Weiler Athenton++: An improved Algorithm of the Weiler Athenton

Xin Guo*
Department of Computer Science
Zhejiang University
Hangzhou, Zhejiang 310058
xinguozju@gmail.com

Abstract

计算任意平面多边形（凸或者凹）的布尔运算（交、并、差）是最基础的平面几何问题之一。Weiler Atherton 算法是计算任意多边形布尔运算的经典算法之一，它可以对任意两个多边形进行“裁剪”操作，进而计算其交、并、差。由于其算法的简易性和通用性，现有大量算法（例如三维消除隐）均将 Weiler Atherton 作为其中的子算法之一，因而 Weiler Atherton 算法也是计算几何领域最为基础和重要的算法之一。传统的 Weiler Atherton 算法只对一般情形的多边形布尔关系进行处理，其无法从理论上对极端情况（例如存在任意边、顶点重合等）的多边形布尔关系进行运算，因而在实际算法实现的时候需要进行大量的 if-else 辅助情况判断。本文针对这一情况，提出了 Weiler Atherton 算法的改进算法 Weiler Athertion++。Weiler Athertion++ 扩展了原 Weiler Athertion 算法中对边及“出点”、“入点”的定义，并详细给出了“出点”和“入点”的判定算法，从而能够处理任意情形的多边形布尔关系。

1 Problem Definition

给定任意平面多边形 S 和任意平面多边形 C，计算交：$S \cap C$，并：$S \cup C$，差：$S - C$

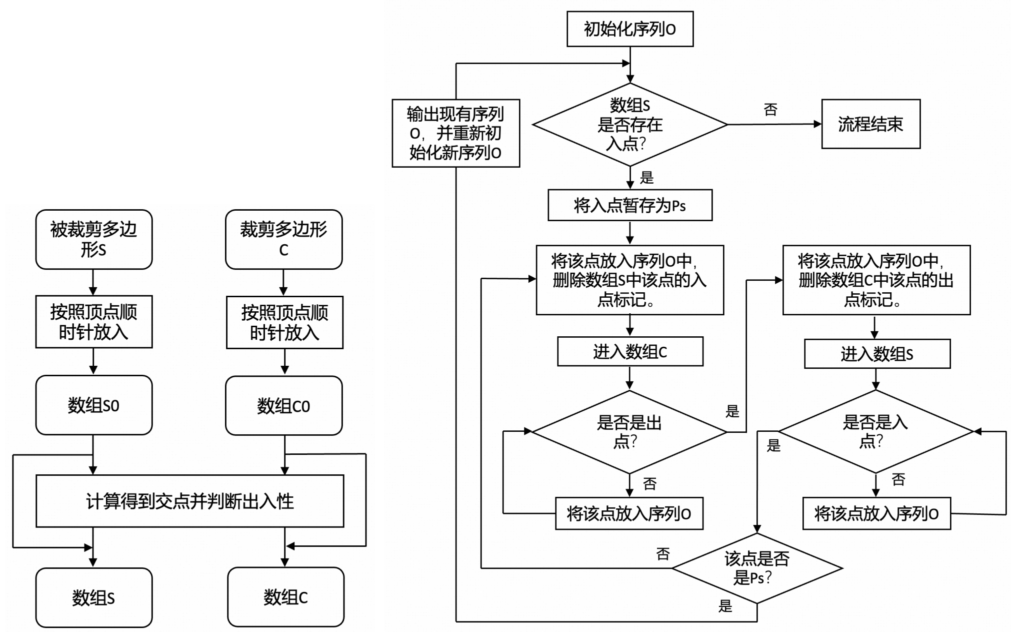
2 Classical Weiler Athenton Algorithm

2.1 Algorithm Description

如图1所示，阐述经典的 Weiler Athenton[1] 算法如下：

*https://sites.google.com/view/xinguo

38th Conference on Neural Information Processing Systems (NeurIPS 2024).



(a) Initiation　　(b) Framework

图 1: Classic Weiler Athenton Algorithm

(1) 顺时针输入被裁剪多边形 S 顶点序列放入数组 S0 中;

(2) 顺时针输入裁剪窗口 C 顶点序列放入数组 C0 中;

(3) 求出被裁剪多边形和裁剪窗口相交的所有交点，并给每个交点打上“入”、“出”标记。然后将交点按顺序插入序列得到新的顶点序列，并放入数组 S 中；同样也将交点按顺序插入序列得到新的顶点序列，放入数组 C 中;

(4) 初始化输出数组 O，令数组 O 为空。接着从数组 S 中寻找“入点”。如果“入点”没找到，程序结束;

