

圆形窗口的凸多边形填充算法*

杜玉越

(聊城师范学院计算机科学系, 山东 252059)

摘要 给出了一种基于圆形窗口的凸多边形填充算法, 它集裁剪与填充功能于一体, 也可完成单纯地裁剪功能。

关键词 圆形窗口, 凸多边形, 裁剪, 填充算法

1 引言

关于多边形的裁剪或填充, 目前已出现了许多有效算法^[1,2], 以及集裁剪与填充功能于一体的快速算法^[3], 它们在实际中得到了广泛应用。

但在图形处理系统中, 也常常遇到圆形窗口的裁剪或填充问题。文献[4]提出的圆形窗口的凸多边形裁剪算法, 仅具有快速裁剪功能, 仍需用另一种方法对其填充。本文研究并给出了基于圆形窗口的凸多边形裁剪与填充算法, 它集裁剪与填充功能于一体, 也可完成单纯地裁剪功能, 并且与文献[4]保持相同的算法时间、复杂度和收敛速度。

2 基本思想

假设在平面坐标系中, 给定一个凸多边形 P 和一个圆形窗口 C 。 P 的 n 个顶点 $p_i = (x_i, y_i)$ ($i=0, 1, \dots, n-1$) 以顺时针方向列出。不失一般性, 假设 C 的圆心坐标为平面坐标系原点, 半径为 R ($R>0$)。

设 a, b 为平面上任意 2 点, 用 \vec{ab} 表示具有从 a 点到 b 点方向的有向线段。本文将 P 的边 $\vec{p_i p_{i+1}}$ 视为有向线段。

P 的边 $\vec{p_i p_{i+1}}$, 用参数方程表示为:

$$\begin{cases} x = x_i + (x_{i+1} - x_i)t \\ y = y_i + (y_{i+1} - y_i)t \end{cases} \quad (0 \leq t \leq 1) \quad (1)$$

C 的参数方程为:

$$\begin{cases} x = R\cos\theta \\ y = R\sin\theta \end{cases} \quad (0 \leq \theta \leq 2\pi) \quad (2)$$

2.1 填充区域边界的确定

定义 1 设 p_i 为凸多边形 P 的顶点, 称 $\vec{p_{i-1} p_i}$ ($p_{-1} = p_{n-1}$) 为顶点 p_i 的入边, 而 $\vec{p_i p_{i+1}}$ ($p_n = p_0$) 为顶点 p_i 的出边。

定义 2 设 P 的顶点 p_i 在 C 之外, 且入、出两边至少有一边与 C 相交 (与 C 相切的边被视为在 C 之外)。若其入边 $\vec{p_{i-1} p_i}$ 与 C 相交, 则记距 p_i 最近的交点为 $p_{i出}$; 若其出边 $\vec{p_i p_{i+1}}$ 与 C 相交, 则记距 p_i 最近的交点为 $p_{i入}$ 。当 $p_{i入}, p_{i出}$ 同时存在时, 规定 $p_{i出}$ 排在 $p_{i入}$ 之前。

若从 P 的某一顶点 (如 p_0) 出发, 沿顺时针方向实施 C 对 P 裁剪, 即对 P 的顶点序列 $\{p_0, p_1, \dots, p_{n-1}\}$ 中的每一顶点 p_i , 按如下规则修改:

(1) 当 p_i 在 C 内时, 在 P 的顶点序列中保留 p_i 。

(2) 当 p_i 在 C 边线上时, 除 p_i 点外, 若 P 的边 $\vec{p_{i-1} p_i}$ 和 $\vec{p_i p_{i+1}}$ 都与 C 还存在交点或在 C 内 (如图 1(a)), 则在 P 的顶点序列中保留 p_i ; 若 P 的边 $\vec{p_{i-1} p_i}$ 与 C 还存在交点或在 C 内, 而边 $\vec{p_i p_{i+1}}$ 在 C 外 (如图 1(b)), 则在 P 的顶点序列中将 p_i 改写为 $p_{i出}$; 若 P 的边 $\vec{p_i p_{i+1}}$ 与 C 还存在交点或在 C 内, 而边 $\vec{p_{i-1} p_i}$ 在 C 外 (如图 1(c)), 则在 P 的顶点序列中将 p_i 改写为 $p_{i入}$; 若 P 的边 $\vec{p_{i-1} p_i}$ 和 $\vec{p_i p_{i+1}}$ 均在 C

