

# 一种宽椭圆生成算法

王文杰<sup>1,2</sup>, 胡 君<sup>1</sup>, 梁凤超<sup>1</sup>

(1. 中科院长春光学精密机械与物理研究所, 吉林 长春 130033; 2. 中国科学院研究生院, 北京 100039)

**摘 要:** 计算机辅助设计领域常涉及宽椭圆生成, 宽椭圆生成算法的优劣直接影响设计效果。为了生成一个圆心在原点的标准宽椭圆, 每次用单像素宽的椭圆中点扫描转换算法, 得到一个单像素宽椭圆上的一个点, 填充一个以该点为中心, 椭圆宽为直径的圆弧, 扫描转换结束后, 生成一个无明显视觉缺陷的第一象限  $\frac{1}{2}$  宽椭圆。

**关 键 词:** 计算机应用; 宽椭圆生成法; 扫描转换; 圆弧平移

**中图分类号:** TP 391

**文献标识码:** A **文 章 编 号:** 1003-0158(2009)04-0076-04

## Algorithm for Generation of Wide Ellipse

WANG Wen-jie<sup>1,2</sup>, HU Jun<sup>1</sup>, LIANG Feng-chao<sup>1</sup>

(1. Changchun Institute of Optics, Fine Mechanism and Physics, Chinese Academy of Sciences, Changchun Jilin 130033, China;

2. Graduate School of the Chinese Academy of Sciences, Beijing 100039, China)

**Abstract:** Ellipse always can be generated in CAD system, but in some times the wide ellipse is needed in some design, so it is necessary to research the algorithm of wide ellipse generation. For a wide ellipse whose center point is on origin point, an arc is drawn when got a point on the ellipse with center scan transformation algorithm around the point till the scanning is end.

**Key words:** computer application; wide ellipse generation; scan transformation; arc moving

作为计算机图形学中基本几何元素之一的椭圆, 其生成算法在几乎所有计算机图形学相关领域都要用到, 尤其在计算机辅助设计中经常涉及。因此, 研究椭圆生成对计算机图形系统十分重要。目前, 已有大量的文献讨论了如何高效生成误差小的椭圆。椭圆的扫描转换法<sup>[1]</sup>就是其中之一, 该算法基于 Da Silva 的算法<sup>[2]</sup>, 运用二阶偏差分 Pitteway<sup>[3]</sup>, Van Aken<sup>[4]</sup>、KAppel<sup>[5]</sup>等所用

的一些技术, 该算法生成的椭圆都是单像素宽的, 而现实中更多时候要生成宽椭圆, 宽椭圆一般定义为沿着垂直两半径为  $a$ 、 $b$  的椭圆弧的两方向, 将此椭圆上的点移动  $\frac{w}{2}$  的距离所形成的两条曲线中间部分, 为了生成宽椭圆, 文献一中方法之一在扫描转换的同时复制椭圆宽度数个像素, 这种方法比较简单, 但造成椭圆切线斜率接

近-1 处显得很细。文献一中方法之二扫描转换两个同心的椭圆, 内椭圆的两个半径分别为  $a - \frac{w}{2}$ ,  $b - \frac{w}{2}$ ; 外椭圆的两个半径为  $a + \frac{w}{2}$ ,  $b + \frac{w}{2}$ ; 然后填充它们间的间隙, 在微分几何中有一个结论: 沿着垂直椭圆弧的方向, 将此椭圆上的点移动  $\frac{w}{2}$  的距离所形成的曲线与原椭圆同心的椭圆, 而是由一个 8 次方程所描述的曲线<sup>[6]</sup>, 因此这种算法也有较大误差, 特别是  $a$  的值接近于  $w$  时。然而, 对这样 8 次函数进行扫描转换, 计算量非常大。圆弧绘制生成宽椭圆算法与椭圆中点扫描转换算法复杂度相当, 且生成的椭圆效果较好, 视觉感受不到明显缺陷。

## 1 算法基本思想

主要考虑中心在原点的标准椭圆  $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$ , 宽度为  $w$ , ( $a > b > w$ ,  $a, b, w \in \mathbb{Z}$ ); 对于中心不在坐标原点的椭圆, 可先作相应的平移变换, 变换为中心在原点的椭圆, 把所得的像素坐标加上一个位移即可得到所求的像素坐标。此外, 只讨论椭圆在第一象限内的生成算法, 其他象限内的点利用椭圆的四分对称性即可得到。

