第 19 卷第 4 期　　　　广西科学院学报　　　　Vol. 19, No. 4
2003 年 11 月　　Journal of Guangxi Academy of Sciences　　November 2003

基于多步法绘制理论的抛物线裁剪算法

Parabola Clipping Algorithm Based on N-Steps Rendering Theory

李建华

Li Jianhua

(广西师范大学数学与计算机科学学院　桂林　541004)

(College of Math. and Comp. Sci., Guangxi Normal University, Guilin, 541004)

摘要　基于多步法绘制原理和 Brensenham 算法，提出一种新的关于抛物线的线性化裁剪算法。该算法首先线性化计算，由给定抛物线生成绘制时所需的两个数组，然后考虑到各种裁剪情况，利用两数组实现抛物线与窗口裁剪线的求交运算，得到相应的裁剪数据，最后再绘制出所求的裁剪图形.

关键词　抛物线　多步数目　多步区间　多步数组　弧边形　裁剪算法

中图法分类号　TP301.6；O182

Abstract　A new clipping algorithm about parabola based on N-step rendering theory and Bresenham algorithm is proposed. The main idea of this algorithm is that two rendering arrays are firstly produced through linear computation by a definite parabola, then while all kinds of clipping situations being considered and the arrays being utilized, the clipping data is obtained by a computation which is a kind of insection computing between parabola and the window clipping line, finally the clipping graphics is rendered.

Key words　parabola, N-step number, N-step interval, N-step array, arc polygon, clipping algorithm

裁剪算法是计算机图形学中的几个主要的基础算法之一，而如何使诸如圆和抛物线一类的常用圆锥曲线的裁剪达到线性化程度，从而避免非线性方程的求根计算，成为许多学者所关注的焦点. 本文试图就抛物线的线性化裁剪问题进行一些有益的探索. 自 1987 年文献[1]提出一种被称为二步法的光栅扫描算法以来，多步的思想就此应运而生，其主要优点是图形的一次扫描转换能绘制多个像素. 文献[2]或文献[3]对有关多步法理论问题进行了一些有意义的探讨，分别对抛物线、圆以及一般函数曲线的多步法绘制问题展开了讨论，提出了多步数目 m_k 和多步区间 MS 等概念，多步法绘制直接体现出离散绘制的思想，符合计算机图形的光栅显示特点.

2003-04-11 收稿，2003-05-26 修回。

本文试图在文献[2]的基础之上，将多步法绘制理论的有关思想进一步应用到的计算机图形学的裁剪算法当中，并就抛物线这一基本曲线的裁剪问题展开进一步讨论.

1　抛物线的多步法裁剪原理

图 1 为多步画抛物线示意图，m_k 为在第 k 条扫描线下，由 $Y_k \pm 0.5$ 产生多步数目. 为使问题简化，仅考虑抛物线 $x = py^2 (p > 0)$，如图 2 为抛物线的裁剪示意图，抛物线 OPA 的 MS（多步区间）部分为 PA，US（单步区间）部分为 OP，两裁剪线 Top、$Left$ 分别与 PA 部分相交产生交点 B、C，弧边 BC 即为裁剪结果. 另外，根据扫描线自上而下绘制图形的特点，从图 2 我们还可看到由 B、C 两点所产生的绘制裁剪图形的关键数据 nb 和 ne. 其中，nb 为 B 点距 PA 段起点 A 的扫描线数；ne 为 C 点距 PA 段起点 A 的扫描线数. 因此，由 nb、ne 和它们之间的多步数目 m_k，就能绘制出弧边 BC.

1.1　定义

定义 1　多步数目集合为对于任一函数曲线内，由多步区间 MS 计算所产生的所有多步数目 m_k 的集合，用 $Mset$ 表示.

定义 2　多步数组为存放 $Mset$ 的数组，对于任一函数曲线 $y = f(x)$，它由两部分组成. 其一为多步区间 MS 部分的 $Mset$；其二为单步区间 US 部分对应其反函数 MS 部分的 $Mset$，称为 US 部分的 $Mset$；抛物线 $x = py^2$ 的多步数组为存放抛物线 MS 和 US 部分的 $Mset$，分别用 $P_m[\,]$ 和 $P_u[\,]$ 表示.

