

文章编号: 1006-544X (2007) 04-0589-04

# 一种凸多边形的交、并求解算法

黄俊华<sup>1,2</sup>, 闫遂军<sup>3</sup>, 朱小龙<sup>3</sup>, 李景文<sup>3</sup>

(1. 武汉大学 遥感信息工程学院, 武汉 430079; 2. 广西区国土资源厅, 南宁 530023;

3. 桂林工学院 土木工程系, 广西 桂林 541004)

**摘 要:** 凸多边形交、并求解的难点在于如何维护结果多边形的顶点序列. 提出了以顶点与多边形的空间拓扑关系为基础, 将不同拓扑关系的顶点进行重新组合的方法来解决任意2个凸多边形交、并的求解问题. 算法易于编程实现, 能够较好的求解二维凸多边形区域各种情况的交与并, 可为GIS应用中矢量多边形之间的关系计算提供算法基础.

**关键词:** 计算几何; 凸多边形; 相交; 相并

**中图分类号:** TP301.6; P208

**文献标志码:** A

研究2个平面凸多边形的交、并求解算法是GIS应用中矢量多边形之间关系计算的基本问题之一<sup>[1,2]</sup>. 现有的一些算法虽然能够解决这一问题, 但是它们多是以多边形的边与另外一个多边形的空间关系为依据, 计算得到交、并多边形, 不仅顶点序列维护困难, 而且需要大量的存储空间. 笔者从多边形的顶点与另外一个多边形的空间关系的角度考虑, 将不同拓扑关系的顶点和2个多边形的交点进行合并, 得到了结果多边形顶点序列, 从而实现了凸多边形的交、并求解运算, 较好的维护顶点序列的存储顺序.

## 1 算法的描述

### 1.1 简单凸多边形的定义

平面多边形是由 $N$ 个点 $V_1, V_2, V_3, \dots, V_n$ 首尾相连的闭合折线段表示,  $V_i (i = 1, 2, \dots, n)$ 为多边形顶点,  $V_1V_2, V_2V_3, \dots, V_{n-1}V_n, V_nV_1$ 为多边形的边. 对于任意非相邻边不相交的多边形称为平面简单多边形. 凸多边形是简单多边形的一种特例, 如果简单多边形的任意顶点 $V_i (i = 1, 2, \dots, n)$ 对应的多边形内角 $\theta_i \in (0, \pi)$ , 则称此多边形为平面简单凸多边形. 除去凸多边形以外的简单多边形都是凹

多边形. 简单凸多边形是客观世界中空间实体常用的描述形式, 其交、并求解算法是空间实体逻辑关系判断的常用方法.

### 1.2 算法思想

2个凸多边形在平面上存在的位置关系有: 分离、包含、相交和相切. 对于2个凸多边形为分离、包含和相切的情况, 其交集和并集较容易表示出来. 本文主要讨论2个凸多边形相交情况下的交、并的求解算法.

算法的基本思想是: 将参与求解过程的2个凸多边形的顶点集按照顶点与多边形的拓扑关系进行双向筛选, 再将筛选出的2组顶点和2个多边形的交点, 按照多边形顶点序列逆时针顺序存储的原则合并共同存储为一个顶点序列, 即可得到结果多边形的顶点序列 (图1).

## 2 算法实现的过程

### 2.1 凸多边形交集求解算法的实现过程

多边形的顶点集采用变长数组进行表示, 按照逆时针顺序存储顶点, 在图2中, 凸多边形的顶点序列表示为 $P(P_1, P_2, P_3, \dots, P_n)$ 和 $Q(Q_1, Q_2, Q_3, \dots, Q_n)$ .

收稿日期: 2007-04-11

基金项目: 国家自然科学基金资助项目 (40574002); 广西自然科学基金项目 (桂科自0448076)

作者简介: 黄俊华 (1963-), 男, 博士研究生, 副教授, 研究方向: GIS空间分析与决策方法.

