一种平面简单多边形核的求解算法

柳　伟　何援军　李震霄
（上海交通大学计算机科学与工程系，上海　200240）

摘　要　平面简单多边形的核是该多边形内部的一个点集，该点集中任意一点与多边形边界上一点的连线都处于这个多边形内部。可见核的这一性质在摄像机定位等问题上得到了应用，本文提出了一种简单多边形核求解的新方法，该方法不仅可以判断核的存在性，而且可以得到核多边形顶点序列。给出的算法容易理解，便于实现，可以广泛地应用于此类问题的求解。

关键词　简单多边形　核　计算几何
中图法分类号：TP391　　**文献标识码**：A　　**文章编号**：1006-8961(2007)06-1098-05

An Algorithm to Calculate the Kernel of A Plane Simple Polygon

LIU Wei, HE Yuan-jun, LI Zhen-xiao
(*Department of Computer Science and Engineering, Shanghai Jiaotong University, Shanghai* 200240)

Abstract　The kernel of a plane simple polygon is a point set in the interior of this polygon, in which any point can be connected to another arbitrary point on the border of the polygon with a line inside the polygon. The character of kernel finds its application in such fields as the positioning of the camera. We propose a new method to calculate the kernel of a single polygon, which can not only judge the existence of the kernel, but also get the point list of the kernel. The given arithmetic is easy to understand and realize, and can be widely applied to such kind of problems.

Keywords　simple polygon, kernel, computational geometry

1　引　言

平面简单多边形是指非相邻的边不相交的平面多边形(本文以后所指的多边形均为平面多边形)。如果给定一个简单多边形中存在一个点集，其中任意一点与对应简单多边形边界上的一点的连线都处在这个多边形内部，则称这个点集为该多边形的核。求解简单多边形核是计算几何中一个经常会碰到的问题。

文献[1]提出了判断平面多边形核是否存在的一个快速算法，该算法构思巧妙，在实际应用中，对于数量较少的简单多边形来说有很高的精确性，并且能很快地判断出该多边形是否有核，大大简化了计算的工作量，但这种方法功能有限，并不能求解核的具体顶点坐标情况。文献[2]结合 z 缓冲器扫描算法，设计了一种核的填充算法，避免了核顶点的计算，能快速地确定简单多边形核的范围，但这种方法只解决了核快速显示的问题，同样不能求解核的顶点坐标。文献[3]、[4]提出了求解平面多边形核心的算法，需要根据顶点的若干种情况进行判断，比较复杂。

在总结前面的相关研究后，本文将提出一种计算简单多边形核的算法，利用多边形边依次剖分初始核多边形即多边形的最小包围矩形来实现，该算法简单明了，对于任何多边形都可以采用同样的处理，速度较快，有一定的实用价值。

2　简单多边形的核计算方法

记简单多边形为 $P\langle v_0, v_1, \cdots, v_{n-1}\rangle$，其中 $v_0, v_1, \cdots, v_{n-1}$ 是多边形的顶点序列，并规定此顶点序列

基金项目：国家自然科学基金项目(60573146)；国家"863"高技术研究发展计划项目(2003AA411310)
收稿日期：2005-09-26；**改回日期**：2006-02-27

第一作者简介：柳伟(1979～　)，男。2004 年获南京航空航天大学机械设计及理论专业硕士学位，现为上海交通大学计算机科学与工程系博士研究生。主要研究方向为计算机图形学。E-mail：Liuwei@cs.sjtu.edu.cn

为逆时针方向。同时记 $e_i=(v_i,v_{i+1})$ $(0\leqslant i\leqslant n-2)$ 为由点 v_i 及 v_{i+1} 所形成的边,同时令 $e_{n-1}=(v_{n-1},v_0)$ 为其顶点首尾相接而形成的边。

给定简单多边形 $P\langle v_0,v_1,\cdots,v_{n-1}\rangle$,对其边界上任意一点 s,设 v 是 P 内部一点,从 v 到 s 的线段 vs 全部在 P 的内部,则具有如此性质的点 v 的集合,称为简单多边形的核。核的一个重要特性是:从核内任意一点都可见多边形的全部边界。直观地讲,如果把多边形边看成是逆时针方向的环,则核的区域即为由每条多边形边切分整个平面而得到的左半平面的交集。一般而言,凸多边形的核就是它本身,而凹多边形的核可能是其内部的一部分,也可能根本就不存在。

