

确定两个任意简单多边形交、并、差的算法

朱雅音<sup>1</sup> 王化文<sup>1</sup> 万 丰<sup>1</sup> 于雷易<sup>2</sup>  
<sup>1</sup>(武汉大学计算机学院 武汉 430072)  
<sup>2</sup>(武汉大学遥感与信息工程学院 武汉 430072)  
(zhuyayin@sina.com)

**摘 要** 提出了把多边形的边分为奇偶边的新思想, 根据输入多边形  $A, B$  之间边的拓扑关系, 划分  $A, B$  边为内边、外边、重叠边 3 种, 揭示  $A, B$  与它们的交、并、差之间边的本质联系, 进而描述了确定任意两个简单多边形交、并、差算法的时间复杂度为  $O((n+m+k)\log(n+m+k))$ , 其中  $n, m$  分别是  $A, B$  的顶点数,  $k$  是两多边形的交点数. 算法建立在数学理论基础之上, 很好地处理了布尔运算的奇异情形, 比如重叠边、边与边相交于边的顶点等情形. 本算法易于编程实现.

**关键词** 多边形; 两个多边形的交、并、差; 计算几何

中图法分类号 TP301.6; TP391

An Algorithm for Determining the Intersection, Union, and Difference of Any Two Polygons

ZHU Ya-Yin<sup>1</sup>, WANG Hua-Wen<sup>1</sup>, WAN Feng<sup>1</sup>, and YU Lei-Yi<sup>2</sup>  
<sup>1</sup>(School of Computer Science, Wuhan University, Wuhan 430072)  
<sup>2</sup>(Institute of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430072)

**Abstract** A new idea about dividing edges of a polygon into odd edges and even edges is presented. Based on the topology relationship between each edge of one polygon and another polygon, edges of both input polygons are divided into three types: interior edges, exterior edges, and overlap edges, and then an algorithm for determining intersection, union, and difference of the two polygons is put forward. The algorithm runs in time  $O((n+m+k)\log(n+m+k))$  in a worst presented case, where  $n$  and  $m$  are the vertex number of the two polygons respectively, and  $k$  is the number of intersection points. Supported by mathematical theorems, the algorithm well resolves the problems in special cases, such as overlapped edges, edges intersection at the vertex of edges, etc. The algorithm is easy to implement.

**Key words** polygon; intersection, union, and difference of two polygons; computational geometry

1 引 言

简单多边形的一般定义是指不自相交的多边形, 可以含 0 个或多个洞. 确定两个任意简单多边形的交并差问题既是计算几何、计算机图形学的基本问题, 也是 GIS 叠加分析的理论基础, 因此, 对这

一问题的研究无论是在理论上, 还是在实践上都有重要意义. 目前, 国内外已经有不少文献介绍确定多边形交并差的算法, 但这些算法大都只能处理凸多边形, 例如 Shamos 算法<sup>[1]</sup> 和 O'Rourke 算法<sup>[2]</sup>. 近来, M. Rivero 提出的算法<sup>[3]</sup> 虽然能处理任意简单多边形的情况, 但其理论复杂, 致使其时间复杂度达到  $O((n+k)(m+k))$ , 其中  $n, m$  分别是两输入

多边形的顶点数,  $k$  是两多边形的交点数. 本文提出的算法基于边的奇偶性质, 根据输入多边形  $A, B$  之间边的拓扑关系, 划分  $A, B$  边为内边、外边、重叠边 3 种, 揭示  $A, B$  与它们的交、并、差之间边的本质联系, 进而描述了确定任意两个简单多边形交、并、差算法. 算法的时间复杂度为  $O((n+m+k)\log(n+m+k))$ , 其中  $n, m, k$  的意义同前. 本算法建立在数学理论基础之上, 很好地处理了布尔运算的奇异情形, 比如重叠边、边与边的相交于边的顶点等情形. 本算法易于编程实现.

## 2 算法的理论基础

为了便于描述, 先定义一些数学符号

$A, B$ : 输入多边形;

$A \cap B, A \cup B, A - B$ :  $A$  与  $B$  的交、并、差;

$\partial A$ : 多边形  $A$  的边的集合, 或  $A$  的边界上点的集合;

$I(A)$ :  $A$  的内域;

$E(A)$ :  $A$  的外域;

$P, Q$ : 点;

$P_x, P_y$ :  $P$  的横、纵坐标;

$e, s$ : 边;

$P \downarrow$ : 过点  $P$  作的垂直向下射线;

$<$ : 点的小于比较符,  $P < Q \Leftrightarrow (P_x < Q_x) \vee (P_x = Q_x \wedge P_y < Q_y)$ ;

$L_p(e)$ :  $e$  的左端点, 即  $e$  两端点中较小的一个;

$R_p(e)$ :  $e$  的右端点, 即  $e$  两端点中较大的一个;

$I(e)$ :  $e$  内点( $e$  上除端点外的其他点)的集合;

$C(P, \partial A)$ : 与  $P \downarrow$  有交点, 且右端点不在  $P \downarrow$  上的  $A$  的边集合. 即  $C(P, \partial A) = \{s \mid s \in \partial A, s \cap P \downarrow \neq \emptyset, R_p(s) \notin P \downarrow\}$ ;

$s_1 \succ_P s_2$ : 边  $s_1, s_2$  在通过  $P$  点的垂直线  $x = P_x$  上的比较, 即  $s_1 \succ_P s_2 \Leftrightarrow (Q_1 \succ Q_2) \vee (Q_1 = Q_2 \wedge K_{s_1} \succ K_{s_2})$ , 其中点  $Q_1, Q_2$  表示边  $s_1, s_2$  与过  $P$  点的垂线的交点,  $K_{s_1}, K_{s_2}$  分别表示  $s_1, s_2$  的斜率;

$\max(C(P, \partial A))$ : 表示  $C(P, \partial A)$  中在  $P$  点处的最大边;

$\vec{e}$ :  $e$  的正方向, 即从  $L_p(e)$  指向  $R_p(e)$  的方向;

$\overleftarrow{e}$ :  $e$  的负方向, 即从  $R_p(e)$  指向  $L_p(e)$  的方向;

$I(A) \blacktriangleleft Left(\vec{e})$ : 表示  $I(A)$  位于  $e$  沿正方向前进时  $e$  的左侧;

$E(A) \blacktriangleleft Right(\overleftarrow{e})$ : 表示  $E(A)$  位于  $e$  沿负方向前进时  $e$  的右侧.

