

doi: 10.7690/bgzd.2018.09.007

基于 Bresenham 的圆弧生成算法及其应用

王荣海, 赵丽梅, 乔之勇

(绵阳职业技术学院信息工程系, 四川 绵阳 621000)

摘要: 针对工件尺寸测量系统中圆弧绘制问题, 结合 Bresenham 圆生成算法, 设计出 3 点绘制顺序圆弧算法。介绍了候选点的生成及选取, 阐述了终止条件及坐标变换, 并将设计的方法与数字微分分析机 (digital differential analyzer, DDA) 算法和直接坐标生成方法进行了比较。实验结果表明: 较 DDA 算法、直接坐标生成方法及部分文献方法, 该算法绘制的圆弧视觉效果更好, 时间效率更高, 达到了工程应用的目的。

关键词: Bresenham 算法; 圆弧生成; 图形学; 人机交互; 尺寸测量

中图分类号: TP391 **文献标志码:** A

Circle Arc Generating Algorithm and Application Based on Bresenham

Wang Ronghai, Zhao Limei, Qiao Zhiyong

(Department of Information Engineering, Mianyang Polytechnic, Mianyang 621000, China)

Abstract: Aiming at the problem of circle arc drawing in work piece size measurement system, a 3 points sequential circle arc drawing algorithm is designed by combining with Bresenham circle generation algorithm. This paper introduces the generation and selection of candidate points. The termination condition and coordinate transformation are expounded. The methods of the design are compared with digital differential analyzer (DDA) algorithm and direct coordinate generation method. The experimental results show that compared with DDA algorithm, direct coordinate generation method and some literature methods, the algorithm has better visual effect and higher time efficiency, and achieves the purpose of engineering application.

Keywords: Bresenham algorithm; circle arc generating; graphics; human computer interaction; size measurement

0 引言

近年来, 随着计算机视觉技术的发展与应用, 视觉测量技术被越来越多地应用到零部件的尺寸测量中。小零件尺寸测量过程中, 不可避免存在圆弧曲线。为了直观体现圆弧检测区域(通过 3 点确定的圆弧加上弧环宽度确定)或检测出的圆弧, 以及一些类似于角度标注等需求, 均需要通过圆弧生成算法绘制出来, 达到现实增强的效果。

在圆弧生成算法中, 主要有正多边形逼近法和插补法^[1-12], 问题的关键点基本都集中在圆弧点的生成及选取上。但是正多边形逼近的 DDA 方法与通过解圆方程的直接坐标生成方法视觉效果较差, 线条比较粗, 影响实际应用效果。传统的大多数方法要么代码实现复杂, 要么逻辑判断过于繁琐, 不利于尺寸测量中圆弧绘制, 特别是同型号零件的批量检测, 圆弧绘制算法视觉效果越好, 绘制速度越快越好^[13-17]。从代码实现难易程度及算法复杂程度来看, 文献[1, 8]比较适合, 但是还需要从复杂程度和简洁程度方面改进。

针对以上情况, 笔者结合工程应用实际, 将问题转化成 3 点绘制圆弧问题。通过 Bresenham 圆生成算法, 结合圆弧的时钟方向, 依据当前动点的象限判断, 获取候选点。通过残差控制从候选点中选取新的动点, 设计出新的终止条件判定方法。算法条理清晰, 优化逻辑判断, 尽量考虑到时间效率, 利于计算机编程实现, 也利于代码移植。

1 候选点的生成及选取

候选点的生成主要利用了圆弧的时钟方向及当前动点所在象限进行生成。而生成的 3 个候选点的选择则使用残差控制, 选择残差评价小的点, 形成新的动点。

1.1 候选点的生成

在候选点的选择上, 主要使用了 Bresenham 圆生成算法的思想。如图 1 所示, 已知圆弧上 3 个不同位置的点 p_1 、 p_2 、 p_3 (起点、中间点、终点), 以这 3 点所确定的圆弧弧心为原点, 建立标准的笛卡尔坐标系。设动点为 p , 生成的候选点分别是 H 、 D 、

收稿日期: 2018-07-03; 修回日期: 2018-07-17

基金项目: 四川省科技计划项目(2018JY0337)

