

一种新的矢量数据多边形的快速裁剪算法

张 钧^{1),2)} 王 鹏¹⁾

¹⁾ (华中科技大学多谱信息处理技术国防科技重点实验室, 武汉 430074) ²⁾ (华中科技大学生命科学与技术学院, 武汉 430074)

摘 要 为实现飞行地理环境中高效的数据调用, 以满足实时性要求, 就需要对飞行地理环境中海量的栅格数据与矢量数据进行统一的数据组织。这种统一的数据组织方法不仅要对海量的栅格数据进行矩形分块组织, 同时也要对海量的矢量数据进行矩形分块组织。为了高效地对海量的矢量数据进行矩形分块组织, 就需要采用高效的矢量数据矩形分块裁剪算法。现有的多边形裁剪算法中, Sutherland-Hodgeman 算法和 Maillot 算法对于裁剪的结果多边形有多个分离部分时都得不到正确的裁剪结果, 而 Weiler-Atherton 算法、Vatti 算法和 Greiner-Hormann 算法却总能得到正确的裁剪结果。后 3 种算法中, 虽然 Greiner-Hormann 算法在空间消耗和时间消耗上都是性能最好的, 但仍不能满足实际工程的要求。为进一步提高裁剪速度, 提出了一种新的快速有效的矩形窗口的多边形裁剪算法。该新算法不仅继承了后 3 种算法在连接形成裁剪的结果多边形时的优点, 而且还对 Greiner-Hormann 算法在插入交点时的处理方式进行了改进, 并采用了比 Greiner-Hormann 算法中应用的双向链表更为简单的单向链表的数据结构。实验结果表明, 新算法不仅能得到正确的裁剪结果, 而且在空间消耗和时间消耗上的性能优于 Greiner-Hormann 算法, 可满足实际工程的要求。

关键词 矢量数据 数据组织 多边形裁剪

中图法分类号: TP391.72 P208 **文献标识码:** A **文章编号:** 1006-8961(2008)12-2409-05

A New Fast Polygon Clipping Algorithm for Vector Data

ZHANG Jun^{1),2)}, WANG Peng¹⁾

¹⁾ (State Key Laboratory for Multi-spectral Information Processing Technologies, Huazhong University of Science and Technology, Wuhan 430074)

²⁾ (School of Life Science and Technology, Huazhong University of Science and Technology, Wuhan 430074)

Abstract To achieve efficient data transfer in real time fly geometric environment (FGE), uniform data organization for massive raster and vector data in the FGE is demanded. In uniform data organization, the rectangular blocking organization is used to deal with massive raster and vector data. To efficiently organize data with rectangular block for vector data needs efficient clipping algorithm using rectangular clipping window. Sutherland-Hodgeman method and Maillot method are this kind of polygon clipping algorithms using rectangular clipping window. But Sutherland-Hodgeman method and Maillot method can not get the correct clipping results if the number of the separate parts in the result polygon is more than one. Weiler-Atherton method, Vatti method and Greiner-Hormann method can get the correct clipping results in this case. Greiner-Hormann method has the best performance in time and space consuming comparing with Weiler-Atherton method and Vatti method. A new fast polygon clipping method using rectangular clipping window is proposed in this paper. The advantage that Weiler-Atherton method, Vatti method and Greiner-Hormann method use linking to form the result polygon has been used as reference in the new method. The new method also improves the mode that Greiner-Hormann method used to insert the cross points. Furthermore, about the data structure, the new method uses the single directional linked list that is simpler than the bidirectional linked list used by Greiner-Hormann method. The experimental results show that the new method can obtain correct clipping result and it has less time and space consumption comparing with Greiner-Hormann method.

Keywords vector data, data organization, polygon clipping

基金项目: 国家自然科学基金项目(60373090)

收稿日期: 2006-09-27; **改回日期:** 2007-08-23

第一作者简介: 张 钧(1966 ~), 男, 副教授。2006 年获华中科技大学工学博士学位, 现在华中科技大学生物医学工程博士后流动站从事博士后研究工作。主要研究兴趣为海量数据组织、计算机视觉、数据挖掘、信号处理等。E-mail: white70121@sina.com

1 引言

为了在飞行地理环境中实现高效的数据调用,以满足实时性要求,就需要对飞行地理环境中海量的栅格数据与矢量数据进行统一的数据组织。这种统一的数据组织方法不仅需要对海量的栅格数据进行矩形分块组织,同时也需要对海量的矢量数据进行矩形分块组织。为高效地对海量的矢量数据进行矩形分块组织,就需要采用高效的矢量数据矩形分块裁剪算法。

