

# 任意多边形裁剪算法的研究及其实现

李志涛<sup>1</sup> 李霖<sup>1</sup> 吴贤良<sup>2</sup> 朱海红<sup>1</sup>

(1 武汉大学资源与环境科学学院,武汉市珞喻路 129 号,430079; 2 湖南省国土资源厅第二测绘院,长沙县暮云镇南托,410119)

**摘要** 介绍了一种改进的 Weiler-Atherton 裁剪算法,简化了算法的实现过程,完善了细节处理,通过在地图符号库设计系统进行实验,获得了满意的结果。

**关键词** 地理信息系统;裁剪;多边形;算法;研究与实现

按照处理对象的不同,裁剪可以分为线段裁剪和多边形裁剪。线段裁剪算法,主要有线段细分裁剪算法<sup>[1]</sup>、中点分割算法、梁友栋-Barsky 算法<sup>[2]</sup>等,这些算法的缺陷就是裁剪区域必须为规则的凸多边形或者矩形窗口<sup>[3]</sup>。多边形裁剪算法除了梁友栋-Barsky 算法外,还有 Suther-Hodgman 算法<sup>[4]</sup>及 Weiler-Atherton<sup>[5]</sup>算法。Suther-Hodgman 算法是用矩形窗口边界直线作为边界对给定多边形进行逐次裁剪,可能产生边界退化的问题,需要进行特殊处理<sup>[6]</sup>。梁友栋-Barsky 多边形裁剪算法是其线段裁剪算法的增强,可以扩展到任意的凸裁剪窗口<sup>[7]</sup>。

在实际应用中所涉及的可能是任意多边形,Weiler-Atherton 算法可以处理凹多边形裁剪区域,实现任意多边形的裁剪,但是其缺点就是实现过程复杂,同时在细节处理方面不够完善。

## 1 Weiler-Atherton 算法主要思想

在 Weiler-Atherton 算法中,多边形采用有序、有方向的顶点环形表描述。当用裁剪区域裁剪被裁剪多边形时,裁剪多边形与被裁剪多边形边界相交的交点成对出现且分为两类,分别称为入点和出点。由入点开始沿被裁剪多边形追踪,当遇到出点时跳转至裁剪多边形继续追踪,如果再次遇到入点则跳转至被裁剪多边形继续追踪。重复以上过程,直至回到起始入点,即完成一个多边形的追踪过程<sup>[8]</sup>。

## 2 对 Weiler-Atherton 算法的改进及实现

Weiler-Atherton 算法的实现过程较为复杂,需要建立层线表、点表及双向指针等。而且有些入点、出点的判断也较繁琐。现对这一算法进行了改进,简化了实现过程,同时在细节处理如两线段的交点、内点外点的判断以及最后获取输出多边形的过程也作了相应的改进。以内裁剪为例,算法描述如下:

第一步,对主多边形及裁剪多边形进行排序。使之按顺时针方向排列,并且确定其首末点相同,亦即闭合的;然后将两多边形由点结构转化为自定义的线段结构,记录每条线段的首末点。

第二步,求多边形的交点。如果要将上一步转化来的两个线段数组中每一项两两求交,则比较耗时,所以就首先对两个线段进行一个外接矩形的比较<sup>[9]</sup>,如果两者不可能相交,就跳过。

任意两条线段之间的交点情况又可细分如下<sup>[10]</sup>。

1) 当两线段都为垂直线时,可能出现交点的情况如图 1 所示。

交点的记录情况是:在图 1(a)中线段  $a$  记录  $b_1$  点,线段  $b$  记录  $a_2$  点;在图 1(b)中,线段  $a$  记录  $b_1$  点,线段  $b$  无交点记录;在图 1(c)中,线段  $a$  记录  $b_1, b_2$  点,线段  $b$  无交点记录;在图 1(d)中,线段  $a$  无交点记录,线段  $b$  记录  $a_2$  点;在图 1(e)中,线段  $a$  记录  $b_2$  点,线段  $b$  无交点记录;在图 1(f)中,线段  $a$

项目来源:国家 863 计划资助项目(2002AA131030)。

GIS 应用等方面的研究。

E-mail:lixinshuang03@163.com

收稿日期:2003-11-12./修回日期:2004-03-03.

