

中图法分类号: TP301.6 文献标志码: A 文章编号: 1006-8961(2010)11-1696-11

检索信息: 朱效民, 赵红超, 刘焱, 廖浩均, 方金云, 韩承德. 矢量地图叠加分析算法研究[J]. 中国图象图形学报, 2010, 15(11):1696-1706

矢量地图叠加分析算法研究

朱效民^{1),2)}, 赵红超¹⁾, 刘焱^{1),2)}, 廖浩均^{1),2)}, 方金云¹⁾, 韩承德¹⁾

¹⁾(中国科学院计算技术研究所, 北京 100190) ²⁾(中国科学院研究生院, 北京 100049)

摘要: 提出了一整套矢量地图叠加分析算法: 提出了大量点与多边形关系的包含性测试方法, 首先对多边形进行预处理, 然后采用射线法对单个点进行包含性测试; 提出了基于双索引的大量线裁剪方法, 分别以线、线段为基础建立两层索引, 有效去除不必要的求交运算; 面叠加, 基于改进的扫描线方法求取交点; 通过对过同一点的线段分布情形的完备分类, 涵盖了所有特殊类型的交点; 在构造环的过程中, 保存整型 ID 信息, 并利用 ID 完成内环外环的匹配以及属性继承。以上几何计算方法及对应的叠加分析功能都已经实现, 与最新研究成果的对比以及与 ArcGIS 对应功能的对比, 都证明了其正确鲁棒、高效可用。上述实现已经应用于实际的 GIS 系统中, 取得了良好的效果。
关键词: 地理信息系统; 空间分析; 叠加分析; 矢量地图叠加; 点的包含性测试; 线裁剪

Research on vector map overlay

Zhu Xiaomin^{1),2)}, Zhao Hongchao¹⁾, Liu Yan^{1),2)}, Liao Haojun^{1),2)}, Fang Jinyun¹⁾, Han Chengde¹⁾

¹⁾(Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190)

²⁾(Graduate University of Chinese Academy of Sciences, Beijing 100049)

Abstract: In this paper, we introduce a series of algorithms for vector map overlay: We proposed an algorithm for large amounts of points inclusion test, which first preprocesses the polygon and then adopts the ray-crossing idea for each point inclusion test. We proposed an algorithm for large amounts of lines clipping, which builds indexes on the whole line and the segment, and avoids unnecessary intersection-computing. For map overlay on polygons, we use the improved plane sweep algorithm to get all the intersection points, and the complete categories on distribution of segments passing the same point cover all the special cases of intersection points. We record the ID in the process of constructing the result rings, which simplifies the two processes-finding outer ring for holes and attribute propagation. All the geometric algorithms and the corresponding overlay functions are implemented, and the comparisons with the state-of-the-art algorithms and ArcGIS's function demonstrate its correctness, robustness, efficiency and usability. The above functions have been used in GIS platform and work well.

Keywords: GIS; spatial analysis; overlay analysis; vector map overlay; point inclusion test; line clipping

0 引言

GIS 自从 20 世纪 60 年代在加拿大和美国诞生

以来,取得了飞速的发展,随着空间分析技术的不断发展,GIS 也将从一般的空间事务处理向分析型空间决策支持方向迈进^[1],GIS 已从数据库型 GIS 进入分析型 GIS 的阶段^[2]。矢量地图叠加分析(也叫

基金项目:国家高技术研究发展计划(863)基金项目(2009AA12Z226)。

收稿日期:2010-03-12;改回日期:2010-05-19

第一作者简介:朱效民(1982—),男。中国科学院计算技术研究所应用技术专业博士研究生。主要研究方向为空间计算、GIS 算法、空间数据挖掘。E-mail: zhuxiaomin@ict.ac.cn。

叠置分析,即 vector map overlay)作为空间分析的一种,在 GIS 中扮演了重要的角色。它是最重要的分析操作之一,是 GIS 中的核心操作之一,是一项组织来自不同数据源的数据的基本技术。而且矢量地图叠加分析算法也是 GIS 中的难点,叠加过程被认为是 GIS 发展过程中的最具挑战性的问题之一;退化情形以及浮点计算的有限精度使得矢量数据叠加计算非常困难。因此,从学科的角度看,地图叠加分析算法是地理信息科学的重点与难点。

一般而言,叠加分析常常是对输入图层和叠加图层(一般为面图层)进行分析。根据输入图层的类型不同,叠加分析一般有 3 种:点面叠加、线面叠加和面面叠加。点面叠加即得到在面内的所有点,并得到点在哪个面内。线面叠加即区分在面内的线和在面外的线;对于跨越面边界的线,将线在边界处打断;对于在面内的线,需要得到线在哪个面内。面面叠加即将两个面图层覆盖的区域区分为 3 部分,即公共部分(交)、只属于第 1 个图层的部分(差)以及只属于第 2 个图层的部分(差)。对于交,需要得到是由哪两个区域相交产生的;对于差的部分,需要计算此区域是由哪一个源区域产生的,至于是被哪些区域裁切产生的,常常不需要知道,因为一个区域产生的差区域常常是此区域被多个区域裁切得到的。

基于叠加分析对于系统及科学的重要性,许多专家学者将矢量地图叠加分析作为学位论文的研究工作做了深入系统的研究^[3-12]。大部分的文献只是简单介绍了叠加分析,并没有提出太多新的计算方法。点面叠加,大部分采用的是经典的射线法^[4,8-9];文献[10]介绍了利用堆栈结构和扫描线方法相结合,扫描结束即可得到所有点与所有多边形的关系;文献[12]提出了基于单调链的点与多边形关系判断的方法。线面叠加方面,阐述的较少。面面叠加,大部分采用的是以单个多边形裁剪为基础^[3,8](常采用出点入点的思路^[7]),一一循环计算的思路;部分采用了一些空间索引策略(如 R+树^[4])。程振林的 Han-E 方法^[5]采用了一种整体计算的思路,但在两组多边形都为复杂多边形集合时,内外环的匹配、属性继承、根据特定操作提取对应的多边形结果等存在一定的难度。