定义 3　弧边形为非线性曲线与直线形成的封闭的图形. 其中非线性部分曲线段为弧边，直线段为线边；抛物线的弧边形为抛物线曲线段与直线形成的封闭的图形. 其中抛物线曲线段部分为弧边，直线段为线边.（如图 2 所示，BC 为弧边，BF 为线边）

定义 4　抛物线的多步三角弧边形为抛物线 MS 或 US 部分的某一弧段上，由该弧段和其在两坐标轴上的投影线段即两直角边构成的图形. 其中，在两直角边当中，对应于纵坐标 y 轴方向的直角边长度值为 my，等于通过该边的扫描线数量；对应于横坐标 x 轴方向的直角边长度值为 mx，其大小为绘制该弧段的若干多步数目 m_k 之和.（如图 1 所示：PBC 为多步三角弧边形，PB 为抛物线 MS 上的某一弧段，PC、BC 为两直角边，其值分别为 mx 和 my）

1.2　抛物线 $x = py^2$（$p > 0$）的多步数组 P_u［$dx1$］和 P_m［$dy1$］的线性化计算

1.2.1　抛物线单步区间（US 部分）和多步区间（MS 部分）的确定

根据在文献[2]的引理 2：对于抛物线 $x = py^2 (p > 0, x \geqslant 0, y \geqslant 0)$

$$MS = \{x, y \mid x \in [\frac{1}{4p}, +\infty), y \in [\frac{1}{2p}, +\infty)\}, \tag{1}$$

$$US = \{x, y \mid x \in [0, \frac{1}{4p}), y \in [0, \frac{1}{2p})\}. \tag{2}$$

根据 MS 和 US 的划分，同时考虑到扫描线自上而下的扫描顺序，我们可将抛物线 $x = py^2 (p > 0)$ 分为 4 个区域（0～3），如图 3 所示.

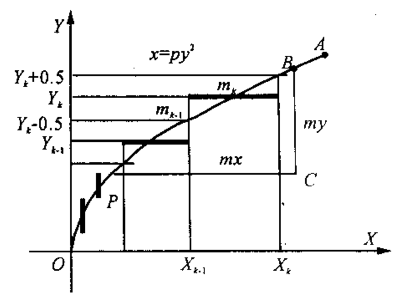


图 1　多步法画抛物线示意图

1.2.2　US 部分多步数组 $P_u[dx1]$ 的线性化计算

可利用文献[4]中所述著名的 Brensenham 算法线性化计算得到 m_k.

设 P 是抛物线 $x = py^2(p > 0, x \geqslant 0, y \geqslant 0)US$ 部分扫描转换所确定的某一 Pixel 点，其坐标为 $P(x, y)$，令 $D(P) = x - py^2$（如图 1、图 4 的 OP 部分）. 假设已选定 m_k 上的点 $P_i(x_i, y_i)$，则下一个候选点为 $U_i(x_i, y_i + 1)$ 或 $R_i(x_i + 1, y_i + 1)$，它们分别在正上方或右上方. 今设 $d_i = D(U_i) + D(R_i)$，则：当 $d_i < 0$ 时选 R_i；当 $d_i \geqslant 0$ 时选 U_i. 参照文献[4]）的圆 d_i 的求法，我们可得到抛物线 d_i 的递推公式为：

$$d_{i+1} = \begin{cases} d_i - 4py_i - 6p + 2, & (d_i < 0) \\ d_i - 4py_i - 6p, & (d_i \geqslant 0) \end{cases} \tag{1}$$

可由 $O(0,0)$ 作为起始点，则有 $D(U_0) = -p, D(R_0) = 1 - p$，因而在不绘制 Pixel 的情况下，可通过(1) 式逐点计算 Pixel 点. 其中，对于任一条垂直线 $x = x_i$，将所有的 U_i 点的数目累加即可得到 m_i，即 m_k，从而得到数组 $P_u[dx1]$. 这里，$dx1$ 为以整个 US 部分为弧边的多步三角弧边形的 mx 值（如图 3).

1.2.3　MS 部分多步数组 $P_m[dy1]$ 的线性化计算

利用文献[2] 中定理 3 可得：

$$m_k = 2pY_k, \tag{2}$$

Y_k 的初值根据 US 部分终端确定，即由 P 点确定：

$$Y_0 = P_u[dx1] + 1. \tag{3}$$

随后，由于扫描线每次递增量为 1，故有：$Y_{k+1} = Y_k + 1$，而 Y_k 的终值即 A 端值，可由预先给定的一个较大的数确定. $dy1$ 大小为以 PA 为弧边的多步三角弧边形的 my 值，故可得到数组 $P_m[dy1]$.