第一作者简介:李新双,硕士研究生,现主要从事遥感图像处理及

## DESIGN OF GROUND OBJECT SPECTRAL LIBRARY

LI Xinshuang ZHANG Liangpei LI Pingxiang

(State Key Laboratory of Information Engineering in surveying, Mapping and Remote Sensing, Wuhan University, 129 Luoyu Road, Wuhan 430079, China)

**ABSTRACT** It is an effective method to establish a ground object spectral library, for managing and analyze varied typical objects' spectra date. This paper discusses the design of spectral library including the data standardization, system structure design and function composition as well.

**KEY WORDS** spectral library; ground object spectra; hyperspectral data

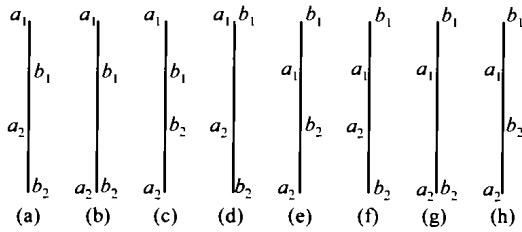


图1 两线段都垂直时可能出现交点的情况

无交点记录, 线段  $b$  记录  $a_1, a_2$  点; 在图 1(g) 中, 线段  $a$  无交点记录, 线段  $b$  记录  $a_1$  点; 在图 1(h) 中, 线段  $a$  记录  $b_2$  点, 线段  $b$  记录  $a_1$  点。

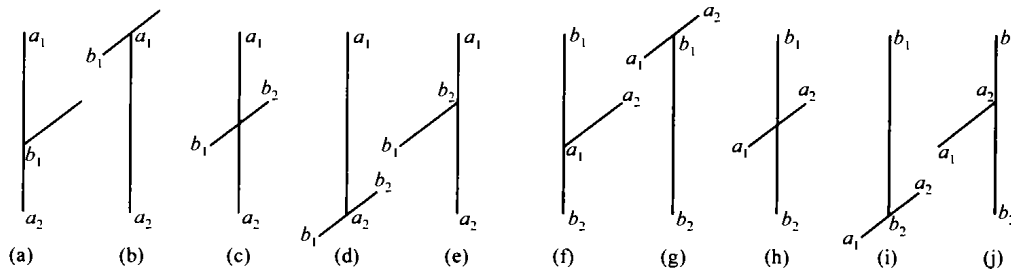


图2 两线段一个垂直另一个不垂直时可能出现交点的情况

2(j) 中, 线段  $a$  无交点记录, 线段  $b$  记录  $a_2$  点。

3) 当两线段皆不是垂直线时, 有两种情况。

两者互相平行。此时, 若两者有部分重合, 则两线段的空间关系及交点的记录情况类似于 1)。若无重合部分, 则没有交点。

两者不平行。处理情况类似于 2)。

记录交点的结构为自定义类型。

```
struct{
    float x; //交点 x 坐标
    float y; //交点 y 坐标
```

```
int flag; //标识符
```

```
int index; //标识符
```

```
}FloatPoint;
```

为主多边形及裁剪多边形各开辟一个数组, 分别记录落在其上的交点。记录交点时  $flag$  全部赋为 0, 而  $index$  则保存记录此交点所在  $Line[i]$  的序号  $i$ , 为下一步的插入操作做准备。