以上是国内近些年对叠加分析方面进行系统研究的相关文献。此之外,针对单个叠加分析算法(点面、线面及面面叠加),也有许多相关研究,这些

相关工作将在下面对应主题中进行介绍。

1 点面叠加

点的包含性测试即判断一个或者一系列点是否位于多边形的内部。随着 GIS 的发展,大量点与多边形的包含性测试成为空间数据处理的一个核心问题。诸多专家学者提出了一系列的算法来解决此问题。这些算法有的不对多边形进行预处理,如角度之和法和射线法^[13]等,有的则预处理多边形,如将多边形分解成许多简单单元(如凸多边形^[14])。根据现有文献对算法的分析,最有效的方法是基于 Cell 的方法^[15]和基于多边形凸剖分的方法^[14]。基于 Cell 的方法将多边形栅格化,然后对不含多边形边的栅格计算其在多边形内部还是外部。对点进行包含性测试时,首先计算点在哪个栅格内;若栅格不含任何边,则可以在 $O(1)$ 时间内得到点与多边形的关系;否则,需要计算点与在当前栅格内边的关系。此方法对部分点的包含性测试时间复杂度是 $O(1)$,因此对于一些简单的多边形,它是非常高效的。基于多边形凸剖分的方法是已知最高效的方法。该方法将多边形进行凸剖分,并用 BSP (binary space partition) 树管理这些凸剖分。对任何一个点,查询 BSP 树,计算可能包含其的凸剖分,然后通过点与凸剖分关系得到点与多边形的关系。其包含性测试较迅速,而其预处理时间相对较多,尤其对于边数较多的复杂多边形。

1.1 本文方法

提出了一种新的大量点的包含性测试方法。方法包含两个步骤:预处理和包含性测试。

预处理即将多边形的 MBR (minimum bounding rectangle) 分成若干相等的竖直条带,并且计算每条边的位置。如图 1 所示,多边形有 16 条边,标记为 0~15;将 MBR 分成 16 个条带,每个条带包含上下两部分,然后计算每条边经过的条带。经过预处理后,可以得到每个条带关联的边。如图 1 所示,条带 0 的上部、下部分别包含的边为 0、15 和 12、13。

经过预处理后,算法对每个点进行包含性测试。首先计算点所在的条带,然后构造辅助射线,并与所有当前条带内的边进行相交测试,根据相交次数的奇偶性得到点与多边形的关系。对于位于 MBR 上半部分的点,辅助射线竖直向上;否则竖直向下。这样的策略以一个判断浮点数大小的 Boolean 操作为

代价,避免了随意构造射线可能造成的更多相交情形。

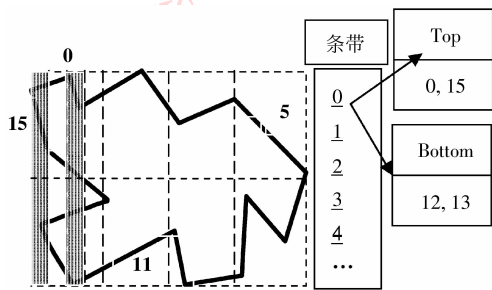


图 1 多边形的预处理

Fig. 1 Preprocessing on a polygon

以射线竖直向上的情形为例,介绍相交判断方法。如图 2 所示,点所在的水平直线与构造的射线将整个平面划分为 3 部分: I、II 和 III。对于每条边,首先计算其两个端点的位置。如果两个端点在同一区域,那么边与射线不相交;否则,边的两个端点位置可能的组合为: I 和 II, I 和 III, II 和 III。两个端点在区域 I 和 II,边必然与射线相交。两个端点在 I 和 III,如果在 III 的端点位于 III 的左半部分,边不与射线相交;否则,假设当前边方向是右下往左上,如箭头所示,那么顶端点 T 一定位于当前边的右侧。因此,只需要计算底部端点(当前点)与当前边的位置关系(左侧或右侧)。如果点位于边的右侧,则边与射线不相交,否则即相交。第 3 种情形与第 2 种情形类似。上述步骤中左右位置关系计算需要乘法和减法操作,其他计算只需要浮点数的大小比较操作,因此具有较高的效率。射线竖直向下时,同理可得。对于一些特殊情形,如:相交在端点或部分重叠等情形,采用微小平移的方法,将构造的射线

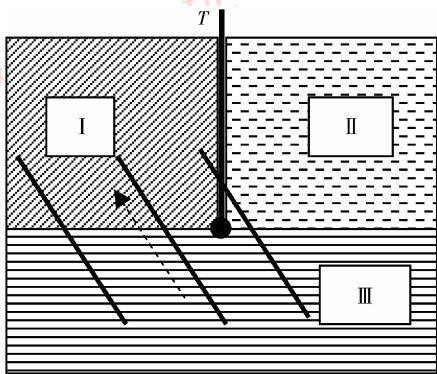


图 2 判断边是否与射线相交

Fig. 2 Judging whether an edge intersect with the ray

左移微小的距离。只要射线平移的方向是一致的,这些特殊的相交情形都能得到正确的判断结果。

1.2 算法分析

下面从最坏和平均两种情形分析算法的时间和空间复杂度。假设多边形的边数为 N ,使用的条带数目是 $N/2$ 。

预处理过程的基础操作即对每条边计算一次,然后将其附加到对应的条带中去。其时间空间复杂度为 $O(N \times M)$,其中 M 表示每个条带包含的边数。因此最坏情形下其复杂度为 $O(N \times N)$ 。真实世界数据中,每个条带包含的边是一定的,其实际空间复杂度是 $O(LM \times N)$, LM 为有限数值,一般与 N 无关,是常数或者接近常数。基于真实地理数据的分析如图 3 所示。3 组数据分别有 2 525、2 501 和 122 552 个多边形。图中“aver”是指平均每个条带内边的数目,即包含性测试的时间复杂度;“times”是所有边的存储次数,即预处理阶段的空间复杂度。如图所示,times 接近线性于 N ,即平均情形下,其时间空间复杂度约为 $O(N)$ 。

包含性测试的空间复杂度为 $O(1)$ 。时间复杂度的最差情形是 $O(N)$ 。然而,基于预处理阶段的空间复杂度分析,我们认为其时间复杂度接近一个有限值,这个值一般远小于 N 。图 4 所示是对前面 3 组数据的时间复杂度分析,也就是 aver 随多边形边数的变化趋势。 X 轴是边数 N ,对应的 aver 是平均每个条带内的边的数目, \log 是指 $\log N$,用来衡量 aver 的。

如 4(a)(b)所示,aver 值是阶段性的,其值与 N 有一定关系,但总体上是一个接近常量的值;而且,aver 总体上小于 \log ,因此我们可以认为其时间复杂度接近 $O(1)$ 。图 4(c)中,可以看到 aver 要大于 \log ,但是小于 \log^2 ,近似为 $O((\log N)^{1.5})$ 。我们认为 $O(\log N \sim ((\log N)^{1.5}))$ 为其时间复杂度。这与数据集 1 和 2 有所不同,究其原因,是因为数据集 1 和 2 都是较为简单的多边形,而数据集 3 的多边形较为复杂,且常带洞。

