

基于交点参数的任意多边形窗口对圆裁剪^①



李晓武, 陈平

(北京科技大学 机械工程学院, 北京 100083)

通信作者: 李晓武, E-mail: xiaowu_li@ustb.edu.cn

摘要: 现有的任意多边形窗口的圆裁剪算法存在算法繁琐等问题, 且没有考虑多边形是带内环的情况, 本文提出了一种基于交点参数分析的多边形窗口的圆裁剪算法, 只需对多边形边与圆的交点在边所在直线的参数值进行比较, 即可判断出交点的进出点特性, 交点排序后, 通过进点 \Rightarrow 出点组合, 即可获得裁剪窗口内的圆弧, 完成裁剪. 编程实践的实例结果也证明本算法是切实可行的, 本文的方法既适用于仅有外环的一般多边形裁剪窗口, 也适用于带内环的任意多边形裁剪窗口的圆裁剪, 因此, 算法更具有通用性.

关键词: 圆裁剪; 任意多边形窗口; 交点进出点特性; 交点参数

引用格式: 李晓武, 陈平. 基于交点参数的任意多边形窗口对圆裁剪. 计算机系统应用, 2022, 31(8): 239-244. <http://www.c-s-a.org.cn/1003-3254/8704.html>

Circle Clipping by Arbitrary Polygon Window Based on Intersection Point Parametric Method

LI Xiao-Wu, CHEN Ping

(School of Mechanical Engineering, University of Science and Technology Beijing, Beijing 100083, China)

Abstract: The existing algorithms of circle clipping against an arbitrary polygon window suffers from complex steps and do not consider the case where the polygon contains an inner ring. Thus, this study presents a new algorithm based on the parameter analysis of intersection points for circle clipping against an arbitrary polygon window. In this algorithm, only by comparing the parameter values of the intersection points on the directional line of the edge, one can classify the intersection points into entry points and exit points. After the intersection points are sorted out, the arcs within the clipping window can be obtained with the combinations of “entry point \Rightarrow exit point.” The proposed algorithm is proved feasible by programming results and is universal to circle clipping against an arbitrary polygon window, even if the polygon contains an inner ring.

Key words: circle clipping; arbitrary polygon window; intersection point of the entry and the exit; intersection point

裁剪是计算机图形学的基本问题之一, 也是 CAD 软件系统的一个重要图形编辑功能, 裁剪一般是指把一个封闭区域作为裁剪窗口, 去裁剪其他图形, 把其他图形中不在窗口内的部分裁剪掉, 只保留窗口内的部分. 裁剪功能除了在 CAD 软件中大量使用外, 在计算机动画、机器人运动学仿真、建筑设计、地理信息系统、平面绘画、影视特技图像处理以及模式识别等领域都有广泛应用.

目前, 裁剪算法的研究主要集中在多边形裁剪方面, 裁剪算法比较丰富, 有适合矩形裁剪窗口的逐边裁剪法、Liang-Barsky 算法、中点分割算法, 适合凸多边形裁剪窗口的线裁剪 Cyrus-Beck 算法、Sutherland-Hodgman 算法、外接矩形判别法和分区判断直接裁剪法以及适合任意多边形裁剪窗口的 Weiler-Atherton 双边裁剪算法等^[1-13].

圆裁剪也是一个重要的图形裁剪功能, 具有广泛

^① 基金项目: 北京科技大学青年教学骨干人才培养计划 (2018JXGGRC)

收稿时间: 2021-10-28; 修改时间: 2021-12-02, 2021-12-10, 2022-01-20, 2022-02-15; 采用时间: 2022-02-18; csa 在线出版时间: 2022-06-16

