南京航空航天大学

论文集

（二〇〇四年） 第14册

信息科学与技术学院

（第2分册）

南京航空航天大学科技部编

二〇〇五年三月

信息科学与技术学院

042 系

目　　录

序号	姓名	职称	单位	论文题目	刊物、会议名称	年、卷、期	类别
01	黄添强 秦小麟	博士 教授	042	Detecting Outliers In Spatial Database	ICIG 2004 Proceedings	2004.	
02	李立言 秦小麟	博士 教授	042	一种基于双网格的空间数据库实现方法	南京航空航天大学学报	2004.36.01	
03	陈　倩 秦小麟	硕士 教授	042	时空数据库中数据建模的研究	计算机工程	2004.30.20	
04	钟　勇 秦小麟	博士 教授	042	基于高可信体系安全锁协议算法研究	南京航空航天大学学报	2004.36.04	
05	钟　勇 秦小麟	博士 教授	042	数据库入侵检测研究综述	计算机科学	2004.31.10	
06	钟　勇 秦小麟	博士 教授	042	Database Intrusion Detection Based on User Query Frequent Itemsets Mining with Item Constraints	International Conference on Information Security 会议上交流		
07	钟　勇 秦小麟	博士 教授	042	Research on Algorithm of User Query Frequent Itemsets Mining	Third International Conference on Machine Learning and Cybernetics 会议上交流		
08	皮德常	副教授	042	基于动态剪枝的关联规则挖掘算法	小型微型计算机系统	2004.25.10	
09	王立松	讲师	042	基于对象关系的物料清单管理技术研究	中国机械工程	2004.15.24	
10	吴　洁	副教授	042	信息系统群演化的综合环境 SEE 的研究	南京航空航天大学学报	2004.36.01	
11	吴　洁	副教授	042	xmITRAM+: an XML-based tool for management of requirements and architectures	系统工程与电子技术	2004.15.04	
12	吴　洁	副教授	042	An Approach for Systems Evolution	IEEE SMC'2004 Conference Proceedings	2004.01.01	
13	袁家斌	副教授	042	微型飞行器新型极化电磁驱动舵机的研究	南京航空航天大学学报	2004.36.01	
14	叶飞跃 王建东	博士 教授	042	一种快速的自适应频繁模式挖掘方法	控制与决策	2004.19.08	
15	叶飞跃 王建东	博士 教授	042	一种挖掘频繁模式的数据库划分新方法	系统工程与电子技术	2004.26.11	

序号	姓名	职称	单位	论文题目	刊物、会议名称	年、卷、期	类别
16	叶飞跃 王建东	博士 教授	042	基于超结构的分布式系统的关联规则挖掘算法	小型微型计算机系统	2004.25.12	
17	叶飞跃 王建东	博士 教授	042	一种基于多数组的频繁模式挖掘算法	计算机工程与应用	2004.40.34	
18	叶飞跃 王建东	博士 教授	042	基于哈希链结构的频繁模式挖掘	计算机工程与应用	2004.40.11	
19	皋　军 王建东	硕士 教授	042	一种基于云模式连续型属性离散化的算法	计算机应用	2004.24.02	
20	郑　泉 王建东	硕士 教授	042	基于 FP-树挖掘大数据库的方法及算法 PCM	计算机工程与应用	2004.40.07	
21	毛宇光	副教授	042	用于关系数据库的多值逻辑研究	计算机科学	2004.31.10	
22	杨　宁 毛宇光	硕士 副教授	042	一种基于 Vague 集的模糊关系代数	计算机科学	2004.31.10	
23	单　峰 毛宇光	硕士 副教授	042	基于预报_校正法的汇率预测模型	计算机应用	2004.24.03	
24	薛　芹 毛宇光	硕士 副教授	042	通用备份系统的设计与实现	计算机应用研究	2004.21.12	
25	潘　娜 毛宇光	硕士 副教授	042	SQL 语言中的空值问题	微机发展	2004.14.12	
26	张玲东 毛宇光	硕士 副教授	042	压力容器行业系统中 BOM 的设计及应用	计算机时代	2004.22.09	
27	陈　兵	副教授	042	非对称 VPN 体系研究	小型微型机计算机系统	2004.25.03	
28	谭晓阳	讲师	042	交互式图像检索中相关反馈技术研究进展	南京大学学报（自然科学）	2004.40.05	
29	谭晓阳	讲师	042	Robust face recognition from a single Training image per Person with Kernel-Based SOM-Face	Lecture Notes in Computer science	2004.3173	
30	刘宁钟	讲师	042	基于中点检测的二维条码识别	小型微型计算机系统	2004.25.02	
31	刘宁钟	讲师	042	基于波形分析的二维条码识别	计算机研究与发展	2004.41.03	
32	刘宁钟	讲师	042	基于选代计算的二值波形反卷积	中国图象图形学报	2004.09.10	
33	夏正友	讲师	042	A Novel Policy and Information Flow Security Model for Active Network	Lecture notes in computer science 3073	2004.3073	

序号	姓名	职称	单位	论文题目	刊物、会议名称	年、卷、期	类别
34	夏正友	讲师	042	A Novel Grid Node-by-Node Security Model	Lecture notes in computer science 3251	2004.3251	
35	夏正友	讲师	042	A Novel Artificial Life Ecosystem Environment Model	Lecture notes in computer science 3305	2004.3305	
36	夏正友	讲师	042	An access control for active network	在 ISCC2004 会议上交流		
37	马维华	副教授	042	基于故障检测的新型集中抄表系统硬件设计	小型微型计算机系统	2004.25.04	
38	马维华	副教授	042	互动式教学方法的探讨	中国高等教育研究杂志	2004.10.112	
39	张红斌 马维华	硕士 副教授	042	基于 VXI 总线的某型雷达电路故障诊断系统	武器装备自动化	2004.23.01	
40	王永安 马维华	硕士 副教授	042	虚拟数字示波器驱动程序的设计与实现	现代科学仪器	2004.23.01	
41	陈河堆 高　航	硕士 副教授	042	一种JDBC连接分享的分类方法及其实现	计算机应用	2004.24.12 增	
42	汪　军 高　航	硕士 副教授	042	基于智能卡的互联网接入认证技术	计算机应用	2004.24.12	
43	徐　涛	教授	042	以“计算作为一门学科”的观点构建“计算机导论”课程	中外教育与研究	2004.16	
44	章秋生 徐　涛	硕士 教授	042	基于主动表观模型（AAM）的图像中物体的定位方法研究	计算机应用	2004.24. 增刊	
45	郁　斌 徐　涛	硕士 教授	042	一个基于HMM和特征重叠抽取技术的对象定位方法研究	在CNNC2004会议上交流	2004	
46	陈松灿	教授	042	Robust image segmentation using FCM with spatial constraints Based on New Kernel-Induced Distance Measure	IEEE Trans SMC Part B	2004.34.04	
47	陈松灿	教授	042	Enhanced $(PC)^2A$ for face recognition with one training image per person	Pattern Recognition Letters	2004.25.10	
48	陈松灿	教授	042	Subpattern-based principle component analysis	Pattern Recognition	2004.37.05	
49	陈松灿	教授	042	Making FLDA applicable to face recognition with one sample per person	Pattern Recognition	2004.37.07	