基于上述思想,多边形核可以按照以下思路求出:逆时针依次用多边形的边剖分多边形所在的平面,留下其左边的部分,舍去其右边部分,最后剩下的便是此多边形的核。这种方法思路简单,但可操作性很差。一种可行的改进办法是对此多边形的最小包围盒进行剖分,而不是其所在的平面。同时这种思路的好处是每次分割后形成的区域为凸多边形,而不是一个无限区域。而且因为直线与凸多边形的交点最多只有两个,从而简化了计算而不像文献[4]那样有多个交点而进行复杂的判断。

将多边形的最小包围盒作为初始核多边形(kernel polygon, KP),依次用多边形的边所在的直线与核多边形进行求交计算,于是可以得到一系列核多边形系列 $KP_1,KP_2,\cdots,KP_{n-1},KP_n$,该多边形系列将逐渐逼近所要求解的核多边形。很显然,KP_n 即为该多边形的核。过程如图 1 所示。

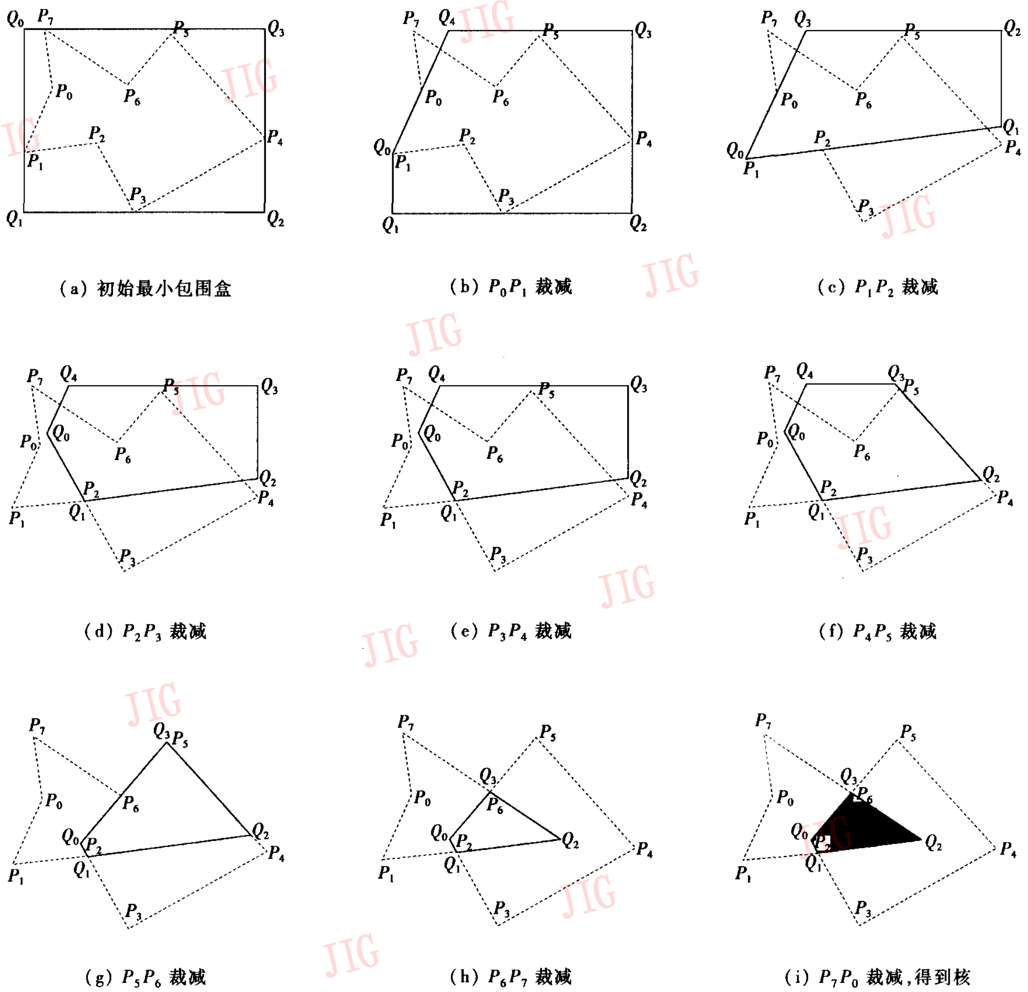


图 1　求核过程示意图(虚线表示简单多边形,实线表示核多边形系列)

Fig. 1　Illustration of process to get the kernel

(Grey lines represent the simple polygon while black lines represent the serial of kernel polygons)

