

凸多边形窗口线裁剪的新算法

孙燮华

(中国计量学院计算机科学与技术系, 杭州 310034)

摘要 凸多边形窗口的线裁剪是用多边形窗口裁剪多边形的基础, 为此, 提出了凸 n 边形窗口的线裁剪新算法. 新算法与 Cyrus-Beck 算法相比, 当 n 较大时, 新算法的乘法大约只有 Cyrus-Beck 算法的 $1/3$ 且仅用 4 次除法. 因此, 新算法大大地加快运算速度.

关键词 计算机图形学(520·6030) 多边形 窗口 线裁剪

中图分类号: TP391 **文献标识码:** A **文章编号:** 1006-8961(2003)12-1475-03

A New Algorithm of Line Clipping for Convex Polygons

SUN Xie-hua

(Department of Computer Science and Technology, China Institute of Metrology, Hanzhou 310034)

Abstract A new algorithm of line clipping for convex polygons with n edges is proposed. Comparing the new algorithm and Cyrus-Beck algorithm, if n is sufficiently large, the number of multiplications used by the new algorithm is one third of the ones used by Cyrus-Beck algorithm and the new algorithm only uses 4 divisions for any convex polygons. Hence, the new algorithm is faster than Cyrus-Beck algorithm.

Keywords Polygon, Window, Segment clipping

0 引言

凸多边形窗口的线裁剪是用多边形窗口裁剪多边形的基础. 其主要的算法有 Cyrus-Beck 算法^[1,2], 梁友栋-Barsky 算法^[2,3] 和 Nicholl-Lee-Nicholl 算法^[2,4]. 以上这些算法有一个共同之处, 即都是用参数方程

$$P(t) = P_1 + (P_2 - P_1)t \quad (0 \leq t \leq 1)$$

表示通过端点 $P_1(x_1, y_1), P_2(x_2, y_2)$ 的线段. 在二维坐标系中, 上述方程对应两个参数方程

$$\begin{cases} x(t) = x_1 + (x_2 - x_1)t \\ y(t) = y_1 + (y_2 - y_1)t \end{cases} \quad (0 \leq t \leq 1)$$

Cyrus-Beck 算法基于凸多边形的如下性质: 点 $P(t)$ 在凸多边形内的充要条件是, 凸多边形边界上任意一点 A 到 P 的向量 \overrightarrow{AP} 和该边内法向量 N 的内积大于零, 即

$$N \cdot \overrightarrow{AP} > 0$$

现在, 考虑凸多边形窗口 R 和线段 P_1P_2 , 求 P_1P_2 落在凸多边形窗口 R 中的部分线段, 如图 1 中 $P(t_i)P(t_u)$ 的线段, 称为可见线段.

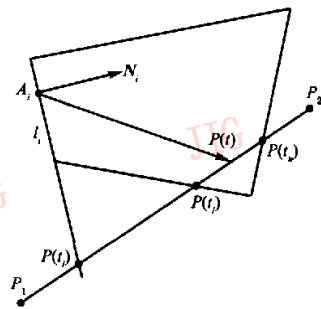


图 1 求 t_i 和 t_u

假定凸多边形有 n 条边, 在每条边 l_i 上任取一点 A_i , 记边 l_i 的内法向量为 N_i . 应寻找满足 $0 \leq t_i \leq t_u \leq 1$ 的最大区间 $[t_i, t_u]$, 使得

$$\begin{cases} N_i \cdot \overrightarrow{A_i P} \geq 0 \\ t_i \leq t \leq t_u \end{cases} \quad (i = 1, \dots, n)$$

即

$$\begin{cases} N_i \cdot \overrightarrow{A_i P_1} + t(N_i \cdot \overrightarrow{P_1 P_2}) \geq 0 \\ t_i \leq t \leq t_u \end{cases} \quad (i=1, \dots, n) \quad (1)$$

对某个 i , 若有

$$N_i \cdot \overrightarrow{P_1 P_2} = 0$$

即内法向量 N_i 与线段 $P_1 P_2$ 垂直, 则多边形的边 l_i 与线段 $P_1 P_2$ 平行, 无交点(如图 2 所示), 这时分两种情形:

(1) $N_i \cdot \overrightarrow{A_i P_1} > 0$, 线段 $P_1 P_2$ 在边 l_i 的内侧(参见图 2(a)). 程序继续, 处理其他边.

(2) $N_i \cdot \overrightarrow{A_i P_1} < 0$, 线段 $P_1 P_2$ 在边 l_i 的外侧(参见图 2(b)). 从而立即断定线段 $P_1 P_2$ 在凸多边形之外, 程序结束.