序号	姓名	职称	单位	论文题目	刊物、会议名称	年、卷、期	类别
50	陈松灿	教授	042	Alternative linear discriminant classfier	Pattern Recognition	2004.37.07	
51	杨绪兵 陈松灿	硕士 教授	042	增强的主分量分类器	复旦学报（自然科学版）	2004.43.05	
52	潘志松 陈松灿	博士 教授	042	一般化的灰 SOM 模型及其性能评估	计算机学报	2004.27.04	
53	潘志松 陈松灿	博士 教授	042	原空间中的核 SOM 分类器	电子学报	2004.32.02	
54	谭可人 陈松灿	硕士 教授	042	Robust image denoising using kernel-induced measures	在 ICPR'04 会议上交流		
55	刘　俊 陈松灿	博士 教授	042	Progressive Principal Component Analysis	Lecture Notes in Computer science	2004.3173	
56	周　鹏	硕士	042	正则化的模糊 Ho-Kashyap 分类器	计算机科学	2004.31.10A	
57	李　剑 万麟瑞	硕士 副教授	042	Web Service 在电子商务中的应用	计算机应用与软件	2004.21.02	
58	赵　悦 万麟瑞	硕士 副教授	042	关系管理软件构架与模板的研究	计算机工程与设计	2004.25.11	
59	秦　俭 万麟瑞	硕士 副教授	042	电子报关软件模板研究	计算机应用研究	2004.21.增刊	
60	戴　群	助教	042	打折最小平方 RBF 网络及其时间序列预测研究	东南大学学报	2004.34.06	
61	黄元元	讲师	042	Binary trademark retrieval using shape and spatial feature	Proceedings of SPIE-The International society for Optical Engineering	2003.5286	
62	黄元元	讲师	042	Binary trademark retrieval using entropy and moments	Proceedings of SPIE-The International society for optical Engineering	2003.5286	
63	张道强	讲师	042	A comment on "Alternative c-means clustering algorithms"	Pattern Recognition	2004.37.02	
64	张道强	讲师	042	A novel kernelized fuzzy C-means algorithm with application in medical image segmentation	Artificial Intelligence in Medicine	2004.32.01	

序号	姓名	职称	单位	论文题目	刊物、会议名称	年、卷、期	类别
65	张道强	讲师	042	在核诱导的鲁棒度量下的模糊 C-均值与可能性 C-均值算法	模式识别与人工智能	2004.17.04	
66	张道强	讲师	042	一般多值双向联想记忆模型及其在 IP 地址识别中的应用	应用科学学报	2004.22.03	
67	张道强	讲师	042	Fuzzy-kernel learning vector quantization	Lecture notes in Computer Science	2004.3173	
68	张道强	讲师	042	Semi-supervised Kernel-Based Fuzzy C-Means	Lecture Notes in computer science	2004.3316	
69	陈 蕾 张道强	硕士 讲师	042	基于小世界体系的核自联想记忆模型	计算机科学	2004.31.10.A	
70	沈国华	讲师	042	Role of Meta-Model in Engineering Data Warehouse	南京航空航天大学学报（英文版）	2004.21.04	
71	沈国华	讲师	042	基于数据仓库技术的工程数据管理系统的研究与实现	小型微型计算机系统	2004.25.01	
72	郑洪源	讲师	042	基于 CORBA-WEB 的多代理供应链管理系统的研究	工业技术经济	2004.23.05	
73	鞠炜刚 郑洪源	硕士 讲师	042	基于知识工程与协同工程的分析型电子商务研究	计算机应用研究	2004.09	
74	曹 晖 郑洪源	硕士 讲师	042	CRM 应用方案分析及解决方法	计算机应用研究	2004.01	
75	徐 岩 郑洪源	博士 讲师	042	面向领域工程的 CRP 系统构件化开发建模	吉林大学学报（信息科学版）	2004.22.06	
76	徐 岩 谢 强	博士 讲师	042	构建 Web 基 CRP 协同服务完善企业质量管理	工业技术经济	2004.23.04	
77	徐 岩 谢 强	博士 讲师	042	质量链管理信息系统重构	武汉大学学报（工学版）	2004.37.05	
78	张 萍 谢 强	硕士 讲师	042	ASP.NET 访问数据库的通用方法	计算机应用	2004.24.06	
79	陈 伟 丁秋林	博士 教授	053 042	一种 XML 相似重复数据的清理方法研究	北京航空航天大学学报	2004.30.09	
80	陈 伟 丁秋林	博士 教授	053 042	具有数据清理功能的交互式数据迁移及应用	吉林大学学报（信息科学版）	2004.22.02	
81	陈 伟 丁秋林	博士 教授	053 042	交互式数据迁移系统及其相似检测效率优化	华南理工大学学报（自然科学版）	2004.32.02	
82	陈 伟 丁秋林	博士 教授	053 042	ERP 发展进程中的数据迁移研究	机械科学与技术	2004.23.05	