的应用,圆裁剪有两种类型,一是圆做裁剪窗口对各种图形进行裁剪,例如圆窗口对直线的裁剪^[14-18],二是多边形或者其他图形作为裁剪窗口对圆进行裁剪,裁剪后只保留窗口内的圆弧部分.在第二种圆裁剪类型中,矩形窗口的圆裁剪在视窗显示时应用较多,裁剪算法有逐边裁剪法、区域编码法及投影法等^[19-22].除了矩形窗口的圆裁剪外,在实际应用中,也存在任意多边形做裁剪窗口的圆裁剪情况,因此,研究任意多边形窗口的圆裁剪也具有实用的价值.关于多边形做裁剪窗口的圆裁剪,文献[23]提出了一种方法:首先,计算多边形每条边与圆的交点,再对交点在圆周方向排序,然后再进一步判断相邻交点间的圆弧是否在窗口内,该方法为多边形窗口的圆裁剪提供了一条有效的途径,但是,文献中的算法需要开平方计算交点和反三角计算检测圆弧的可见性,效率低且耗资源,文献[24]将圆裁剪分成相交性检测求交点、点在多边形内外判断、交点排序以及可见性检测等5个步骤,在圆与多边形相交性检测时利用投影和射线法相结合,降低了运算的耗费,对交点排序和圆弧可见性检测方法也做了相应改进,算法的效率相比文献[23]有所提高.文献[25]提出了和文献[24]类似的方法,利用X-扫描线法分析圆与多边形的位置关系,再利用多边形顶点与圆的相对位置对交点排序.文献中提出的方法,除了细节不同外,算法的整体思路基本上是相同的:均是先求出多边形与圆的交点,再对交点在圆矢量方向上排序,然后,判断相邻两个交点之间的圆弧段是否在多边形裁剪窗口内(即可见性检测),从而实现圆的裁剪.在上述思路中,除了计算交点外,交点排序和相邻交点之间圆弧的可见性检测也是非常重要的环节,文献采用的交点排序方法有三三排序法^[24]、比较顶点和圆心距离的排序法^[25]等,但是,这些排序方法只适用于多边形裁剪窗口只有外环的情况,当多边形裁剪窗口除了外环外还含有一个或多个内环多边形时,文献中的交点排序方法就会出错,则后续的圆弧可见性检测方法也会失效,因此,文献中的方法不适合有内环的任意多边形做裁剪窗口的圆裁剪情况.

笔者在进行相关课题研究时,对任意多边形裁剪窗口的圆裁剪问题也进行了探讨.通过分析发现,当多边形裁剪窗口和圆相交时,利用交点相对裁剪窗口的进出点特性,即可判断圆弧段的可见性,不需要再对圆弧的可见性单独进行检测,而通过比较圆与多边形边所在直线相交的参数值大小就能够直接确定交点的进出点特性信息,也不再需要其他额外的计算判断.

为此,本文提出了一种基于交点参数值分析的任意多边形裁剪窗口的圆裁剪算法,与文献中的方法相比,本算法只需对交点参数值进行比较,即可判断出裁剪窗口内的圆弧部分,判断步骤简洁明了,效率要明显高于文献中的方法,适用于包括含内环多边形在内的任意多边形做裁剪窗口对圆进行裁剪的情况.

1 基于进出点特性的多边形窗口对圆裁剪

在任意多边形裁剪窗口对多边形的裁剪算法中,基于交点进出点特性的 Weiler- Atherton 双边裁剪算法是一种非常有效的解决方案,该方法通过将两个多边形设置为有向多边形,当两个多边形有交点时,沿着多边形的方向,通过判断交点是进入裁剪窗口的进点还是离开裁剪窗口的出点,并保留进点和出点之间的多边形,实现裁剪.该方法利用交点的进出点特性获得裁剪窗口内的部分,对多边形裁剪窗口的圆裁剪具有借鉴意义.