3 具体求解过程

3.1 多边形边系数矩阵

设多边形某条边 e_i 的两个顶点的坐标分别为 (x_i, y_i) 及 (x_{i+1}, y_{i+1})，该边所在直线的方程为 $a_i x + b_i y + c_i = 0$，则其 3 个参数可初步设定为

$$
\begin{aligned}
a_i &= y_{i+1} - y_i \\
b_i &= -x_{i+1} + x_i \\
c_i &= x_{i+1} y_i - x_i y_{i+1}
\end{aligned}
$$

对于一个 n 边形，可用一参数矩阵

$$
\boldsymbol{M}_1 = \begin{bmatrix} a_0 & a_1 & \cdots & a_{n-2} & a_{n-1} \\ b_0 & b_1 & \cdots & b_{n-2} & b_{n-1} \\ c_0 & c_1 & \cdots & c_{n-2} & c_{n-1} \end{bmatrix}
$$

记录其各边所在直线的参数情况。矩阵中每一列代表对应边所在直线的系数。多边形所在平面内的点用齐次坐标中的一个坐标矢量 $(x, y, 1)$ 来表示，设此矢量与参数矩阵 $\boldsymbol{M}_1$ 的乘积为向量 $(flag_0, flag_1, \cdots, flag_{n-1})$，则此向量中各分量的符号代表该点位于对应边的那一侧。约定若该点位于此多边形内部，则所有 $flag_i$ 的符号应该为正。因此对上述矩阵元素应加以调整，调整的方法如下：

对每一条边 $e_i = (v_i, v_{i+1})$，设其前一点 v_{i-1} 的齐次坐标矢量与参数矩阵 $\boldsymbol{M}_1$ 的乘积中第 i 个分量为 $flag_i$，若顶点 v_i 为凸点且 $flag_i < 0$ 时或顶点 v_i 为凹点且 $flag_i > 0$ 时，同时改变 a_i, b_i, c_i 的符号，即令 $a_i = -a_i, b_i = -b_i, c_i = -c_i$。

3.2 逐边裁减

由于是用直线依次分割多边形，且初始多边形为矩形，因此核多边形系列 $KP_1, KP_2, \cdots, KP_n$ 均为凸多边形。边 e_i 所在的直线剖分多边形 KP_i 求交得到多边形 KP_{i+1}，e_i 与 KP_i 的关系有 3 种情况：(1) e_i 完全包含在 KP_i 之中（如图 1 中 P_0P_1, P_1P_2）；(2) e_i 与 KP_i 相交（如图 1 中 $P_2P_3, P_4P_5, P_5P_6, P_6P_7$）；(3) e_i 与 KP_i 分离（如图 1 中 P_3P_4, P_7P_1）。在求交过程中，第 2 种情况可以转化为第 1 种情况，方法是在 e_i 所在的直线上取包含在 KP_i 中两点取代 e_i 的两个端点，并且使得其方向与 e_i 的方向保持一致。经过上述处理，现在就可以只考虑第 1 种和第 3 种情况。

这两种情况的判断方法为：将多边形 KP_i 的顶点齐次坐标右乘参数矩阵 $\boldsymbol{M}_1$，若对于所有的顶点，$flag_i$ 的方向一致，则表明 KP_i 的所有顶点位于 e_i 的一侧，即 KP_i 与 e_i 分离。否则可归为第 1 种情况。若 $flag_i$ 均大于 0，则表示多边形 KP_i 的所有顶点在边 e_i 正侧，则此次计算结束，求解下一条边；若 $flag_i$ 均小于 0，则表示多边形 KP_i 的所有顶点在边 e_i 负侧，则整个计算结束，核为空，即不存在。图 2 即为核不存在的一例。

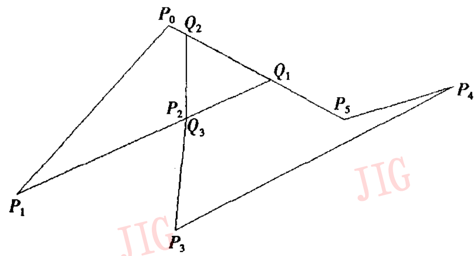


图 2 多边形 $P_0P_1P_2P_3P_4P_5$ 核不存在

Fig. 2 The kernel of polygon $P_0P_1P_2P_3P_4P_5$ doesn't exist

当分别用 $P_0P_1, P_1P_2, P_2P_3, P_3P_4$ 裁减后，得到多边形 $Q_1Q_2Q_3$，这时它们都位于边 P_4P_5 的负侧，这时可以判断该多边形核为空。与之相反，多边形 $Q_1Q_2Q_3$ 在边 P_3P_4 的正侧，经过判断后不予处理。