综上,对于点与有 N 条边的多边形进行包含性测试时,平均情形下,预处理的时间和空间复杂度都约为 $O(N)$,包含性测试的空间复杂度为 $O(1)$,时间复杂度对于简单多边形接近 $O(1)$,对于复杂多边形约为 $O((\log N)^{1.5})$ 。

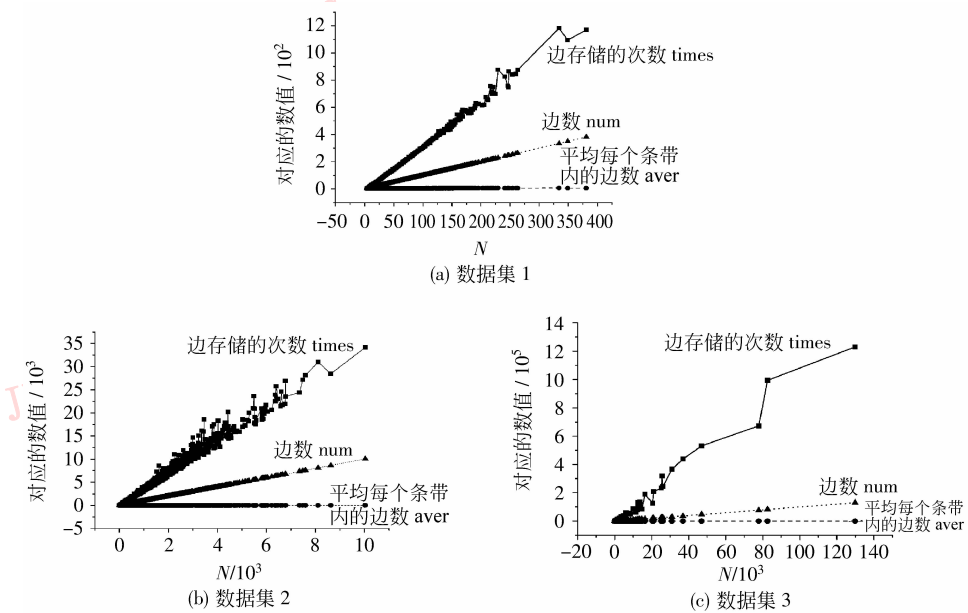


图 3 预处理阶段存储空间的变化曲线
 Fig. 3 Transformation curve of storage space in preprocessing

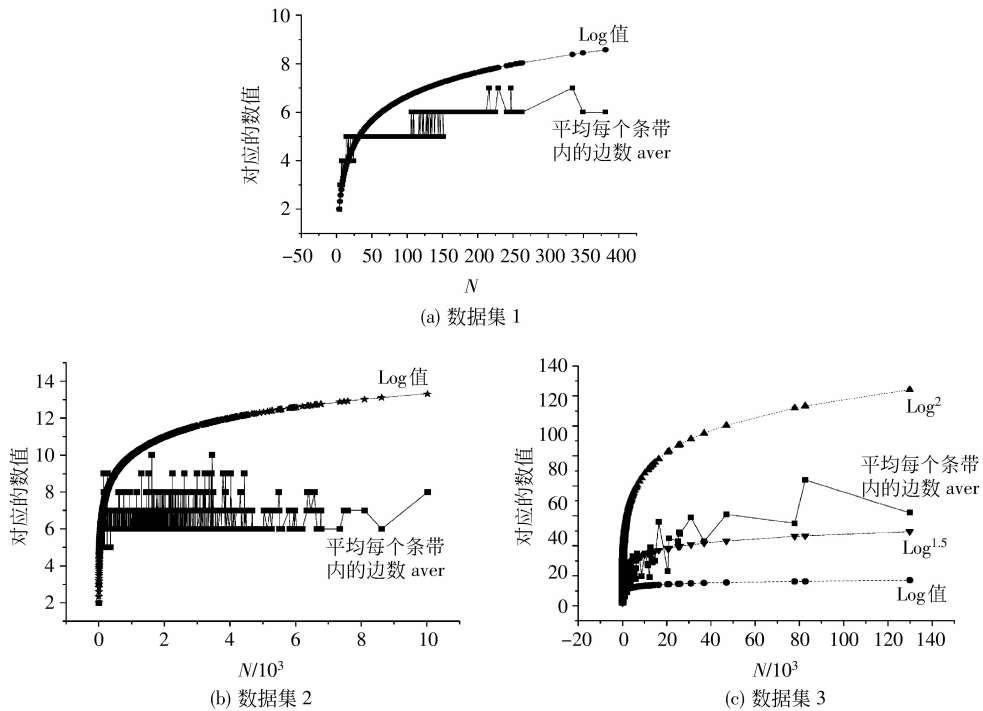


图 4 条带内的边数变化曲线
 Fig. 4 Transformation curve of edge number in stripes

1.3 算法对比测试

测试的输入是由多边形 MBR 过滤得到的点。本文算法 (Our) 与 Cell^[15] 算法 (其源码为地址 <http://www.iimg.org/index.php/publisher/articleview/>

frmArticleID/104) 进行了对比。对比测试在同一台 IBM X31 笔记本 (Intel PM 处理器 1.7GHz, 512M 内存) 上进行, 并且两个算法在 VC++.NET 中实现, 使用相同的编译、链接、执行环境。对比的结果如

表 1 所示。3 个多边形是上述源码带的数 据, 点数据是一个均匀分布于多边形的 MBR 中的 $1\ 000 \times 1\ 000$ 的点集。从表 1 对比可得, 本文方法预处理及点包含性测试的时间都少于 Cell 方法。此外, 本文方法得到的结果是精确的, Cell 的计算结果正确率是 98% [15]。基于凸剖分的方法较为复杂, 不同的实现有不同的效率, 因此我们没有实现其算法。凸剖分的方法也用这 3 个多边形做了测试, 其测试结果在文献[14]中。文献[14]中的 Grid 方法即 Cell 方法, 而且对比也是基于文献[15]带的源码及数据。以 Grid 法为基准, 计算可得对应 3 个数据的预处理以及包含性测试的相对值如表 2 所示, 可得本文算法(Our)具有较少的初始化时间, 对于复杂多边形更为明显。点的包含性测试而言, 本文算法也稍优于方法基于凸剖分(Convex A, Convex B)的方法。