图1所示为一个多边形裁剪窗口对圆裁剪,其中,多边形为 $P_0P_1P_2P_3P_4P_5P_6P_7P_8$,顺时针走向, O 是被裁剪圆,逆时针走向.圆 O 与多边形相交的交点为: I_0 、 I_1 、 I_2 、 I_3 、 I_4 和 I_5 .根据交点的进出点分析方法,沿着圆的走向,进入多边形裁剪窗口内的交点称为进点,离开多边形裁剪窗口内的交点称为出点,由于圆本身是封闭的,裁剪窗口的内部区域也是连通的,所以,进点和出点成对出现,数量相同.沿着圆的走向,从进点到出点之间的圆弧位于多边形裁剪窗口内部,在裁剪时应予以保留,从出点到进点之间的圆弧在多边形裁剪窗口的外部,应裁剪掉.因此,在求出多边形裁剪窗口和圆的交点后,首先,沿着圆的走向对交点进行排序,然后,对排序后的交点,进行进点 \Rightarrow 出点的组合,这样就可以获得多边形裁剪窗口内部的圆弧.例如,图1中,将多边形裁剪窗口和圆的交点沿着圆的逆时针走向排序,顺序为: I_0 (进点) $\Rightarrow I_5$ (出点) $\Rightarrow I_4$ (进点) $\Rightarrow I_3$ (出点) $\Rightarrow I_2$ (进点) $\Rightarrow I_1$ (出点),则进点 \Rightarrow 出点的组合为: $I_0 \Rightarrow I_5$ 、 $I_4 \Rightarrow I_3$ 和 $I_2 \Rightarrow I_1$,对应这些进点到出点之间的圆弧为裁剪后需保留的部分.

图1中的多边形裁剪窗口是只有一个外环的一般多边形的情况.下面对形状更为复杂的含内环的任意多边形裁剪窗口的圆裁剪情况进行分析.

图2为一个含内环的多边形对圆进行裁剪,多边形的外环顶点依次为 $P_0P_1P_2P_3P_4P_5P_6$,外环按顺时针走向,内环顶点依次为 $Q_0Q_1Q_2Q_3Q_4$,内环按逆时针走向, O 是被裁剪圆,逆时针走向.圆 O 分别与多边形裁剪窗口的内外环相交,圆与外环的交点以及交点的进出

点特性为: I_0 (进点)、 I_3 (出点)、 I_2 (进点) 和 I_1 (出点), 圆与内环的交点以及交点的进出点特性为: I_4 (出点) 和 I_5 (进点). 沿着圆的逆时针走向, 对所有交点进行排序, 顺序为: I_0 (进点) $\Rightarrow I_4$ (出点) $\Rightarrow I_5$ (进点) $\Rightarrow I_3$ (出点) $\Rightarrow I_2$ (进点) $\Rightarrow I_1$ (出点). 则排序后的交点进行进点 \Rightarrow 出点组合为: $I_0 \Rightarrow I_4$ 、 $I_5 \Rightarrow I_3$ 和 $I_2 \Rightarrow I_1$, 对应这些进点到出点之间的圆弧也为裁剪后需保留的部分.

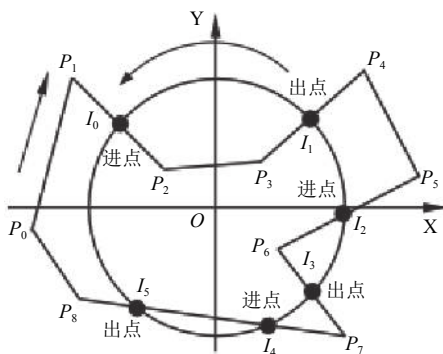


图1 多边形窗口的圆裁剪进出交点分析

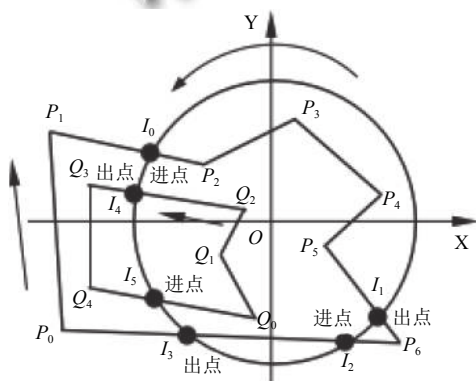


图2 含内环的任意多边形裁剪窗口的圆裁剪

因此, 利用交点的进出点特性也可以实现对含内环多边形在内的任意多边形裁剪窗口的圆裁剪.