在将被裁剪的多边形(又称为实体多边形)位于窗口(又称为裁剪多边形)之外的部分裁剪掉时(裁剪的结果称为结果多边形),采用的算法称为多边形裁剪算法。其中,Sutherland-Hodgeman 算法(简称 S 算法)、Liang-Barsky 算法、Maillot 算法(简称 M 算法)等算法^[1~3]要求裁剪多边形是矩形,而实体多边形则要求是凸多边形。根据飞行地理环境数据的特点,只有能处理实体多边形为一般多边形多边形裁剪算法才在飞行地理环境的数据组织中具有实际的应用价值。在这类算法中,具有代表性的 Weiler-Atherton 算法^[4](简称 W 算法)和近年出现的 Vatti 算法^[5](简称 V 算法)及 Greiner-Hormann 算法^[6](简称 G 算法)在合理的时间内,对于一般多边形都可以得到正确的裁剪结果。其中,W 算法使用的是树形数据结构,而 V 算法和 G 算法由于使用的是双向链表的数据结构,所以后两者在复杂性及运行速度方面都优于前者。可这些算法还是不能满足实际工程的要求。

本文提出了一种新的多边形裁剪的快速算法。其中的裁剪多边形是矩形,而实体多边形则是一般多边形。由于该算法只使用单向链表数据结构,所以具有数据结构简单、占用空间少的特点。算法最终通过简单的遍历一个线性链表,就可以得到裁剪的结果多边形。本文最后对新算法与目前被认为最有效的 G 算法等的性能进行了比较。比较的结果表明,新算法优于上述算法。

2 多边形裁剪的基本概念与定义

2.1 多边形的边界与多边形的内部与外部

按照 Shapefile 矢量数据格式的定义,折线由单向相连的点表示,折线中的点称为节点,相邻节点之间的直线段称为边,而折线中节点的存放顺序则表

示折线的方向。

众所周知,多边形可由一条或多条构成多边形边界的封闭折线表示。每条边界都是一条封闭的,不会自相交的折线。而折线的方向就是多边形的边界的方向。

当一个观察者沿多边形的边界按照边界的方向行走时,观察者的右手侧为多边形的内部,观察者的左手侧为多边形的外部。多边形的边界也被视为多边形的内部。

2.2 进点和出点的定义

当 S 为实体多边形及 C 为裁剪多边形时。设 I 是多边形 S 和多边形 C 的一个交点。如果沿着 S 的边界的方向,可以在点 I 处从 C 的外部进入 C 的内部,则称点 I 为一个进点。如果沿着 S 的边界的方向,可以在点 I 处从 C 的内部出到 C 的外部,则称点 I 为一个出点。

2.3 可见多边折线

实体多边形中,落在裁剪多边形窗口内的那部分尽可能连接在一起的线段(其端点包括实体多边形的顶点和交点),称为可见多边折线。

将可见多边折线按照正确的方式连接起来,就可以得到裁剪的结果多边形。

3 一种新的多边形裁剪算法

在飞行地理环境的数据组织中,裁剪多边形为矩形,其边界方向为顺时针方向。实体多边形为一般多边形,可以有多条边界,还可以有多个孔洞。在这种情况下,笔者提出了一种新的多边形裁剪算法。这种新算法不仅吸收了现有的多边形裁剪算法中的优点,同时克服了现有的多边形裁剪算法中的缺点,并且适用于裁剪多边形为凸多边形的情形。新算法的优点如下:

(1)新算法吸收了 S 算法将窗口裁剪问题转化为一连串的小型裁剪问题的思想,将关于原始一般多边形的裁剪问题转化为一连串的由裁剪多边形的一条封闭边界对实体多边形的一条封闭边界进行裁剪的问题。这样,任意多边形之间的裁剪问题就被转化为一连串的只有一条边界的多边形之间的裁剪问题。

(2)新算法借鉴了 M 算法中的可见边的概念,提出了可见多边折线的概念,同时吸收了 W 算法、V 算法和 G 算法中的不需判断裁剪多边形的节点

是否在实体多边形内部的优点,新算法仅需要在中间链表中进行搜索就能正确地得到裁剪的结果多边形。

(3)新算法在不改变实体多边形和裁剪多边形节点链表的情况下,可一次性地求取所有交点,并且可通过排序算法一次性地生成一个中间链表,用来记录所有的交点和裁剪多边形的原始节点。而W算法、V算法和G算法,在将进点和出点插入到实体多边形和裁剪多边形的节点链表中的时候,由于两个多边形的节点表都在不断变化,即节点数在不断增加,且参与两个多边形之间求交点计算的边数也在不断增加,因此计算量非常大。