### 2.1 边的奇偶定义及其定理、性质

定义 1. 对于多边形  $A$  的任一条边, 设  $P$  是  $e$  的内点, 如果  $C(P, \partial A)$  有奇数条边, 则称  $e$  是  $A$  的**奇边**, 简记为  $+e$ ; 否则称  $e$  是  $A$  的偶边, 简记为  $-e$ .

例如, 图 1 中  $Q_1 \in I(s_9)$ ,  $C(Q_1, \partial A) = \{s_9, s_1\}$ , 所以  $s_9$  是  $A$  的偶边, 记做  $-s_9$ . 图 1 中其他边的奇偶性均已标出, 请读者自行推理.

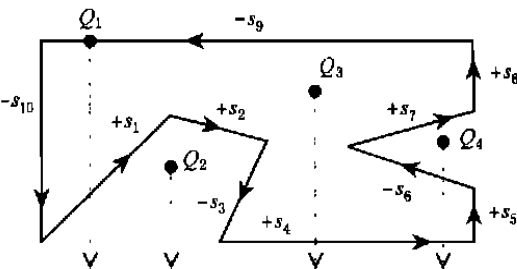


图 1 多边形  $A$  及其边的奇偶性

性质 1. 对任一给定简单多边形  $A$ , 其边的奇偶性与定义 1 中  $P$  点的选择无关.

证明. 设想  $P$  从  $e$  的左端点滑向右端点时, 如果存在着边  $s \in \partial A$  与集合  $C(P, \partial A)$  的关系发生变化(从  $s \in C(P, \partial A)$  到  $s \notin C(P, \partial A)$ , 或者相反), 则变化只能发生在当  $P \downarrow$  经过  $s, s'$  两边端点  $Q$  的瞬间, 可分为 4 种情况讨论:

- ①  $Q = L_p(s) = L_p(s')$ ; 这时有  $C(P, \partial A) = C(P, \partial A) + \{s, s'\}$ , 所以  $e$  的奇偶性不变;
- ②  $Q = R_p(s) = R_p(s')$ ; 这时有  $C(P, \partial A) = C(P, \partial A) - \{s, s'\}$ , 所以  $e$  的奇偶性不变;
- ③  $Q = L_p(s) = R_p(s')$ ; 这时有  $C(P, \partial A) = C(P, \partial A) - \{s'\} + \{s\}$ , 所以  $e$  的奇偶性不变;
- ④  $Q = R_p(s) = L_p(s')$ ; 同③类似. 证毕.

**定理 1.** 对于任意两条边  $s_1, s_2 \in C(P, \partial A)$ , 如果  $s_2$  是在  $P$  点处小于  $s_1$  的最大边, 即  $s_1 \succ_P s_2$ , 且不存在边  $e \in C(P, \partial A)$ , 使得  $s_1 \succ_P e, e \succ_P s_2$  同时成立, 则  $s_1, s_2$  在  $A$  中的**奇偶相异**.

例如图 1 中,  $s_9, s_1 \in C(Q_1, \partial A)$ ,  $s_1$  是在  $Q_1$  点处小于  $s_9$  的最大边, 所以  $s_9, s_1$  奇偶相异.

**定理 2.** 对于任一不在多边形  $A$  边界上的点  $P$ , 如果过点  $P$  所作的垂直向下的射线与多边形  $A$  的相交的**最大边是偶边**, 或**没有与  $A$  的任何边相交**, 则  $P$  在多边形  $A$  的**外域**, 其逆亦然; 如果射线与  $A$  相交的最大边是奇边, 则  $P$  在多边形  $A$  的**内域**.

其逆亦然

定理 2 亦可用数学方式表达为: 对于任一  $P \notin \partial A$ , 有:

①  $\max(C(P, \partial A))$  是偶边或  $C(P, \partial A) = \emptyset \Leftrightarrow P \in E(A)$ ;

②  $\max(C(P, \partial A))$  是奇边  $\Leftrightarrow P \in I(A)$ .

例如图 1 中,  $C(Q_2, \partial A) = \emptyset$ ,  $\max(C(Q_3, \partial A)) = +s_4$ ,  $\max(C(Q_4, \partial A)) = \max(\{-s_6, +s_4\}) = -s_6$ , 所以  $Q_2 \in E(A)$ ,  $Q_3 \in I(A)$ ,  $Q_4 \in E(A)$ .

**定理 3** 边的奇偶性、方向性与多边形内域、外域的联系:

① 如果  $e$  是多边形的奇边, 那么, 当沿着  $e$  的左端点走向右端点时, 多边形的内域位于左手, 外域位于右手, 其逆亦成立

② 如果  $e$  是多边形的偶边, 那么, 当沿着  $e$  的右端点走向左端点时, 多边形的内域位于左手, 外域位于右手, 其逆亦成立

定理 3 亦可用数学方式表达为:

①  $e$  是  $A$  的奇边  $\Leftrightarrow I(A) \Leftarrow \text{Left}(e^{\rightarrow}) \Leftrightarrow E(A) \Leftarrow \text{Right}(e^{\rightarrow})$ ;

②  $e$  是  $A$  的偶边  $\Leftrightarrow I(A) \Leftarrow \text{Left}(e^{\leftarrow}) \Leftrightarrow E(A) \Leftarrow \text{Right}(e^{\leftarrow})$ .