本裁剪算法的步骤归纳如下:

- 第1步. 依次判断多边形外环和内环的每条边与圆是否相交, 如相交, 则计算交点;
- 第2步. 判断交点是进点还是出点, 并在交点信息中记录其进出点特性;
- 第3步. 沿圆的走向将交点进行排序;
- 第4步. 根据交点的进出点特性, 将排序后的交点进行进点 \Rightarrow 出点组合, 保留进点 \Rightarrow 出点之间对应的圆弧.

2 交点进出点特性和直线参数的关系

在上述的裁剪算法中, 判断交点的进出点特性是

整个算法的关键. 对于多边形裁剪窗口对多边形的裁剪, 可以通过被裁剪边的两个顶点和裁剪边的相对位置关系来判断交点是进点还是出点, 但是对于圆裁剪, 由于圆本身没有顶点, 所以, 不能再借鉴多边形的进出点判断法来分析圆与多边形交点的进出点特性, 因此, 还需要寻找新的方法来判断交点的进出点信息.

设线段 P_0P_1 是多边形裁剪窗口的一条边, 两个端点分别为 $P_0(x_0, y_0)$ 和 $P_1(x_1, y_1)$, 边的方向是 P_0 指向 P_1 , 则该边所在直线的参数方程表示为:

$$\begin{cases} x = (1-t)x_0 + tx_1 \\ y = (1-t)y_0 + ty_1 \end{cases} \quad (1)$$

如图3所示, 参数将边所在直线划分为3个区域: $(-\infty, 0)$ 、 $[0, 1]$ 和 $(1, +\infty)$, 其中, 当参数值 $t \in [0, 1]$ 时, 对应直线上的点在 P_0 和 P_1 之间的线段上, 即在多边形的边上, 而且参数值从 P_0 到 P_1 逐渐增大; 当 $t \notin [0, 1]$ 时, 对应直线上点不在 P_0 和 P_1 之间的线段上, 而是在该线段两侧的延长线上; 当 $t < 0$ 时, 点在 P_1 指向 P_0 的线段延长线上, t 值越来越小, 当 $t > 1$ 时, 点在 P_0 指向 P_1 的线段延长线上, t 值越来越大, 也即直线上的点在沿着 P_0 指向 P_1 的方向上, 对应的参数值 t 越来越大, 反之, 参数值 t 越来越小.

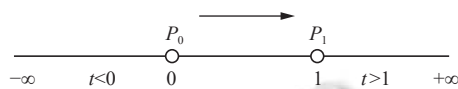


图3 边所在直线的参数表示

设多边形为有向多边形, 其中, 外环为顺时针走向, 内环为逆时针走向, 则多边形内部所在区域具有这样的特点: 不管是外环边还是内环边, 沿着边的走向前进, 即沿着 P_0 点指向 P_1 点的方向前进, 这时, 边的右侧均为多边形的内侧区域. 即使多边形的各边具有不同的倾斜方向, 该特点仍然存在, 如图4所示.

图5所示是多边形的一个边所在直线与圆相交的情况, 圆的走向是逆时针, 边直线的方向是 P_0 点指向 P_1 点, 则沿着直线走向的右侧是多边形的内侧区域. 若圆与直线 P_0P_1 相交, 交点记为 I_0 和 I_1 , 则从图中可以看到, 圆在交点 I_0 处进入多边形内侧区域, 即 I_0 为进点, 在交点 I_1 处离开多边形内侧区域, 即 I_1 为出点, 设交点 I_0 和 I_1 在直线的参数方程式 (1) 中对应的参数值分别为 t_0 和 t_1 , 这时, 参数值 t_0 和 t_1 的大小关系是:

$$t_0 < t_1 \quad (2)$$

式 (2) 也是图4中有向边所在直线第一种倾斜情

况下,和圆相交时交点的进出点特性和交点在直线上参数值大小的对应关系。

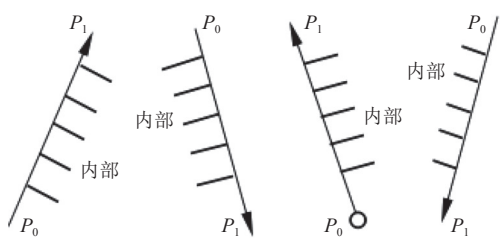


图4 有向边的右侧为多边形内侧区域

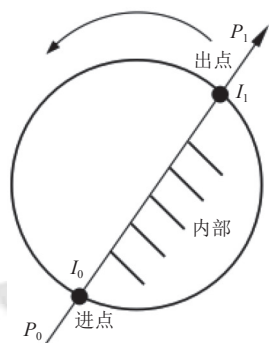


图5 有向边直线和圆相交

图6是图4中有向边所在直线的另外3种倾斜方向下和逆时针走向圆相交的情况,直线的方向是P0点指向P1点,I0和I1是两个交点,其中,I0为进点,I1为出点,从图6也可以得出和图5相同的结论:沿着边所在直线的走向,如果右侧是多边形裁剪窗口的内侧区域,圆沿逆时针走向,圆和直线相交的两个交点分别是进点和出点,则进点在直线上的参数值小于出点在直线上的参数值,即进点和出点在直线上的参数值具有和式(2)相同的大小关系。

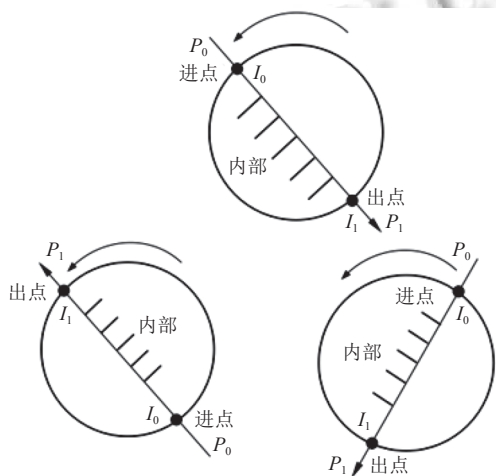


图6 各种倾斜位置有向边直线和圆相交

因此,圆和多边形裁剪窗口相交时交点的进出点特性,通过判断交点在对应边所在直线的参数值大小即可确定下来,这样,就利用第1节提出的基于交点进出点特性的多边形窗口的圆裁剪算法,即可实现对圆的裁剪。

3 圆与边所在直线的交点计算及处理

上述的圆与多边形边相交是指圆与多边形边所在直线相交,这时,交点可能在多边形的边上,也可能在边的延长线上.只有落在多边形的实际边上而不是在其延长线上的交点才用于裁剪,这个交点称为实交点,在边的延长线的交点称为虚交点.当边与圆没有实交点时,圆和该边不存在裁剪,不必计算.只有存在实交点时,才计算交点的参数值,并比较大小,判断该实交点是进点还是出点。

判断直线与圆是否存在交点的方法如下:

设圆的半径是R,圆心为Pc(xc,yc),逆时针方向,圆方程为:

$$(x-x_c)^2+(y-y_c)^2=R^2 \tag{3}$$

将式(1)中的直线参数化表示代入式(3)中,建立参数t的一元二次方程.该方程的解为:

$$t=\frac{B\pm\sqrt{B^2-4AC}}{2A} \tag{4}$$

其中,

$$A=(x_1-x_0)^2+(y_1-y_0)^2$$

$$B=2[(x_1-x_0)(x_c-x_0)+(y_1-y_0)(y_c-y_0)]$$

$$C=(x_c-x_0)^2+(y_c-y_0)^2-R^2$$

令 $\Delta=B^2-4AC$,则 $t=\frac{B\pm\sqrt{\Delta}}{2A}$.

