dm_k \geq 0 \text{ and } db_k \geq 0 \\ \alpha(dm_k + 2x_{k+1} + 1) + \beta(db_k + 4(x_k - y_k) + 10), & dm_k < 0 \text{ and } db_k \geq 0 \\ \alpha(dm_k + 2x_{k+1} - 2y_{k-1} + 1) + \beta(db_k + 4x_{k+1} + 6), & dm_k \geq 0 \text{ and } db_k < 0 \end{cases} \quad (14)$$

Substituting the circular function at the starting position coordinate  $(x_k, y_k) = (0, r)$  into the above circular function to obtain the initial value of the decision parameter.

$$d_0 = \alpha\left(\frac{5}{4} - r\right) + \beta(3 - 2r) \quad (15)$$

It is verified by experiments that when  $\alpha$  is 0.62 and  $\beta$  is 0.38, the actual experimental results of decision parameters are relatively good, so equation (15) can be rewritten as

$$d_0 = 0.62 \times \left(\frac{5}{4} - r\right) + 0.38 \times (3 - 2r) = 1.915 - 1.38r \quad (16)$$

Equation (16) still involves floating-point operations, so simple rounding of floating-point numbers is obtained.

$$d_0 = 0.62 \times \left(\frac{5}{4} - r\right) + 0.38 \times (3 - 2r) = 2 - r \quad (17)$$

Although the decision to select pixel points using equation (17) is more accurate, the parameters  $\alpha$  and  $\beta$  can still be adjusted in real life to achieve the best experimental results.

Therefore, the steps of the midpoint distance circle generation algorithm are as follows:

1. Use the radius  $r$  and the center coordinates  $(x_c, y_c)$  of the circle and the first point  $(0, r)$  on the circumference as the known quantity

2. Calculate the initial value of the decision parameter  $d_0 = 2 - r$

3. Starting from the coordinates of the starting position, the x-axis coordinate is incremented by one pixel at a time. The value of the corresponding decision parameter is calculated according to the value of the previous decision parameter by (14), and the next pixel is judged according to the formula (4). coordinate

4. Calculate the coordinates of the corresponding symmetrical points of the remaining 7 octants

5. Move the calculated pixel point  $(x_k, y_k)$  through the translational transformation of the coordinates to the circumference of the center:  $x_k = x_c + x_k, y_k = y_c + y_k$

6. Repeat steps 3-5 until  $y \leq x$ , that is, complete the drawing of the circle.

## 6. Simulation

Opengl programming is used to realize the midpoint circle algorithm, Bresenham circle algorithm and the midpoint distance algorithm of this paper, and the experiment was run 16 times. The experimental results show that the proposed algorithm is a circle generation algorithm with high efficiency, strong stability and high flexibility. The result is shown in Figure 4.

## 7. Conclusions

The midpoint distance circle generation algorithm based on the midpoint circle algorithm and the Bresenham algorithm proposed in this paper is an easy to implement and simple calculation algorithm. Compared with the midpoint circle algorithm, the calculation process of floating point numbers is omitted. Compared with the Bresenham algorithm, it is more flexible and controllable. Therefore, the proposed algorithm is a feasible circle generation algorithm.

## 8. Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

## 9. Data availability

Data used to support the findings of this study are included with in the article.

## References

- [1] JIA Yin-liang, ZHANG Huane-hun, JING Ya-zhiA. Modified Bresenham Algorithm of Line Drawing[J]. JOURNAL OF IMAGE AND GRAPHICS, 2008, 13(1):163-166.
- [2] LAI Yuan, FENG Huanqing, ZHUGE Bin. An Algorithm of Mesh Point Generation for Surface Reconstruction[J]. COMPUTER ENGINEERING AND APPLICATIONS, 2002, 38(3):83-84.
- [3] Jianguang Sun.Computer Graphics[M].Beijing:Tsinghua university press,1998.
- [4] ZHU Xiao-lin, GAO Cheng-hui, HE Bing-wei, HUANG Min-chun, CHEN Jie. An Improved Method of Hough Transform Circle Detection Based on the Midpoint Circle-Producing Algorithm[J]. JOURNAL OF ENGINEERING GRAPHICS, 2010, 31(6):29-33.
- [5] Wang Zhixi, Wang Runyun. Improvement of Bresenham's Circle Generation Algorithm[J]. COMPUTER ENGINEERING, 2004, 30(12):178-180.
- [6] Shi Zhixin. Research on Basic Raster Graphics Generation Algorithm[D]. ShanDong University, 2007.
- [7] Lin Yun, Tan Wei-xian, Hong Wen, Wang Yan-ping, Wu Yi-rong. Polar Format Algorithm for Circular Synthetic Aperture Radar[J]. JOURNAL OF ELECTRONICS & INFORMATION TECHNOLOGY, 2007(9):58-58.
- [8] Wang Xide, Fang Qingyan. General conic curve algorithm based on octant arc[J]. HELONGJIANG SCIENCE AND TECHNOLOGY INFORMATION, 2007(9):58-58.
- [9] Wang Runyun, Wang Zhixi. A Kind of Fast and Accurate Algorithm for Circle Generation[J]. JOURNAL OF SICHUAN UNIVERSITY OF SCIENCE AND TECHNOLOGY, 2004, 23(3):43-44.
- [10] Donald Hearn,M Baker.Computer Graphics with OpenGL [M]. Prentice Hall,2003.