序号	姓名	职称	单位	论文题目	刊物、会议名称	年、卷、期	类别
83	阎志华 丁秋林	博士 教授	053 042	用蜂群算法实现动态作业车间调度	组合机床与自动化加工技术	2004.02	
84	阎志华 丁秋林	博士 教授	053 042	基于蜂群算法的作业车间调度研究	机械科学与技术	2004.23.09	
85	阎志华 丁秋林	博士 教授	053 042	基于 OAGIS 的制造执行系统的研究	机械科学与技术	2004.23.09	
86	何月顺 丁秋林	博士 教授	042	一种有效的优化数据仓库性能的解决方案	南京航空航天大学学报	2004.36.01	
87	何月顺 丁秋林	博士 教授	042	调整优化 Oracle 9i 数据库的性能	计算机应用与软件	2004.21.06	
88	宋允辉 丁秋林	硕士 教授	042	集成质量管理	小型微型计算机系统	2004.25.04	
89	王有远 丁秋林	博士 教授	053 042	供应链管理信息集成框架	航空制造技术	2004.08	
90	凌兴宏 丁秋林	博士 教授	053 042	基于协商的 Multi-Agent 生产计划与调度系统	机械科学与技术	2004.23.2	
91	殷　平 丁秋林	硕士 教授	042	推理技术在决策支持系统中的应用	计算机应用	2004.07	
92	林中伟 丁秋林	博士 教授	053 042	扩展企业资源计划体系结构研究	计算机工程与应用	2004.03	
93	杜国平	博士	042	显示法证明分析	哲学研究	2004.06	

ICIG 2004 Proceedings

Detecting Outliers In Spatial Database

Tianqiang Huang, Xiaolin Qin
Department of Computer Science and Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, 210016, PR China
tianqianghuang@163.com

Abstract

Detecting outlier in spatial database is important for many KDD applications. Existing works in outlier detection don't distinguish between spatial dimension and non-spatial dimension or have poor efficiency. In this paper, we proposed a new measure to identify spatial outliers. We defined spatial outlier factor (SOF) to detect spatial outliers efficiently, and proposed a algorithm (SOFind) to identify them. SOF can successfully identify significant outliers and filtrate some meaningless outliers but can't do it by other methods. The experimental results show that our approach is effective and efficient.

1. Introduction

This paper focuses on spatial outliers in spatial database, i.e., observations that appear to be inconsistent with their neighborhoods in spatial database. Detecting spatial outliers is useful in many applications of geographic information systems and spatial databases [1,2]. These application domains include transportation, ecology, public safety, public health, climatology, and location based services.

Existing works in outlier detection usually don't constitute the distinction between spatial dimensionality and no-spatial dimensionality, but spatial attribute is different from the other attribute in complexity of spatial data types, spatial relationships, and spatial autocorrelation etc.

Recently, Shekhar and Lu presented a new approach [3,4] to detect spatial outlier, which capture spatial attribute. However, there are several shortcomings. First, the approach of using a statistical test is useful for discovering global outliers but may not be able to discover local outliers, which are likely to be of more interest. Second, this approach supposes multidimensional attributes fit multivariate normal distribution, which is not always tenable.

In this paper, we make the best of spatial information. We defined a new measure, spatial outlier factor (SOF), to detect spatial outlier, and proposed an algorithm (SOFind) to identify it. We analyzed it computational costs, and demonstrated the effectiveness and the efficiency in experiment.

2. The measure of spatial outlier

In this section, we develop a formal definition of spatial outlier, which avoids the shortcomings presented in the Section 1. The key difference between our notion and existing notions of outliers is that we define outlier in impact neighborhood. We assign to each object an outlier factor, which is the degree that the object is being deviating as like LOF presented by Breunig [5]. However, spatial data exhibits spatial autocorrelation and heteroscedasticity, so there are key different between them.

Given a reflexive and symmetric spatial relation R, we can define impact neighborhoods of a location p as follows [6]:

Definition 1: Impact Neighborhood of p, denoted as $IN(p)$, is a set of locations $P = \{p_1, ..., p_k\}$ such that p_i is a neighbor of p, i.e. $(p, p_i) \in R(\forall i \in k)$.

Definition 2: Let $P \in R^{m+n}$, i.e., for any location $p \in P$, p

have m spatial attributes and p have n non-spatial attributes. Let $p, q \in P$, The **Attribute Distance** between p and q, denoted as *A_Dist(p, q)*, is a distance function between p and q in the n dimensional Euclidean space.

Note that A_Dist(p, q) is not a distance between p and q in spatial attribute dimension, but in non-spatial attribute dimensions.

Definition 3: The **Local Density** of p is defined as

$$LD(p)=|IN(p)| / \sum_{o \in IN(p)} A\_Dist(p,o)$$

A_Dist(p, o) is a distance between p and o in n dimensional non-spatial attribute Euclidean space. *|IN(p)|* represents cardinality of Impact Neighborhood of p, i.e., the number of location in neighborhood of p. Intuitively, the Local Density of an object p is the inverse of the average distance based on the neighbors of p.

Definition 4: The **Spatial Outlier Factor** is defined as

$$SOF(p) = \sum_{o \in IN(p)} (LD(o)/LD(p)) \ / |IN(p)|$$

The **spatial outlier factor (SOF)** of spatial object p represents the degree that outlier is deviating its neighbors. It is the average of the ratio of the local density of p and those of p's neighbors. It is easy to see that the lower p's local density is, and the higher the local densities of p's neighbors are, the higher is the SOF value of p.

Resembling the formula of LOF, value of SOF increases as the deviation degree of outlier increases for an object. We can derive a lemma that is similar to the one exhibited in [5] for LOF to show the correctness of SOF for clustered points. Similar to [5], we assume inside the cluster, the maximum distance between neighbors and minimum distance between neighbors are very close in values. Then objects deep inside clusters have values of SOF approximately equal to 1.

Lemma 1:

Let C be a set of objects forming a cluster,

$$Min\_A\_Dist = \min \{A\_dist(p, q) \mid p, q \in C\},$$
$$Max\_A\_Dist = \max \{A\_dist(p, q) \mid p, q \in C\},$$
$$\theta = Min\_A\_Dist / Max\_A\_Dist,$$

Assume θ is close to 1. Let $p \in C$ be an object embedded inside the cluster C. Then SOF(p) is approximately 1.