在第一象限内的四分之一椭圆分为两个区域来处理。两个区域之间以斜率为-1 的点(即法向量两个分量相等的点)作为分界。对于区域 I, 以点(0,  $b$ )为始点,  $x$  方向单位长作为步长。向右生成曲线。

假设当前扫描计算得到的点为批  $p_0(x, y)$ , 扫描转换的下一个点  $p_1$  可能为  $s_1(x+1, y)$  或  $s_2(x+1, y-1)$ , 判断  $s_1, s_2$  的中间点  $m(x+1, y+1/2)$  在椭圆的内还是外, 选择下一个扫描点。

如果中点  $m$  在内( $m$  点代入椭圆方程值小于 1), 则下一个点  $p_i$  为  $(x+1, y)$ , 否则为  $(x+1, y-1)$ ;

当斜率变为小于-1时,转向区域II,此时以 $(a, 0)$ 为始点。 $y$ 方向作为单位步长,向左生成曲线。

假设当前扫描计算得到的点为  $p_0'(x, y)$ , 扫描转换的下一个点可能为  $s_1'(x, y+1)$  或  $s_2'(x-1, y+1)$ , 判断  $s_1', s_2'$  的中间点  $m(x - \frac{1}{2}, y + \frac{1}{2})$

在椭圆的内还是外，选择下一个扫描点。

如果中点在内, 则下一个点  $p_1'$  为  $(x, y+1)$  否则为  $(x-1, y+1)$ 。

以上简单介绍了椭圆中点扫描生成算法,接下来基于中点扫描算法来生成宽椭圆。

宽椭圆生成算法思路: 先以中点圆算法<sup>[1]</sup>生一个圆心在原点直径为  $w$  的圆, 填充该圆, 然后以椭圆扫描转换算法[1]求得椭圆  $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$  上的点, 将填充的圆平移到扫描转换得到的各点处 (如图 1)。

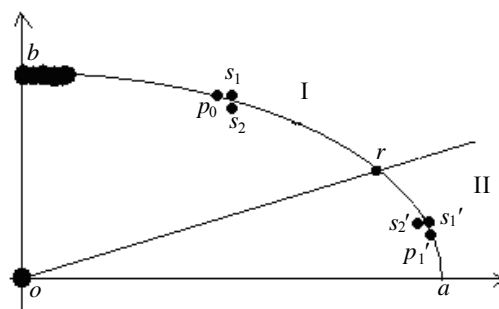


图 1 生成椭圆第一象限

## 2 算法实现过程

如图 2, 椭圆扫描转化得到的相邻点  $P_0(x_0, y_0)$ 、 $P_1(x_1, y_1)$ , 它们对应的圆形填充区为  $U_1$ 、 $U_0$ , 填充圆于造成中间大量的像素  $U_1 \cap U_2$  重复绘制, 算法复杂性较高, 实际上只须绘制  $U_1 - U_0$ , 即图中黑色的部分, 这些点全在圆边上, 为“半圆弧”, 这个半圆弧相对扫描点  $P_1$  的位置以及它的形状取决于扫描推进的方向, 也就是  $P_1$  与  $P_0$  的位置关系。图 1 所示, 根据椭圆扫描转换分 4 种来考虑它们的位置 ( $\theta$  为圆心角):

(1) 扫描第一区间(切线斜率  $r>-1$ ),  $p_1$  取  $s_1$ ,  $x_1=x_0+1$ ,  $y_1=y_0$  时, 如图 2 所示。

$U_1-U_0$  为图中黑色右半圆弧, 绘制右半圆弧的算法:

1) 从最左的一个点开始, 取  $1/8$  圆  $\left(\frac{\pi}{4} \leq \theta \leq \frac{\pi}{2}\right)$  上的一个点  $q$ ;

2) 由对称性, 计算  $q$  对应  $\pi/2$  圆  $\left(-\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}\right)$  的 3 个对称点  $q_0, q_1, q_2$ ;

3) 将  $q$ 、 $q_0$ 、 $q_1$ 、 $q_2$  平移  $(x_1 i, y_1 j)$ ,  $(x_1, y_1)$  为  $p_1$  的坐标;