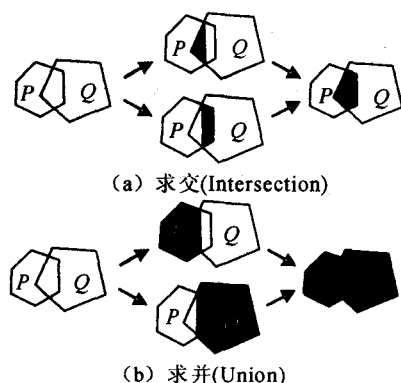


图1 算法思想

Fig. 1 Basic idea of the algorithm

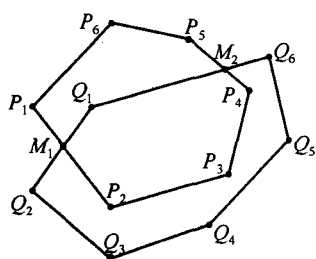


图2 凸多边形

Fig. 2 Two convex polygons

凸多边形交集求解算法的多边形顶点的筛选条件:判断顶点与另一个多边形的拓扑关系,如果顶点为内部点,则存储待用。定义点序列  $P\_Q$  由多边形  $P$  的顶点序列中包含在多边形  $Q$  内部的顶点(图2中的  $P_2, P_3, P_4$ )和多边形  $P, Q$  的交点(图2中的  $M_1, M_2$ )组成;定义点序列  $Q\_P$  由多边形  $Q$  的顶点序列中在多边形  $P$  内部的顶点和多边形  $P, Q$  的交点组成。点序列  $P\_Q, Q\_P$  合并即可得到交集多边形的顶点序列。

2.1.1 求解点序列  $P\_Q$  和  $Q\_P$  点序列  $P\_Q$  的求解算法描述如下。

Step 1:初始化  $i = 1, n$  为多边形  $P$  当前的顶点数目,  $m$  为多边形  $Q$  当前的顶点数目;然后在顶点序列  $P, Q$  尾部分别插入点  $P_1, Q_1$ 。

Step 2:判断  $P_i$  是否在多边形  $Q$  内部,如果在,则将  $P_i$  点存储到点序列  $P\_Q$  中。

Step 3:定义变量  $j = 1$ 。

Step 4:判断线段  $P_i P_{i+1}, Q_j Q_{j+1}$  是否有交点,如果有交点,则将交点插入到点序列  $P\_Q$  中。

Step 5:判断  $j$  是否与  $m$  相等,如果不相等,则执行  $j++$ ,再跳回 Step 4。

Step 6:判断  $i$  是否与  $n$  相等,如果不相等,则执行  $i++$ ,再跳回 Step 2。

Step 7:分别将顶点序列  $P, Q$  最后一个顶点(在 Step 1 中插入的  $P_1, Q_1$ )删除。

Step 8:求解过程完成,退出。

按照此算法的点序列  $P\_Q$  为  $(M_1, P_2, P_3, P_4, M_2)$ ;同理可以得到点序列  $Q\_P$  为  $(Q_1, M_1, M_2)$ 。

2.1.2 合并点序列  $P\_Q$  和  $Q\_P$  合并点序列  $P\_Q, Q\_P$  即可得到多边形  $P, Q$  的交集多边形的顶点序列,合并过程中必须维护交集多边形顶点序列的逆时针存储方式,而多边形  $P, Q$  的交点是点序列  $P\_Q, Q\_P$  合并的依据。

本文所采用的合并算法是将点序列  $Q\_P$  的每个顶点插入到点序列  $P\_Q$  中相应的位置,从而得到交集多边形的顶点序列。合并算法的步骤如下。

Step 1:对点序列  $Q\_P$  进行调整,调整的目标是让顶点序列中首顶点成为点序列  $P\_Q$  中的一个顶点。调整的原则是:将首顶点取出来并将其后边的值依次向前移动一个位置,最后再将刚才取出来的顶点插入到末顶点后边。

Step 2:对点序列  $P\_Q$  进行调整,调整目标是使点序列  $P\_Q$  的末顶点与点序列  $Q\_P$  的首顶点相等,调整的原则同 Step 1 中的调整原则。