推论 1. 边的奇偶性、方向性与多边形方向性的联系:

① **奇边正方向, 偶边负方向**排列端点  $\Leftrightarrow$  多边形的外接多边形顶点按逆时针排列, 洞按顺时针排列.

② **奇边负方向, 偶边正方向**排列端点  $\Leftrightarrow$  多边形的外接多边形顶点按顺时针排列, 洞按逆时针排列.

例如图 1, 4~6, 8~10 中, 其奇边均为正方向, 偶边为负方向排列, 多边形顶点的排列顺序请读者自行判断.

限于篇幅, 上述定理、性质、推论的证明略去, 读者可结合图形理解

## 2.2 输入多边形边的 3 种基本拓扑类型: 内边、外边、重叠边

任一输入多边形的边, 不妨设为  $B$  的边  $e$ , 相对于另一输入多边形  $A$ , 有且仅有 4 种拓扑关系, 如图 2 所示

① **内边**:  $e$  的所有内点均位于  $A$  的内域,  $\forall P \in I(e) \Rightarrow P \in I(A)$ . 如边  $s_1$  所示

② **外边**:  $e$  的所有内点均位于  $A$  的外域,  $\forall P \in I(e) \Rightarrow P \in E(A)$ . 如边  $s_2$  所示

③ **重叠边**:  $\exists s \in \partial A$ ,  $\forall P, P \in e \Leftrightarrow P \in s$ . 如边  $s_3$  所示

上述 3 种边统称为简单边

④ **复杂边**: 不属于简单边的其他边, 如边  $P_1P_2$  所示

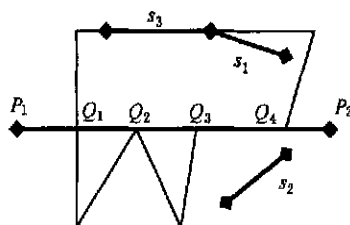


图 2 边与另一多边形拓扑关系

注意到复杂边的特点是: 与另一多边形相交于内点. 由于按交点大小顺序, 可以把复杂边分解成若干段简单边. 例如边  $P_1P_2$  可以分解为  $P_1Q_1$ ,  $Q_1Q_2$ ,  $Q_2Q_3$ ,  $Q_3Q_4$ ,  $Q_4P_2$  五段简单边. 因此复杂边的分解可以归结为确定两多边形交点问题, 解决该问题的一个常用方法是**平面扫描算法**, 关于该算法的详细描述已超出了本文的范围, 请参考文献 [4, 5].

综上所述, 根据一多边形的边相对于另一多边形的拓扑关系,  $A, B$  边可分为 3 种基本拓扑类型: 内边、外边、重叠边

## 3 $A, B$ 边的奇偶性以及拓扑类型的确定算法

同分解复杂边一样,  $A, B$  边的奇偶性和拓扑类型也可以在平面扫描过程中确定. 平面扫描算法用到两个基本的数据结构: 事件点表  $eps$  和扫描线状态  $sls$ .  $eps$  是按从小到大的顺序存放的  $A, B$  的顶点和交点序列.  $sls$  是在事件点  $Q$  处按比较符 ' $>$ ' 建立的,  $A, B$  边的序列是以平衡二叉树实现的  $sls$ , 其结点数据结构可设计如下:

```
typedef struct {
    int A-Point[2], B-Point[2]; /* 分别存放
    A, B 边的两端点 */
    bool A-OddEven, B-OddEven; /* 分别标识
    A, B 边的奇偶性 */
    int A-Type, B-Type; /* 分别标识 A, B 边
    的 3 种拓扑类型: 内边、外边、重叠边 */
    slsEdgeNode *parent, leftSon, rightSon;
    /* 分别指向父结点和左、右子结点 */
} slsEdgeNode;
```

算法规定重叠边存放在相同结点. 函数  $Bellow-SamePgn(e, sls)$ : 返回  $sls$  中小于  $e$  的最大边, 如果这种边有两条: 重叠边, 则返回与  $e$  属于同一多边形的

那条边 例如图 3 中, 在事件点  $Q=L_p(s_2)$  处,  $sls=\{e_1, e_5, s_2, e_4, e_2\}$  (注意, 这里  $R_p(s_1)=Q$ , 所以  $s_1$  已从  $Q$  的  $sls$  中删去), 其中重叠边:  $s_2=Qe_4$ .  $Bellow-SamePgn(e_1, sls)=e_5$ ,  $Bellow-SamePgn(e_5, sls)=e_4$ ,  $Bellow-SamePgn(s_2, sls)=Bellow-SamePgn(e_4, sls)=e_2$ ,  $Bellow-SamePgn(e_2, sls)$  不存在 函数  $Bellow-DifferentPgn(e, sls)$  的值与  $Bellow-SamePgn$  相等, 只是如果小于  $e$  的最大边是重叠边, 则返回与  $e$  属于不同多边形的那条边 例如  $Bellow-SamePgn(e_5, sls)=s_2$ , 而不是  $e_4$ .