4) 判断是不是最后一个点, 不是的话, 取下一个点; 回到 2); 是则圆弧绘完。

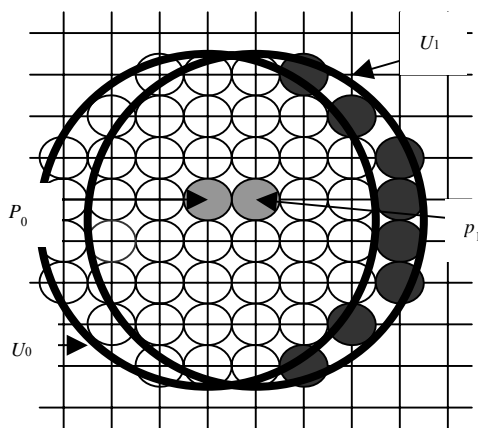


图2 中点扫描向右推进时相邻两次填充区的关系

(2) 扫描第一区间 (切线斜率  $r > -1$ ),  $p_1$  取  $s_2$ ,  $x_1 = x_0 + 1$ ,  $y_1 = y_0 - 1$ , 如图3所示。

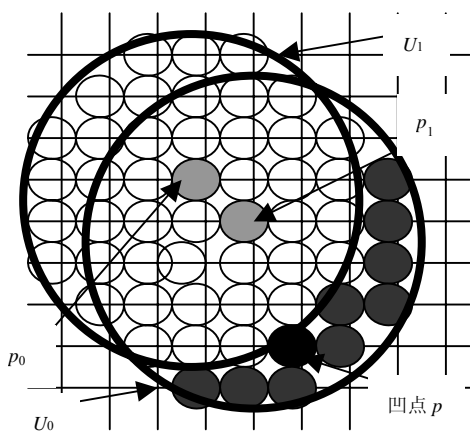


图3 中点扫描斜下推进时相邻两次填充区的关系

$U_1 - U_0$  为图中黑色半圆弧  $\left(-\frac{3\pi}{4} \leq \theta \leq \frac{\pi}{4}\right)$ , 图中类似  $p$  点的凹点要绘制, 绘制右下半圆弧的算法:

- 1) 从最左的一个点开始, 取  $1/8$  圆  $\left(\frac{\pi}{4} \leq \theta \leq \frac{\pi}{2}\right)$  上的一个点  $q$ ;
- 2) 由对称性, 计算  $q$  对应在  $1/2$  圆  $\left(-\frac{3\pi}{4} \leq \theta \leq \frac{\pi}{4}\right)$  的 3 个对称点  $q_0$ ,  $q_1$ ,  $q_2$ ;
- 3) 将  $q$ 、 $q_0$ 、 $q_1$ 、 $q_2$  平移  $(x_1 i, y_1 j)$ ,  $(x_1, y_1)$

为  $p_1$  的坐标;

4) 判断  $q$  边有没有凹点, 有则取凹点执行 2)、3);

5) 判断是不是最后一个点, 不是的话, 取  $q$  下一个点; 回到 2); 是则圆弧绘完。

用同样的方法可以填充第二区间的椭圆。

(3) 扫描第二区间 (切线斜率  $r \leq -1$ ), 当  $p_1$  取  $s_1$ ,  $x_1 = x_0$ ,  $y_1 = y_0 - 1$  时, 如图4所示。

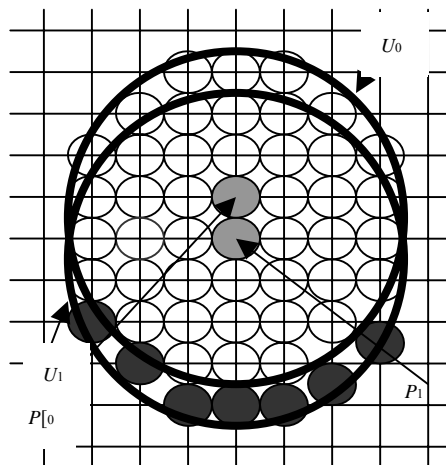


图4 中点扫描向下推进时相邻两次填充区的关系

$U_1 - U_0$  为图中黑色半圆弧  $(-\pi \leq \theta \leq 0)$ , 绘制下半圆弧的算法:

- 1) 从最左的一个点开始, 取  $1/8$  圆  $\left(\frac{\pi}{4} \leq \theta \leq \frac{\pi}{2}\right)$  上的一个点  $q$ ;
- 2) 由对称性, 计算  $q$  对应在  $1/2$  圆  $(-\pi \leq \theta \leq 0)$  的 3 个对称点  $q_0$ ,  $q_1$ ,  $q_2$ ;
- 3) 将  $q$ 、 $q_0$ 、 $q_1$ 、 $q_2$  平移  $(x_1 i, y_1 j)$ ,  $(x_1, y_1)$  为  $p_1$  的坐标;
- 4) 判断是不是最后一个点, 不是的话, 取下一个点; 回到 2); 是则圆弧绘完。