* 本文得到山东省自然科学基金资助

收稿日期: 1997-10-08; 收到修改稿日期: 1997-11-24

外(如图 1(d)),则在 P 的顶点序列中删除 p_i 。

(3)当 p_i 在 C 外,且其入、出边至少有一边与 C 相交时,在 P 的顶点序列中,根据定义 2 用交点替

换顶点 p_i 。

(4)当 p_i 在 C 外,且其入、出边均不与 C 相交时,在 P 的顶点序列中删除 p_i 。

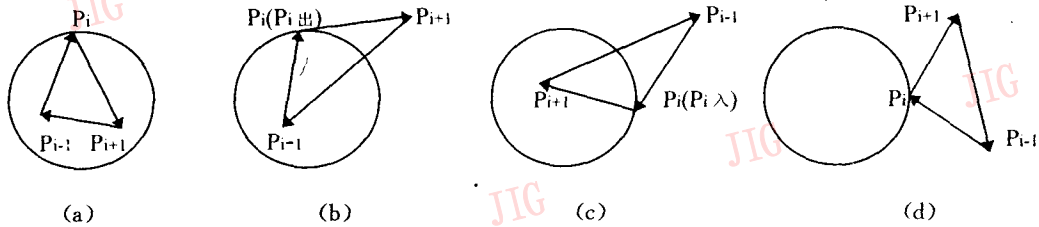


图 1

当重复上述过程,再回到始点时,则产生 P 的一个新顶点表,记为 P' 。如图 2 所示, P 被 C 裁剪后得到的新顶点表 P' 为: $p_{0入} p_1 p_2 p_{3出} p_{3入} p_4出$ 。

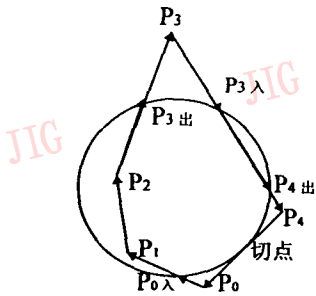


图 2

本文将 P' 中的交点 $p_{i出}$ 或 $p_{i入}$ 也视为顶点。

定理^[4] 设 P 被 C 裁剪后得到的新顶点表为 P' 。若将 P' 中各顶点用直线顺序连接并闭合(即 P' 中首尾 2 顶点亦连接),但其间遇 $p_{i出}$ 与 $p_{j入}$ ($0 \leq i, j \leq n-1$) 相连时,用 C 的边线(即圆弧 $p_{i出} p_{j入}$) 连接,则所得图形是 P 在 C 之内的部分。

因此,当 P' 非空时,根据定理可作出 P 在 C 内的填充区域边界。而当 P' 空时,还需判定 P 是否包含 C 。若 P 包含 C ,则填充区域边界为 C ;否则, P 与 C 没有填充区域。

2.2 区域填充

设 P 的具有最大 y 值的顶点为 $p_{max} = (x_{max}, y_{max})$,具有最小 y 值的顶点为 $p_{min} = (x_{min}, y_{min})$ 。当 P 有 2 个或 2 个以上具有最大(小) y 值的顶点时,在它们中选取 x 值最小的一个作为 $p_{max}(p_{min})$ 。现将 P 的顶点表分成 2 部分:左顶点表 LVT(Left Vertex Table)和右顶点表 RVT(Right Vertex Table),其中 $LVT = \{p_{max}, p_{(max-1+n) \bmod n}, \dots, p_{min}\}$, $RVT =$

$p_{max}, p_{(max+1) \bmod n}, \dots, p_{min}\}$ 。对 P 裁剪与填充时,若将由 LVT 中顶点产生的新顶点链入 $P'_{左}$,而由 RVT 中顶点产生的新顶点链入 $P'_{右}$,则 P 的新顶点表 P' 由 $P'_{左}$ 和 $P'_{右}$ 顺序组合而成,即 $P' = \{P'_{左}, P'_{右}\}$ 。