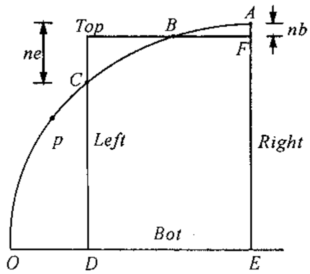


图 2　抛物线的裁剪示意图

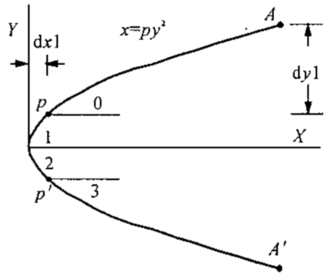


图 3　抛物线 MS、US 区域划分(0 ～ 3) 示意图

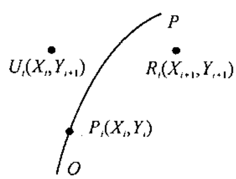


图 4　Brensenham 算法示意图

1.3　抛物线的裁剪

抛物线的裁剪问题主要讨论一般窗口裁剪问题，即是 Top、Bot 水平线和 $Left$、$Right$ 垂直线与抛物线相交问题. 为使问题简化，首先仅考虑 $x = py^2(p > 0)$ 的情况，如图 2 为抛物线的裁剪示意图，显然，根据抛物线的对称性，可分为 4 个弧线段（参考图 3). 图 2 中仅画出 2 个弧线段即 OP 和 PA. 另外，Bot 线与 $y = 0$ 线重合，$Right$ 线与抛物线边界端点 A 相交，属特殊情况. 由对称性特点，我们又可将 4 段弧线统一映射到 2 个弧段 OP、PA 上，这样便可利用数组 $P_u[]$ 或 $P_m[]$ 解决 OP 或 PA 与窗口裁剪线的求交问题，最终统一归结为多步数组 $P_u[n]$ 或 $P_m[n]$ 的下标起点 nb 或终点 ne 的求取问题. 从另一方面来看，抛物线的裁剪问题，可看成为一个弧边形的绘制问题，弧边形（如图 2 的 $BCDEF$）由线边（图 2 裁剪线段）和弧边（图 2 BC 段）组成，线边仅为直线绘制，因此，抛物线弧线段的求取和绘制就成为弧边形绘制的关键. 需要指出的是，将抛物线的裁剪问题统一到 $P_u[]$ 或 $P_m[]$ 上来的另一优点是几何不变性（定义参见文献[5]）的保持，即某些几何信息不随坐标变换而变化的性质，仅仅在最后绘制时再考虑坐标位置.

1.4　线性求交

1.4.1　裁剪线与抛物线交点的求取

在求交之前，先确定出抛物线的多步数组 $P_m[\,]$ 和 $P_u[\,]$ 及其位置数据(如图 3 中的 A、P、O、P'、A' 点数据)，然后由窗口裁剪线数据和抛物线的位置数据，确定出可能相交的抛物线区域，并求出对应于此相交区域的 mx 和 my 的值. 具体做法是，以图 3 中 0 号区域为例，若裁剪线 Top 交于此区域(如图 2 所示)，可根据 Top 信息和 A 点的数据立即确定出相交弧段 AB 的 my 值，即图 2 中的 nb 值，并将其代入下面(5)式右边，则又可求出 mx，即边 BF 的长度值. 这样，在绘制时便可利用 A 点坐标和对应于 AB 弧段的 mx、my 值确定 B 点坐标；同样，若裁剪线 $Left$ 交于此区域，可首先可由 $Left$ 和 A 点数据确定出 AC 弧段的 mx 值，并将其代入下面(5)式左边，则又可求出 my，即图中的 ne. 与此类似，我们便可求出对于任何裁剪线和任何抛物线区域相交所产生的 mx 或 my，以便绘制裁剪图形时使用.

关于交点的求取公式，先考虑 MS 段情况：当已知多步三角弧边形的 mx 或 my 时，可根据 $P_m[\,]$ 数据，在有限次迭加 m_k 的情况下(最多不超过 $dy1$ 次)，可分别对应求出 my 和 mx(参考图 1)

$$my = \min i \quad \left(mx <= \sum_{i=0}^{i=dy1-1} P_m[i]\right), mx = \sum_{i=0}^{i=my-1} P_m[i]. \tag{5}$$

另外，US 段情况与此相同.

1.4.2　裁剪线与抛物线 $x = py^2 (P > 0)$ 相交情况分析

对于 Top、Bot 线与抛物线弧相交，可分别产生 nb 和 ne 数据(如图 2)；对于 $Left$ 线，与抛物线上半段($y \geqq 0$)相交可求出 ne(如图 2 所示)，与下半段($y < 0$)相交可求出 nb；对于 $Right$ 线，nb 和 ne 情况正好与 $Left$ 线相反.

1.4.3　多条裁剪线相交情况考虑

当 2 条及以上的相互垂直的裁剪线相交时，有 2 种情况需要考虑：其一是，交点在抛物线之内，此时，除考虑裁剪线与抛物线的交点以外，还需考虑裁剪线段的取舍. 如图 2 所示，裁剪线 $Left$ 和 Bot 相交，线边 DC 和 DE 为裁剪结果；其二是交点在抛物线之外，此时，只需考虑裁剪线与抛物线的相交情况. 如图 2 Top 和 $Left$ 相交，弧边 BC 为裁剪结果.