(5) 如果找到“入”点，则将“入”点记为 Ps 中暂存;

(6) 将“入”点录入到输出数组 O 中。并从数组 S 中将该“入”点的“入”点标记删去;

(7) 沿数组 S 顺序取顶点：如果顶点不是“出点”，则将顶点录入到输出数组 O 中，流程转第 7 步。否则，流程转第 8 步;

(8) 沿数组 C 顺序取顶点：如果顶点不是“入点”，则将顶点录入到输出数组 O 中，流程转第 8 步。否则，流程转第 9 步;

(9) 如果顶点不等于起始点 Ps，流程转第 6 步，继续跟踪数组 S。否则，将数组 O 输出; 流程转第 4 步，寻找可能存在的分裂多边形。算法在第 4 步：满足“入”点没找到的条件时，算法结束。

2.2 由求“交”扩展为“交并补”

基本假设：多边形 S 不包含边界，多边形 C 包含边界。

具体求交并差时的调整如下：

1）交：构建 S 链，构建 C 链

2）并：S 链出入交换，C 链出入交换

3）差：S 链出入交换，C 链反向（包含出入交换）

经典 Weiler Atherton “裁切”算法中的“入点”和“出点”定义较为简单：“入”点为被裁剪多边形由此点进入裁剪窗口；“出点”为被裁剪多边形由此点离开裁剪窗口。按照上述定义，并且执行经典 Weiler Atherton “裁切”算法，我们可以得到任意平面多边形 S 和任意平面多边形 C 的交：$S\bigcap C$。

其次，为了进一步推导得到多边形的并、差算法，我们需要对算法中的数组 S 和数组 C 进行调整如下：

1）$S\bigcup C$：将 S 数组中的“出点”标记改为“入点”标记，“入点”标记改为“出点”标记；将 C 数组中的“出点”标记改为“入点”标记，“入点”标记改为“出点”标记

2）$S-C$：将 S 数组中的“出点”标记改为“入点”标记，“入点”标记改为“出点”标记；将 C 数组中的全部点的顺序反向（从而“出”“入”标记也反向）。

2.3 Examples of the classic Weiler Atherton algorithm

2.3.1 正例（求交）：求 $ABCD\bigcap EFGH$

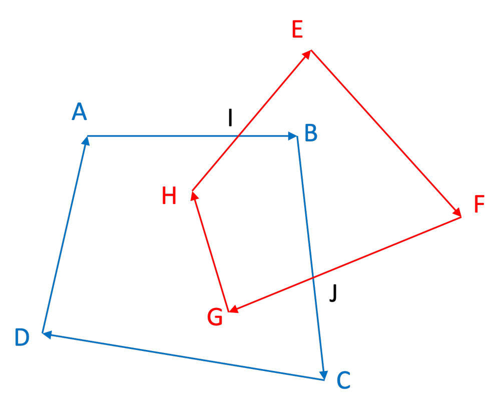


图 2: 正例（求交）

如图2所示，我们想要求四边形 ABCD 和四边形 EFGH 的交。

构建 S 数组：A->I(出)->B->J(入)->C->D->A

构建 C 数组：E->F->J(入)->G->H->I(出)->E

O 数组：J->G->H->I->B->J

得到最终的交多边形为五边形 JGHIB

2.3.2 反例（求交）：求 $ABCD \bigcap EFGH$

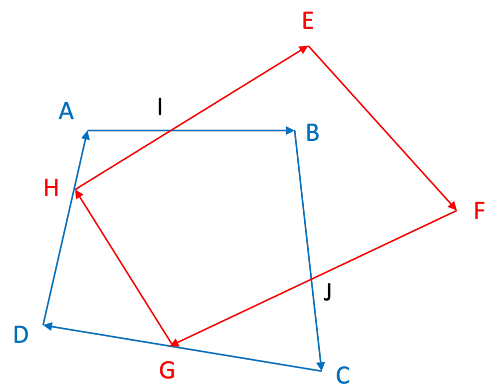


图 3: 反例（求交）

如图3所示，我们想要求四边形 ABCD 和四边形 EFGH 的交。

构建 S 数组：A->I(出)->B->J(入)->C->G(?)->D->H(?)->A

构建 C 数组：E->F->J(入)->G(?)->H(?)->I(出)->E

但在构建 S 数组和 C 数组时，我们发现经典的 Weiler Atherton 算法无法运行下去，因为它无法给出 H 和 G 的出入性判断。

3 Weiler Athenton++

3.1 Algorithm Description

经典的 Weiler Atherton 算法仅仅给出了一般情况的多边形布尔运算算法，而对于一些极端情况下（例如两个多边形存在顶点或者边重合）等情况，算法将给出错误结果，甚至无法完成运算。为此，我们提出了改进的 Weiler Atherton++ 算法。