设水平扫描线间隔为 $d > 0$,扫描自 $top := \text{MIN}(y_{max}, R) - d$ 开始,到 $bottom := \text{MAX}(y_{min}, -R)$ 结束。因此,当扫描线 $Y = top$ 时, Y 不会与 P 的边重合。

设当前水平扫描线为 $Y = y (y \neq bottom)$,若 Y 与 P 的边 $p_i p_{i+1}$ 相交,则交点的 x 坐标值,由式(1)计算得

$$x = x_i + \Delta_i(y - y_i) \tag{3}$$

其中 $\Delta_i = (x_{i+1} - x_i) / (y_{i+1} - y_i)$ 为边 $p_i p_{i+1}$ 斜率的倒数。若水平扫描线 Y 与 C 相交,则存在 2 个交点,交点的 x 坐标值,由下式计算:

$$x = \pm \sqrt{R^2 - y^2} \tag{4}$$

定义 3 设当前扫描线为 $Y = y$ 。若 P 的顶点 $p_i = (x_i, y_i)$ 位于当前扫描间隔内,即 $y + d > y_i \geq y$,则称顶点 p_i 是被水平扫描线 $Y = y$ 激活的顶点,简称活顶点。特别地,当 $Y = top$ 时,称满足 $y_i \geq top$ 的顶点 p_i 为活顶点。

如图 3 所示, P 的顶点 p_0, p_1, p_3 位于当前扫描间隔内,由定义 3,顶点 p_0, p_1, p_3 为当前扫描线 Y 的活顶点。

显然,以扫描线间隔 d 填充时,对于当前扫描线 $Y = y$,可由公式(3)和(4)求出 Y 与 P 的交点及

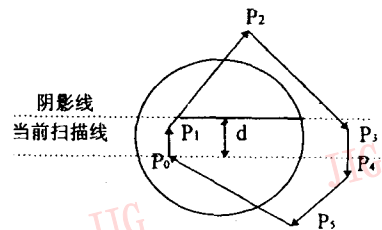


图 3

Y 与 C 的交点,然后将交点按 x 值由小到大排列。若 x 值为 $-\sqrt{R^2-y^2}$ 的交点处于偶数位,则自该交点开始,否则自其下一交点开始,与后面 x 值不大于 $\sqrt{R^2-y^2}$ 的交点,在当前扫描线上连接起来,即为填充区域在当前扫描线上的阴影线。

由于 P 为凸多边形,当水平扫描线 Y 满足: $top \geq Y > bottom$ 时, P 仅有 2 条边与 Y 相交,且这 2 条边的端点分别属于顶点表 LVT 和 RVT。将端点在 LVT 中与当前扫描线 Y 相交的边记为 E_l ,而端点在 RVT 中与当前扫描线 Y 相交的边记为 E_r ,则称 E_l, E_r 为 P 在当前扫描线 Y 的活边。因此, P 与当前扫描线 Y 的交点,就是 P 的活边 E_l 和 E_r 与 Y 的交点。当 $Y=bottom$ 时, Y 与 P 的活边 E_l 或 E_r 有可能重合,但此时扫描线 Y 已到达 P 与 C 填充区域边界,不用作出填充区域在 Y 上的阴影线,即不需求 E_l 或 E_r 与 Y 的交点。

2.3 新顶点表 P' 的形成

新顶点表 P' 是在区域填充过程中形成的。

当以扫描间隔 d 逐次扫描时,对当前扫描线 Y,在顶点表 LVT 和 RVT 中,分别依次检查是否存在活顶点,直至遇非活顶点结束:对 LVT(RVT)被激活的每一顶点 p_i , (1)若 $x_i^2 + y_i^2 < R^2$,即 p_i 在 C 内,则将 p_i 链入 $P'_L(P'_R)$ 。(2)若 p_i 的入(出)边与 C 相交,则求出交点。当 p_i 在 C 内时,该交点以入(出)边交点链入 $P'_L(P'_R)$;当 p_i 在 C 外时,若存在一个交点,则该交点以出(入)边交点链入 $P'_L(P'_R)$,若存在 2 个交点,则 y 坐标值较大的交点以出(入)边交点链入 $P'_L(P'_R)$,另一交点以入(出)边交点链入 $P'_L(P'_R)$ 。(3)在 LVT(RVT)中,删除 p_i 。