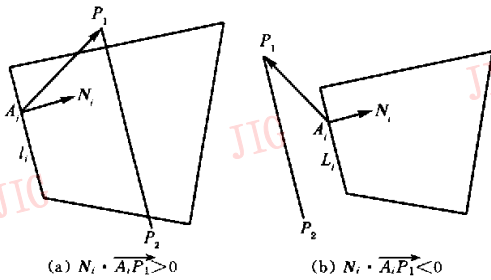


图 2 l_i 与 $P_1 P_2$ 平行

当 $N_i \cdot \overrightarrow{P_1 P_2} \neq 0$ 时, 记

$$t_i = \frac{N_i \cdot \overrightarrow{A_i P_1}}{N_i \cdot \overrightarrow{P_1 P_2}} \quad (2)$$

上面的 t_i 为线段 $P_1 P_2$ 与第 i 条边或其延长线的交点的参数(见图 1). 由式(1), 得

$$\begin{cases} t \geq t_i & N_i \cdot \overrightarrow{P_1 P_2} > 0 \\ t \leq t_i & N_i \cdot \overrightarrow{P_1 P_2} < 0 \end{cases} \quad 0 \leq t \leq 1 \quad (3)$$

显然, 式(3)解的最小值和最大值是

$$t_l = \max\{0, \max\{t_i; N_i \cdot \overrightarrow{P_1 P_2} > 0, i=1, \dots, n\}\}$$

$$t_u = \min\{1, \min\{t_i; N_i \cdot \overrightarrow{P_1 P_2} < 0, i=1, \dots, n\}\}$$

最后, 判断解的合理性. 若 $t_l \leq t_u$, 则 t_l, t_u 是可见线段的端点参数. 否则, 即 $t_l > t_u$, 则整条线段 $P_1 P_2$ 在凸多边形窗口外部.

在上面的算法中, 在计算各边内法向量 $N_i (i=0, 1, \dots, n-1)$ 时需要 $2n+2$ 次乘法, $2n+3$ 次加减法.

在 n 次求交点参数 t_i 运算式(2)中, 需要 $4n$ 次乘法, n 次除法, $3n$ 次加减法.

在用交点参数 t_i 计算交点中, 有 4 次乘法, 8 次加减法.

因此, Cyrus-Beck 算法总共需要 $6n+6$ 次乘

法, n 次除法和 $5n+11$ 次加法.

分析 Cyrus-Beck 算法, 在计算各边内法向量和计算各边与线段 $P_1 P_2$ 交点参数中使用了大量的乘法和除法, 却不管有关的边与线段实际上是否相交. 显然, 若有方法能避免计算与线段不相交边的交点, 那么, 就能大大地减少乘法和除法.

2 线裁剪新算法

本文的算法不采用线段的参数化表示. 用通常的直角坐标系方程表示. 其算法思想基于下面的定理.

定理 1 直线 AB 与线段 PQ 相交的充要条件是线段 PQ 的端点 P 和 Q 不在直线 AB 的同侧

如图 3 所示, 线段 $P_3 P_4$ 与直线 AB 相交, 端点 $P_3 P_4$, 在直线 AB 的两侧. 线段 $P_0 P_1$ 和 $P_5 P_6$ 都与直线 AB 相交, 端点 P_0 在直线 AB 上, 它们的两端点不在直线 AB 的同侧. 线段 $P_1 P_2, P_2 P_3, P_4 P_5$ 不与直线 AB 相交, 它们的两端点在直线 AB 的同侧.

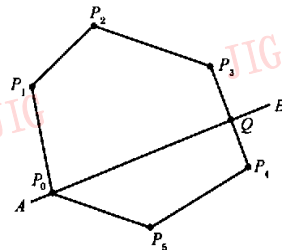


图 3 定理 1 示意图

定理 2 一线段与凸多边形至多有两个交点

新算法的基本思想是检查凸多边形各顶点在直线 AB 的哪一侧, 仅对两端点在直线 AB 异侧的边求它与直线 AB 的交点, 然后舍去在线段 AB 延长线上的交点, 判别出可见线段(图 4).

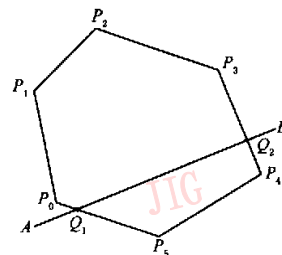


图 4 算法示意图之 1

该算法不再要求凸多边形各边的内法向量, 也不必求直线 AB 与每条边的交点, 从而舍去了大

量的求交点运算.