Weiler Atherton++ 算法包含了两个基本假设，阐述如下。

假设一：边界假设

对于被裁剪多边形 S，我们认为 S 的内部不包含其边界。对于裁剪多边形 C，我们认为 C 的内部包含其边界。这个假设与经典 Weiler Atherton 算法不同之处在于，经典 Weiler Atherton 算法没有对 S 和 C 的内部进行一般性说明，即算法默认 S 和 C 的内部均不包含边界。

假设二：出入点假设

Weiler Atherton++ 给了全新规则用于“出点”和“入点”判断。

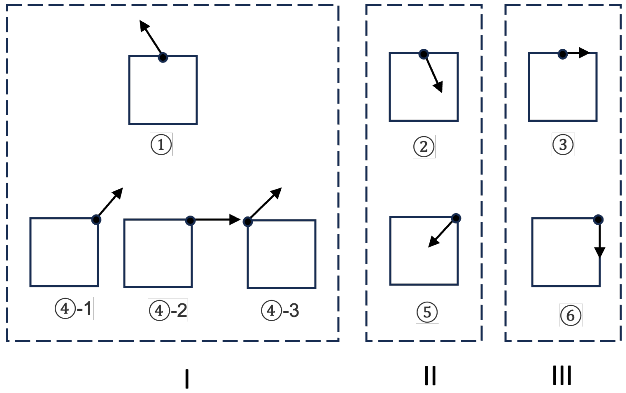


图 4: 射线与多边形关系

1）我们首先阐述如何对顶点在多边形边上的射线和多边形的关系进行分类。如图4所示，我们将射线与多边形的关系分为 I、II、III 三大类和 6 小类。其中在第一大类中，射线从多边形的边上射到多边形的外部（不含边）；第二大类中，射线从多边形的边上射到多边形的内部（不含边）；第三大类中，射线从多边形的边上射到多边形的边（及其延长线）上。

2）根据上述对顶点在多边形边上的射线分类，我们可以将位于多边形边上的点分为如下 18 小类，其中我们又进一步将其划分为 9 个等价类（如图5所示，第（1）小类和第（10）小类为等价类）。其中 A、B、C 是多边形按照顺序依次存在的三个相邻顶点。

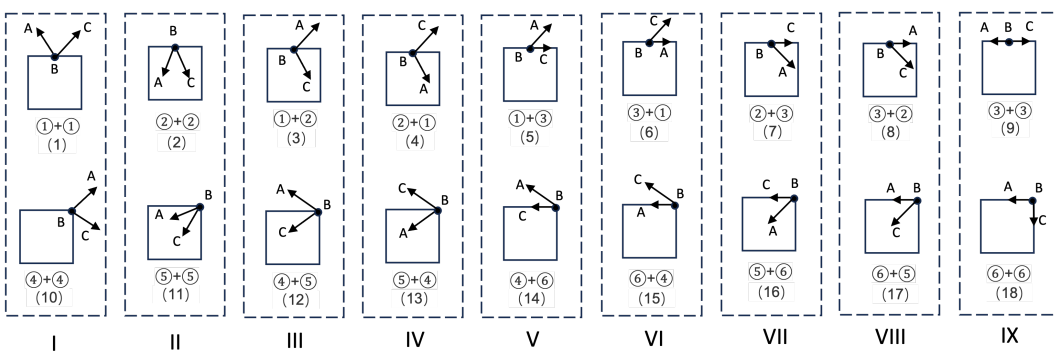


图 5: 点与多边形的关系

具体来说，a) 当我们认为多边形的内部包含其边界时：

“入点”的情况有：III、V

“出点”的情况有：IV、VI

既不是“入点”又不是“出点”有：I、II、VII、VIII、IX

b) 当我们认为多边形内部不包含其边界时：

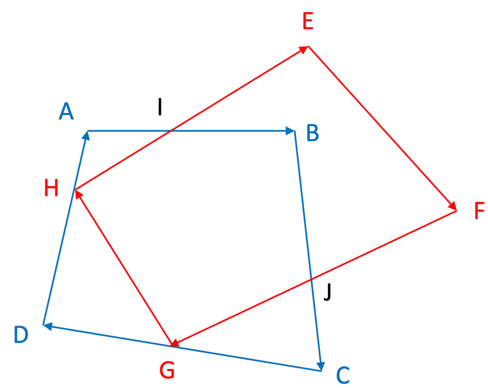
“入点”的情况有：III、VIII

“出点” 的情况有：IV、VII

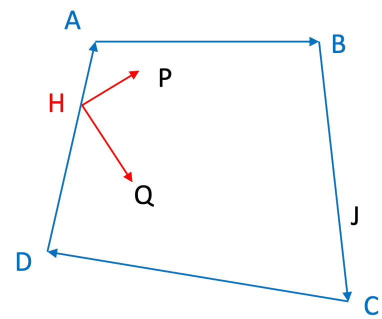
既不是 “入点” 又不是 “出点” 有：I、II、V、VI、IX

3.2 Examples of the Weiler Atherton++ algorithm

3.2.1 例一（求交）



(a)