(4)新算法在生成裁剪的结果多边形时,只需在中间链表中跟踪搜索。而W算法、V算法和G算法,则需要通过在实体多边形和裁剪多边形的两个节点表之间来回切换跟踪搜索来生成裁剪的结果多边形,其工作量相当于新算法的两倍。

通过以上改进,与现有方法比,新算法不仅能正确裁剪一般多边形,还能节省运算时间。因此,新算法在保证裁剪正确性的前提下,大大缩短了飞行地理环境的数据组织的时间,可满足实际工程的要求。

3.1 新算法的数据结构

大家知道,多边形裁剪算法需要一个合适的数据结构来存储和操作多边形以及交点。在W算法中采用的是树形结构;而在G算法中采用的则是双向链表的结构,即每个多边形用一个双向链表来表示,每找到一个交点,就将其分别插入到实体多边形和裁剪多边形的两个双向链表中,并在这两个双向链表中所插入的同一个交点之间建立起链接关系。新算法采用的是更为简单的单向链表,这样不仅进一步降低了数据结构的复杂性,而且占用更少的存储空间。

新算法内部数据结构如下:

(1)可见多边折线:它是相邻的进点和出点之间的一段折线,也是实体多边形的一段位于裁剪多边形内部的边界。头节点为进点,尾节点为出点。

(2)中间链表:中间链表就是将进点、出点和裁剪多边形的节点,先按照沿裁剪多边形的边界路径各点到裁剪多边形的第1个节点的距离排序后,再按升序连接而成的一个单向循环链表。

3.2 新算法的算法步骤

新算法将关于原始一般多边形的裁剪问题转化为一连串的由裁剪多边形的一条封闭边界对实体多

边形的一条封闭边界进行裁剪的问题。最后,由裁剪多边形的每条封闭边界关于实体多边形的每条封闭边界形成的一连串的裁剪的结果多边形的多条封闭边界就构成了原始一般多边形的裁剪的结果多边形。

因此,以下的描述中,假定裁剪多边形和实体多边形都只有一条封闭边界。本文将裁剪多边形的这条封闭边界简称为裁剪多边形,而将实体多边形的这条封闭边界简称为实体多边形。

由于新算法是为飞行地理环境中的数据组织服务的,假定裁剪多边形为顺时针方向的矩形。

新算法分为以下6个步骤:

(1)寻找始算点,即由实体多边形的第1个节点开始,寻找位于裁剪多边形外的实体多边形的1个节点,并标记这个节点为始算点。

若实体多边形的所有节点都位于裁剪多边形内,则实体多边形位于裁剪多边形内。此时,裁剪的结果多边形与实体多边形完全相同,可直接输出裁剪的结果多边形。

(2)寻找始算边,即由始算点开始,寻找实体多边形与裁剪多边形有交点的第1条边,并标记这条边为始算边。显然,始算边的头节点一定在裁剪多边形外。

若实体多边形的每一条边都与裁剪多边形没有交点,则只有以下两种情形:①如果有1个裁剪多边形的节点位于实体多边形内,则裁剪多边形位于实体多边形内,即裁剪的结果多边形与裁剪多边形完全相同;②如果有1个裁剪多边形的节点位于实体多边形外,则裁剪多边形位于实体多边形外,即裁剪的结果多边形为空。在这两种情形下,都可以直接输出裁剪的结果多边形。

(3)记录交点和可见折线边。由始算边开始,按照求交策略寻找实体多边形与裁剪多边形的交点,同时判定各交点的进出性,并将相邻的一对进出点之间的实体多边形的一段折线记录为可见多边折线,然后计算沿裁剪多边形边界路径由裁剪多边形的第1个节点到交点的距离值,并将其作为排序参数。

(4)生成中间链表。依照排序参数,按照升序对步骤(3)中的交点和裁剪多边形的节点进行排序,生成中间链表,并将排序参数相同的点,按节点、出点和进点的顺序存入中间链表,同时将中间链表记录为单向循环链表。