表 1 与 Cell 方法的对比测试
Tab. 1 Comparisons with the cell-based method


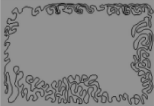

多边形	端点个数	预处理时间/ms		10 ⁶ 个点的包含性测试时间/ms	
		Cell	Our	Cell	Our
	100	0	0	912	210
	1 294	10	0	831	330
	28 105	90	30	3 495	520

表 2 以 Grid 法为基准的相对对比结果

Tab. 2 The relative comparison result based on the Grid method

端点个数	预处理时间相对值				包含性测试相对值/%			
	Grid	Convex A	Convex B	Our	Grid	Covex A	Convex B	Our
100	1	0.17	0.06	0	1	29.5	35.9	23.0
1 294	1	2.44	0.897	0	1	54.4	92.6	39.7
28 105	1	11.89	7.99	0.22	1	19.1	28.5	14.9

结合以上两表的绝对对比与相对对比数据, 可以得到一个凸剖分方法的近似计算值。以相对值为基础, 可得其预处理、包含性测试的时间变化曲线分别如图 5、图 6 所示。从图示可得, 本文方法在预处

理和包含性测试方面都要优于 Cell、Convex A、Convex B 方法。

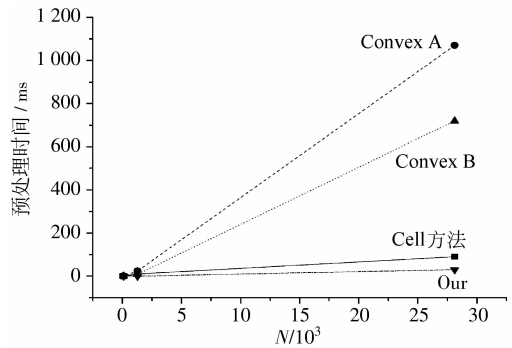


图 5 预处理时间变化曲线

Fig. 5 Transformation curve of preprocessing time cost

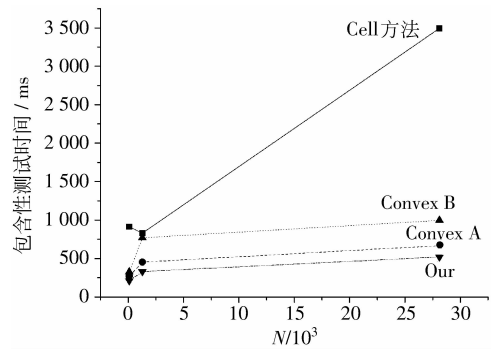


图 6 点的包含性测试时间变化曲线

Fig. 6 Transformation curve of point inclusion test time

以此几何计算算法为基础, 辅以数据读写模块, 实现了基于 Shape File 文件的矢量点面叠加分析功能。与 ArcGIS 的点面叠加分析(点面求交功能)功能对比如表 3 所示。正确性方面, 本文方法(VegaGIS)与 ArcGIS 得到的结果要素数是一致的; 效率方面, 本文的叠加分析功能整体计算时间为 ArcGIS 同样功能的 20% ~ 25%。以 MBR 内的点数

表 3 点面叠加分析功能对比测试表

Tab. 3 Comparison table of point-polygon-overlay

MBR 中点的个数	多边形个数	时间/s		结果要素数		裁剪区域
		VegaGIS	ArcGIS	VegaGIS	ArcGIS	
26 209	1	2.2	9.0	19 710	19 710	省域
264 722	1	12.0	51.0	134 611	134 611	区域 1
417 726	1	21.9	96	272 536	272 536	区域 2
597 531	2	32.9	144	407 147	407 147	工作范围
649 567	2 525	52.9	223	639 684	639 684	行政区划

为基础, 可得其随着点数变化计算时间的变化曲线如图 7 所示。

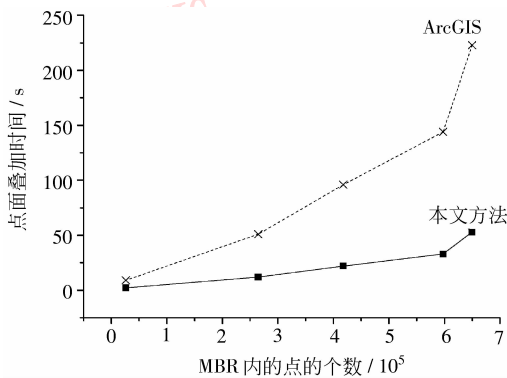


图 7 点面叠加分析时间随点数变化曲线图

Fig. 7 Transformation curve of point-polygon overlay time cost on point number

2 线面叠加

GIS 中的线裁剪的对象往往是大量的线, 其核心问题是如何避免不必要的求交运算、快速求取所有交点。如果采用传统的线裁剪算法, 则需要一一计算, 即使采用了高效的单条线裁剪算法, 如采用错切变换法^[16]求交点, 或者将多边形进行凸剖分等^[17], 仍然不够高效。

有一系列高效的算法解决了大量线段的求交问题, 如扫描线算法^[18]、基于空间索引的方法等。文献[19]分类总结了这些高效的求交点算法。真实地理数据中, 有很多线不与多边形的任何边相交, 因此这些线无需参与求交运算。传统的扫描线算法、空间索引算法等, 对所有输入线段不加区别的进行处理, 这些线同样参与了运算—在扫描线算法中, 它们被插入、删除; 在空间索引方法中, 它们需要参与建立索引以及查询的时间—这些都导致了不必要的运算。

2.1 本文方法

基于以上分析, 采用双层的索引机制, 基于线、线段建立索引。本文方法依赖于一个高效的 R-tree 实现。R-tree 提供索引建立以及索引查询的功能。

第 1 层索引是基于整条线 MBR 的。索引建立之后, 基于多边形的每条边的 BR (bounding rectangle) 进行查询, 得到了所有可能与多边形相交的线。对于这些可能与多边形相交的线, 其线段的

数量仍然是相当多的, 因此基于线段的第 2 层索引将被建立。然后针对多边形的每条边进行查询, 得到可能与当前边相交的所有线段, 并计算交点。之后对这些交点按照所属线的不同以及在线上出现的顺序进行排序, 计算交点的出入属性, 并根据奇偶性设置所有交点的属性, 最后根据不同的操作, 截取在多边形内部或者外部的部分。对于那些没有产生任何交点的线, 为每条线选取一个点作为代表点, 然后执行一次点与多边形的包含性测试, 即可得到点代表的线与多边形的关系, 然后将其完整地插入到内部或者外部结果集合中去。