Step 3:点序列  $Q\_P$  的首顶点删除,后边的顶点向前。

Step 4:判断点序列  $Q\_P$  的首顶点是否包含在点序列  $P\_Q$  的顶点集中,如果不包含,则将首顶点插入到点序列  $P\_Q$  的末顶点后边,并跳回 Step 3。

Step 5:如果点序列  $Q\_P$  的顶点集不为空,则跳回 Step 2。

Step 6:点序列  $P\_Q$  即为多边形  $P, Q$  相交的结果多边形的顶点序列。

按照算法运算,实例合并过程中状态演示如表 1。

表1 顶点序列合并过程演示  
Table 1 Union of ordered vertices

状态	点序列 $P\_Q$	点序列 $Q\_P$
0	$(M_1, P_2, P_3, P_4, M_2)$	$(Q_1, M_1, M_2)$
1	$(M_1, P_2, P_3, P_4, M_2)$	$(M_1, M_2, Q_1)$
2	$(P_2, P_3, P_4, M_2, M_1)$	$(M_1, M_2, Q_1)$
3	$(P_2, P_3, P_4, M_2, M_1)$	$(M_2, Q_1)$
4	$(M_1, P_2, P_3, P_4, M_2)$	$(M_2, Q_1)$
5	$(M_1, P_2, P_3, P_4, M_2)$	$(Q_1)$
6	$(M_1, P_2, P_3, P_4, M_2, Q_1)$	$()$
7	退出。点序列 $P\_Q(M_1, P_2, P_3, P_4, M_2, Q_1)$ 即为结果多边形的顶点序列	

2.1.3 存在的特殊情况及解决方法 如图3所示,多边形  $P(P_1, P_2, P_3, P_4, P_5)$ , 多边形  $Q(Q_1, Q_2, Q_3, Q_4, Q_5, Q_6, Q_7)$ . 如果按前面介绍的求解点序列  $P_Q$ 、 $Q_P$  的算法计算,可以得到点序列  $P_Q(M_1, P_2, P_3, M_2, M_3, P_5, M_4)$ , 点序列  $Q_P(M_1, M_4, M_2, Q_6, M_3)$ .

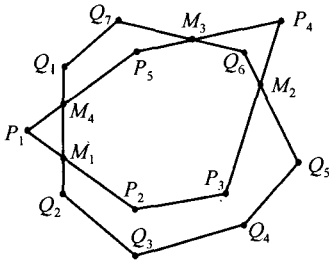


图3 算法中存在的特殊情况  
Fig. 3 Special case of the algorithm

此时,点序列  $Q_P$  的顶点序列中  $M_1$  和  $M_4$  存储顺序违背了逆时针存储的原则,因此在上述算法的基础上对相关步骤进行了如下改进。

(1) 在 Step 1 中定义变量  $number$  初始化为 0, 用于标识一条线段与另一个多边形的交点个数。

(2) 在 Step 4 判断线段  $P_iP_{i+1}$ 、 $Q_jQ_{j+1}$  的关系, 如果存在交点则进行一次  $number++$  操作。

(3) 在 Step 5 后插入一步运算,即判断  $number$  的大小:当  $number > 1$ ,将点序列  $P_Q$  的顶点集的最后  $number$  个顶点按照与顶点  $P_i$  的距离从小到大的顺序重新存储;当  $number \leq 1$  时,直接进入下一步。在下一步的操作中将  $number$  重置为 0。

用改进算法重新计算得到的点序列  $Q_P(M_4, M_1, M_2, Q_6, M_3)$ , 满足了存储顺序的要求。图4为改进后的算法流程图。

## 2.2 凸多边形并集求解算法的实现

凸多边形并集求解算法中多边形顶点的筛选条件为判断顶点与另一个多边形的拓扑关系,如果顶点为外部点,则存储待用。定义点序列  $P_Q$  由多边形  $P$  的顶点序列中包含在多边形  $Q$  外部的顶点和多边形  $P$ 、 $Q$  的交点组成;定义点序列  $Q_P$  由多边形  $Q$  的顶点序列中包含在多边形  $P$  外部的顶点和多边形  $P$ 、 $Q$  的交点组成。将点序列  $P_Q$ 、 $Q_P$  进行合并就可以得出结果多边形的定点序列。