Proof:(Because of restriction of the layout, we Have omitted the course of proving)

3. Algorithm of detecting outlier and computational complexity

In this section, we provide a computationally efficient algorithm for detecting outliers and then analyzing computational complexity. Every point in spatial point set P corresponds to a vector of attributes in A, i.e., P_i possess attributes $a_i, a_i \in R^q$ (the q dimensional Euclidean space).

Given the location set P correspond to A, neighbor relation R, For each spatial location $p_i \in P$, algorithm questing the Impact Neighborhood $IN(p_i)$; For every Impact Neighborhood, firstly it computes their local density, then computes their $SOF(P_i)$. Finally algorithm sorts the objects and reports the top-n outliers.

Spatial outlier detecting algorithm (SOFind):

Input:

- P={p_1, p_2, . . ., p_n} is a set of spatial points;
- A={a_1, a_2, . . ., a_n} is a set of attributes correspond to the set of spatial points P;
- R is neighborhood relation;
- N is number of outliers

Output: outlier_set.

Method:

```
(1) for (i = 1; i <= |P|; i++){
(2)     p_i = Get_one_locaction(i, P);   /* Select each
                                            location from P */
(3)     Find_Neighbor_Set(p_i, R, P); /* Find neighbor of
                p_i from P, label it and compute number
                of objects in neighborhood of p_i. */
        }
(4)for (j = 1; j <= |IN|; j++){    /* |IN| is number of
                                      Impact Neighborhood */
(5)     for (k = 1; k <= Num_j; k++){   /* compute the
                local density of every point in Impact
                Neighborhood, Num_j is number of the j-th
                neighborhood*/
(6)         for (m = 1; m <= Num_j; m++){   /* compute
                the summary of attribute distance
(7)         Sum_ A_Dist(k)= Sum_ A_Dist(k) +
                A_Dist(P_k, P_m);
            }
(8)         LD(k) = (Num_k / Sum_ A_Dist(k));   /*
                        compute the Local Density of p_k
(9)     Sum_ LD(j) = Sum_ LD(j) + LD(k);
```

```
    }
(10)  for(n=1; n<= Num_j; n++){
(11)  SOF(p_j) = (Sum_ LD(j) / Num_j)/LD(P_j);
(12)  if (SOF(p_j) in Top 5) Add_Element(outlier_set, P_j );
       /* Add the element to outlier_set
       }
  }
```

The algorithm time complexity can divide two partitions, one is the time in partitioning Impact Neighborhood, and the other is computing SOF. The time complexity of computing Impact Neighborhood is then based on that of neighborhood query. For the neighborhood query, there are two choices. We can use a grid-based approach, which processes neighborhood query in constant time if the grid directory resides in memory, leading to a complexity of O(n). If an index structure (e.g. R-tree) exists for the spatial data set, spatial index can be used to process neighborhood query, which has complexity of O(*logn*), leading to a complexity of O(*nlogn*). Suppose spatial database was divided *k* Impact Neighborhoods, so average number of Impact Neighborhood is *n/k*. The computation of SOF costs $O(k*(n/k)^2)$, i.e., $O(n^2/k)$. In summary, the total computational cost of the algorithm is $O(n) + O(n^2/k)$ for grid-based structure, i.e., $O(n^2/k)$, or $O(nlogn) + O(n^2/k)$ for index-based structure.

4. Experiments

The criteria evaluating outlier detection approaches can be divided into two parts: effectiveness and efficiency. First, we use artificial data to explain effectiveness of our approach. Second, we use real-world spatial data to verify the effectiveness. Experiments show that our ideas can be used to successfully identify significant outliers and filtrate some meaningless outlier but cannot done by other methods (for example: LOF). Last, we use large numbers of environmental data to test, which display our approach is efficient. Because of restriction of the layout, we only present the second experiment.

The real-word database was used in the second experiment. We computed the local outliers for the database of environmental noise monitor information from the Environmental Information Center of the Environment Protection Bureau in Fujian province. The database contains lots of records and many fields. For simplifying experiment we only select 286 recorders and keep six fields in one-day record on November 12, 2000. These fields are code of monitor station, grid number (represents geo-spatial location), and monitor values (Leq, L10, L50, and L90. These are noise descriptors, which represent average noise level, noise levels exceeded 10%, noise levels exceeded 50% and noise levels exceeded 90% of the time respectively).

We use three approaches to detect outlier by SOF, LOF [5], and SLZ [3, 4]. We referred to the method that proposed by Shekhar, Lu and Zhang as SLZ. Detecting results display in Table 1, Table 2, and Table 3.

Table 1. Outliers detected by SOF

The code of monitor station	Grid number	Leq	L10	L50	L90
3501267	268	56.7	59	54	52
3501072	19	70.6	74	68	65
3501144	84	69.1	71	69	55
3501143	83	65.9	69	62	60
3501004	117	59.1	61	56	53
3501213	255	77	68	61	57

Table 2. Outliers detected by LOF

The code of monitor station	Grid number	Leq	L10	L50	L90
3501144	84	69.1	71	69	55
3501072	19	70.6	74	68	65
3501231	245	60	68	59	61
3501267	268	56.7	59	54	52
3501244	8	70	55	43	65

Table 3. Outliers detected by SLZ

The code of monitor station	Grid number	Leq	L10	L50	L90
3501144	84	69.1	71	69	55
3501267	268	56.7	59	54	52
3501072	19	70.6	74	68	65
3501213	255	77	68	61	57
3501143	83	65.9	69	62	60
3501004	117	59.1	61	56	53

We computed the LOF values in the MinPts range of 20 to 30. Below we discuss all the local outliers with LOF > 1.5. SLZ detected the top six outliers at a confidence interval of 95 percent. According these outliers we look up relative datum and consult these with domain specialist.

The six outliers detected by SOF are all correct and have actual meaning in environmental noise monitor, and SLZ does the same, but LOF identified two incorrect outliers (3501231, 3501244) and SLZ run more computational cost. These displayed in Table 4. Computational cost, which contains CPU and I/O, was compared in the table 5. All experiments were conducted on a Pentium-4-2.2G machine with 256 MB of RAM and running Windows XP professional.