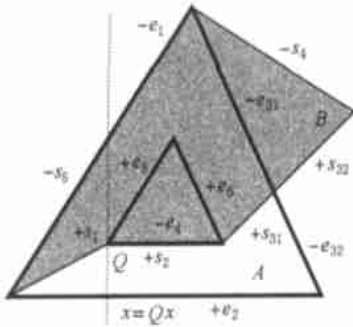


图 3 输入多边形  $A, B$

**算法 1** 在平面扫描算法的  $sls$  中确定边  $e$  的奇偶性:

- ①  $s=Bellow-SamePgn(e, sls)$ ;
- ② if  $s$  不存在 then  $e$  是奇边;
- ③ else if  $s$  和  $e$  属于同一个多边形 then  $e$  的奇偶性与  $s$  的奇偶性相反;
- ④ else if  $s$  为内边 then  $e$  为偶边;
- ⑤ else  $e$  为奇边

说明: 行②: 如果  $s$  不存在, 则  $e$  是  $sls$  中最小的一条边, 不失一般性, 设  $e \in \partial A$ ,  $P \in I(e)$ , 则  $C(P, \partial A)=\{e\}$ , 所以  $e$  是奇边 行③:  $e$  和  $s$  属于同一多边形,  $s$  是  $e$  在事件点处小于  $e$  的最大边, 根据是定理 1,  $e, s$  奇偶相异 行④隐含条件:  $s$  和  $e$  属于不同多边形 不妨设  $e \in \partial A$ ,  $s \in \partial B$ , 对于任一点  $P \in I(e)$ ,  $s$  为内边  $\Rightarrow P \in I(A)$ ; 又根据定理 2,  $P \in I(A) \Rightarrow \max(C(P, \partial A))$  是奇边;  $e$  与  $\max(C(P, \partial A))$  属于同一多边形, 且  $\max(C(P, \partial A))$  是在事件点处小于  $e$  的最大边, 根据定理 1,  $e$  与  $\max(C(P, \partial A))$  奇偶相异, 所以  $e$  为偶边 行⑤隐含条件:  $s$  和  $e$  属于不同多边形,  $s$  为外边 其判断根据同行④.

**算法 2** 在平面扫描算法的  $sls$  中确定边  $e$  的拓扑类型:

- ① if  $sls$  中有边与  $e$  大小相等 then  $e$  是重叠边;
- ② else {

- ③  $s=Bellow-DifferentPgn(e, sls)$ ;
- ④ if  $s$  不存在 then  $e$  是外边;
- ⑤ else if  $s$  和  $e$  属于同一多边形 then  $e$  与  $s$  的拓扑类型相同;
- ⑥ else if  $s$  是偶边 then  $e$  是外边;
- ⑦ else  $e$  是内边;
- ⑧ }

证明 行④: 如果  $s$  不存在, 则  $e$  是  $sls$  中最小的一条边 不妨设  $e \in \partial A$ , 则对于任一  $P \in I(e)$ , 有  $C(P, \partial B)=\emptyset$ . 又根据定理 2,  $C(P, \partial B)=\emptyset \Rightarrow P \in E(B)$ , 所以  $e$  是外边 行⑤:  $s$  和  $e$  是  $sls$  中属于同一输入多边形的紧挨着的两邻边, 它们相对于另一输入多边形的拓扑类型相同 行⑥隐含条件:  $s$  和  $e$  属于不同多边形  $\max(C(P, \partial B))=s$ , 不妨设  $e \in \partial A$ ,  $s \in \partial B$ , 对于任一  $P \in I(e)$ , 根据定理 2,  $\max(C(P, \partial B))$  是偶边  $\Rightarrow P \in E(B)$ , 已知  $s$  是偶边, 所以  $e$  是外边 行⑦隐含条件:  $s$  和  $e$  属于不同多边形,  $s$  为奇边 其判断根据同行⑥.

#### 4 $A \cap B, A \cup B, A - B$ 边与 $A, B$ 边的内在联系

标记  $Odd(e, A)$  表示边  $e$  在  $A$  中的奇偶性,  $\neg Odd(e, A)$  的值与  $Odd(e, A)$  相反.

**定理 4** 构成多边形  $A \cap B, A \cup B$  的边在  $A \cap B, A \cup B$  中的奇偶性, 与该边在原来所属的多边形  $A, B$  的奇偶性相同; 对于构成  $A - B$  的边, 如果该边原来所属的多边形是  $A$ , 则其在  $A - B$  中的奇偶性与在  $A$  中的奇偶性相同 如果该边原来所属的多边形是  $B$ , 则其在  $A - B$  中的奇偶性与在  $B$  中的奇偶性相异

定理 4 揭示了  $A \cap B, A \cup B, A - B$  中边的奇偶性与  $A, B$  中边的奇偶性的本质联系, 其亦可形式化描述为:

- ①  $\forall e, e \in \partial(A \cap B), e \in \partial A \Rightarrow Odd(e, A \cap B) = Odd(e, A)$ ;
- ②  $\forall e, e \in \partial(A \cap B), e \in \partial B \Rightarrow Odd(e, A \cap B) = Odd(e, B)$ ;
- ③  $\forall e, e \in \partial(A \cup B), e \in \partial A \Rightarrow Odd(e, A \cup B) = Odd(e, A)$ ;
- ④  $\forall e, e \in \partial(A \cup B), e \in \partial B \Rightarrow Odd(e, A \cup B) = Odd(e, B)$ ;
- ⑤  $\forall e, e \in \partial(A - B), e \in \partial A \Rightarrow Odd(e, A - B) = Odd(e, A)$ ;
- ⑥  $\forall e, e \in \partial(A - B), e \in \partial B \Rightarrow Odd(e, A - B) = \neg Odd(e, B)$

$= \lceil \text{Odd}(e, B)$ .