2.2 算法分析

就复杂度而言, 线裁剪算法中复杂度最高的即为求交点部分。对于 M 条线段, 假设平均具有 N 个点, 这些线段被一个具有 P 条边的多边形进行裁剪, 其求交部分的复杂度为 $O(M \times N \times P)$ 。对最坏的情形(任何一条线的 MBR 都与多边形相交, 线的任意一条线段都与多边形相交), 本文方法具有相同的时间复杂度。然而在真实世界的地理数据中, 最普遍的情况即用一个多边形对大量线进行裁剪, 相对于多边形的 MBR, 线的 MBR 一般是很小的; 只有一部分线与多边形的边相交; 对于多边形相交的线, 只有部分线段与多边形的边相交。

从理论上分析, 面积的比值在一定程度上代表了过滤率。将 MBR 的面积(多边形所有边的 MBR 的面积与整个多边形 MBR 的面积)作为一个衡量因素, 来分析过滤率。分别选取两组地理数据, 一组是描述行政区域的多边形数据, 边数较少, 较为简单; 另一组是描述土地利用的数据, 边数较多且多含有洞, 较为复杂。图 8 中两图分别描述了简单、复杂多边形的过滤率随着多边形边数的变化而变化的趋势。可得大部分多边形的面积比值都在 0.1 左右, 从而可以利用双重索引策略, 达到一个较为理想的过滤率, 减少不必要的运算。

2.3 线裁剪功能测试

我们没有从算法角度与线面一一裁剪的方法进行对比, 而是辅以必要的读写过程, 实现了整个线面叠加分析方法, 并与 ArcGIS 的对应功能进行了对比, 如表 4 所示。

计算时间随要素个数的变化趋势图如图 9 所示。可得, 随着要素数增多, 计算时间呈近线性增

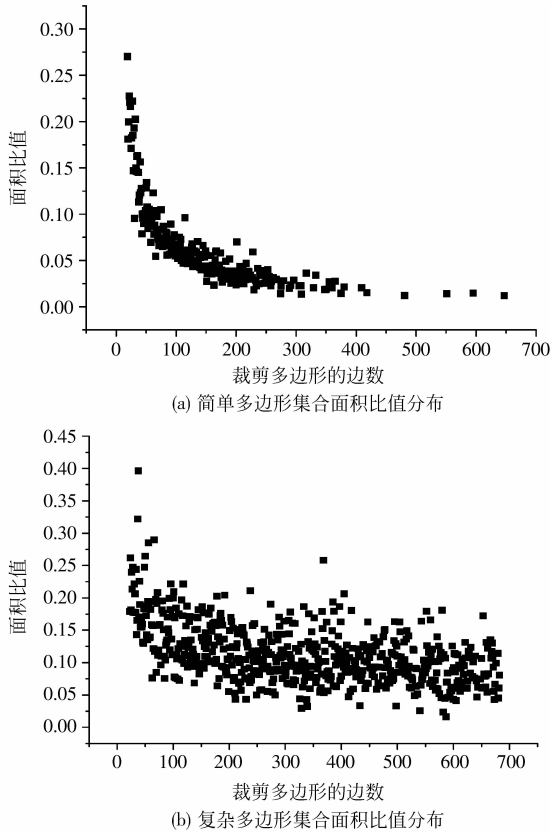


图 8 面积比值变化随边数变化图

Fig. 8 Transformation graph of area ratio on edge number

表 4 线面叠加分析功能对比测试表

Tab. 4 Comparisons of line-polygon-overlay

裁剪多边形	MBR 内要素个数	计算时间/s		时间比值%	结果要素数		差值百分比/%
		VegaGIS	ArcGIS		VegaGIS	ArcGIS	
省域	65 357	17.3	30.0	57.7	49 446	49 446	0
区域 1	436 774	95.9	336	28.5	205 269	205 260	0.002 9
区域 2	510 272	119.3	488	24.4	317 237	317 234	0.000 95
工作范围	832 801	187.9	717	26.2	522 506	522 494	0.002 9

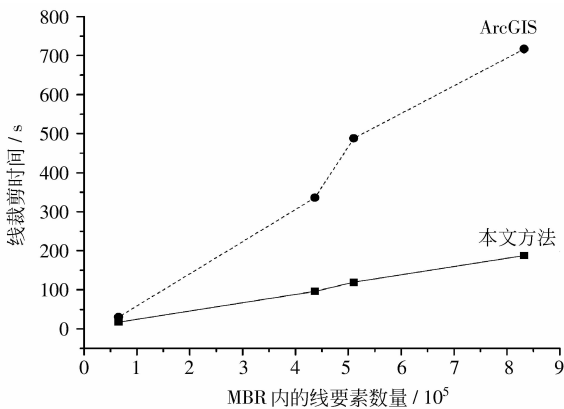


图 9 线面叠加功能时间耗费随线数量的变化趋势

Fig. 9 Transformation curve of line-polygon overlay time cost on line number

长。本文方法的时间约为 ArcGIS 时间的 25% ~ 60% 左右,尤其当要素数较多时,本文方法优势更为明显,约为 ArcGIS 的 25% 左右。算法的正确性方面,要素个数的差值百分比小于 0.01%,可以满足计算需求。

3 面面叠加

3.1 相关工作

因为矢量地图叠加分析的重要性和困难性,有一系列矢量地图叠加分析技术被提出^[20-28]。其中部分是基于拓扑结构的^[20-21],其余的是基于几何对象的。而空间数据最流行的组织方式是基于空间对象的(比如 Shape File 格式),并且基于对象的数据和基于拓扑的数据的转化需要额外的时间耗费,因此,基于对象的矢量地图叠加分析是目前最流行的,是研究的热点与难点。

基于对象的矢量地图叠加分析方法有两种思路:一是通过一一循环计算^[28],采用一些空间获取方法(空间索引^[22]),二是将所有对象视为一个整体来进行计算^[23-27],常采用平面扫描法。一一循环的方法一次仅仅计算两个对象的叠加,任何一个对象可能多次参与运算,因此效率较低。采用整体扫描计算的方法^[24-27]中,都是通过扩展传统的扫描线方法,定义在不同类型交点(bend, start, end or intersection)处的动作。只是,这些方法对特殊情形的处理仍然不够全面。对于这类整体计算的方法,内外环的匹配以及属性继承是必不可少的。在上述使用平面扫描的方法中,没有给出如何实现这两步计算;在文献[23]中,采用的是 point-in-ring-test 为内环找到对应的外环;在文献[20]中,同样采用了 point-in-polygon-test 来进行源多边形的查找,这些都需要额外的计算。