当 $\Delta < 0$ 时,方程无解,即直线和圆没有交点。

当 $\Delta = 0$ 时,方程只有一个解,即直线和圆相切,此时也不必考虑裁剪。

当 $\Delta > 0$ 时,直线与圆有两个交点,这时需进一步判断是否存在实交点,当有实交点时,计算方程的两个解 t_1 和 t_2 :

$$t_1=\frac{B-\sqrt{\Delta}}{2A}, t_2=\frac{B+\sqrt{\Delta}}{2A} \tag{5}$$

由于 $t_1 < t_2$,所以, t_1 对应的交点为圆在直线上的进点, t_2 对应的交点为圆在直线上的出点。

当 $0 \leq t_1, t_2 \leq 1$ 时,则 t_1 和 t_2 对应的两个交点均为实交点,将 t_1 和 t_2 代入式(1),即得交点值,将两个交点插入

交点序列中,并标识每个交点的进出点特性。

当 t_1 和 t_2 中仅有一个值在 $[0,1]$ 区间时,则在 $[0,1]$ 区间的 t 值对应的交点为实交点,将该交点插入交点序列中,并标识其进出点特性。

4 圆周方向交点排序

所有的实交点沿着圆周方向排序是实现裁剪的重要一步。文献中的交点排序方法是比较每个点在圆的逆时针方向对应的圆弧角大小,需要进行反三角运算,比较耗性能。由于所有的交点都在圆上,所以,本文采用比较交点与 $(R,0)$ 或者 $(-R,0)$ 距离的方法进行排序,如图7所示。

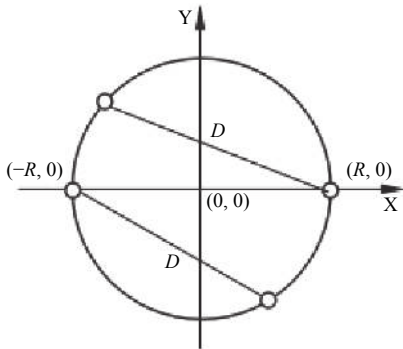


图7 圆周方向交点排序

当交点的坐标值 $y \geq 0$ (即交点在圆上 0° 到 180° 之间)时,按交点到 $(R,0)$ 的距离大小排序:

$$D^2 = (x - R)^2 + y^2 \quad (6)$$

当交点坐标值 $y < 0$ (即交点在圆上 180° 到 360° 之间)时,按交点到圆上 $(-R,0)$ 的距离大小排序:

$$D^2 = (x + R)^2 + y^2 \quad (7)$$

然后,将两组交点序列合在一起,获得所有交点的排序。

5 算法实例及与文献算法的分析比较

为了验证本文提出的基于交点参数的多边形裁剪窗口的圆裁剪方法,笔者在 Visual C++ 平台下进行了图形编程,在生成直线、圆、圆弧以及各种类型多边形的基础上,实现了本文第1节提出的基于交点进出点特性的多边形窗口的圆裁剪算法,在算法中,利用第3节的方法计算多边形边与圆的交点,交点的进出点特性则根据第2节提出的交点所在边直线的参数判断,并按第4节的方法在圆周方向对交点进行排序。

图8和图9是利用本算法实现的裁剪效果实例,其中,图8是仅有外环多边形裁剪窗口对圆的裁剪,其中,图8(a)是裁剪前的多边形裁剪窗口和被裁剪的圆,图8(b)是裁剪后效果。图9是带内环的多边形裁剪窗口对圆的裁剪,其中,图9(a)是裁剪前的多边形裁剪窗口和被裁剪的圆,图9(b)是裁剪后效果。图8和图9中的多边形均通过随机交互方式构造生成,无任何特殊设定,裁剪结果证实本文提出的算法是切实可行的。