限于篇幅,下面只证明第 1 点,其他 5 点同理可证

证明. 已知  $e \in \partial(A \cap B)$ , 不失一般性, 设  $e$  是  $A \cap B$  奇边 根据定理 3,  $I(A \cap B) \blacktriangleleft \text{Left}(e^+)$ , 又因为  $I(A \cap B) \subseteq I(A)$ , 所以  $I(A) \blacktriangleleft \text{Left}(e^+)$ , 又因为  $e \in \partial A$ , 根据定理 3,  $e$  是  $A$  的奇边. 即  $\text{Odd}(e, A \cap B) = \text{Odd}(e, A)$ . 证毕

**定理 5** 构成多边形  $A \cap B$  的边, 由且仅由输入多边形  $A$  或  $B$  中的以下 2 种边组成: ①  $A$  或  $B$  的内边, ②  $A, B$  的重叠边, 且该重叠边在  $A, B$  的奇偶性相同; 构成多边形  $A \cup B$  的边, 由且仅由  $A, B$  中的以下 2 种边组成: ①  $A$  或  $B$  的外边, ②  $A, B$  的重叠边, 且该重叠边在  $A, B$  的奇偶性相同; 构成多边形  $A - B$  的边, 由且仅由  $A, B$  中的以下 3 种边组成: ①  $A$  的外边, ②  $B$  的内边, ③  $A, B$  的重叠边, 且该重叠边在  $A, B$  的奇偶性相异

定理 5 揭示了  $A \cap B, A \cup B, A - B$  的边与  $A, B$  边的本质联系, 其亦可形式化描述为:

- ①  $\partial(A \cap B) = \{e \mid e \text{ 是 } A \text{ 或 } B \text{ 的内边} \vee ((e \text{ 是重叠边}) \wedge (\text{Odd}(e, A) = \text{Odd}(e, B)))\}$ ;
- ②  $\partial(A \cup B) = \{e \mid e \text{ 是 } A \text{ 或 } B \text{ 的外边} \vee ((e \text{ 是重叠边}) \wedge (\text{Odd}(e, A) = \text{Odd}(e, B)))\}$ ;
- ③  $\partial(A - B) = \{e \mid (e \text{ 是 } A \text{ 的外边}) \vee (e \text{ 是 } B \text{ 的内边}) \vee (e \text{ 是重叠边} \wedge (\text{Odd}(e, A) \neq \text{Odd}(e, B)))\}$ .

限于篇幅,下面只证明形式化描述中的第 1 点,其他两点同理可证

证明  
因为  $A, B$  边有且仅有 3 种基本拓扑类型: 内边、外边、重叠边. 对于内边  $\in \partial(A \cap B)$ , 外边  $\notin \partial(A \cap B)$ , 容易根据定义证明. 下面只分析重叠边的情况, 可分为两步证明:  
第 1 步, 证明:  $\forall e, e$  是重叠边,  $\text{Odd}(e, A) = \text{Odd}(e, B) \Rightarrow e \in \partial(A \cap B)$ .  
已知  $e$  是重叠边,  $\text{Odd}(e, A) = \text{Odd}(e, B)$ , 不失一般性, 设  $e$  是  $A, B$  的奇边, 根据定理 3,  $I(A) \blacktriangleleft \text{Left}(e^+), I(B) \blacktriangleleft \text{Left}(e^+)$ , 又因为  $I(A \cap B) = I(A) \cap I(B)$ , 所以  $I(A \cap B) \blacktriangleleft \text{Left}(e^+)$ . 同理可证:  $E(A \cap B) \blacktriangleleft \text{Right}(e^+)$ . 所以  $e$  是  $A \cap B$  内域、外域的分界线, 即  $e \in \partial(A \cap B)$ ;

第 2 步, 证明:  $\forall e, e$  是重叠边,  $e \in \partial(A \cap B) \Rightarrow \text{Odd}(e, A) = \text{Odd}(e, B)$ ;

已知  $e$  是重叠边,  $e \in \partial(A \cap B)$ , 根据定理 4 第

①, ② 两点, 可得  $\text{Odd}(e, A \cap B) = \text{Odd}(e, A)$ ,  $\text{Odd}(e, A \cap B) = \text{Odd}(e, B)$ , 所以  $\text{Odd}(e, A) = \text{Odd}(e, B)$ . 证毕

例 1. 图 3 中,  $\partial A = \{-e_1, +e_2, -e_{31}, -e_{32}, -e_4, +e_5, +e_6\}$ ,  $\partial B = \{+s_1, +s_2, +s_{31}, +s_{32}, -s_4, -s_5\}$ , 其中  $+e_2, -e_{32}, +s_{32}, -s_4$  是外边,  $-e_1, -s_5$  和  $-e_4, +s_2$  是两对重叠边, 其余为内边

根据定理 5 有:  
 $\partial(A \cap B) = \{-e_1, +s_1, +e_5, +e_6, +s_{31}, -e_{31}\}$ , 如图 4 所示:

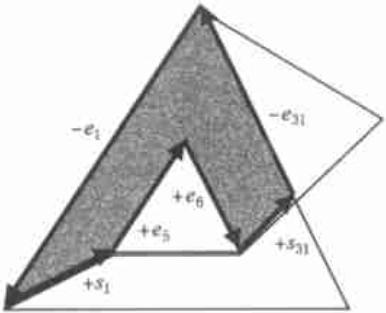


图 4 输出多边形  $A \cap B$

$\partial(A \cup B) = \{-e_1, +e_2, -e_{32}, +s_{32}, -s_4\}$ , 如图 5 所示:

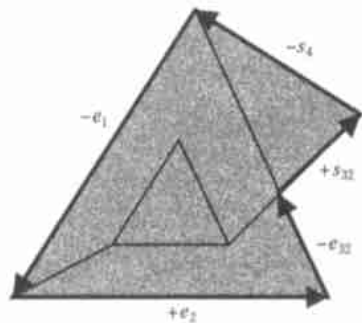


图 5 输出多边形  $A \cup B$

$\partial(A - B) = \{+e_2, -e_{32}, -s_{31}, -e_4, -s_1\}$ , 如图 6 所示:

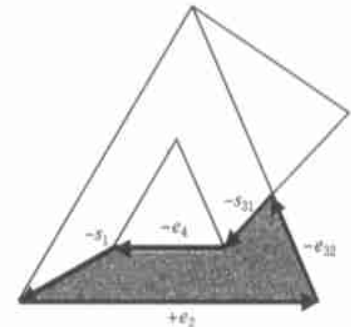


图 6 输出多边形  $A - B$

例 2 图 7 中,  $\partial A = \{-e_1, +e_2, +e_3, -e_{43}, -e_{42}, -e_{41}, -e_5, -e_6, +e_{71}, +e_{72}, +e_{73}, +e_{74}, +e_{75},$



$-e_8\}$ ,  $\partial B = \{+s_1, +s_{21}, +s_{22}, +s_{23}, -s_3, -s_{43}, -s_{42}, -s_{41}, -s_{52}, -s_{51}, +s_6, -s_{72}, -s_{71}\}$ , 其中  $-e_{42}, +e_{72}, +e_{74}, +s_{22}, -s_{42}, -s_{51}, +s_6, -s_{72}$  是内边, 其余均为外边

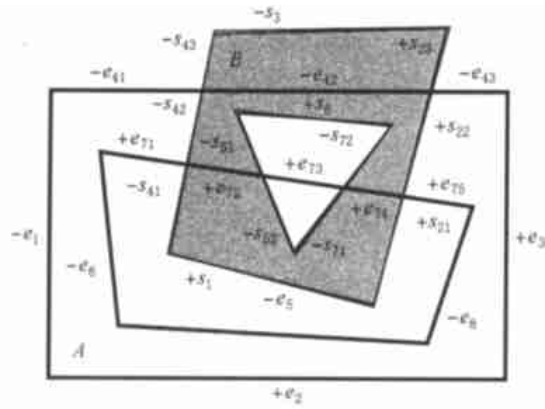


图 7 输入多边形  $A, B$

根据定理 5 有:  
 $\partial(A \cap B) = \{+e_{72}, -s_{51}, +s_6, -s_{72}, +e_{74}, +s_{22}, -e_{42}, -s_{42}\}$ , 如图 8 所示:

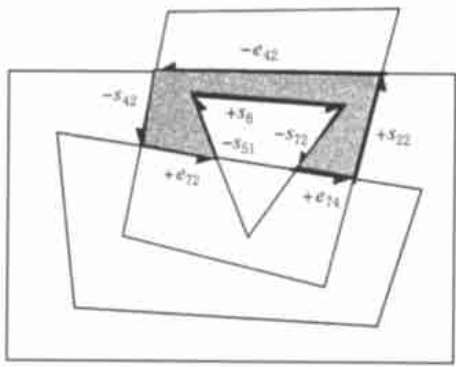


图 8 输出多边形  $A \cap B$

$\partial(A \cup B) = \{-e_1, +e_2, +e_3, -e_{43}, +s_{23}, -s_3, -s_{43}, -e_{41}, -e_5, -e_6, +e_{71}, -s_{41}, +s_1, +s_{21}, +e_{75}, -e_8, -s_{52}, +e_{73}, -s_{71}\}$ , 如图 9 所示:

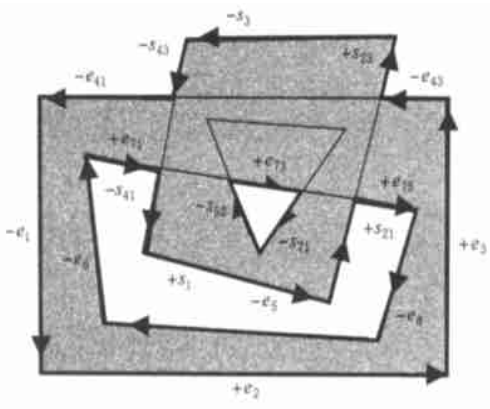


图 9 输出多边形  $A \cup B$

$\partial(A - B) = \{-e_1, +e_2, +e_3, -e_{43}, -s_{22}, +e_{75}, -e_8, -e_5, -e_6, +e_{71}, +s_{42}, -e_{41}, +s_5, +e_{73}, +s_{72}, -s_6\}$ , 如图 10 所示:

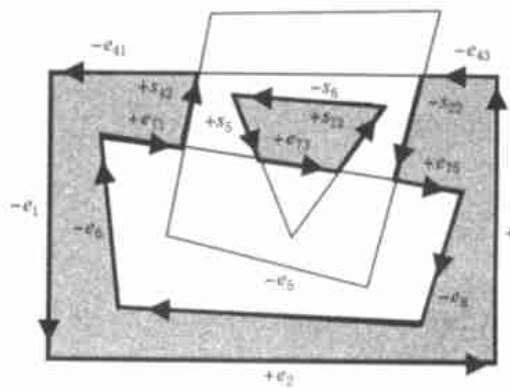


图 10 输出多边形  $A - B$

### 5 确定 $A \cap B, A \cup B, A - B$ 的算法描述

算法 3. 确定  $A \cap B, A \cup B, A - B$ .

Setp1. 在平面扫描过程中, 计算  $A, B$  的交点 (包括切点), 分解复杂边为简单边, 同时根据算法 1, 2, 确定  $A, B$  边的奇偶性及其拓扑类型, 并记录在数据结构 **simpleEdge** 中;

Setp2. 针对多边形交、并、差的具体计算特点, 分别根据算法 4 ~ 6 进行边的跟踪, 输出构成  $A \cap B, A \cup B, A - B$  的中间多边形;

Setp3. 依次构造各中间多边形的 *Border*, 确定中间多边形的方向性 根据推论 1, 判断其是洞还是外接多边形;

Setp4. 判断洞 *Border* 与外接多边形 *Border* 的包含被包含关系, 确定洞归属于哪个外接多边形, 进而确定  $A \cap B, A \cup B, A - B$ .