作者简介: 王荣海(1969—), 男, 四川人, 工程硕士, 教授, 从事电子与通信工程研究。

V ，可以推导出候选点的坐标在 p 点基础上的增量关系，如表 1 所示。

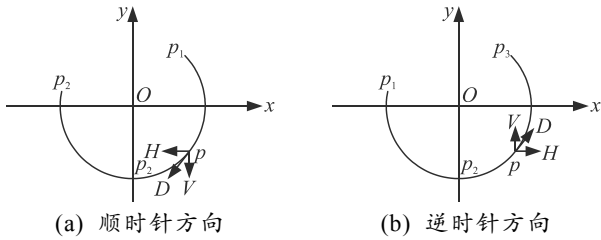


图 1 圆弧方向与候选点的关系

表 1 候选点与动点的关系

象限(顺)	象限(逆)	H点增量	D点增量	V点增量
1	3	$\Delta x=1, \Delta y=0$	$\Delta x=1, \Delta y=-1$	$\Delta x=0, \Delta y=-1$
2	4	$\Delta x=1, \Delta y=0$	$\Delta x=1, \Delta y=1$	$\Delta x=0, \Delta y=1$
3	1	$\Delta x=-1, \Delta y=0$	$\Delta x=-1, \Delta y=1$	$\Delta x=0, \Delta y=1$
4	2	$\Delta x=-1, \Delta y=0$	$\Delta x=-1, \Delta y=-1$	$\Delta x=0, \Delta y=-1$

如果象限 1~4 的编码分别是 (ab 表示) 11, 01, 00, 10, 时针的顺逆 (c 表示) 分别是 1, 0, 那么可以得到如表 2、表 3 的卡诺图(卡诺图某项值为真, 则增量为+1, 否则为-1)。

$$f(H_{\Delta x}) = b \wedge c + \bar{b} \wedge \bar{c}; \tag{1}$$

$$f(V_{\Delta y}) = \bar{a} \wedge c + a \wedge \bar{c}. \tag{2}$$

表 2 $H_{\Delta x}$ 的卡诺图

c	ab			
	00	01	11	10
1		1	1	
0	1			1

表 3 $V_{\Delta y}$ 的卡诺图

c	ab			
	00	01	11	10
1	1	1		
0			1	1

因为 $D_{\Delta x}$ 和 $D_{\Delta y}$ 的卡诺图分别与 $H_{\Delta x}$ 的卡诺图和 $V_{\Delta y}$ 的卡诺图一致。所以有：

$$f(D_{\Delta x}) = b \wedge c + \bar{b} \wedge \bar{c}; \tag{3}$$

$$f(D_{\Delta y}) = \bar{a} \wedge c + a \wedge \bar{c}. \tag{4}$$

这样就可以使用这组编码的逻辑运算来判定 3 个候选点的增量取值，从而在具体编写代码时就可以减少逻辑判断。

1.2 残差控制

获取到 H 、 D 、 V 这 3 个候选点后，可以通过使用残差函数从这 3 点中得到最理想的点，使得这一点尽可能地靠近圆弧的理论轨迹。设选取的点是 p_{next} ，那么满足 $f(p_{next}) = |\sqrt{x_p^2 + y_p^2} - R|$ 最小化，为

了减少开方的运算，可以使用 $f(p_{next}) = |x_p^2 + y_p^2 - R^2|$ 作为残差的判别函数。但是如果为了避免平方运算，可以使用文献[1]中的误差项判别函数。并且，由于这里是使用弧心为原点建立的笛卡尔坐标系，因此在倍数运算时，还避免了减法运算后再倍数运算。最后再统一坐标变换到绘图坐标系中。由于计算机硬件的发展，以前面临的存储与运算速度的问题已有很大的改变。

2 终止条件及坐标变换

终止条件选取最经典的做法是使用当前动点 p 与终止点的重合度评价。常见的有点点距离判定方法，以及点点 x 坐标之差绝对值与 y 坐标绝对值之和判断。除此之外，笔者新设计了 3 种行之有效的终止条件。迭代次数的最大次数可以设置为弧长的理论长度。