本文中提出一个鲁棒的矢量地图叠加分析法。该方法是一种整体计算的方法。首先通过扫描线方法一次扫描计算得到所有的交点;并且在交点计算过程中,改进了传统的扫描线方法,一次对过同一点的所有线段进行计算;通过对过同一点的线段的完备的分类,涵盖了所有特殊类型的交点,增强了算法的鲁棒性。此外,得到的交点记录了产生交点的多边形的 ID 信息,并将此信息延伸到构造的结果环以及多边形中, ID 简化了内外环匹配以及属性继承过程。

3.2 算法步骤

文本方法有如下几个步骤:

1) 初始化多边形, 即矫正多边形的方向外环逆时针内环顺时针。对于某些数据格式如 Shape File, 这一步不是必需的。

2) 交点计算

改进了传统的扫描线算法, 在任何一个事件点处, 得到所有过此事件点的线段, 并对这些线段进行分组分类, 然后进行局部计算。对于这些过同一点的线段, 首先根据其所属的多边形集合的不同分为两组, 然后对每一组根据其所属的多边形以及环的 ID 的不同分为若干部分。这样, 每部分至多包含两条线段, 因此过同一点的线段相交可能有以下 4 种情形, 如图 10 所示, 粗线条为 Sub, 细线条为 Clip。

Sub1_Clip1: 两个多边形各有一条线段过当前点, 这是最常见的交点类型。Sub1_Clip2: 即当前交点为 Clip 的一个端点。根据 Sub 的线段与 Clip 的两条过当前端点的边的位置关系的不同, 有 3 种分

布类型, 其中第 3 种又有 3 个不同的种类。Sub2_Clip1 与 Sub1_Clip2 是类似的。Sub2_Clip2: 即当前点是同时为 Sub 和 Clip 的端点。

对于 Sub1_Clip1 的情形, 因为多边形的边的方向已经初始化, 因此只需要计算 Sub1 线段的末端点位于 Clip1 线段的左侧还是右侧, 得到交点的出入属性。如图 11(a) 所示, 交点为一个入点, Sub 进入 Clip 内部。即通过计算线段的左右侧得到线段所属多边形的内侧外侧区域。而对于有两条线段的情形, 如图 11(b) 所示, 两条线段将边附近的区域划分为两部分(三角形部分和弧形部分), 一部分为多边形的外部区域, 另一部分为内部区域。任意一条或者两条边若跨越两个区域则必然跨越多边形, 也就是具有出入属性, 如上 Sub2_Clip1, Sub1_Clip2 和 Sub2_Clip2 中的情形 2 所示。一般而言, 若此两线段的方向(即构成的三角形的方向)为逆时针, 则三角形区域为内部区域; 否则弧形区域为内部区域。然后判断线段的末端点位于哪个区域内。

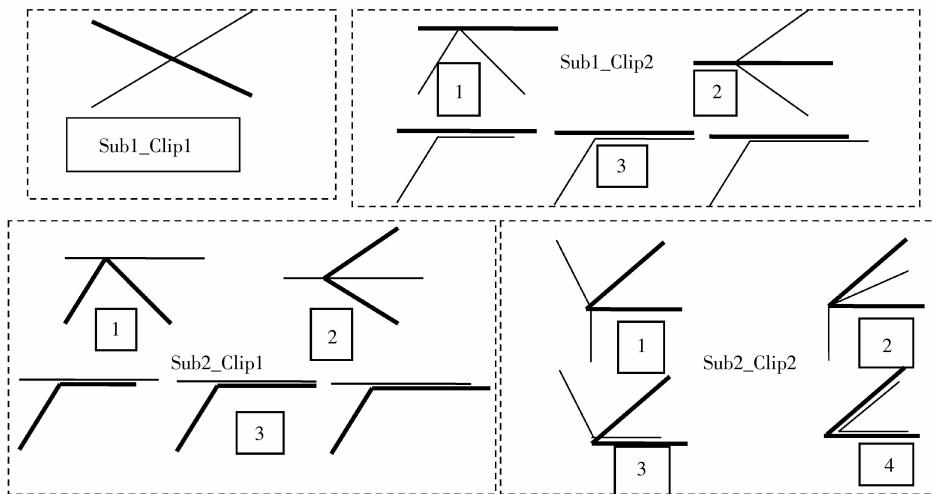


图 10 面面叠加中两个多边形交点的所有可能情形

Fig. 10 All the possible cases of intersection points in polygon overlay

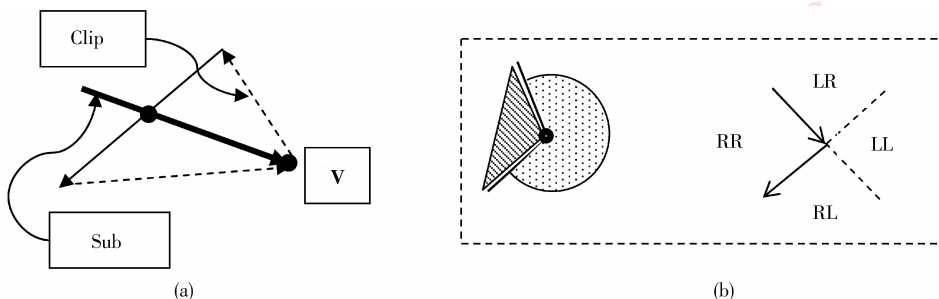


图 11 交点属性计算方法

Fig. 11 Computing method for entry/exit property

对于不跨越两线段,而仅仅与两线段“擦肩而过”的情形 (Sub2_Clip1, Sub1_Clip2 和 Sub2_Clip2 中的情形 1), 这样的点不作为交点。对于存在重叠的情形 (Sub2_Clip1, Sub1_Clip2 和 Sub2_Clip2 中的情形 3), 也是通过计算不与边重叠的端点 (对于末端点恰好位于 Clip 线段上的情形, 则选择始端点进行计算, 最后将结果反置即可) 位于两线段所在的多边形的内侧还是外侧。对于 Sub2_Clip2 的情形 4, 此时两组线段在端点部分完全重合, 整体而言, 不是出点也不是入点, 可以记录此交点, 并标记其为特殊的完全重合的交点。在这样的交点位置时, 无须采取任何动作, 这样的交点也不参与出入交点交替的规律。

以上分类涵盖了所有可能的交点类型, 并且对每种类型都提供了合理的计算方法。通过以上计算方法, 能够正确得到两个多边形的所有交点及其出入属性。对于多边形边重叠、交点在端点等特殊类型, 在不改变数据的前提下, 精确、鲁棒地得到了所有交点。