(5) 输出结果多边形。在中间链表中搜索未标记过的进点, 执行以下步骤: ① 建立一个空的节点表用来记录裁剪的结果多边形的一个新的分离部分, 并将这个未标记过的进点标记为这个节点表的始节点 (第 1 个节点); ② 将进点标记为已经搜索过, 并将以该进点为头节点的可见多边折线输出到裁剪的结果多边形的这个分离部分的节点表中; ③ 在中间链表中, 从刚刚输出的可见多边折线的尾节点 (出点) 开始, 继续跟踪中间链表。每遇到裁剪多边形的原始节点, 就将其输出到裁剪的结果多边形的这个分离部分的节点表中, 直至遇到新的进点; ④ 重复步骤②③步, 直至在中间链表中回到裁剪的结果多边形的这个分离部分的始节点为止。

(6) 重复步骤(5), 直至中间链表中没有未标记过的进点为止。裁剪结束。

3.3 新算法与现有算法的比较

S 算法和 M 算法都是矩形裁剪窗口针对凸多边形的裁剪算法。其中, S 算法需要进行 4 次裁剪 (即按照矩形窗口的 4 个边界依次裁剪), 在进行每条边界的裁剪过程中, 可将本次裁剪的结果多边形作为下一条边界裁剪时的实体多边形。而 M 算法则只需要通过 1 次裁剪, 便能得到裁剪的结果多边形, 只不过在裁剪过程中加入了拐点测试, 以确定矩形裁剪窗口的节点的包入情况。这两种算法在空间消耗上都只占了一个多边形的存储空间, 而在时间消耗上, M 算法要明显优于 S 算法。在裁剪一般多边形时, 这两种算法虽然在空间消耗和时间消耗上比新算法要好, 但是在裁剪的结果多边形有多个分离部分的时候, 这两种算法却因增加了一些冗余的边而不能得到正确的裁剪结果。

W 算法、V 算法和 G 算法都能在实体多边形为一般多边形的情形下输出正确的裁剪的结果多边形。这 3 种算法中, G 算法在空间消耗和时间消耗上都是性能最好的。

本文新算法能在实体多边形为一般多边形的情形下输出正确的裁剪的结果多边形。新算法在空间消耗和时间消耗上都优于 G 算法, 而在裁剪结果的正确性上又优于 S 算法和 M 算法。

本文共进行了 48 组实验, 其中 37 组实验的实体多边形是自行设计的, 11 组实验的实体多边形来自参考文献。下面列出的实验中, 编号为 4-11 的实体多

边形是自行设计的。编号为 m-c, 99-a 等实体多边形来自文献[2]~[7]。对每一种多边形都进行 1 000 次裁剪, 统计 1 000 次裁剪中的最短时间消耗, 最长时间消耗和平均时间消耗 (时间单位为 ms)。

下面是一些裁剪实验的结果图。每组实验结果图中都给出了裁剪矩形和实体多边形的示意图以及 S 算法、M 算法、G 算法和新算法得到的裁剪结果多边形。其中, 细点划线为顺时针方向的裁剪矩形边界, 细虚线为顺时针方向的实体多边形边界, 粗实线为顺时针方向的裁剪结果多边形的边界。

图 1(a)~图 3(a) 均为裁剪矩形和实体多边形示意图; 图 1(b)~图 3(b) 为 S 算法和 M 算法得到的相同的裁剪结果的示意图; 图 1(c)~图 3(c) 为 G 算法和本文新算法得到的相同的裁剪结果示意图。

图 1 是编号为 m-c 的实体多边形的裁剪结果, 正确的裁剪结果如图 1(b) 和图 1(c) 所示, 裁剪的结果多边形只有 1 个分离部分, 是 1 个 20 边形。

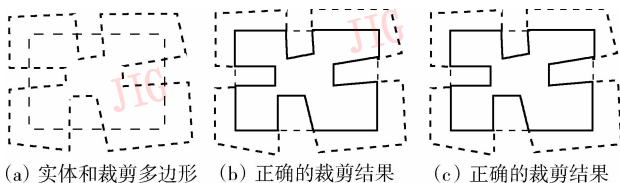


图 1 编号为 m-c 的实体多边形的裁剪结果示意图

Fig. 1 Clipping result of test m-c

图 2 是编号为 4-11 的实体多边形的裁剪结果, 正确的裁剪结果如图 2(c) 所示, 裁剪的结果多边形由 2 个分离部分构成, 分别是位于裁剪矩形的右上角的 1 个四边形和左下角的 1 个三角形。而图 2(b) 所示的错误裁剪结果中, 在裁剪的结果多边形的 2 个分离部分之间多出了 2 条边 (分别位于裁剪矩形的右下边界和下右边界) 将分离部分连接了起来。