2.1 圆弧方向判断

如图 2(a)所示，在圆弧上的点，其弧(p_1, p, p_3)的时钟方向始终与弧(p_1, p_2, p_3)是同一方向，不在弧上的点不满足这条性质；因此，可以使用这个条件作为终止条件的选取。起止点及其附近的点可以做特殊处理，以提高算法由于浮点运算误差带来的不良影响。

2.2 弧心角判断

如图 2(b)所示，在圆弧上的点 p ，可以构成弧(p_1, p, p_3)，其弧心角始终与弧(p_1, p_2, p_3)的弧心角相等。基于该条件，可以设置另一个终止条件。该方式还可以使用分段画弧的方式等效代替，如果弧(p_1, p_2, p_3)是劣弧或平弧，设向量 op_1, op_3 夹角为 θ ，设向量 op_1, op 夹角为 θ_1 ，向量 op, op_3 的夹角为 θ_2 ，那么有 $\theta = \theta_1 + \theta_2$ ；如果弧(p_1, p_2, p_3)是优弧，那么就可以先计算出 p_1 关于原点的对称点 p_1' ，这样可以得到一条平弧及劣弧，先利用上面的方式画一段平弧，然后再绘制剩下的劣弧。上面直接使用弧心角判断一次性绘制圆弧和分 2 次绘制圆弧是等价的。为了更好地规避计算过程中的截断误差影响，不宜使用弧度。

2.3 弦判断

如图 2(c)所示，弧的起点和终点构成弧的一条弦 $\overline{p_1 p_3}$ ，那么在圆弧上的点 p 与点 p_2 始终在弦 $\overline{p_1 p_3}$ 的同侧。利用这个点与弦的相对位置关系可以设置成一个终止条件，因此只需要判断点与直线的相对位置关系即可。

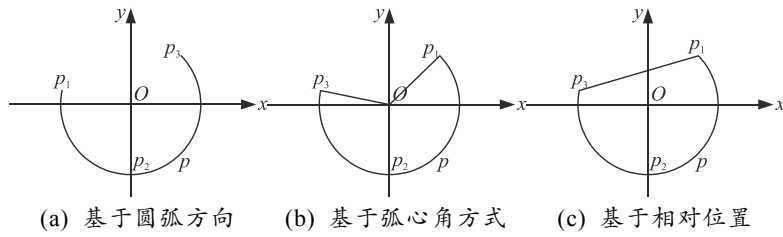


图 2 3 种不同的终止条件

2.4 点重合度判断

由于在圆弧上的动点 p ，由起点 p_1 和圆弧方向严格约束了其生成，所以从理论上讲， p 点会与终点 p_3 重合；因此，终止条件可以判定 p 与 p_3 的重合度判断，可以使用 p_1 与 p_3 的距离判别重合度。为了规避开方运算，可以使用 $(x_p - x_{p_3})^2 + (y_p - y_{p_3})^2 < 1$ 来判断。为了规避平方运算，也可以使用文献[1]的方法，即 $|x_p - x_{p_3}| < 1$ 且 $|y_p - y_{p_3}| < 1$ 。

2.5 坐标变换

得到当前动点 p 的坐标后，这个点坐标是以弧心为原点建立标准笛卡尔坐标系下的点，需要变换到绘图坐标系下进行绘图。由于 2 个坐标系最多只是 x 或 y 轴的方向相反，因此不会存在因坐标变换时旋转带来的二次截断误差。式(5)是从建立笛卡尔坐标系，变换到标准的绘图坐标系 (x 轴正向为向右， y 轴正向为向下，原点在绘图区左上角) 的变换关系式。设弧心 O 点在屏幕坐标系下的坐标点为 (x_0, y_0) ， p 点在笛卡尔坐标系下的坐标点为 (x_p, y_p) ，在屏幕坐标系下的坐标点为 (x'_p, y'_p) ，则变换表达式为：

$$\begin{bmatrix} x'_p \\ y'_p \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_p \\ y_p \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \quad (5)$$