剖分多边形 KP_i 的一个最重要的步骤就是求解 e_i 与 KP_i 的交点。下面讨论 e_i 在 KP_i 内部情况下的求交运算。将边 e_i 所在的直线方程用参数方程 $P = P_i + (P_{i+1} - P_i)t$ 表示，起点 P_i 终点 P_{i+1} 按多边形的逆时针方式确定。设多边形 KP_i 有 n_i 条边，则 e_i 所在的直线与多边形 KP_i 的交点个数为 n_i，记为 $t_0, t_1, \cdots, t_{n_i-1}$。因为多边形 KP_i 内含 e_i，故 t 值的范围只能在 $(-\infty, 0]$ 及 $[1, +\infty)$ 之间，据此将这 n_i 个 t 值分成两个组：下限组在 $(-\infty, 0]$ 范围内，上限组在 $[1, +\infty)$ 范围内。下限组取最大值 $t_{\max}$，上限组取最小值 $t_{\min}$，并将这两个 t 值依次代入参数方程 $P = P_i + (P_{i+1} - P_i)t$ 中，即得到所求交点。

这种求交算法比 Cyrus-Beck 算法[5]的一个优势在于通过将线段的端点移入需剖分的多边形内，就可以通过 n_i 次求交得到交点，而不需要通过计算多边形每条边法矢与线段向量点乘结果的正负确定上限组和下限组。

3.3 顶点替换

多边形 $KP_1, KP_2, \cdots, KP_{n-1}, KP_n$ 依次形成的过程本质上是交点替换多边形部分顶点的过程。首先要解决的问题是分清哪些顶点需要删除，交点如何插入。对于 KP_i 位于 e_i 的一侧的情况，要么判断核

不存在，处理结束，要么不予处理。而对于相交情况，情况就要复杂一些。对于第 i 次替换，可以将多边形 KP_i 的顶点序列组成一个齐次坐标矩阵 $\boldsymbol{M}_{2i}$。即

$$\boldsymbol{M}_{2i} = \begin{bmatrix} x_{i0} & y_{i0} & 1 \\ x_{i1} & y_{i1} & 1 \\ \cdots & \cdots & \cdots \\ x_{i(n_i-1)} & y_{i(n_i-1)} & 1 \end{bmatrix}$$

将矩阵 $\boldsymbol{M}_{2i}$ 右乘向量 $(a_i, b_i, c_i)^{\mathrm{T}}$（$(a_i, b_i, c_i)^{\mathrm{T}}$ 为本文 2.1 节中得到的边 e_i 对应的列向量），设其结果为一列向量 $\boldsymbol{\alpha} = (\mathrm{sign}_{i0}, \mathrm{sign}_{i1}, \cdots, \mathrm{sign}_{i(n_i-1)})^{\mathrm{T}}$，若 $\boldsymbol{\alpha}$ 中某分量小于零，则表示这一分量所对应的顶点位于边 e_i 相对于核的外测，故这个顶点应该删除。一般来说，边 e_i 将多边形 KP_i 分成两个部分，因此，需要删除的顶点是连续顶点。下一个步骤就是将求交过程中生成的顶点插入到删除点的位置，从而生成新的多边形 KP_{i+1}。图 1 详细表示了顶点替换的过程。

3.4　算法描述

通过上面 3 个关键步骤的分析，整个求核过程就很清晰了。下面即为本算法的具体描述。

Step1：输入多边形的顶点，求出多边形参数矩阵 $\boldsymbol{M}_1$，求出多边形的最小包围盒作为初始核多边形 KP_0。

对多边形每条边进行如下操作：

Step2：判断多边形边 e_i 与多边形 KP_i 的关系，若为上述的第 1 种或第 2 种情况，则计算与多边形 KP_i 的交点；若为第 3 种情况，则判断是否 KP_i 保持不变还是核为空。