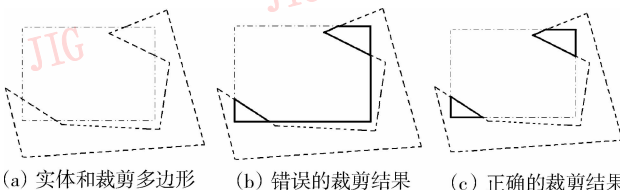


图 2 编号为 4-11 的实体多边形的裁剪结果示意图

Fig. 2 Clipping result of test 4-11

图 3 是编号为 99-a 的实体多边形的裁剪结果,

正确的裁剪结果如图 3(c)所示,裁剪的结果多边形由 3 个分离部分构成,分别是位于裁剪矩形的右上角的 1 个三角形,右下角的 1 个三角形和左下角的 1 个五边形。而图 3(b)中所示的错误的裁剪结果中,在裁剪的结果多边形的 3 个分离部分之间多出了 2 条边(分别位于裁剪矩形的右边界和下边界)将分离部分连接了起来。

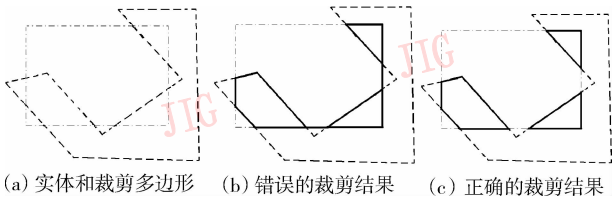


图 3 编号为 99-a 的实体多边形的裁剪结果示意图
Fig. 3 Clipping result of test 99-a

表 1 不同多边形裁剪算法的时间消耗对比表
Tab. 1 Comparison table of the consumed time of the polygon clipping algorithms

实体多边形编号	算法	实体多边形的节点数	性能参数	
			结果多边形分离部分的个数	平均耗时间(ms)
m-c	S 算法	20	1	13
	M 算法		1	9
	G 算法		1	29
	新算法		1	19
4-11	S 算法	8	1	8
	M 算法		1	4
	G 算法		2	15
	新算法		2	11
99-a	S 算法	8	1	8
	M 算法		1	4
	G 算法		3	20
	新算法		3	15

从表 1 和图 1~图 3 可看出,在所有的实验中,G 算法和新算法都能给出正确的裁剪的结果多边形。当裁剪的结果多边形中分离部分的个数为 1 时,S 算法和 M 算法也可以给出正确的裁剪的结果

多边形,而 M 算法在 4 种算法中的时间消耗性能最优。当裁剪的结果多边形中分离部分的个数大于 1 时,S 算法和 M 算法都不能给出正确的裁剪的结果多边形,而此时,新算法和 G 算法虽都能给出正确的裁剪的结果多边形,但新算法在时间消耗上的性能优于 G 算法。

由此可见,与现有方法比,新算法在保证正确裁剪的前提下,还能节省运算时间,从而缩短了飞行地理环境的数据组织时间,可满足实际工程的要求。

4 结 论

大家知道,要高效地实现矢量数据的分块组织,就需要采用高效的矢量数据的分块裁剪算法。本文在借鉴现有多边形裁剪算法的思想以及优点的基础上,提出了一种快速有效的多边形裁剪算法。这种新的多边形裁剪算法不仅能得到正确的裁剪结果,而且在时间消耗和空间消耗上的性能也优于 G 算法。

参考文献 (References)

1 Sutherland Ivan E, Hodgman Gary W. Reentrant polygon clipping [J]. Communications of ACM, 1974, 17(1):32~42.

2 Liang You-dong, Barsky Brian A. An analysis and algorithm for polygon clipping [J]. Communications of the ACM, 1983, 26(11): 868~877.

3 Maillot Patrick-Gilles. A new, fast method for 2D polygon clipping: Analysis and software implementation [J]. ACM Transactions on Graphics, 1992,11(3):276~290.

4 Weiler Kevin, Atherton Peter. Hidden surface removal using polygon area sorting [A]. In:Proceedings of Siggraph'77[C], San Jose, California, USA, 1977: 214~222.

5 Vatti Bala R. A generic solution to polygon clipping [J]. Communications of the ACM, 1992, 35(7): 57~63.

6 Greiner Gunther, Hormann Kai. Efficient clipping of arbitrary polygons [J]. ACM Transactions on Graphics, 1998, 17(2): 71~83.

7 Liu Yong-kui, Gao Yun, Huang You-qun. An Efficient algorithm for polygon clipping [J]. Journal of Software, 2003, 14(4):845~856.

[刘勇奎,高云,黄有群.一种有效的多边形裁剪算法[J].软件学报,2003,14(4):845~856.]