(b)

图 6: Weiler Atherton 反例修正

如图6，求 $ABCD\bigcap EFGH$。本案例是上述经典 Weiler Atherton 无法处理的案例。

举例说明，以数组 S0（A->B->C->D->A）和数组 C0（E->F->G->H->E）为例，如果想判断 H 点关于多边形 ABCD 为出点还是入点，那么我们能够获得的局部关系图为:

此关系图符合 II 中的模式，无论多边形是否包含其边界，我们看到，H 点均既不是 “入点” 又不是 “出点”。因此，针对上述经典 Weiler Atherton 无法处理的反例。我们构建数组如下：构建 S 数组：A->I(出)->B->J(入)->C->D->A 构建 C 数组：E->F->J(入)->G->H->I(出)->E

我们得到 O 数组：J->G->H->I->B->J，因而多边形的交为 JGHIB

3.2.2 例二（求差）

求 $ABCGHID-EFD$，其中两个多边形存在边 DI 和 GC 重合，存在顶点 D 重合。此处我们给出了一个及其困难的例子以表明 Weiler Atherton++ 算法的普适性。

按照裁切算法，我们构建得到：

S 数组：A->P(入)->Q(出)->B->R(入)->C->G->H->I->D(出)->A

C 数组：E->Q-(入)>R(出)->F->C->G->I->D(入)->P(出)->E

因为我们求的是差，需要将 S 数组和 C 数组进行修正得到：

S 数组：A->P(出)->Q(入)->B->R(出)->C->G->H->I->D(入)->A

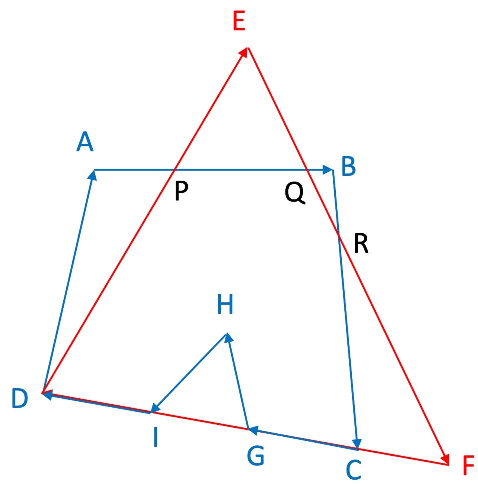


图 7: 例二（求差）