并集求解算法中点序列  $P_Q$  的求解流程图与图4中交集求解算法中点序列  $P_Q$  的计算流程图不同之处:将流程图中(\*)步的判断条件改为“ $P_i$  是

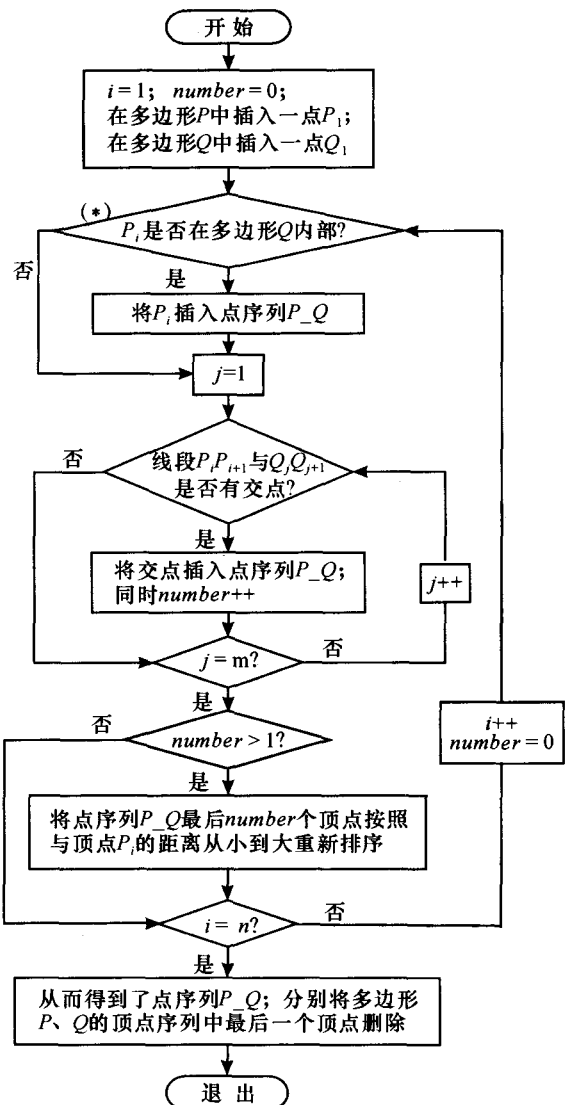


图4 交集求解算法中点序列  $P_Q$  的计算流程图

Fig. 4 Flowing chart of solving  $P_Q$

否在多边形  $Q$  外部”，即如果  $P_i$  在多边形  $Q$  外部时, 则将  $P_i$  点插入到点序列  $P_Q$  中。

如图2, 凸多边形  $P(P_1, P_2, P_3, \dots, P_n)$  和凸多边形  $Q(Q_1, Q_2, Q_3, \dots, Q_n)$ . 通过上面的算法可以得到点序列  $P_Q$  为  $(P_1, M_1, M_2, P_5, P_6)$ ; 同理可以得到点序列  $Q_P$  为  $(M_1, Q_2, Q_3, Q_4, Q_5, Q_6, M_2)$ . 点序列  $P_Q$ 、 $Q_P$  的顶点合并后得到并集多边形的定点序列为  $(P_5, P_6, M_1, Q_2, Q_3, Q_4, Q_5, Q_6, M_2)$ .