第三步, 将交点数组插入到原始多边形中。将求得的交点数组分别插入到主多边形及裁剪多边形中, 其操作过程如图 3 所示。

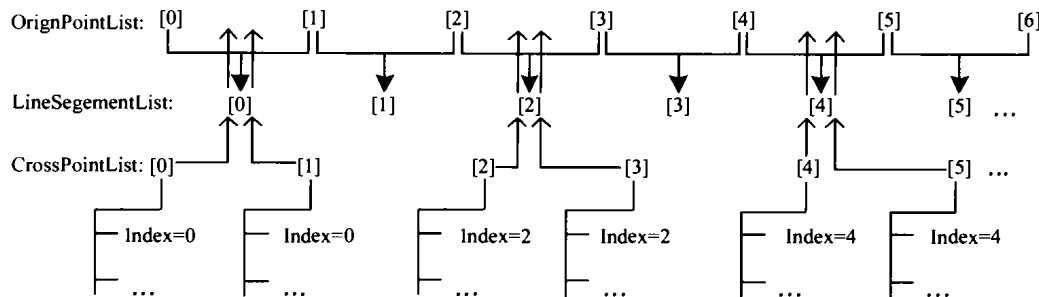


图3 交点插入到原始多边形

根据交点的  $index$  值得到产生其线段的序号, 而每一线段都对应着原多边形顶点数组的相邻的两项, 因此只需将交点插入到顶点数组中这相邻两项之间即可。当同时有几个点需要插入时, 需对这些交点进行排序。进行这一步操作时要将原多边形由普通的点转化为自定义的  $FloatPoint$  型, 其  $flag$  全部赋为 0; 记插入交点后的主多边形及裁剪多边形分别为  $ClippingList$  及  $ClippedList$ 。

第四步, 内点外点的判断。不再是单纯判断其为内点外点并存到不同数组中, 而是通过每一个点结构中的  $flag$  值来标识每一个点的状态, 内点外点的判断如图 4 所示。其中图 4(a) 点落在多边形的内部, 其  $flag$  为 1; 图 4(b) 点落在多边形外部, 其  $flag$  为 0; 图 4(c) 点落在多边形边上, 并且与下一点的中点落在多边形边上, 其  $flag$  为 1; 图 4(d) 点落在多边形边上, 并且与下一点的中点落在多边形外部, 其  $flag$  为 -1。

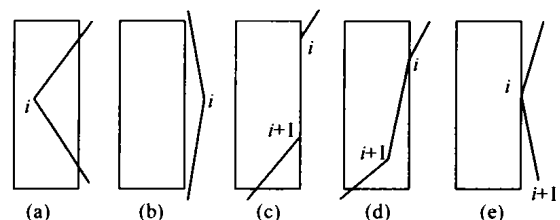


图4 内点外点的判断

按上述原则对两多边形  $ClippingList$  和  $ClippedList$  上每个点进行上述的  $flag$  赋值操作。

第五步, 裁剪。经过以上各步处理, 每个点都有一个

flag 值,并且是根据特定规则得出的,只要在主多边形和裁剪多边形间依次追踪 flag 值为 1 的点,最后使区域闭合,就得到了所要的多边形。开辟两数组,一个用以纪录裁剪所得的多边形的顶点,另一个用来纪录裁剪所得各个多边形的分界点的位置,用以实现分块提取。两个简单多边形内裁剪如图 5 所示。

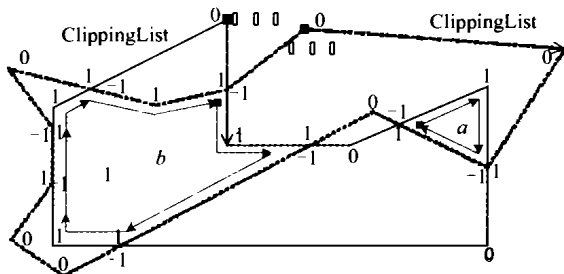


图 5 多边形裁剪示意图

在图 5 中两多边形裁剪时,首先裁出的是区域 b,由主多边形起始点开始遍历,从第一个 flag 为 1 的点开始记录,此后在主多边形和裁剪多边形中跳转,依次记录 flag 值为 1 的点,最后封闭区域 b。然后再跳回到主多边形,从第一次跳转到裁剪多边形处开始遍历,裁出区域 a。当再次跳回到主多边形并进行遍历时,主多边形中已经没有 flag 值为 1 且没有记录过的点。这时裁剪便结束了。