2　算法设计

步骤 1　用 1.2 的线性化计算方法分别求取给定抛物线的多步数组 $P_u[dx1]$ 和 $P_m[dy1]$；

步骤 2　就窗口的 4 条裁剪线：Top、$Left$、$Right$ 和 Bot 线，分别考虑与 4 段抛物线弧线相交情况：

1) 求取交点和所在抛物线弧线段编号(0 ～ 3)，并确定对应弧线段的全画(0)、部分画(1)和不画(－1)的信息；

2) 求取抛物线的多步三角弧边形的两直角边 mx 和 my；

3) 求取抛物线弧边所对应的 nb、ne 等数据；

步骤 3　根据 步骤 2 信息分别确定弧边形的弧边数据和线边数据，其中：

1) 根据步骤 2 抛物线弧线段绘制信息和相应的 nb、ne 数据，确定弧边数据；

2) 根据步骤 2 抛物线弧线段的不画信息和有关的交点数据，确定线边数据；

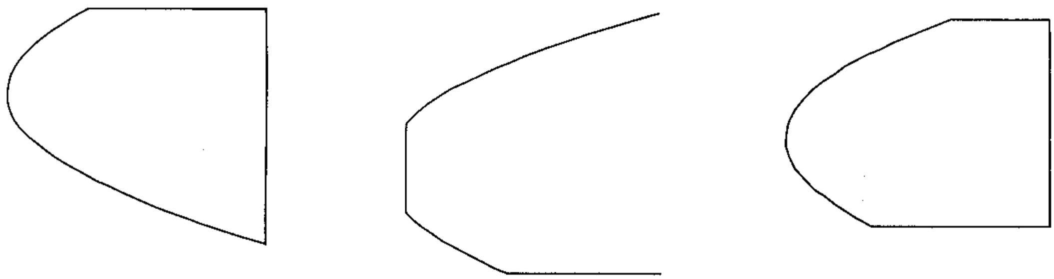
步骤4　根据以上有关数据和窗口、抛物线的相应实参，分别绘制弧边形对应的弧边和线边，其中：

1）根据步骤2中的1）弧线段编号和全画信息(0)及其相应实参绘制弧边；

2）根据步骤2中的1）弧线段编号和部分画信息(1)、步骤2的2)、步骤3的1）及其相应实参绘制弧边；

3）根据步骤3中的2）绘制线边.

3　运行结果举例



(a)　　(b)　　(c)

图5　抛物线的裁剪效果图($p=0.02$，窗口参数可调)

(a)*Top*、*Right*内相交裁剪结果；(b)*Left*、*Bot*外相交裁剪结果；(c)*Top*、*Right*、*Bot*相交裁剪结果

如图5所示为用本文算法程序运行所得的抛物线的裁剪效果图.其中图5(a)为*Top*、*Right*线在抛物线内相交时的裁剪结果，中间两条弧边全画($id=0$)，上下两条弧边部分画($id=1$)；图5(b)为*Left*、*Bot*两条裁剪线在抛物线外相交时的裁剪结果，该图*Right*裁剪线在抛物线右边界以外，故右边不封闭.另外，中间两条弧边不画($id=-1$)，上下两条弧边部分画($id=1$)，结果表明*Left*线在中间两弧边以右；图5(c)为*Right*线与*Top*、*Bot*两线在抛物线内相交，需2次考虑裁剪线取舍.

4　结束语

运行结果表明，该算法是可行的，特别是当抛物线$x=py^2(p>0)$的p较大，而抛物线的边界不是太大时，用若干少部分多步数目m_k的连续相加，即可对抛物线求交，避免了对抛物线非线性方程的平方根运算.另外，将该算法稍加改造，利用相同的$P_m[]$和$P_u[]$，也可适应各类抛物线的裁剪(如$p<0$时和$y=px^2$的情况)，还能用于抛物线裁剪图形的填充，该算法还可以进一步改进，有关问题还需进行深入研究.

参考文献

1　Wu X，Rokne J G. Double-step incremental generation of lines and circles. CVGIP，1987，37：331～334.

2　李建华.抛物线的多步法绘制.计算机应用，2002，增刊：40～42.

3　李建华.圆的多步法绘制理论和算法研究.广西科学院学报，2002，18(4)：173～177.

4　唐荣锡，汪嘉业，彭群生等.计算机图形学教程(修订版).北京：科学出版社，2000.38～40.

5　孙家广.计算机图形学(第三版).北京：清华大学出版社，1998.303.

(责任编辑：黎贞崇)