2.6 算法实现

通过以上描述，算法的具体实现流程如图 3。

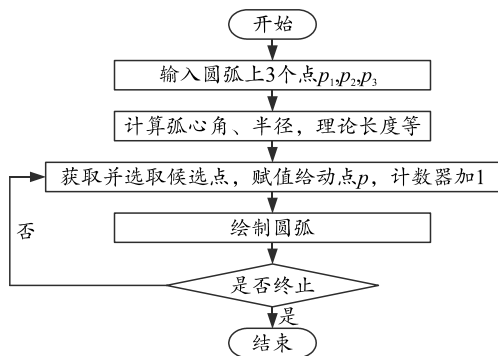


图 3 算法流程

算法具体描述如下：

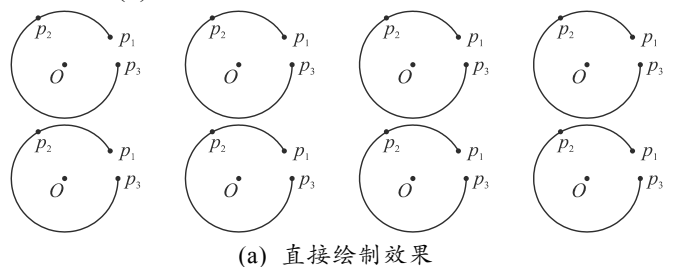
- 1) 输入圆弧上的 3 个不同点， p_1, p_2, p_3 ，分别是起始点、中间点和终止点。必要时可以进行坐标转换。计算出圆弧的弧心角 θ ，半径 R ，弧心点 O ，理论长度 L ，圆弧始终方向。初始化动点 p 为 p_1 并绘制。
- 2) 依据 1.1 节获取候选点，依据 1.2 节选取候选点，赋值给动点 p 。计数器加 1。
- 3) 将候选点坐标变换到屏幕坐标系下绘制。
- 4) 选取 2.1~2.4 节的某个终止判定条件，判定动点 p 停止与否。
- 5) 计数器 $\geq L$ 。
- 6) 计算 4) 和 5) 的逻辑或，若为假，跳转到 2)。
- 7) 算法结束。

3 实验及应用

将设计的方法与 DDA、直接坐标生成实现，使用 C++ 编程语言，在 Visual Studio 2015 上编程实现，并进行算法比较。实验的软硬件平台为英特尔 I7 4720HQ 处理器，2.6 GHz，64bit Windows10 操作系统，DDR3 1600 Hz 16.0 GB 内存。

选取固定的 3 个点，半径 60，使用 8 种不同方法直接绘制出效果。

图 4(a) 对应的算法分别是 2.1~2.4 节、文献[1,8]、DDA 以及坐标直接生成算法。其中，文献[8]使用了其中的正负圆生成算法，DDA 算法使用正多边形逼近的方法，坐标直接生成是通过横向和纵向扫描解圆方程，筛选解方程得到的点进行绘制。图 4(b) 是将获取的圆弧点与半径之差扩大 10 倍后绘制的效果。图 4(c) 为将极大圆、极小圆和理论圆绘制出来。



(a) 直接绘制效果

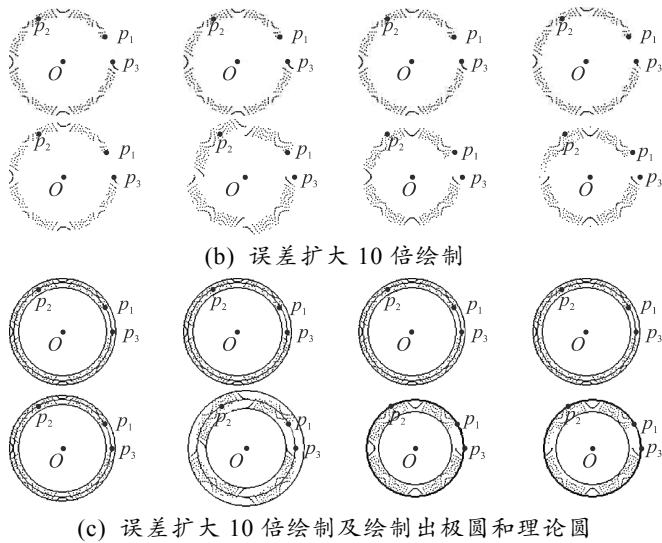


图 4 不同方法效果对比

表 4 是统计获取的圆弧点与半径之差的数据，对应的图形是图 3，由于 2.4 节与文献[1]使用的是逐点筛选，终止条件控制极为严格，可以看成是圆弧栅格标准图形，用于对比。从表 4 可以看出，提出的方法比 DDA、直接坐标生成方法，文献[8]的方法标准差更小，均值在 0 附近，结合图 4(b)、(c) 可以看出：绘制的像素点比较均匀地分布在理论圆