Table 4. Performance comprise with SOF, LOF, and SLZ

	Numberof outliers	Correctness	Cost (sec)
SOF	6	100%	4.5
LOF	5	60%	4.9
SLZ	6	100%	6.1

For comparing computational cost, we used one-week, one-month, six-month, and one-year recorders of the environmental noise database, which contains 2002, 8580, 52052 and 104390 recorders respectively. An R-tree indexing structure is provided for speeding up the neighbor queries. R-tree is chosen because it is an index structure for efficient query processing of high-dimensional data. Their performance comprised as Figure 1. We can see that SOF scheme have better time cost than the other.

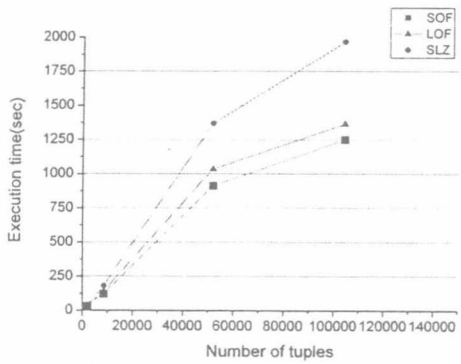


Figure 1. Execution time of LOF, SLZ and SOF for the environmental noise

5. Conclusions

According explosive growth of spatial data and widespread use of spatial databases, finding spatial outliers is an important task for many KDD applications. Existing work in outlier detection doesn't constitute the distinction between spatial dimension and no-spatial dimension, which causes outliers are missed or detecting meaningless outliers. We introduce the notion of the spatial outlier factor (SOF), which effectively meaningful identify outliers. Experimental results display that our scheme is effectiveness and efficiency. Contributions as follows:

- A measure for identifying the degree of each object being an outlier is presented, which is called SOF. This measure captures both spatial autocorrelation and spatial heteroscedasticity, so it can identify more meaningful local spatial outlier.
- We present an efficient algorithm for mining spatial local outliers, which based on our measure.
- We analyze that this approach is more effective and efficient than previous approaches, and experimental results show it is true.

For future work, we will integrate the SOFind algorithm more tightly with Impact Neighborhood query to make the detecting process more efficient.

References

[1] S.Shekhar and S.Chawla A Tour of Spatial Databases. Prentice Hall, 2002.

[2] S.Shekhar, S. Chawla, S. Ravada, A. Fetterer, X. Liu, and C,T. Lu. "Spatial Databases: Accomplishments and Research Needs", IEEE Transactions on Knowledge and Data Engineering, 11(1), 1999, pp. 45-55.

[3] S. Shekhar, C.T. Lu, and P. Zhang. "A unified approach to detecting spatial outliers. GeoInformatica", 7(2), 2003, pp. 139–166.

[4] C.T. Lu, D. Chen, and Y. Kou. "Detecting spatial outliers with multiple attributes", In Proceedings of 15th IEEE Inter -national Conference on Tools with Artificial Intelligence (ICTAI 2003), 3-5 November 2003, Sacramento, California, USA, IEEE Computer Society, 2003, pp. 122–128.

[5] M.M. Breunig, H.P.Kriegel, R.T.Ng, and J. Sander. "LOF: Identifying density-based local outliers", In: Proceedings of SIGMOD'00, Dallas, Texas, USA, 2000, pp. 427 - 438.

[6] S. Shekhar and Y. Huang, "Discovering Spatial Co-location Patterns: A Summary of Results", Proc. of 7th International Symposium on Spatial and Temporal Databases (SSTD01), L.A., CA, 2001.

第36卷第1期
2004年2月

南京航空航天大学学报
Journal of Nanjing University of Aeronautics & Astronautics

Vol. 36 No. 1
Feb. 2004

一种基于双网格的空间数据库实现方法

李立言，秦小麟
（南京航空航天大学信息科学与技术学院，南京，210016）

摘要：双网格是一种基于离散几何的空间对象表示方法。与使用整数坐标系统的Realms不同，双网格使用了有限精度的有理数坐标系统。本文分析了Realms和双网格这两种空间对象表示方法对空间拓扑性的影响以及各自的优缺点，讨论了基于双网格和主存技术实现空间管理子系统的方法。本文还介绍了空间管理子系统与PostgreSQL的集成技术，并设计和实现了基于双网格的空间分析DBMS-SADBS Ⅱ。该系统能够有效地管理空间数据和非空间数据，并具有ROSE代数定义的全部空间分析操作功能。

关键词：数据库管理系统；计算几何；地理信息系统；双网格；Realms；空间数据库
中图分类号：TP311.132.3；TP311.12　　**文献标识码**：A　　**文章编号**：1005-2615(2004)01-0103-05

Dual Grid Approach to Implementing Spatial Databases

LI Li-yan, *QIN Xiao-lin*
(College of Information Science and Technology,
Nanjing University of Aeronautics & Astronautics, Nanjing, 210016, China)

Abstract: Dual grids are an approach to expressing spatial objects based on discrete geometry. Different with a realm using integer coordinate system, the dual grids adopt finite precision rational coordinate system. This paper analyzes the difference between the two approaches based on the realm and dual grids, and compares their effects on spatial topological correctness. The implementation methods for spatial management subsystem based on the dual grids and main memory techniques are discussed. The integration technology for spatial management subsystem and PostgreSQL is presented, and a spatial analysis DBMS, SADBS Ⅱ, based on dual grid is implemented. The system can manage spatial and non-spatial data effectively, and has all spatial analysis functions defined in ROSE algebra.

Key words: DBMS; computational geometry; GIS; dual grids; Realms; spatial database

传统空间数据库在空间操作能力和健壮性方面的缺陷已越来越明显。首先，由于基于欧几里德空间的数据表示缺乏足够的约束，导致空间对象之间的拓扑关系分析十分复杂，而空间运算过高的复杂度限制了空间操作的能力。更为严重的是，由于计算机中的浮点数在表示精度上有限制，无法准确地映射欧几里德空间，使空间运算的健壮性得不到保证。例如，要表示二维空间中的一个点对象，就用两个有限精度的浮点数来近似表示它的坐标；线对象中的每条线段用两个点对象来分别表示端点；而区域对象中的每个多边形则用连续的端点对象来表示。这种表示方法将使得某些空间操作无法满足封闭性，因为两个线段的交点坐标完全有可能超出系统所用的浮点数的精度范围。为了满足封闭性，就只能近似表示，从而导致求出的近似交点有时会带来拓扑正确性问题，可能会出现“高速公路与河的交点（桥）不在河上”这样的错误。

