1000-9825/2002/13(07)1309-04 ©2002 Journal of Software 软 件 学 报 Vol.13, No.7

任意多边形顶点凸、凹性判别的简捷算法*

刘润涛

(哈尔滨理工大学 计算机应用技术研究所,黑龙江 哈尔滨 150080)

E-mail: liurt@0451.com

http://www.hrbust.edu.cn

摘要: 给出了一种确定任意多边形顶点凸、凹性的简捷算法.该算法只需要 $2n+4$ 次乘法,$5n+10$ 次加、减法及 $2n+3$ 次比较即可完成(n 是多边形顶点的个数).同时,给出了任意简单多边形走向的充要条件.

关 键 词: 多边形;凸凹性;算法;走向;充要条件

中图法分类号: TP391 文献标识码: A

在模式识别、图像处理、曲面插值等方面常常遇到对多边形区域或离散点进行分割的问题,如能预先确定每个多边形顶点的凸、凹性,就可使该问题的解决得到简化[2,3].对于判别任意多边形顶点的凸、凹性的问题,文献[1]给出了一个算法,其时间复杂性为 $O(n^2\log n)$次乘法和 $O(n^2)$次比较.本文所给出的算法,其时间复杂度仅为 $2n+2$ 次乘法,$2n$ 次加、减法及 $3n$ 次比较,时间复杂度减少.

本文首先给出几个相关的概念,然后讨论任意简单多边形走向判别的充要条件.之后给出任意多边形顶点凸、凹性判别的算法.最后对该算法的时间复杂度进行分析.

1 基本概念

为叙述方便,先给出几个相关的定义.

定义 **1**. 设 $p_i=(x_i,y_i),i=1,2,3,\ldots,n,p_{n+1}=p_1$ 是给定多边形的 n 个顶点,若对任意 $i,j(i\neq j),i,j=1,2,3,\ldots,n$,线段 p_ip_{i+1} 与 p_jp_{j+1} 或是相邻且相交于一端点或不相交,则称该多边形为简单多边形.

定义 **2**. 设 $p_1,\ldots,p_n,p_{n+1}=p_1$ 是一个简单多边形.若线段 $p_{i-1}p_i$ 与线段 p_ip_{i+1} 所形成的内角(即由该多边形所围有界区域内所形成的角)是一个不超过 180°的角,则称顶点 p_i 是凸的,否则,称 p_i 是凹的.

由此定义可知,对任意一个简单多边形,其每个顶点或是凸的,或是凹的.

定义 **3**.设 $p_1,p_2,\ldots,p_n,p_{n+1}=p_1$ 是一个简单多边形.若沿 $p_1\to p_2\ldots p_n\to p_{n+1}$ 方向走,该简单多边形所围的有界区域总在左边,则称该多边形的走向是逆时针的;反之,称其走向是顺时针的.

2 简单多边形走向的充要条件

给定一个简单多边形,其顶点为 $p_1,p_2,\ldots,p_n,p_{n+1}=p_1$,它的走向是逆时针的还是顺时针的,对于判别每个顶点的凸、凹性是很重要的.如何用较小的计算量解决这个问题是确定每个顶点的凸、凹性的一个关键步骤.

解决该问题的思路是:先求出给定的多边形 n 个顶点的 x 值或 y 值最大或最小的点(称其为极值点),使该多边形落入由这些 x 值和 y 值最大或最小的点构成的矩形内(注意,一定有点落在 4 条边上),如图 1 所示.然后,依据该多边形在每个极值点处与相邻两顶点的位置关系,就可以确定该多边形的走向.

下面以 y 值最大的点为例说明该方法的实现过程.

* 收稿日期: 2000-11-29; 修改日期: 2001-04-26

基金项目: 国家自然科学基金资助项目(69705004,10171025);黑龙江省自然科学基金资助项目(F9706)

作者简介: 刘润涛(1961-),男,黑龙江东宁人,副研究员,主要研究领域为计算机辅助几何设计,计算机图形算法.