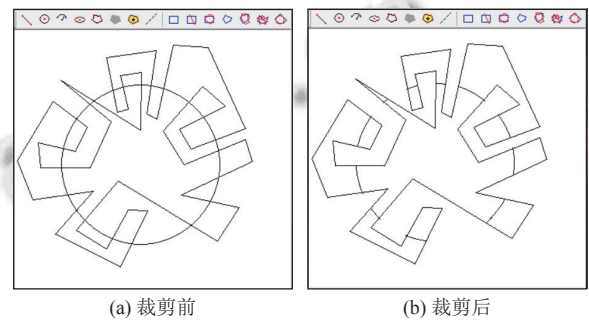


图8 仅有外环的多边形裁剪窗口的圆裁剪

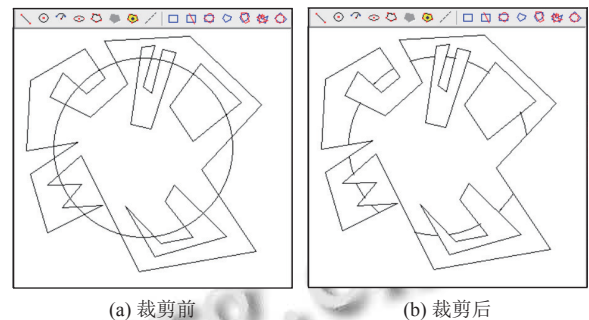


图9 带内环的多边形裁剪窗口的圆裁剪

本实验中采用的直线、多边形、圆以及圆弧这些图形的生成方法均为笔者自己开发的算法实现,没有借助 Visual C++ 平台或者其他如 OpenGL 图形库现有的绘图函数生成,为了说明本文算法的效率和性能,下面列出本文方法和文献方法在步骤上的区别比较。

表1是本文算法与文献算法的比较。其中,在判断和计算裁剪边界直线和圆的交点时,本文算法的时间复杂度是 $O(n)$,其中 n 为多边形的边的数量,空间复杂度为 $O(k)$,其中 k 为实交点的数量,对交点排序时的复杂度为 $O(k)$,空间复杂度为 $O(1)$,也即本文和文献在共有的算法步骤中的时间复杂度和空间复杂度是相同的。但是,除了共有的步骤外,文献算法还需要其他步骤才能实现,而本文的算法,仅根据交点所在直线的参数值大小,判断出交点的进出点特性,即可进行裁剪,不需要太多步骤,因此效率更高。另外,文献也没

有考虑多边形裁剪窗口带内环的情况,本文算法不仅适用于仅有外环的一般多边形裁剪窗口,也适用于带内环的任意多边形裁剪窗口的圆裁剪,因此,算法更具有通用性。

表1 本文算法和文献算法^[23-25]的比较

比较内容	文献算法	本文算法
相交性检测	通用算法或改进的专门算法	和文献方法类似
求交点	根据参数 t 计算实交点	和文献方法类似
交点排序法	不适用于带内环的多边形	可适用于任意多边形类型
判断点在多边形内外	需要	不需要
优劣弧判断	需要	不需要
适用的多边形裁剪窗口类型	仅有外环的一般多边形	带外环和内环的任意多边形

6 结论

本文提出了一种基于交点参数信息分析的任意多边形裁剪窗口的圆裁剪算法,在计算多边形边与圆交点的同时,通过交点在边所在直线的参数信息,即可识别出交点的进出点特性,从而实现对圆的裁剪.与文献中的方法相比较,本算法步骤较少,编程实例结果也验证了本文算法的可行性.本文的算法不仅对仅有外环的一般多边形裁剪窗口的圆裁剪适用,也适用于带内环的任意多边形裁剪窗口的圆裁剪.

参考文献

- 李晓武. 计算机图形学—原理、算法及实践. 北京: 清华大学出版社, 2018: 79-115.
- 何援军. 计算机图形学. 第3版, 北京: 机械工业出版社, 2016: 113-126.
- 刘勇奎, 颜叶, 石教英. 一个有效的多边形窗口的线裁剪算法. 计算机学报, 1999, 22(11): 1209-1214. [doi: 10.3321/j.issn:0254-4164.1999.11.016]
- 刘勇奎. 计算机图形学的基础算法. 北京: 科学出版社, 2001: 62-70.
- 赵平, 冯春, 李柏林. 一般多边形窗口的有效线裁剪算法. 西南交通大学学报, 2004, 39(1): 64-68. [doi: 10.3969/j.issn.0258-2724.2004.01.015]
- 朱亚臣, 谭建荣, 陆国栋, 等. 基于连续分区与串联编码的线段裁剪新算法. 中国图象图形学报, 2007, 12(4): 732-739. [doi: 10.3969/j.issn.1006-8961.2007.04.029]
- 李竹林, 雷岗. 一种改进的 Sutherland-Cohen 裁剪算法. 计算机工程与应用, 2012, 48(34): 175-178. [doi: 10.3778/j.issn.1002-8331.1105-0213]