如果我们需要得到差的操作结果 ($A - B$), 那么需要记录 B 的邻接点。因此如果过同一点的所有线段属于同一个多边形集合, 那么当前的交点一定是一个 Touching 点, 将其加到 touching_Intersection_Sub 或 touching_Intersection_Clip, 即同色“交点”集合。这些交点将用来在求差操作时, 将线段打断, 以删除重叠且反向的线段。否则, 过同一点的线段即属于两个多边形集合, 此时的点将不会作为交点插入到同色交点集合中去。在此点, 只需要计算 Sub 和 Clip 的交点即可。

3) 处理没有产生交点的环

首先需要计算当前没有产生交点的环是否位于对应的多边形内, 通常可用环的代表点与多边形关系判断来实现。然后根据点与多边形的关系、当前的具体操作、当前环为内环还是外环来决定对当前环的操作; 对于不对称的求差操作, 还需要区分环是属于 Sub 还是 Clip。具体处理方法如表 5 所示。○即环不是结果的组成部分, Outer 或者 Inner 表示此环作为外环或者内环插入到结果集合中。对于求差操作的 Clip 多边形对应的环, 因为外环要作为内环插入, 内环要作为外环插入, 插入到结果内环集合或者外环集合时, 常常首先反置环的端点顺序, 使得结

果环的方向符合一般的规则。叠加的并操作即交、差、差 3 部分操作之和, 而表 5 所示的几何并为外轮廓。

表 5 没有产生交点的环的处理策略

Tab. 5 Handing method of rings with no intersections

没有产生交点的环	根据不同的操作是否添加到结果中			与对应多边形的位置关系
	交	几何并即外轮廓	差	
Subject Polygon 的外环	Outer	○	○	内部
	○	Outer	Outer	外部
Subject Polygon 的洞	Inner	○	○	内部
	○	Inner	Inner	外部
Clip Polygon 的外环	Outer	○	Inner	内部
	○	Outer	○	外部
Clip Polygon 的洞	Inner	○	Outer	内部
	Inner	Inner	○	外部

4) 根据交点构造环

首先即对交点排序, 这样可以很容易得到当前交点的下一个交点。一般而言, 求交操作时, 因为任何一个交是由两个多边形构成的, 排序时可以按照两个 ID 综合排序, 之后即在不同出入属性交点处, 根据不同的操作, 或者向前, 或者向后, 或者切换到产生交点的另外一个多边形。而求差时, 则一般按照一个 ID 进行排序, 即 $A - B$ 时, 按照 A_1, A_2, \dots 的顺序对所有交点进行排序, 属于同一个 A 的交点则按其环上的出现的顺序顺次排序; 还需要用同色“交点”将环打断, 删除重叠且反向的折线, 然后对所有具有相同 ID 的打断的折线进行首尾相连, 生成差的结果。根据交点得到环之后, 即根据环的方向 (与输入多边形环的方向规则一致), 将其加入到临时的内环或者外环集合中去。

5) 内外环匹配及属性继承

因为构造的环都记录了 ID 信息, 我们可以利用 ID 的匹配来完成内外环的匹配, 只有 ID 相等的环才属于同一个多边形。而 ID 又是从输入多边形继承而来, 因此可以利用 ID 来查找其源多边形继承属性。

算法的时间复杂度即为求交点部分的算法复杂度, 也就是扫描线的算法复杂度。

3.3 算法对比测试

对比测试是基于真实世界的地理数据的, 两个面图层的要素数分别为: 122 552 和 2 525。所有的

测试都是相同的环境下进行的。

首先,与一一循环(采用 r-tree 索引,以多边形裁剪为基础)计算方法进行了对比。对比结果如表 6 所示,计算时间仅仅是几何计算的时间。在求交点时,分别采取了不同的求交点方法。

表 6 与一一循环方法的对比,采用不同的求交点方法
Tab.6 Comparisons with the one-by-one method, adopting different intersection computing methods

求交点方法		几何计算时间/s
本文方法	利用扫描线整体计算交点的方法	63
一一循环方法	对每一个多边形查询索引,得到相关的多边形,然后一次扫描,求得交点	97
	对每一个多边形查询索引,得到相关的多边形,然后一一计算,扫描线法求交点	352
	对每一个多边形查询索引,得到相关的多边形,然后一一计算,暴力法求交点	1 608

如表 6 所示,本文方法效率高于一对一或者一对多循环的方法,即使采用了高效的平面扫描求交点方法。具体原因是:不论是一对一计算还是一对多计算,基于一个多边形的计算次数不仅仅是一次;而采用一次扫描的方法,对每个多边形仅仅扫描一次。

辅以必要的读写过程,实现了叠加分析功能,并与 ArcGIS 进行了对比,结果如表 7 所示。根据测试数据可得,就整个叠加功能而言,本文方法(Vega)效率要优于 ArcGIS 的对应功能。

表 7 面面叠加分析功能对比测试表
Tab.7 Comparisons of polygon overlay

操作	结果要素数	时间耗费/s		采用的方法
		总体时间	几何计算	
1 Intersect ($D1 \cap D2$)	157 454	75	63	Vega
	157 454	144	/	Arc
2 Difference ($D1 - D2$)	411	63	61	Vega
	411	132	/	Arc
3 Difference ($D2 - D1$)	2 703	59	56	Vega
	2 703	134	/	Arc
4 Symmetric difference ($D1 - D2, D2 - D1$)	3 114	64	61	Vega
	3 114	133	/	Arc
5 Union ($D1 \cup D2$)	160 568	86	75	Vega
	160 568	144	/	Arc

4 结 论

本文在综合分析了现有的综合性研究以及单个主题的研究基础上,提出了一整套矢量地图叠加分析算法,包括点面、线面和面面叠加分析算法。这些算法都已经实现,并应用于实际的系统中,取得了良好的效果。

在本文的基础上,还有以下工作需要做进一步的研究与实现:

1) 不同并行环境下(单机多核、集群等)算法的并行实现思路。

2) 计算的智能性研究。对于一些分析方法,当数据规模、数据特点不同时,各个算法具有不同的相对效率高低,因此如何在面对特定的数据时,根据数据的特点、规模等,计算得到合适的参数,并根据参数选择最高效的分析方法。

参考文献 (References)