图3中的特殊情况也会在求并集的过程中出现, 但是在求交集过程中对特殊情况的处理方法同样在此适用。

凸多边形交、并求解算法的运行效果如图5所示。

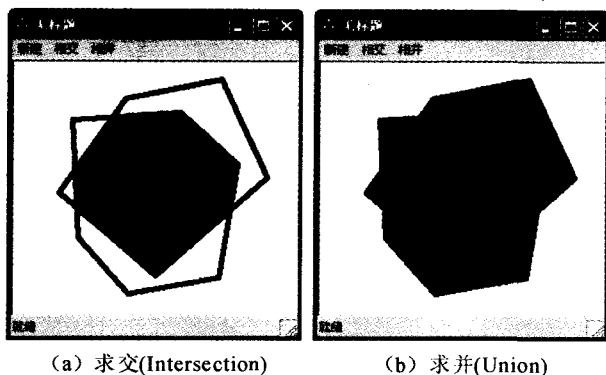


图5 算法的运行结果

Fig. 5 Results of the algorithm

### 2.3 凸多边形交、并求解算法的扩展

本文在对凸多边形的交、并问题研究过程中,对凹多边形的情况作了探讨:

1) 当运算的2个多边形中有1个为凹多边形时,首先利用剖分算法<sup>[3,4]</sup>将凹多边形剖分为1个凸多边形集合,然后将剖分后得到的凸多边形集合的每一个元素与凸多边形进行求交、并求解运算。

2) 当运算的2个多边形都为凹多边形,将2个凹多边形分别剖分为凸多边形集合,然后将剖分后得到的2个凸多边形集合交叉进行交、并求解运算。

图6a中,将凹多边形Q进行剖分,得到图6b,然后将剖分后的多边形1、2、3分别与凸多边形P求交。交集求解的结果由2个部分组成,它们分别是多边形1、2与多边形P求交的结果,如图6c。

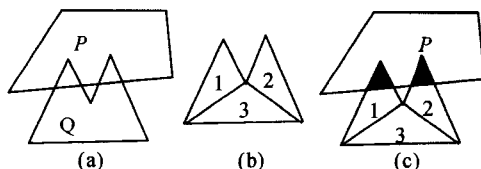


图6 凹多边形的交集运算

Fig. 6 Intersection of the concave polygon

## 3 结 论

(1) 本文提出的凸多边形的交、并求解算法能够较好的解决二维矢量多边形之间的关系计算问题,为GIS应用中几何图形的叠置分析计算提供参考。

(2) 算法采用 Visual C++ 6.0 编程实现,实例证明此算法不仅解决了顶点序列维护困难的问题,而且减少了存储空间,便于快速解决1:N多边形之间的交、并求解问题。

### 参考文献:

- [1] Preparate F, Shamos M. Computational Geometry: An Introduction [M]. Springer - Verlag, 1985.
- [2] 庞明勇, 卢章平. 计算两凸多边形的并集多边形及其面积的计算机算法与实现 [J]. 工程图学学报, 2004 (1): 90 - 94.
- [3] 王正旋. 一个加权剖分简单多边形为凸多边形的算法 [J]. 计算机学报, 1998, 21 (3): 229 - 233.
- [4] 肖忠辉. 简单多边形凸单元剖分的编码算法 [J]. 计算机学报, 1996, 19 (6): 477 - 481.

## Algorithm for Intersection and Union Between Convex Polygons

HUANG Jun-hua<sup>1,2</sup>, YAN Sui-jun<sup>3</sup>, ZHU Xiao-long<sup>3</sup>, LI Jing-wen<sup>3</sup>

(1. School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China;

2. Guangxi Bureau of Land and Resources, Nanning 530023, China;

3. Department of Civil Engineering, Guilin University of Technology, Guilin 541004, China)

**Abstract:** The difficulty of the intersection and union of the convex polygons is maintaining the order of vertices. The algorithm is based on the spatial topological relationships between the vertex and the polygon, and resets different topological vertices in order to solve the problem of the intersection and union of the convex polygons. The algorithm is easy in practice and well solves the 2D's problems of the intersection and union of the convex polygons, while providing the basis of algorithm for calculating the relationship between vector polygons in the application of GIS.

**Key words:** computational geometry; convex polygon; intersection; union