C 数组：E-> P(入)-> D(出)->I->G->C->F-> R(入)-> Q(出)-> E

从而，我们得到两个 O 数组：1）Q->B->R->Q 2）D->A->P->D，即求差的结果为两个三角形：QBR 和 DAP。

在具体的算法实现中，可能会涉及到两个多边形关系的初判、点与多边形状关系、线段与线段关系。为此，我将这些判断方法单独放在附录中。

A 两个多边形关系初判

如图8所示，我们将多边形关系初判分成三大类：1）完全嵌套：嵌套且无边重合；2）重合嵌套：嵌套且有边重合；3）相离。

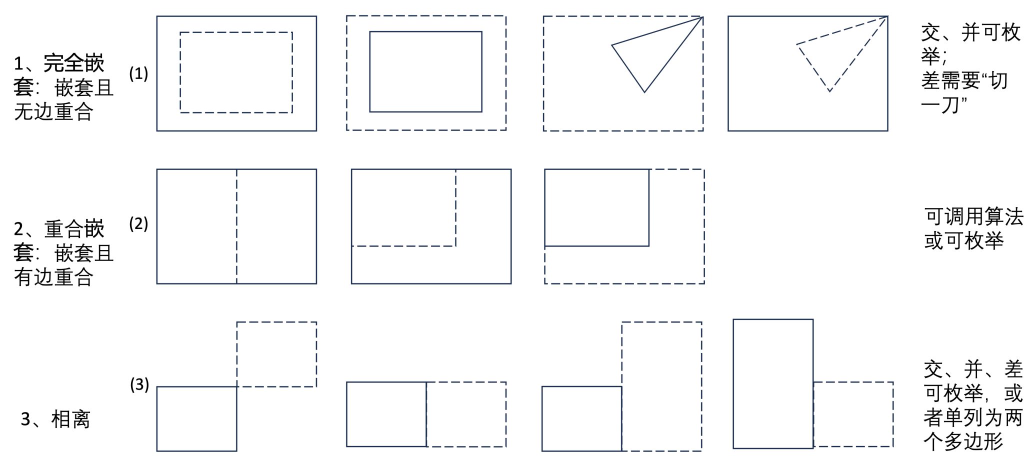


图 8: 两个多边形关系初判

B　点与多边形关系

如图9所示，我们使用右侧射线法判断点是否在多边形内，并将点和多边形关系分为 9 类。

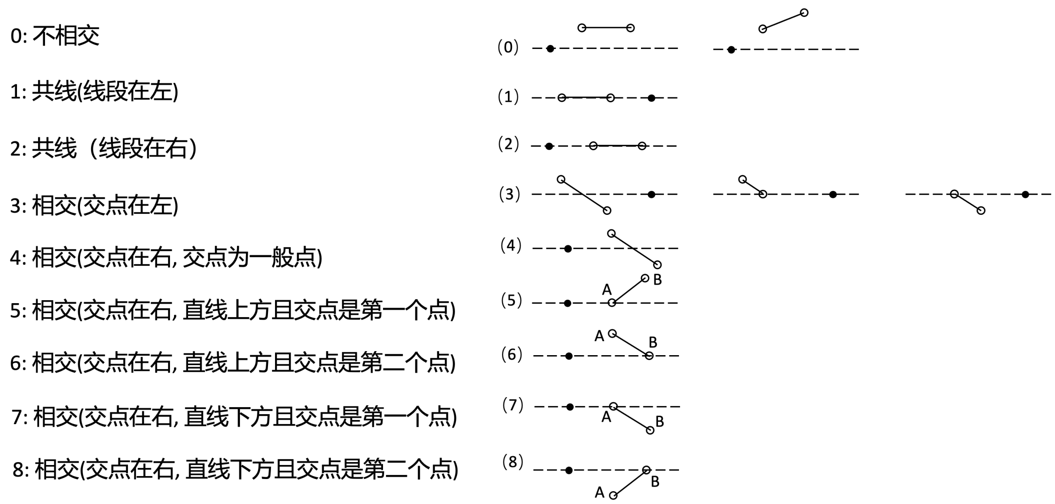


图 9: 点与多边形关系

C　线段与线段关系

如图10所示，我们将线段和线段关系分为三大类。

1）cross_flag = 0; # 延长线相交

2）cross_flag = 1; # 相交 (包含交点为顶点，不共线)

3）cross_flag = 2; # 平行 or 共线 (可能重叠)

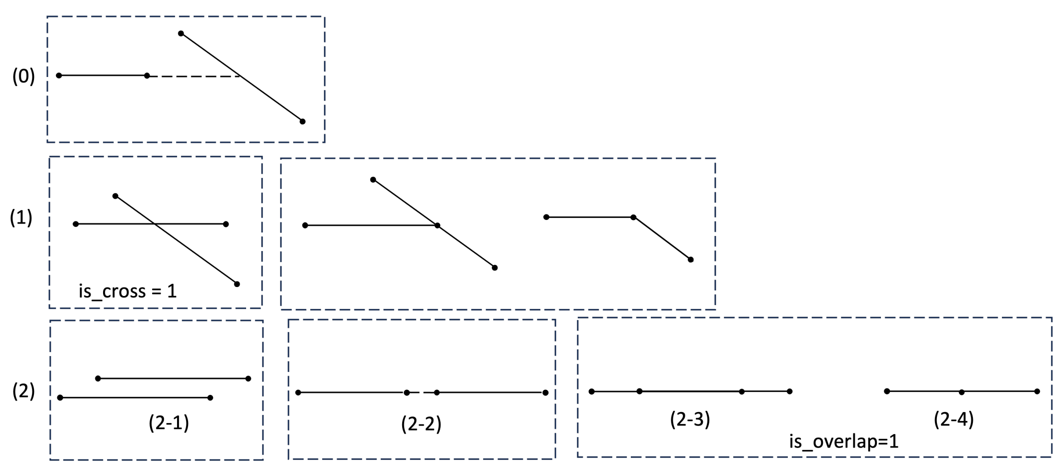


图 10: 线段与线段关系

References

[1] Kevin Weiler and Peter Atherton. Hidden surface removal using polygon area sorting. *ACM SIGGRAPH computer graphics*, 11(2):214–222, 1977.