在填充结束后,对 P'_L 中的顶点逆置,即将 P'_L 中的顶点次序前后颠倒,逆置后的顶点表仍记为 P'_L ,再将 P'_L 和 P'_R 中的顶点,顺序组合起来,便构成了新顶点表 P',即 $P' = P'_L$ 和 P'_R 。

2.4 P 与 C 的交点计算

若 $p_i \in LVT(RVT)$,当 p_i 被当前扫描线 $Y=y$ 激活时,应判断 p_i 的入(出)边是否与 C 相交,相交时,还需求出交点坐标。

根据式(1)、(2),若 P 的边 $\overrightarrow{p_i p_{i+1}}$ 与 C 相交,则交点处的参数 t 应满足:

$$A_i t^2 + 2B_i t + C_i = 0 \quad (5)$$

其中 $0 \leq t \leq 1, A_i = (x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2, B_i =$

$$x_i(x_{i+1} - x_i) + y_i(y_{i+1} - y_i), C_i = x_i^2 + y_i^2 - R^2.$$

记 $\delta_i = B_i^2 - A_i C_i$ 。

若 $\delta_i \leq 0$,由定义 2,边 $\overrightarrow{p_i p_{i+1}}$ 与 C 不相交。

若 $\delta_i > 0$,则得参数 t 的 2 个实根:

$$\begin{cases} t_1 = (-B_i - \sqrt{\delta_i})/A_i \\ t_2 = (-B_i + \sqrt{\delta_i})/A_i \end{cases} \quad (6)$$

当 $t_1, t_2 \in [0, 1]$ 时, $\overrightarrow{p_i p_{i+1}}$ 与 C 不相交,否则 $\overrightarrow{p_i p_{i+1}}$ 与 C 存在交点,且交点坐标可由式(2)计算。当 t_1 (或 t_2) $\in [0, 1]$ 时, $\overrightarrow{p_i p_{i+1}}$ 与 C 存在一个交点。当 $t_1, t_2 \in [0, 1]$ 时, $\overrightarrow{p_i p_{i+1}}$ 与 C 有 2 个交点。

3 算法步骤

(1) 计算 P 的具有最大 y 值的顶点 $p_{max} = (x_{max}, y_{max})$, 最小 y 值的顶点 $p_{min} = (x_{min}, y_{min})$ 。

(2) 由 P 的顶点序列表,产生左顶点表 LVT 和右顶点表 RVT。

(3) 令 $top := \text{MIN}(y_{max}, R) - d, bottom := \text{MAX}(y_{min}, -R), FLAG := 0$ 。

(4) 水平扫描线自 $Y := top$ 开始扫描,扫描线每下移一次,即 $Y := Y - d$,均完成下列操作:

(4.1) 若 $Y < bottom$,则置

$$d := d + Y - bottom, Y := bottom.$$

(4.2) 对于当前扫描线 Y,在 LVT 和 RVT 中,分别依次找出活顶点,直至遇到非活顶点或 LVT (RVT)的最后一个顶点,并对每个活顶点 p_i ,完成下述操作:

(4.2.1) 若 $p_i \in LVT$,令 $E_l := \overrightarrow{p_{i-1} p_i}$,①当 p_i 在 C 内时,则将 p_i 链入 P'_L 。若 E_l 与 C 相交,则由式(6)及(1)计算交点,交点以入边交点链入 P'_L 。②当 p_i 在 C 外且 E_l 与 C 相交时,若仅存在一个交点,则交点以出边交点链入 P'_L ;否则 t 值大的交点为出边交点, t 值小的为入边交点,然后再将其顺序链入 P'_L 。③在 LVT 中删除顶点 p_i 。

(4.2.2) 若 $p_i \in RVT$,令 $E_r := \overrightarrow{p_i p_{i+1}}$,①当 p_i 在 C 外时,则将 p_i 链入 P'_R ,又若 E_r 与 C 相交,则由式(6)及(1)计算交点,交点以出边交点链入 P'_L 。②当 p_i 在 C 外且 E_r 与 C 相交时,若存在一个交点,则交点以入边交点链入 P'_R ;否则, t 值小的交点为入边交点, t 值大的为出边交点,然后再将其顺序链入 P'_R 。③在 RVT 中删除顶点 p_i 。