其中, 存储  $A, B$  边信息的数据结构 *simpleEdge* 可设计如下:

```
typedef struct {
    int pgn;
    /* 标记边是属于多边形 A、还是多边形 B */
    int leftPoint, rightPoint;
    /* 标记边的左、右端点 */
    bool visited;
    /* 标记边是否已访问过, 初始值为 false */
    int type;
    /* 标记边的拓扑类型: 内边、外边、重叠边 */
    bool OddEven;
    /* 标记边的奇偶性 */
    simpleEdge * lp_AdjEg, * rp_AdjEg; /* 指向连接左、右邻边的边 */
};
```

```
右端点的本输入多边形的邻边的指针 */
simpleEdge * lp-otherPgn-AdjEg[ 2], rp-otherPgn-
AdjEg[ 2]; /* 指向连接左、右端点的另一输入多边
形的两邻边的指针 */
} simpleEdge.
```

算法 4. 进行边的选择和跟踪, 输出构成  $A \cap B$  的中间多边形.

- ① for each  $e$  of  $A, B$  的边 {
- ② if  $e$  没有访问过  $\wedge (e$  是内边  $\vee ((e$  是重叠边)  $\wedge$   
 $(Odd(e, A) = Odd(e, B))))$  then
- ③ do{
- ④ 标记  $e$  为已访问; 如果  $e$  是重叠边, 还要标记  
其重叠边  $e'$  为已访问;
- ⑤ if  $e$  是奇边 then {
- ⑥ 把  $e$  的左端点添加到顶点链表中;
- ⑦  $P \leftarrow e$  的右端点;
- ⑧ } else { // 隐含条件  $e$  是偶边
- ⑨ 把  $e$  的右端点添加到顶点链表中;
- ⑩  $P \leftarrow e$  的左端点;
- ⑪ }
- ⑫ if (在连接  $P$  的另一输入多边形的邻边中, 存  
在着满足条件的内边  $s$ ;
- ⑬  $s$  是奇边且  $P = Lp(s)$ , 或者  $s$  是偶边  
且  $P = Rp(s)$ ) then  $e \leftarrow s$ ;
- ⑭ else  $e \leftarrow$  连接  $P$  的本输入多边形的邻边;
- ⑮ } until  $P$  与顶点链表中的第 1 个顶点相等;
- ⑯ 输出顶点链表中顶点组成的中间多边形, 并清空  
顶点链表;
- ⑰ } //end if
- ⑱ //end for

说明: 算法 4 的第①行遍历  $A, B$  的每条边, 第②行是根据定理 5 选择符合构成  $A \cap B$  条件的没有被访问过的边. 第⑤~⑪行, 确保奇边按正方向, 偶边按负方向排列端点, 从而保证外接多边形顶点按逆时针排列, 洞按顺时针排列. 第⑮行判断中间多边形是否闭合.

第⑫~⑭行根据定理 5 和推论 1, 在与  $P$  点相连的  $A, B$  邻边中选择下一条边, 注意到可能存在着两条(可以证明至多只有两条, 至少存在一条)符合条件的邻边, 这时需要作出选择, 如果选择不当可能导致输出多边形不是简单多边形. 如图 11 中, 设  $+s_1$  的左端点已添加到顶点链表,  $P$  为  $+s_1$  的右端点, 则作为  $+s_1$  的下一条边, 在  $P$  点处有两条满足条件的邻边可供选择:  $+e_2$  和  $-s_2$ . 算法 4 选择了  $+e_2$ , 使  $A \cap B$  由两个简单多边形  $(s_1, e_2, e_3), (s_2, e_5, e_1)$  构成. 如果在  $P$  点处选择边  $-s_2$ , 就会导致  $A \cap B$

由一个非简单多边形  $(s_1, e_2, e_3, s_2, e_5, e_1)$  构成. 同理在算法 5, 6 中, 也存在这种选择问题. 如图 12 所示,  $A \cup B$  可表示为两个简单多边形:  $(e_1, e_{21}, e_{22}, e_3), (s_1, s_2, s_3)$ , 也可以表示为一个非简单多边形:  $(e_1, e_{21}, s_1, s_2, s_3, e_{22}, e_3)$ . 如图 13 所示,  $A - B$  可表示为 3 个简单多边形:  $(e_{12}, e_{21}, s_1), (e_{22}, e_{32}, s_2), (e_{11}, s_3, e_{31})$ , 也可以表示为一个含洞的非简单多边形, 其外接多边形为  $(e_{11}, e_{12}, e_{21}, e_{22}, e_{32}, e_{31})$ , 洞为  $(s_1, s_2, s_3)$ . 算法 5, 6 选择了简单多边形的表示方式.

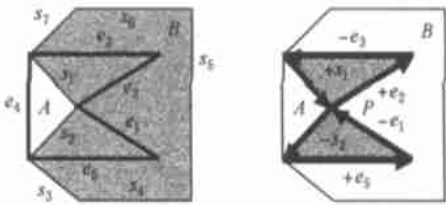


图 11 多边形  $A, B$  及  $A \cap B$

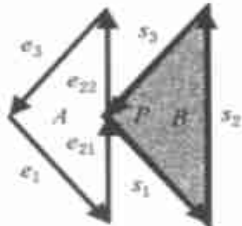


图 12 多边形  $A \cup B$

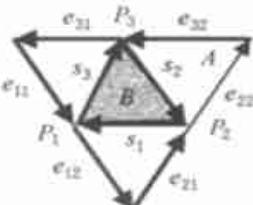


图 13 多边形  $A - B$

算法 5, 6 的结构与算法 4 相同, 限于篇幅, 不再作算法说明.

算法 5. 进行边的选择和跟踪, 输出构成  $A \cup B$  的中间多边形.