两侧，而 DDA 算法与直接坐标生成方法，点基本分布在理论圆以内，从视觉上体现则更光滑连续。从表 4 的极差及图 4(c) 可以看出，本文中方法获取的线条更细。从而可以得出：较 DDA、直接坐标生成方法与文献[8]方法，该方法绘制的圆弧更加连续光滑，线条更细，视觉效果更好。与文献[1]效果对比，设计的新方法效果至少持平，也证明了设计终止条件的有效性，但由于精度控制问题，存在极小的像素点数误差。

使用随机点测试，在某个区域中每次随机产生 3 个点，然后筛选出能够确定出圆弧的组点，总规模 100 次。得到的实验数据为表 4 所示。

由表 5 的数据可知，该方法绘制效率比较快速，普遍比 DDA 和直接坐标生成的方法时间效率更高，并且 DDA 与直接坐标生成方法大量生成重复点，文献[8]的方法也存在部分重复点。设计的新方法在时间效率上与主流标准方法一致。

使用圆弧检测功能检测出的圆弧如图 5 所示。其中阴影部分由鼠标点下的 3 点及弧环宽度决定圆弧检测区域。检测出的亚像素点拟合出的圆弧由虚线绘制。

表 4 不同方法绘制数据比较

方法	像素点数	最大值/像素	最小值/像素	均值/像素	方差	标准差/像素	极差/像素
2.1	311	0.492 8	-0.465 3	0.044 1	0.064 2	0.253 3	0.958 1
2.2	310	0.492 8	-0.465 3	0.045 6	0.065 3	0.255 6	0.958 1
2.3	309	0.492 8	-0.465 3	0.044 6	0.064 5	0.254 0	0.958 1
2.4	309	0.492 8	-0.465 3	0.044 6	0.064 5	0.254 0	0.958 1
文献[1]	309	0.492 8	-0.465 3	0.044 6	0.064 5	0.254 0	0.958 1
文献[8]	444	1.008 1	-0.991 5	-0.071 7	0.227 3	0.476 7	1.999 7
DDA 算法	909	0.000 0	-1.331 4	-0.686 0	0.081 6	0.285 6	1.331 4
直接坐标生成	824	-0.058 4	-1.339 7	-0.698 9	0.095 9	0.309 7	1.341 3

表 5 不同方法比较(矩形区域随机获取 3 点)

方法	时间/ms	平均像素点距离	平均像素点个数	平均理论长度	平均弧心角/(°)	平均半径
2.1	7.53	609.21	528.66			
2.2	7.96	610.91	529.12			
2.3	8.28	610.75	528.93			
2.4	7.73	608.63	528.08			
文献[1]	8.59	608.63	528.08	577.16	112.35	363.33
文献[8]	7.68	726.58	727.58			
DDA 算法	13.23	577.27	1 515.90			
直接坐标生成	12.91	—	1 480.14			