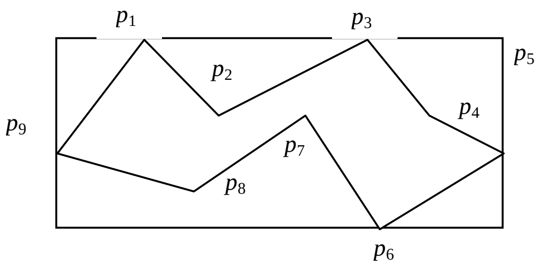

Fig.1
图 1

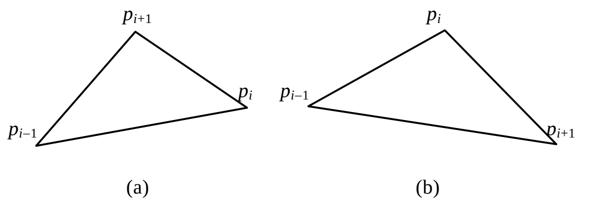

Fig.2
图 2

记 p_i 为 y 值最大的点,即 $y_i=\max[y_j]$,做辅助点 p_i':

当 p_{i-1},p_i,p_{i+1} 共线时,令 $p_i'=p_i-\alpha(0,1)$(α为任一正数,可取$\alpha=|p_{i-1}p_i|$).

否则,令 $p_i'=(p_{i-1}+p_{i+1})/2$.

作以 p_i'为原点通过 p_i 的射线 l(注意,l 是有方向的),则 p_{i-1},p_{i+1} 位于 l 的两侧.易见,若 p_{i-1} 位于 l 的左侧,则该多边形是顺时针的.反之亦然.

因此,得到以下结论,该多边形走向是顺时针的$\Leftrightarrow p_{i-1}$ 位于 l 的左侧.进行类似的讨论可得,该多边形走向是逆时针的$\Leftrightarrow p_{i-1}$ 位于 l 的右侧.那么,如何判别 p_{i-1} 是位于 l 的左侧还是右侧呢?

为此,引入函数 $S(p_{i-1},p_i,p_{i+1})=\begin{vmatrix} x_{i-1} & y_{i-1} & 1 \\ x_i & y_i & 1 \\ x_{i+1} & y_{i+1} & 1 \end{vmatrix}=\Delta p_{i-1}p_ip_{i+1}$ 的有向面积的 2 倍.

若 p_{i-1} 在由 p_i 经 p_{i+1} 的射线的左侧,则$\Delta p_{i-1}p_ip_{i+1}$ 的有向面积为正,否则为负,如图 2 所示,图 2(a)为 p_{i-1} 在由 p_i 经 p_{i+1} 的射线的左侧,图 2(b)为 p_{i-1} 在由 p_i 经 p_{i+1} 的射线的右侧.

经简单计算可得

$$S(p_{i-1},p_i,p_{i+1})=(x_i-x_{i-1})(y_i-y_{i+1})-(y_i-y_{i-1})(x_i-x_{i+1}),$$

因此,多边形走向是顺时针的$\Leftrightarrow S(p_{i-1},p_i',p_{i+1})>0$,多边形走向是逆时针的$\Leftrightarrow S(p_{i-1},p_i',p_{i+1})<0$.

对其他的极值点处也有类似的结果,惟一的区别就在于 p_i'的构造上,这里不再赘述.

3 多边形顶点凸、凹性判别及算法

确定了多边形的走向以后,判定每个顶点的凸、凹性问题就容易解决了.

对每个顶点 $p_i(i=1,2,3,\ldots,n)$,只需先计算 $S(p_{i-1},p_i,p_{i+1})$,然后,根据该值的正、负性及该多边形的走向就能确定该点的凸、凹性.

具体地,若多边形走向是逆时针的,则当 $S(p_{i-1},p_i,p_{i+1})\geq 0$ 时,p_i 点就是凸的,而当 $S(p_{i-1},p_i,p_{i+1})<0$ 时,p_i 点就是凹的.

对于多边形走向是顺时针的情况,判别条件与多边形走向是逆时针的情况相反,即:

当 $S(p_{i-1},p_i,p_{i+1})\leq 0$ 时,p_i 点就是凸的,而当 $S(p_{i-1},p_i,p_{i+1})>0$ 时,p_i 点就是凹的.