算法 5 与算法 4 基本相同, 除第②, ⑫, ⑬用以下 3 行代替外:

- ② if  $e$  没有访问过  $\wedge (e$  是外边  $\vee ((e$  是重叠边)  $\wedge$   
 $(Odd(e, A) = Odd(e, B))))$  then {
- ⑫ if 连接  $P$  的本输入多边形的邻边  $s$  是外边 then  $e \leftarrow s$ ;
- ⑬ else if 连接  $P$  的另一输入多边形的邻边中, 存在着  
外边  $s'$  then  $e \leftarrow s'$ .

算法 6. 进行边的选择和跟踪, 输出构成  $A - B$  的中间多边形.

算法 6 与算法 4 除第②~⑬行不同外, 其余相同

- ② if  $e$  没有访问过  $\wedge ((e \text{ 是 } A \text{ 的外边}) \vee (e \text{ 是 } B \text{ 的内边}) \vee (e \text{ 是重叠边} \wedge (Odd(e, A) = \neg Odd(e, B))))$  then {
- do {
- ③ 记  $e$  为已访问; 如果  $e$  是重叠边, 还要标记其重叠边  $e'$  为已访问;
- ④ if  $e$  是  $A$  的奇边或是  $B$  的偶边 then {
- ⑤ 把  $e$  的左端点添加到顶点链表中;
- ⑥  $P \leftarrow e$  的右端点;
- ⑦ } else { //隐含条件:  $e$  是  $A$  的偶边或是  $B$  的奇边
- ⑧ 把  $e$  的右端点添加到顶点链表中;
- ⑨  $P \leftarrow e$  的左端点;
- }
- ⑩ if  $e$  是  $A$  的边, 与  $P$  相连的属于  $B$  的邻边中, 存在着满足条件的内边  $s$ ;  $s$  是  $B$  的奇边且  $P$  是  $s$  的右端点, 或者  $s$  是偶边且  $P$  是  $s$  的左端点
- ⑪ then  $e \leftarrow s$ ;
- ⑫ else if  $e$  是  $B$  的边, 与  $P$  相连的属于  $A$  的邻边中, 存在着满足条件的外边  $s'$ ;  $s'$  是奇边且  $P$  是  $s'$  的左端点, 或  $s'$  是偶边且  $P$  是  $s'$  的右端点
- ⑬ then  $e \leftarrow s'$ .

## 6 算法分析

本文建立在数学基础上, 通过一系列的定理和推论, 揭示了  $A, B$  与  $A \cap B, A \cup B, A - B$  之间边的本质联系, 对算法理论中较难理解的部分给出了证明, 保证了算法的正确性

算法 1, 2 可在常数时间内完成 算法 3 的步①可在一次平面扫描过程中完成, 其最坏情况下的时间复杂度为  $O((n+m+k)\log(n+m+k))^{[4,9]}$ ; 步②, 即算法 4~6 至多遍历  $A, B$  每条边 2 次, 其时间复杂度为  $O(n+m+k)$ ; 步③需要遍历中间多边形的每条边, 因为这些边的总数不会超过  $A, B$  边的总数, 所以其时间复杂度亦为  $O(n+m+k)$ ; 步④的时间复杂度为  $O(d \times f)$ , 其中  $d$  和  $f$  分别表示步②输出的中间多边形中, 洞和外接多边形的个数 综上所述, 算法的时间复杂度为  $O((n+m+k)\log(n+m+k) + d \times f)$ , 考虑到  $d \times f$  一般远小于  $(n+m+k)\log(n+m+k)$ , 且步④没有乘除运算, 所以算法的时间复杂度可以简化为  $O((n+m+k)\log(n+m+k))$ .

本算法对输入多边形的方向性不作要求, 输出多边形中外接多边形按逆时针排列, 洞按顺时针排列, 符合一般工程约定 算法很好地处理了多边形布尔运算的奇异情形(边界情形), 比如重叠边、边与

边交于边的顶点等情形, 从而具有很好的鲁棒性. 算法结构清晰, 易于编程实现. 本算法经过实践检验, 效果良好. 算法的主要思想对确定两个多面体的交、并、差问题亦有参考价值

## 参 考 文 献

- 1 F P Preparata, M I Shamos. Computational Geometry: An Introduction. Berlin: Springer, 1985
- 2 J O'Rourke. Computational Geometry in C. Cambridge: Cambridge University Press, 1985
- 3 M Rivero, F R Feito. Boolean operations on general planar polygons. Computers & Graphics, 2000, 24(6): 881~898
- 4 Borut Zalik. Two efficient algorithms for determining intersection points between simple polygons. Computer & Geoscience, 2000, 26(2): 137~151
- 5 周培德, 卢开澄. 计算几何. 北京: 清华大学出版社, 2000 (Zhou Peide, Lu Kaicheng. Computational Geometry (in Chinese). Beijing: Tsinghua University Press, 2000)
- 6 B Zalik. Merging a set of polygons. Computers & Graphics, 2001, 25(1): 77~88
- 7 王文成, 吴恩华. 判断检测点是否是多边形或多面体内的新方法. 软件学报, 2000, 11(12): 1614~1619 (Wang Wencheng, Wu Enhua. A new method deciding whether a point is in a polygon or a polyhedron. Journal of Software(in Chinese), 2000, 11(12): 1614~1619)



朱雅音 女, 1971 年生, 硕士研究生, 主要研究方向为计算几何、软件工程、多媒体, 神经网络



王化文 男, 1945 年生, 教授, 主要从事 CIMS、多媒体、系统集成等方面的研究



万 丰 男, 1972 年生, 硕士研究生, 主要研究方向为计算几何、空间运算、空间查询、GIS



于雷易 男, 1975 年生, 博士研究生, 主要研究方向为地理信息系统