(4.3) 对活边 E_l 和 E_r

(4.3.1) 当 $Y = bottom$ 时, 填充区域内的阴影线已全部作出且 P'_L, P'_R 已经形成, 转(5)步。

(4.3.2) 根据式(3)计算扫描线 Y 与 E_L, E_R 交点的 x 坐标, 由式(4)计算 Y 与 C 的交点, 并将交点按 x 值由小到大排列。若 x 值为 $-\sqrt{R^2 - Y^2}$ 的交点处于偶数位, 则自该交点开始, 否则自其下一交点开始, 与后面 x 值不大于 $\sqrt{R^2 - Y^2}$ 的交点(若该交点存在, 令 $FLAG = 1$), 在当前扫描线 Y 上连接起来。

(5) 将 P'_L 中的顶点逆置并与 P'_R 中的顶点顺序组合在一起, 构成新顶点表 P' 。

(6) 根据 P' 形成填充区域边界:

(6.1) 当 P' 非空时, 依据定理作出填充区域边界。

(6.2) 当 P' 空时, 若 $FLAG = 1$, 则作出 C 的边线即为填充区域边界; 若 $FLAG = 0$, 则根据文献[4]中定理 2 确定填充区域边界。

(7) 算法结束。

4 讨 论

上述算法, 已用 C 语言编程实现, 经算例验证, 算法可靠性高, 速度快。

一般情况下, 圆形窗口 C 与凸多边形 P 的新

顶点表 P' 是非空的, 这种情况下, 算法(6)步中的(6.2)步可省略。由于算法第 4 步, 在当前扫描间隔 d 内, G 依次找出 LVT 和 RVT 中的活顶点直至遇非活顶点或 LVT(RVT)的最后一个顶点结束, 且每激活一顶点均将其从 LVT 或 RVT 中删除, 因此, 算法在最坏情况下的时间复杂度仍为 $O(n)$ 。本算法思想亦适用于圆形窗口对简单多边形的填充算法。

参 考 文 献

- 1 Weiler KJ, Atherton PR. Hidden Surface Removal Using Polygon Area Sorting. *Computer Graphics*, 1997, 11(2).
- 2 Foley JD, Dam AV. *Fundamentals of Interactive Computer Graphics*. Addison-Wesley, Reading, Mass, 1982.
- 3 熊一华等. 一个隐含有修剪功能的多边形填充算法: C-Y-X 算法, *计算机学报*, 1988, 11(7): 442~447.
- 4 杜玉越. 圆形窗口的凸多边形裁剪. *计算机应用与软件*, 1995, 12(4): 48~53.



杜玉越 副教授, 1982 年 7 月山东工业大学数理系毕业, 1991 年 3 月南京航空航天大学获理学硕士学位。现主要从事计算机图形学、CAD、算法设计与分析等方面的教学研究工作。

A Filling Algorithm of Circular Window for Convex Polygons

Du Yuyue

(Department of Computer Science, Liaocheng Teachers College, Shandong 252059)

Abstract A filling algorithm of circular window for convex polygons is presented. It serves the needs of filling with clipping as well as clipping-only functions.

Keywords Circular window, Convex polygon, Clipping, Filling algorithm

IBM 显示器全新登场

继通用型 G 系列和专业型 P 系列显示器之后, IBM 日前又发布了新款的 P 系列显示器以及大屏幕平板式液晶显示屏。其中, 14.5 英寸和 16.1 英寸大屏幕平板式液晶显示屏还是第一次在内地面市, 它可以用做金融、零售、医疗、服务以及专业设计机构等业务系统的显示媒介。

而 IBM 此次发布的 P 系列专业型大屏幕显示器, 共包括 P72(17 英寸)、P92(19 英寸)和 P202(21 英寸)3 种类型型号。相对于普通显示器, 此次发布的新款 P 系列专业型显示器不仅具有分辨率高、色彩丰富以及无闪烁等性能, 更具有节约能耗、降低辐射和节省空间的三大优点。