记凸多边形的各顶点为 $P_i(x_i, y_i) (i=0, 1, \dots, n-1)$, 定义 $P_n=P_0$. 设 $A(u_1, v_1), B(u_2, v_2)$, 则过点 A, B 的直线方程为

$$ax + by + c = 0$$

这里

$$a=v_2-v_1, b=u_1-u_2, c=bv_1-au_1$$

记 $e_i=ax_i+by_i+c$ 若 $e_i e_{i+1} < 0$, 则 P_i, P_{i+1} 分别在直线 AB 的两侧(图 4). 新算法的具体步骤如下:

(1) 计算 e_0 . 若 $e_0=0, P_0$ 是直线 AB 与多边形的一个交点. 交点计数增加 1.

(2) 对 $i=0, 1, \dots, n-1$ 计算 e_{i+1} . 若 $e_{i+1}=0, P_{i+1}$ 是直线 AB 与多边形的一个交点 $Q_i(x_i, y_i)$, 交点计数增加 1. 当交点计数等于 2 时, 退出循环.

(3) 若 $e_i e_{i+1} < 0$, 求直线 AB 与边 $P_i P_{i+1}$ 的交点 $Q(x, y)$. 交点计数增加 1, 当交点计数等于 2 时, 退出循环, 否则, 返回第 2 步.

(4) 求出线段 AB 在多边形边内的可见线段.

记线段 AB 与多边形各边的交点为 $Q_1(x, y), Q_2(x_2, y_2)$. 有可能无交点或只有一个交点.

不妨设 $u_1 < u_2$. 对其他情形, 可类似地处理. 若 $u_1 \leq x_1, x_2 \leq u_2$, 且交点计数等于 2, 则 $Q_1 Q_2$ 为可见线段(参见图 4).

若 $u_1 \leq x_1 \leq u_2, x_2 \geq u_2$ 且交点计数等于 2, 则 $Q_1 B$ 为可见线段(参见图 5). 这时另一交点 Q_2 在线段 AB 的延长线上.

若 $u_1 \leq x_2 \leq u_2, x_1 < u_1$ 且交点计数等于 2, 则 $A Q_2$ 为可见线段. 这时另一交点 Q_1 在线段 AB 的延长线上. 对于其他的情形, 类似地处理.

若交点计数等于 0 或 1, 无可见线段.

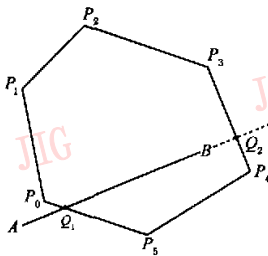


图 5 算法示意图之 2

3 算法分析与实验结果

该算法在一般情况下, 判别 $e_i (i=0, 1, \dots, n-1)$

的正负号中至多使用 $2n+2$ 次乘法和 $2n+3$ 次加减法.

在二次求交点运算中需要 16 乘法, 4 次除法和 12 次加减法.

于是, 本算法至多使用 $2n+18$ 次乘法, 4 次除法和 $2n+15$ 次加减法.

由此可见, 新算法当 n 较大时, 其乘法大约只有 Cyrus-Beck 算法的 $1/3$, 且大大地减少了除法.

表 1 是分别用 Cyrus-Beck 算法程序 Clip_CB.C 和新算法程序 Clip_NW.C 对同一个半径为 150 个像素的正 1 000 边形的线裁剪 20 000 次的速度比较. 其中线段 1 的端点是(230, 250), (240, 460), 线段 2 的端点是(530, 250), (150, 290).

表 1 C-B 算法与新算法速度比较 单位:s

	对线段 1 裁剪	对线段 2 裁剪
C-B 算法	13	15
新算法	6	7

由此可见, 新算法比 Cyrus-Beck 算法要快得多. 且速度之比与理论分析接近.

参 考 文 献

- 1 Cyrus M, Beck J. Generalized two-and three-dimensional clipping [J]. Computers and Graphics, 1978, 3(1):23~28.
- 2 Rogers D F. 计算机图形学的算法基础(第二版)[M]. 北京:机械工业出版社, 2002.
- 3 Liang Y D, Barsky B A. A new concept and method for line clipping[J]. ACM Transactions on Graphics, 1984, 3(1):1~22.
- 4 Nicholl T M, Lee D T, Nicholl R A. An efficient new algorithm for 2D line clipping: Its development and analysis[J]. Computer Graphics, 1987, 21(4):253~262.
- 5 刘斌, 王勇, 黄树槐. 二维线段的矩形窗口逐边裁剪算法研究 [J]. 计算机应用研究, 1997, 1, 15~17.
- 6 陆国栋, 吴晖. 基于变窗口过滤技术的线段裁剪中点分割算法 [J]. 计算机辅助设计与图形学学报, 2002, 14(6):513~517.



孙燮华 1945 年生, 软件研究所所长, 中国计量学院计算机系教授, 1981 年于杭州大学信息与计算科学系硕士研究生毕业, 1990~1991 美国麻省理工学院(MIT)高级访问学者. 主要研究领域为计算机图形学、图象处理、算法分析和应用数学等.