- 刘勇奎. 图形裁剪算法研究. 计算机工程与应用, 2005, 41(21): 18-23. [doi: 10.3321/j.issn:1002-8331.2005.21.006]
- 刘勇奎, 高云, 黄有群. 一个有效的多边形裁剪算法. 软件学报, 2003, 14(4): 845-856.
- 银红霞, 杜四春, 蔡立军, 等. 计算机图形学. 北京: 中国水利水电出版社, 2005: 40-80.
- Foster EL, Hormann K, Popa RT. Clipping simple polygons with degenerate intersections. Computers & Graphics: X, 2019, 2: 100007.
- 杨朝斌, 刘丽峰. 启发式教学理念与方法探究——以“计算机图形学”为例. 教育教学论坛, 2021, (3): 145-148.
- 何援军. 国产 CAD 软件重启之路. 计算机集成制造系统, 2021, 27(11): 3057-3075.
- 姚涵珍, 宋鹏, 张国安. 圆形窗口裁剪算法的研究与实践. 计算机辅助设计与图形学学报, 1992, (3): 14-20.
- 孙长嵩, 李丽洁, 宋阳. 一个基于六角网格的圆形窗口的裁剪算法. 科技与经济, 2006, (1): 78-80.
- 刘勇奎. 圆形及椭圆形裁剪窗口. 计算机工程设计, 1994, (4): 33-37.
- 蔡敏, 袁春风, 宋继强, 等. 一种快速的圆形窗口裁剪算法. 计算机辅助设计与图形学学报, 2001, 13(12): 1063-1067. [doi: 10.3321/j.issn:1003-9775.2001.12.002]
- 沈庆云, 周末水, 周儒荣. 一种圆形窗口裁剪的新方法. 计算机辅助设计与图形学学报, 1997, 9(6): 538-542. [doi: 10.3321/j.issn:1003-9775.1997.06.010]
- 黄文钧, 谢宁新. 免解二次方程的圆形窗口裁剪算法. 广西科学院学报, 2003, 19(4): 159-161, 164. [doi: 10.3969/j.issn.1002-7378.2003.04.004]
- 孙岩, 唐棣. 矩形窗口的曲线裁剪算法. 计算机应用与软件, 2003, 20(5): 35-36, 56. [doi: 10.3969/j.issn.1000-386X.2003.05.016]
- 唐棣, 孙岩. 基于像素的圆窗口的图形裁剪算法. 计算机应用与软件, 2003, 20(10): 68-69. [doi: 10.3969/j.issn.1000-386X.2003.10.030]
- 王蕊, 阎晓敏, 唐棣. 基于矩形窗口分区编码的圆形裁剪新算法. 辽宁大学学报: 自然科学版, 2011, 38(2): 177-180.
- 杭后俊, 孙丽萍. 任意多边形窗口的圆裁剪算法. 计算机技术与发展, 2009, 19(5): 235-237, 241. [doi: 10.3969/j.issn.1673-629X.2009.05.066]
- 苗永春, 唐权华. 任意多边形窗口的矢量圆裁剪算法. 计算机辅助设计与图形学学报, 2016, 28(9): 1451-1458. [doi: 10.3969/j.issn.1003-9775.2016.09.007]
- 杨琴, 李宁, 王亮亮. 任意多边形窗口的圆裁剪算法. 计算机系统应用, 2018, 27(8): 170-175. [doi: 10.15888/j.cnki.csa.006437]

(校对责编: 孙君艳)