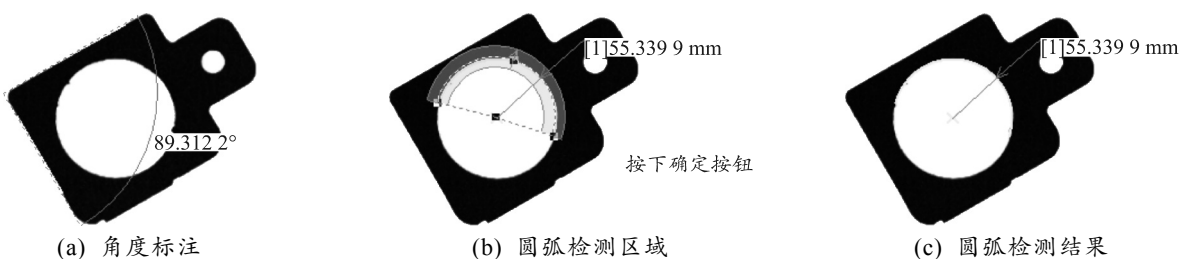


图 5 部分应用效果

4 结束语

笔者使用 Bresenham 圆生成算法, 结合动点象限判断以及圆弧方向等条件, 尽可能生成准确的圆弧点, 从而使圆弧绘制的准确性得到保障, 在候选点生成上简化了逻辑运算, 从而使编码更简洁。在圆弧可选择的终止条件方面, 笔者不仅使用了传统判定方法, 还设计出了新的判定方法, 在精度上符合标准终止条件方法。由于绘制的圆弧视觉效果及时间效率普遍好于传统算法, 并且生成的点是按圆弧时钟方向顺序生成, 因而更加适合有特殊需求的圆弧点集。将该算法应用到工件尺寸测量系统中, 绘制效果的综合性能至少与标准方法持平, 达到了工程应用的目的。

参考文献:

- [1] 江巨浪, 张佑生. 连续生成任意角度的新算法[J]. 合肥工业大学学报(自然科学版), 2003, 5(5): 1398-1441.
- [2] 孔逸敏, 凌秀泽, 徐寅林. 基于符号判别法的逐点比较法圆弧插补算法的研究[J]. 南京师范大学学报(工程技术版), 2012, 12(2): 64-68.
- [3] 牛连强, 薛瑾, 朱天翔. 快速绘制圆弧的行程算法[J]. 沈阳工业大学学报, 2010, 32(4): 411-416.
- [4] 程锦, 陆国栋, 谭建荣. 一种快速圆弧绘制算法[J]. 软件学报, 2002, 13(12): 2275-2280.
- [5] 权秀敏, 尹显明. 一种简捷的快速圆弧插补新算法的设计与计算[J]. 机械研究与应用, 2008, 21(1): 81-83.
- [6] 丁宇辰. 圆弧的生成算法研究[J]. 南京工程学院学报(自然科学版), 2010, 8(2): 59-62.
- [7] 张胜男, 刘连强, 陈欣. 有效的混合圆弧扫描转换算法[J]. 图形学报, 2012, 33(6): 50-58.
- [8] 孙崇璇. 圆弧生成方法探究与实现[J]. 云南师范大学学报, 2011, 31(4): 64-68.
- [9] 牛连强, 冯海文, 吴澎. 利用残差控制的快速圆弧生成与反走样算法[J]. 计算机辅助设计与图形学学报, 2011, 23(20): 232-239.
- [10] 牛连强, 张胜男, 钟玲. 圆弧扫描转换的混合算法[J]. 沈阳工业大学学报, 2013, 35(1): 72-78.
- [11] 刘勇奎. 线生成圆弧的最佳逼近新算法[J]. 计算机辅助设计与图形学学报, 1990(2): 24-27.
- [12] 卓守鹏. 一种快速生成圆弧的算法[J]. 浙江大学学报, 1986, 20(4): 160-165.
- [13] 李伟光, 曾顺星, 叶国强, 等. 一种无直线逼近的高速高精度圆弧插补算法[J]. 组合机床与自动化加工技术, 2014(3): 53-57.
- [14] 刘焕, 刘强, 周胜凯, 等. 基于微分模型的空间圆弧与椭圆弧插补研究[J]. 农业机械学报, 2015, 46(8): 338-343.
- [15] BRESENHAM J E. A linear algorithm for incremental digital display of circular arcs [J]. Communications ACM, 1977, 20(2): 100-106.
- [16] WANG W, JOE B, WANG C Y. On the difference method for drawing conic arcs[J]. CVGIP: Graphical Models and Image Processing, 1994, 56(1): 8-18.
- [17] BRESENHAM J E. Pixel-processing fundamentals[J]. IEEE Computer Graphics and Applications, 1996, 16(1): 74-82.