(4) 扫描第二区间 (切线斜率  $r \leq -1$ ), 当  $p_1$  取  $s_2$ ,  $x_1 = x_0 + 1$ ,  $y_1 = y_0 - 1$ , 此时  $p_0$  与  $p_1$  位置与 (2) 同, 见图2, 必须绘制右下半圆弧, 其算法见绘制右下半圆弧的算法。

### 3 宽椭圆具体的生成过程

(1) 生成直径为椭圆宽度  $w$  的  $1/8$  圆, 以数组  $y[n]$  记下  $1/8$  圆的纵坐标值;

(2) 填充圆心在  $(0, b)$  处直径为  $w$  的圆,

$p_0$  为  $(0, b)$ ;

(3) 判断  $p_0$  是否为分界点, 是转(7), 不是转入(4);

(4) 由椭圆的扫描转换法求得  $p_1$ ;

(5) 判断  $p_1$  与  $p_0$  是对角关系还是水平相邻, 如果是水平相邻, 绘制右半圆, 如果斜对角, 绘制右下半圆;

(6)  $p_0 = p_1$ , 转(3);

(7) 填充圆心在  $(0, b)$  处直径为  $w$  的圆,  $p_0$  为  $(a, 0)$ ;

(8) 判断  $p_0$  是否为分界点, 不是分界点转入(9), 是则结束;

(9) 由椭圆的扫描转换法求得  $p_1$ ;

(10) 判断  $p_1$  与  $p_0$  是对角关系还是垂直相邻, 如果是垂直相邻, 绘制上半圆, 如果斜对角, 绘制左上半圆;

(11)  $P_0 = P_1$ , 转(8)。

## 4 结 束 语

Windows XP 系统下, Visual C++实现本算法生成的一个横半轴为 100 像素, 纵半轴为 60 像素, 宽为 10 像素的椭圆见图 5, 对比列复制算法和矩阵画刷平移法, 本算法生成的宽椭圆整体更加匀称, 局部更平滑。

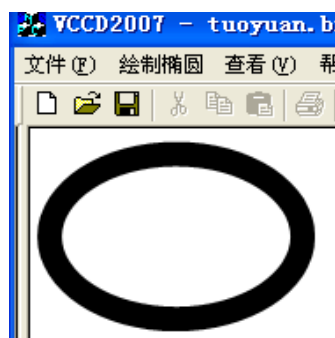


图 5 采用本算法生成的一个宽椭圆

另外, 本算法用一个数组记下了  $1/8$  圆的纵坐标, 牺牲较小的存储空间, 以画弧线方式代替圆形画刷填充方式, 大大减小了复杂度, 生成的椭圆效果较好, 尤其是较大较宽的椭圆。

## 参 考 文 献

- [1] [美]James D Foley 等著. 计算机图形学及实践 C 语言描述(原书第 2 版)[M]. 唐泽华等译. 北京: 机械工业出版社, 2004. 60-64.
- [2] DA S. Raster algorithms for 2D primitives [D]. Computer Science Department, Brown University, Providence, RI, 1989.
- [3] PITTEWAY M, Algorithm for drawing ellipse or hyperbolae with a digital plotter [J]. Computer J., 1967, 10(3): 282-289.
- [4] VAN A. An efficient ellipse-drawing algorithm [J]. CG&A, 1984, 9(4): 24-35.
- [5] KAPPEL M. An ellipse-drawing algorithm for raster display in earnshaw, R.,ed [J]. Fundamental Algorithms for Computer Graphics, NATO ASI Series, Springer Verlag, Berlin, 1985, 5(1): 257-280.
- [6] SALMON G. Treatise on conic sections (10th edition) [M]. London: Longmans, Green, &Co., 1896. 11-14.
- [7] 阎 双, 唐 棣. 椭圆的双步生成算法[J]. 计算机工程与应用, 2006, 33(4): 66-67.
- [8] 赵京东. 一个椭圆生成算法[J]. 计算机工程与应用, 2006, 35(6): 27-29.
- [9] 陈震岳, 朱桂林. 基于几何关系的椭圆图形生成算法[J]. 计算机工程与科学, 2004, 26(8): 56-59.