传统的空间数据库系统没有解决这个问题，而

基金项目：国家自然科学基金（49971063）资助项目；江苏省自然科学基金（BK2001045）资助项目。
收稿日期：2002-11-20；**修订日期**：2002-12-31
作者简介：李立言，男，硕士研究生，1978年3月生；秦小麟，男，教授，博士生导师，1953年6月生，E-mail：qinxcs@nuaa.edu.cn。

是把问题留给了应用程序。Illustra 的二维空间扩展包提供了空间数据类型和最基本的组合操作，但没有相关的空间操作功能[1]；Oracle 的空间扩展包虽然提供了一些空间操作函数，却只能返回运算结果的近似值[2]。

为了解决空间运算的健壮性问题，Güting 和 Schneider 提出了Realms[3]和基于Realms 的ROSE 代数[4]。Realms 使用离散的网格空间来代替欧几里德空间，所有的空间对象都定义在网格上，其空间运算一致性和封闭性由一组约束和空间调整来保证。不过空间调整有明显的副作用，使得这种技术难以广泛应用。在此基础上，Lema 和Güting 又提出了“双网格(Dual grid)[5]”的概念，使得空间调整的复杂性大大降低。

作者先后实现了基于Realms 的空间数据库原型系统SADBS[6]和基于双网格的空间数据库原型系统SADBS Ⅱ，本文主要介绍基于双网格的空间管理子系统实现方法，以及空间管理子系统与可扩充数据库集成实现空间分析DBMS(数据库管理系统)的技术。

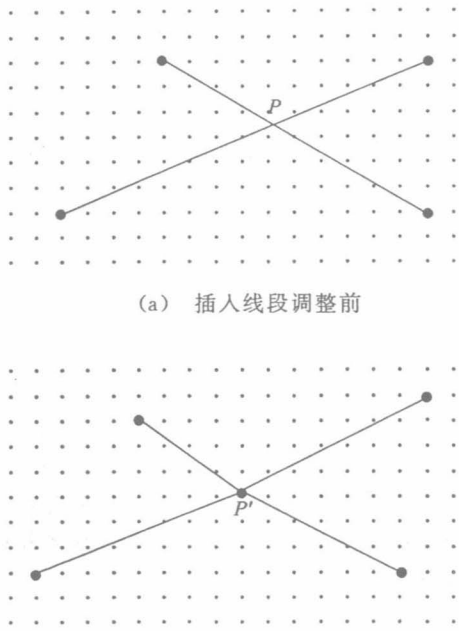
1 空间数据的表示与操作

1.1 Realms 和ROSE 代数

Realms 是建立在一个$N\times N$离散网格上的点和线段的集合[3]，其特性是：任意两条线段均不相交(除了端点相交外)，任意点不能在任意线段当中(至多在端点上)。ROSE 代数定义了Points，Lines 和Regions 三种空间对象，同时定义了较完善的空间操作[4]。由于所有的空间对象和空间操作都定义在Realms 上，且满足Realms 的约束条件，因此能够保证空间运算的一致性和封闭性。

传统空间表示方法的拓扑误差问题主要是因为没有足够的精度表示线段的交点，而在Realms 空间中，线段不会相交，从而保证了空间拓扑正确性。实际上，为了满足Realms 中线段不相交的约束条件，空间对象必须先通过适当的空间调整才能被加入到Realms 空间中去。如图1 所示，两条线段相交于点P，先把交点P调整到临近的网格点P'上，再把原来的两条线段分裂成4 条线段，就可以满足Realms 约束。

实际中的空间调整非常复杂，如图1 所示，线段在调整后改变了位置，这会引起线段与在它附近的空间对象之间的拓扑关系发生改变。要保证空间对象之间的拓扑关系正确并防止线段在多次调整后发生漂移，必须考虑线段的闭包(线段的闭包是指紧邻线段的网格点的集合)。还需要加强Realms 约束，禁止线段调整到闭包以外的网格点上，并且不允许点在线段的闭包上，否则就按照点在线段当中处理。而当线段调整后，新的线段又可能与Realms 中的原有线段相交，从而引起连锁调整。



(a) 插入线段调整前

(b) 插入线段调整后

图1 Reamls 中线段的插入与调整

在作者最初实现的SADBS 原型系统中，由于空间调整的复杂性及其副作用而没有完全实现Realms 约束，空间数据的合法性由应用层保证[6]。这样虽然保证了执行性能，却破坏了空间数据库系统的完整性，降低了其实用性。

1.2 双网格技术

Realms 的主要优点得益于无交点特性，以及用整数表示带来的无拓扑误差特性。但其主要缺点也是由于整数表示，使得不在网格点上的线段交点无法准确表示，从而引起复杂的空间调整。要解决这个问题，需要找到一种空间对象的离散表示方法能够准确表示线段的交点，这样就可以避免改动空间对象的值。

双网格技术是把空间对象从以整数为坐标的Realms 空间，扩展到以有理数为坐标的双网格空间[5]。由于有理数对于加、减、乘、除四则运算满足闭包性，因此如果线段的端点坐标都是有理数，则两条线段的交点坐标仍然是有理数。

在双网格的基础上，如图1 所示的情况就无需调整线段交点的位置，只需要把原来的两条线段在

P 点处分裂成4条线段。由于没有改变线段的位置，因此不会出现拓扑关系错误和线段漂移，更不会发生连锁调整。

计算机表示的有理数是有一定范围的，要保证线段交点坐标是可表示的，还需要对线段作进一步的约束。计算机中表示有理数一般采用分数形式，即使用两个整数分别表示分子和分母，这两个整数的范围就决定了有理数的表示范围。设分子分母分别用 n 和 m 个比特的整数来表示，有理数的表示范围记为 Rational(n, m)，则双网格中点的值域就是所有能用 Rational(n,m) 表示其坐标的点的集合；双网格中的线段除了端点坐标能用 Rational(n,m) 表示外，还必须满足以下约束[5]：线段所在的直线方程可以表示成 $A*x+B*y=C$，其中 A,B,C 为整数，$|A|$ 和 $|B|$ 小于 $\sqrt{2^{m-1}}$，$|C|$ 小于 $\frac{2^{n-1}}{\sqrt{2^{m-1}}}$。