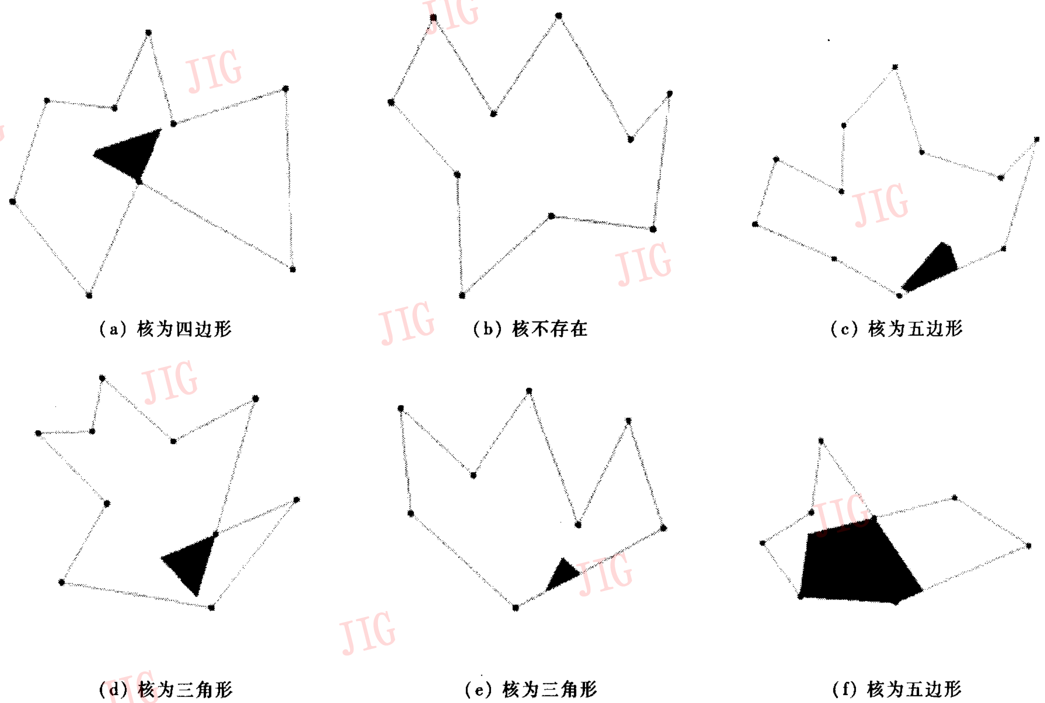
Step3：根据第 2 步计算的交点，进行顶点替换，得到多边形 KP_{i+1}。

Step4：若每条边计算完毕或判断核为空，结束程序，否则转到 Step2。

4　实验与结论

根据上述思想，我们用 Visual C + +6.0 实现了一个可视化的任意多边形核心求解系统，图 3 是几个具体的算例。

可以看出，本文提出的算法可以有效地解决求解多边形的核心问题，不仅可以判断核的存在性，而且可以求解出核的顶点坐标系列。本算法的一个优点是判断较少，适应性比较强，因此很容易直接推广到球面多边形核心及空间多面体核心的求解问题上。



(a) 核为四边形　(b) 核不存在　(c) 核为五边形

(d) 核为三角形　(e) 核为三角形　(f) 核为五边形

图 3　几个算例

Fig. 3　Several examples

参考文献(References)

1 Wang Zheng-xuan, Xu Chang-qing, Pang Yun-jie. A fast algorithm to determine whether the kernel of simple polygon is empty [J]. Journey of Computer Aided Design and Computer Graphics, 2000, **12**(9): 656 ~ 659. [王钲旋,徐长青,庞云阶. 判断简单多边形的核是否为空的一个快速算法[J]. 计算机辅助设计与图形学学报, 2000, **12**(9):656 ~ 659.]

2 Lu Nan, Chen Bing-fa. The scanning beam algorithm of filling the visible core of polygon[J]. Modern Computer, 2003, **13**(9): 123 ~ 126. [陆楠,陈炳发. 简单多边形的可见核的扫描线填充算法[J]. 现代计算机, 2003,**13**(9):123 ~ 126.]

3 Wang Hai-xiang. An algorithm for deciding and construction of kernel of simple polygon in computational geometry [J]. Computer Engineering, 1995, **20**(2): 482 ~ 484. [王相海. 计算几何中简单多边形核的存在判定及构造的一个算法[J]. 计算机工程, 1995, **20**(2):482 ~ 484.]

4 Zhou Pei-de. An algorithm for deciding the kernel of polygon [J]. Journey of Engineering Graphics, 1995, **11**(2): 28 ~ 30. [周培德. 确定任意多边形的核的算法[J]. 工程图学学报, 1995, **11**(2): 28 ~ 30.]

5 David F Rogers. Procedural Elements for Computer Graphics (Second Edition) [M]. Beijing: China Machine Press, 2000: 148 ~ 157. [David F Rogers. 计算机图形学的算法基础(第 2 版)[M]. 北京: 机械工业出版社, 2000: 148 ~ 157.]