这只是两个简单多边形的内裁剪,还可以扩展到裁剪多边形和被裁剪多边形都带孔的情况。带有孔的任意多边形进行裁剪时,需要将孔按照逆时针排序,同时 flag 值的设置也有相应的改动。外裁剪获取的是裁剪多边形落在主多边形外部的部分,需要将主多边形按照逆时针排序,若主多边形包含孔,则孔按顺时针排序。裁剪多边形不作改变,同时设置各点 flag 时要做对应的改动。

### 3 在地图符号库中的应用实验

在地图符号库中,由于地图符号本身不仅有封闭的多边形,还有不封闭的线段,因此还需要对线段进行裁剪。同样因为主多边形的任意性,前面提到的线段裁剪算法不能适用。可以通过对 Weiler-Atherton 算法作一些改动便可实现。同多边形裁剪相比较,线段的裁剪较为简单,不需要将主多边形作处理,只要将交点插入到裁剪线段中,通过判断裁

剪线段的 flag 值便可确定线段跟主多边形的关系,得到最后的裁剪结果。

### 4 结束语:

本文所讨论的算法是在 Weiler-Atherton 算法的基础上所作的一些改进后在 VC++ 6.0 的环境下实现的。同原算法相比,简化了实现过程,避免了使用双向指针等复杂技术,同时还详细分析了特殊情况的处理,可处理各种复杂的情况,最后生成的多边形可分块提取输出。地图符号库的应用实验表明,改进后的算法对各种复杂情况的裁剪都能精确计算出结果,并且稳定可靠。□

### 参考文献

- [1] 吴有富. Cohen-Sutherland 算法的改进及其推广[J]. 贵州工业大学学报, 1998, 27(4): 6~9
- [2] 梁友栋, Barsky B A. An Analysis and Algorithm for Polygon Clipping [J]. Communications of the ACM, 1983, 26(11): 868~877
- [3] Rogers D F. Procedural Elements for Computer Graphics (2E) [M]. 北京: 机械工业出版社, 2002
- [4] Sutherland E E, Hodgeman G W. Reentrant Polygon Clipping [J]. Communications of the ACM, 1974, 17(1): 32~42
- [5] Kevin W, Peter A. Hidden Surface Removal Using Polygon Area Sorting [J]. Computer Graphics, 1977, 11(2): 214~222
- [6] Hearn D, Baker M P. Computer Graphics (2E) [M]. 北京: 电子工业出版社, 2002
- [7] 沈纪桂, 蔡英平, 程刚. 凹多边形裁剪[J]. 浙江大学学报, 1989, 23(1): 145~152
- [8] 吴兵, 尹伟强. 具有拓扑关系的任意多边形裁剪算法[J]. 小型微型计算机系统, 2000, 21(11): 1166~1168
- [9] 杨维芳. 两个复杂多边形求交的矢量算法[J]. 兰州铁道学院学报(自然科学版), 2002, 21(1): 108~110
- [10] Vatti B R. A Generic Solution to Polygon Clipping [J]. Communications of the ACM, 1992, 35(7): 56~63

收稿日期: 2003-11-28. / 修回日期: 2004-02-27.

第一作者简介: 李志涛, 硕士研究生, 现主要研究方向为计算机图形图像处理在地理信息系统中的应用。

E-mail: leezhitao\_ture@163.com

## RESEARCH AND IMPLEMENTATION OF POLYGON CLIPPING ALGORITHM

LI Zhitao<sup>1</sup> LI Lin<sup>1</sup> WU XianLiang<sup>2</sup> ZHU HaiHong<sup>1</sup>

(<sup>1</sup> School of Resource and Environment Science, Wuhan University, 129 Luoyu Road, Wuhan 430079, China;

<sup>2</sup> The 2nd Surveying and Mapping Institute of the Land and Resource Department of Hunan Province, Nantuo, Changsha 410119, China)

**ABSTRACT** This paper discusses an approach to improving Weiler-Atherton clipping algorithm in the process of determination of start point and end point, processing of line-line intersections and acquisition of resulting polygon etc. The approach simplifies the algorithm and perfects the details' handling. Last, the approach is tested in a symbol system.

**KEY WORDS** geographic information system; clipping; polygon; algorithm; research and implementation