下面给出该算法的实现过程.为表示上的一致性,令 $p_{-1}=p_n$,算法表述如下:

Step 1. 求 y 值最大的点 p_k,使 $y_k=\max\{y_i\}$

令 $k=1,y_{\max}=y_1$

对 $m=1$ 到 n

若 $y_m>Q$ 则 $k=m,Q=y_m$

经过该步骤,即 n 次比较,就可得到所需的点及相应的下标 k.

Step 2. 判别多边形的走向

用 $Nflag=0$ 表示走向为逆时针,$Nflag=1$ 表示走向为顺时针.先给定$\varepsilon>0$(用来判定两个点的接近共线的

情况),

令 $Nflag=0$

计算 $v=S(p_{k-1},p_k,p_{k+1})$

$=(x_k-x_{k-1})(y_k-y_{k+1})-(y_k-y_{k-1})(x_k-x_{k+1})$

若$|v|<\varepsilon$,则令 $p'=p_k-|p_k-p_{k-1}|(0,1)$

否则,令 $p'=(p_{k-1}+p_{k+1})/2$

计算 $\omega=S(p_{k-1},p_k',p_{k+1})$

若 $\omega>0$,则令 $Nflag=1$

Step 3. 确定各顶点的凸、凹性

用一个数组 $symble(n)$来记载各顶点的凸、凹性,$symble(i)=0$ 表示 p_i 是凸的;$symble(n)=1$ 表示 p_i 是凹的.

若 $Nfalg=0$ 则对 $i=1$ 到 n

计算 $v=S(p_{i-1},p_i,p_{i+1})$

若 $v\geq 0$ 则 $symble(i)=0$

否则 $symble(i)=1$

若 $Nfalg=1$ 则对 $i=1\sim n$

计算 $v=S(p_{i-1},p_i,p_{i+1})$

若 $v\leq 0$ 则 $symble(i)=0$

否则 $symble(i)=1$

经过该算法的 3 个步骤之后,各个顶点的凸、凹性就全部得到了,其标识存放在数组 $symble(n)$中.

4 算法的计算复杂性分析及结论

Step 1 中用了 n 次判断,Step 2 中至多用 4 次乘法,2 次判断,10 次加减法,而 Step 3 用了 $2n$ 次乘法,$5n$ 次减法和 $n+1$ 次判断.因此,整个算法共需要 $2n+4$ 次乘法,$5n+10$ 次加减法运算及 $2n+3$ 次判断.本算法已应用于我们自行开发的三维几何造型软件系统中,算法稳定.

References:

[1] Zhou Pei-de. An algorithm for determining convex-concave vertices of arbitrary polygon. Journal of Software, 1995,6(5):276~279 (in Chinese).

[2] Zhou Pei-de. An algorithm for determining the vertices of a convex hull. Journal of Beijing Institute of Technology, 1993,13(1): 69~72 (in Chinese).

[3] Liu, H., Srinath, M.D. Corner detection from chain-code. Pattern Recognition, 1990,23(1,2):51~68.

附中文参考文献:

[1] 周培德.确定任意多边形凸、凹顶点的算法.软件学报,1995,6(5):276~279.

[2] 周培德.求凸壳顶点的一种算法.北京理工大学学报,1993,13(1):69~72.

A Simple and Fast Algorithm for Detecting the Convexity and Concavity of Vertices for an Arbitrary Polygon*

LIU Run-tao

(*Institute of Computer Application Technology*, *Harbin University of Science and Technology*, *Harbin* 150080, *China*)

E-mail: liurt@0451.com

http://www.hrbust.edu.cn

Abstract: A simple and fast algorithm for detecting the convexity and concavity of vertices for an arbitrary polygon is presented, which needs 2*n*+4 multiplications, 5*n*+10 additions or subtractions and 2*n*+3 comparisons (*n* is the number of the vertices of the polygon). Meanwhile, the sufficient and necessary condition for the orientation of an arbitrary simple polygon is given.

Key words: polygon; convexity and concavity; algorithm; orientation; sufficient and necessary condition

* Received November 29, 2000; accepted April 26, 2001

Supported by the National Natural Science Foundation of China under Grant Nos.69705004, 10171025; the Natural Science Foundation of Heilongjiang Province of China under Grant No.F9706