在空间数据库中逐一验证线段的直线方程将是非常复杂的，因此如果原始的空间数据没有对线段的直线方程进行约束，则应该提高双网格的精度来满足其约束[5]。

双网格技术运用了有限精度的计算几何知识[7]，改变了 Realms 的基础，既保留了无交点特性和无拓扑误差特性，同时在很大程度上解决了空间调整的效率问题。从而能够确保新插入的空间对象至多只会引起数据库中某些空间对象的表示发生变化，而不会调整它们的值。因此空间对象的一致性得到保证，同时也避免了连锁调整。

2　基于双网格的空间管理子系统

2.1　空间数据组织

最基本的空间数据是点对象。原先基于 Realms 的空间管理子系统使用两个无符号32位整数来表示点的坐标值，现在需要改成用两个有限精度的有理数来表示，有理数的分子分母需要用位数更高的整数来表示。作者使用了 C++Boost 库中的有理数类[8]，使用无符号64位长整数来表示分子和分母，这样的双网格精度可以接受长度为20个比特的整数坐标。其他的空间对象都建立在点对象之上，因此所有的空间数据结构都是基于有理数坐标的。除此以外，具体的空间数据组织形式都与基于 Realms 空间管理子系统一样[6,9]。

空间分析功能需要用到坐标值的比较和运算，由于 Boost 库中的有理数类已经重载了加、减、乘、除等运算符，因此也无需做出修改，按照 ROSE 代数中的分析函数，直接以有理数代替整数进行运算即可。

在 Realms 中所有的交点都必须对应一个网格点，否则就会得到错误的输出。为了有效实现基于双网格的空间管理子系统，原有在 Realms 上的一些约束必须保留。若限制输入数据使用长度为 n 个比特的整数坐标，双网格精度采用 Rational($3n+3$, $2n+3$)，则无需检查线段的直线方程就可满足双网格约束[5]；然后再使用类似空间调整的方法，在对象插入数据库时求出所有交点，并显式表示出来，就可以满足 Realms 约束。这里所用到的空间调整不会改变空间对象的值，也不会引起连锁调整，至多只会改变空间对象的表示，因此要简单得多。

在空间对象的输入过程中，先将整数坐标转换成相等的有理数坐标，然后使用 Realms 约束，找出所有线段相交和点在线段上的情况，通过线段分割解决。当输出空间对象时，考虑到用户程序一般没有处理有理数的能力，所以把有理数坐标转换成高精度的浮点数返回给用户。由于空间分析操作都由空间管理子系统完成，用户只需处理空间数据显示问题，有理数转换成浮点数所引起的微小误差不会影响空间对象的拓扑正确性。

在空间管理子系统满足双网格约束和 Realms 约束的基础上，可以有效地进行空间数据的输入、输出和分析操作。由于这种约束是由空间管理子系统保证的，对用户程序没有额外的限制，因此，空间数据的一致性、拓扑正确性，空间数据库的健壮性和效率都得到了有效保证。

必须解决的一个问题是由于输入、输出函数的不对称，可能会破坏数据库系统的完备性。例如，求两个空间区域的公共部分，返回的是一个区域对象，该对象的所有组成元素都是空间数据库中已有的点和线段，显然是满足约束条件的。但是，要把这样一个空间对象插入到空间数据库中却并不容易。这是因为从输出函数返回的空间对象已经是用浮点数来表示坐标了，而输入函数只接受整数坐标形式。为了解决这个问题，我们给每个返回空间对象的空间操作函数增加了另外一个版本，新的函数把输出和输入步骤合并，直接把新产生的空间对象以内部表示形式插入空间数据库中，同时返回该对象的引用标志。

2.2　空间数据管理

为了方便管理和提高性能，空间管理子系统设

计成基于主存技术的数据库。空间管理子系统在连续的存储空间内存放，分为数据库头部和数据库体两个部分。数据库体中的内容又分成3个部分：空间对象存储区、空间对象描述符区和空闲区。数据库头部用来对空间管理子系统的总体进行描述和控制。对象存储区则用来存储各种空间对象，在SADBS Ⅱ中，空间对象的数据结构仍采用平衡二叉树。由于空间对象的大小是不可确定的，在管理空间对象时，引入了长度固定的空间对象引用标识，称为空间对象描述符，它们存储在空间对象描述符区。

引入空间对象描述符的概念使空间管理子系统的用户无需直接使用空间对象的指针，只要使用一个整数指出空间对象描述符的相对偏移量就可以通过一种统一的方式操作各种空间对象。这个整数是空间对象的一个惟一标志，可以用来代替各种类型的空间对象指针进行空间对象的访问，因此称为对象标志符。这种方法隔离了空间对象的实现细节，方便了用户程序，更使得有利于空间管理子系统与可扩充DBMS集成实现空间分析DBMS。

主存技术使各种空间对象的存储和访问能够在内存中进行，既降低了I/O的交换频率，也提高了空间分析操作算法的效率。

空间数据的管理功能主要由以下几个函数来实现：

· load_sadb，close_sadb，pack_sadb：完成空间管理子系统的装入，存储和内存整理功能，其中load_sadb以文件名作为输入参数，返回值均为布尔类型；

· create_object：解析字符串类型的参数，把空间对象的外部表示转换成内部表示，并插入空间管理子系统中，返回对象标志符；

· drop_object：以对象标志符作为输入参数，删除相应的空间对象；

· update_object：以对象标志符作和字符串作为参数，相当于先执行drop_object，再执行create_object，返回新的对象标志符；

· unref：以对象标志符作为输入参数，返回空间对象的字符串描述形式。

3 空间数据库集成

与SADBS一样，SADBS Ⅱ仍然采用将空间管理子系统与可扩充DBMS集成来生成基于双网格的空间分析DBMS。实验已经证明，空间管理子系统可以和任何具有可扩充接口的DBMS集成，生成空间分析DBMS。目前，作者是将空间管理子系统与PostgreSQL[10]集成来生成SADBS Ⅱ。下面简要讨论空间数据库集成的相关问题。