- [1] Huang Xingyuan, Ma Jinsong, Tang Qin. Introduction on Geographical Information System [M]. Beijing: Science Press, 2001. [黄杏元, 马劲松, 汤勤. 地理信息系统概论 [M]. 北京: 高等教育出版社, 2001.]
- [2] Guo Renzhong. Spatial Analysis [M]. Beijing: Science Press, 2008. [郭仁忠. 空间分析 [M]. 北京: 高等教育出版社, 2008.]
- [3] Wan Feng. Spatial analysis algorithms and implementation [D]. Wuhan: Wuhan University, 2002. [万丰. 空间分析算法与实现 [D]. 武汉: 武汉大学, 2002.]
- [4] Dong Peng. Research on efficient query and analysis system of distributed spatial information [D]. Beijing: Chinese Academy of Sciences, 2004. [董鹏. 分布式空间信息的高效查询与分析系统研究 [D]. 北京: 中国科学院研究生院, 2004.]
- [5] Cheng Zhenlin. Research on key algorithms of vector spatial analysis in VegaGIS [D]. Beijing: Chinese Academy of Sciences, 2006. [程振林. VegaGIS 中矢量空间分析的关键算法研究 [D]. 北京: 中国科学院研究生院, 2006.]
- [6] Zhao Hongchao. Research and implementation on spatial relation [D]. Beijing: Chinese Academy of Sciences, 2006. [赵红超. 空间关系的研究和实现 [D]. 北京: 中国科学院研究生院, 2006.]
- [7] Zhang Wenyi. Buffer and overlay analysis in GIS [D]. Changsha: Zhongnan University, 2007. [张文艺. GIS 缓冲区和叠加分析 [D]. 长沙: 中南大学, 2007.]
- [8] Li Mingcong. Design and realization of prototype system based on spatial overlay analysis theory and technology [D]. Dalian: Dalian University of Technology, 2007. [李明聪. 基于叠置分析

- 技术的原型系统的设计与实现[D]. 大连:大连理工大学, 2007.]
- [9] Peng Jiannan. Research on GIS spatial analysis method [D]. Changchun: Jilin University, 2007. [彭剑楠. GIS 空间分析方法研究[D]. 长春:吉林大学,2007.]
- [10] Mao Dingshan. Research on vector data overlay algorithm based on computational geometry [D]. Taian: Shandong University of Science and Technology, 2007. [毛定山. 基于计算几何的矢量数据叠加分析算法研究[D]. 泰安:山东科技大学,2007.]
- [11] Li Xiaojun. Research on GIS spatial analysis method [D]. Hangzhou: Zhejiang University, 2007. [李晓军. GIS 空间分析方法研究[D]. 杭州:浙江大学,2007.]
- [12] Wang Liqing. Research on computational geometry algorithms in GIS [D]. Changsha: Zhongnan University, 2008. [王丽青. GIS 中的计算几何算法研究[D]. 长沙:中南大学,2008.]
- [13] Haines E. Point in polygon strategies [C]//Academic Press Graphics Gems Series, Graphics gems IV. San Diego, CA: Academic Press Professional, Inc. ,1994;24-46.
- [14] Li J, Wang W, Wu E. Point-in-polygon tests by convex decomposition [J]. Computers and Graphics, 2007, 31 (4): 636-648.
- [15] Zalik B, Kolingerova I. A cell-based point-in-polygon algorithm suitable for large sets of points [J]. Computers and Geosciences, 2001, 27(10) :1135-1145.
- [16] Huang Y Q, Liu Y K. An algorithm for line clipping against a polygon based on shearing transformation [J]. Computer Graphics Forum, 2002, 21(4) : 683-688.
- [17] Li Jing, Wang Wencheng, Wu Enhua. Line clipping against a polygon based on convex decomposition [J]. Journal of Computer-aided Design and Computer Graphics, 2007, 19(4) : 425-429. [李静,王文成,吴恩华. 基于凸剖分的多边形窗口线裁剪算法[J]. 计算机辅助设计与图形学学报,2007, 19(4):425-429.]
- [18] Bentley J L, Ottmann T A. Algorithms for reporting and counting geometric intersections [J]. IEEE Transactions on Computers, 1979, 28(9) :643-647.
- [19] Andrews D S, Snoeyink J, Boritz J, et al. Further comparisons of algorithms for geometric intersection problems [C]//Proceedings of the 6th International Symposium on Spatial Data Handling, Edinburgh, Scotland: CRC Press, 1994 : 709-724.
- [20] Finke U, Hinrichs K. A spatial data model and a topological sweep algorithm for map overlay [C]//Proceedings of the 3rd International Symposium on Advances in Spatial Databases. Singapore: Springer Berlin / Heidelberg, 1993 : 162-177.
- [21] Van Roessel J W. A new approach to plane-sweep overlay: topological structuring and line-segment classification [J]. Cartography and Geographic Information Systems, 1991, 18 (1): 49-67.
- [22] Dong Peng, Li Jinping, Bai Yuqi, et al. Vector map overlay algorithm based on improved quadtree indexing [J]. Journal of Computer-aided Design and Computer Graphics, 2004, 16(4) : 530-535. [董鹏,李津平,白予琦,等. 基于改进四叉树索引的矢量地图叠加分析算法[J]. 计算机辅助设计与图形学报, 2004,16(4):530-535.]
- [23] Oosterom P. An r-tree based map-overlay algorithm [C]// Proceedings of the 5th European Conference on Geographical Information Systems. Paris, France: EGIS Foundation, 1994 : 318-327.
- [24] Nievergelt J, Preparata F P. Plane-sweep algorithms for intersecting geometric figures [J]. Communications of the ACM, 1982, 25(10) :739-747.
- [25] Kriegel H, Brinkhoff T, Schneider R. An efficient map overlay algorithm based on spatial access methods and computational geometry [C]//Proceedings of the International Workshop on DBMS's for Geographic Applications. Berlin, Germany: Springer-Verlag, 1992;194-211.
- [26] Chan E P F, Jimmy N H. A general and efficient implementation of geometric operators and predicates [C]//Proceedings of the 5th International Symposium on Advances in Spatial Databases. Berlin, Germany: Springer-Verlag, 1997 : 69-93.
- [27] Becker L, Giesen A, Hinrichs K H, et al. Algorithms for performing polygonal map overlay and spatial join on massive data sets [C]//Proceedings of the International Symposium on Advances in Spatial Databases. Hong Kong, China: Springer-Verlag, 1999;270-285.
- [28] Rodrigues V L, Andrade M V A, Queiroz G B D, et al. A more efficient method for map overlay in Terralib [C]//Advances in Geoinformatics, VIII Brazilian Symposium on GeoInformatics. Campos do Jordão (SP). Brazil: Springer, Berlin: Heidelberg, 2006;37-46.