可扩充数据库（面向对象数据库、对象关系数据库）的扩展主要是4个方面：(1) 扩充用户数据类型，主要是定义新类型在数据库中的存储方式；(2) 扩充新类型上的操作函数，其中不可缺少的是输入输出函数；(3) 扩充新类型上的索引；(4) 扩充相关的优化机制。

在SADBS Ⅱ中，空间数据的类型是对用户透明的，用户只需使用对象标志符就可以访问空间数据。对象标志符直接对应于PostgreSQL中的int类型，因此，无需扩充新的数据类型，直接把对象标志符当作整型值存储在数据库表中即可。当数据库中需要用到空间数据时，只需使用整数类型，例如：

```
create table highway (name varchar (20), route int);
```

其中，route字段就是空间对象类型。

在扩充空间数据操作函数时，先把所有的空间数据操作函数用create function命令注册到PostgreSQL中去，然后就可以直接调用。不过，对于数据库用户而言，对象标志符也应该是透明访问的，这就需要对管理函数使用复合SQL语句，例如：

```
insert into highway values ('No1', create_object('L(10,10,50,20)(50,20,100,100)'));
delete from highway where name = 'No1' and drop_object(route);
update highway set route=update_object (route, 'L (10, 10, 100, 100)') where name = 'No1';
select unref (route) from highway where name= 'No1';
```

其中，"L(10,10,50,20)(50,20,100,100)"表示从坐标(10,10)到(50,20)再到(100,100)的Lines类型的空间对象。通过复合SQL语句，数据库用户避免了直接操作对象标志符，同时还保证了空间数据与非空间数据的一致性。

对于任何数据库而言，索引和相关的优化都是最难而且最重要的部分，它在很大程度上影响着数据库的性能。空间数据的索引一般采用R树，它可以看成是B树在多维空间上的扩展。PostgreSQL内置了对R树索引的支持，能够直接处理以R树为索引的数据库操作的优化问题[10]。因此，只要把空间管理子系统和PostgreSQL的R树索引机制集成

起来，就可以利用PostgreSQL内部的优化机制，进一步提高空间查询的性能。

PostgreSQL中的R树索引是建立在内置的BOX类型基础上的，而作者实现的基于双网格的空间管理子系统在建立每个空间对象时，均为每个空间对象建立了一个包含该空间对象的最小矩形值MB(Minimal box)。所不同的是，PostgreSQL中的BOX是以double类型为坐标值的，而空间管理子系统中的MB是以有理数类型为坐标值的。由于索引只是对空间对象的近似表示进行比较，所以有理数转换成浮点数的精度损失不会影响空间操作的最终结果。

本文引入了一个新函数get_mb，其参数为对象标志符，返回R树索引所需要的BOX类型。这样，当创建和使用空间索引时都需要显式地指定这个函数，例如：

```
create index highway_ index on highway using rtree (get_mb(route));
select h1. name from highway as h1, highway as h2
where get_ mb(h1. route) && get_ mb(h2. route) and ll_intersects(h1. route, h2. route);
```

其中操作符&&以两组BOX对象作为输入参数，利用索引highway_index进行空间连接运算，找出所有满足“BOX相交”条件的元组；ll_intersects用来判断两个Lines对象是否相交。

SADBSⅡ能够有效地管理空间数据和非空间数据，两者之间采用前向连接[11]。其SQL语言可以无缝地处理空间数据和非空间数据，并能处理50多个空间分析操作，包括：Meet，Inside，Overlay，Intersect，Disjoint，Adjacent，Plus，Minus等。

此外，作者还用Java语言和JDBC接口实现了一个用标准SQL来访问空间数据库的演示程序，验证了SADBSⅡ对空间数据的管理与处理能力。

4　结束语

基于Realms的ROSE代数确保了空间运算的封闭性和健壮性，但由于空间调整的复杂性，使它的实用性受到了很大的限制。双网格技术的引入有效地解决了这个问题，它不仅继承了Realms中的所有优点，还避免了复杂的空间调整，进一步确保了空间对象的一致性。

作者设计和实现了基于双网格技术的空间管理子系统，能够有效地表示、存储、管理和处理空间对象，并与PostgreSQL集成实现了空间分析DBMS——SADBSⅡ。用其查询语言能够有效地完成诸如“江苏省与哪些省市邻接”、“312国道穿过哪省市”这样复杂的空间分析操作，表明了通过双网格技术实现空间分析数据库管理系统的可行性和有效性。

参考文献：

[1] Illustra Information Technologies. Illustra 2D spatial dataBlade guide release 1.3[M]. Chapter 3: 2D Spatial Functions, 1994.

[2] Corporation O. Oracle spatial user's guide and reference release 9.0.1[EB/OL]. Chapter 12: Geometry Functions. http://otn. oracle. com/docs/products/spatial/content. html, 2001.

[3] Güting R H, Schneider M. Realms: a foundation for spatial data types in database systems[A]. Proc of the 3rd international Symposium on Large Spatial Databases[C]. 1993. 14～35.

[4] Güting R H, Schneider M. Realm-based spatial data types: the ROSE algebra[J]. VLDB Journal, 1995, 4(2): 243～286.

[5] Lema J A, Güting R H. Dual grid: a new approach for robust spatial algebra implementation [A]. FernUniversitat Hagen[C]. Informatik-Report 268, 2000.

[6] 周　毅. 基于Realms的空间分析数据库系统SADBS的实现[D]. 南京：南京航空航天大学信息科学与技术学院，2001.

[7] Greene D H, Yao F F. Finite-resolution computational geometry[A]. Proc of the 27th IEEE Annual Syrup. Foundation of Computer Science[C], 1986. 143～152.

[8] Moore P. C++Boost rational numbers[EB/OL]. http://www. boost. org/libs/rational/rational. html, 2001.

[9] 卜令科. 基于Realms和主存数据库技术的空间数据存储管理子系统的研究和实现[D]. 南京：南京航空航天大学信息科学与技术学院，2000.

[10] The PostgreSQL Global Development Group. PostgreSQL 7.1 Programmer's Guide [EB/OL]. Chapter 12～15. http://www. ca. postgresql. org/ftpsite/doc/7.1/programmer. pdf, 2001. 176～210.

[11] 秦小麟. 空间分析数据库的研究方法和技术[J]. 中国图象图形学报，2000, 5(9): 711～715.