

支持关键词搜索的空间连接查询研究

陈德华 刘良旭 乐嘉锦

(东华大学计算机科学与技术学院 上海 201620)

摘要 近年来,许多实际应用不仅需要支持空间连接查询而且需要具备关键词搜索功能,以帮助用户查找那些既满足空间连接条件又包含指定关键词的空间对象组合。正是在这种需求的驱动之下,定义了一种具备关键词搜索功能的空间连接查询(Spatial Join with Keyword Search,缩写 SJKS),并提出了一种基于 IR²-Tree 的 SJKS 查询处理算法(IR²-TreeSJKS 算法),旨在实现关键词搜索与空间连接查询的高效结合。实验表明,本算法可有效支持具有关键词搜索功能的空间连接查询处理。

关键词 关键词搜索,空间连接,IR²-Tree,IR²-TreeSJKS 算法

中图法分类号 TP309 **文献标识码** A

Research on a Spatial Join Query with Keyword Search

CHEN De-hua LIU Liang-xu LE Jia-jin

(College of Computer Science and Technology, University of Donghua, Shanghai 201620, China)

Abstract In recent years, many real applications are required to support both spatial joins and keyword-based search. Motivated by such requirements, this paper defined a new kind of Spatial Join query with Keyword Search(SJKS) which combines both keyword-based search and spatial join. To answer SJKS efficiently, this paper constructed IR²-Tree(Information Retrieval R-Tree) indexing structures on the datasets, and presented an IR²-Tree-based SJKS processing algorithm named as IR²-TreeSJKS. The results of experiments show that IR²-TreeSJKS is effective for spatial join queries with keyword search.

Keywords Keyword search, Spatial join, IR²-Tree, IR²-TreeSJKS algorithm

空间连接(Spatial join)是一种重要的空间数据查询操作,常被用于在两个空间数据集中查找满足某些空间连接谓词(如重叠、交叉等)的对象组合。一个典型的空间连接应用就是在道路和城镇两个数据集中,查找所有相互交叉的道路与城镇组合。另外,在空间数据库中,一个空间对象通常不仅具有空间位置属性(如坐标、经纬度、走向等),而且包含非空间位置属性尤其是文本描述属性,如名字、类型、基本介绍等^[1,2]。以上述的城市数据集为例,每个城市除了存储它的多边形空间信息之外,还会存有该城市的名称、地理人文、历史沿革等文本信息,而且后者当前的信息量仍在不断地迅速增长。因此,许多实际的空间数据库应用在进行空间连接查询的同时,允许用户使用若干个关键词,以返回那些文本属性信息包含所指定关键词的对象组合。例如,用户在上述道路与城市两个数据集中查找那些文本描述属性包含用户所指定的关键词且相互交叉的道路-城市组合。本文称这类查询为具有关键词搜索的空间连接查询(Spatial Join query with Keyword Search, SJKS)。

具体而言,一个 SJKS 查询 Q 由空间连接谓词 P_f (如交叉或重叠等)与一组关键词列表 $K\{k_1, \dots, k_m\}$ 组成。则 Q 的查询结果为一组形式为 (m, n) 的对象组合,其中 m 与 n 分别

属于两个不同的数据集 M 和 N 。此外,结果中的任何一个对象组合 (m, n) 需要满足如下两个条件:(1)对象 m 与 n 之间的空间关系满足 P_f ;(2)两个对象的文本信息 $m.t$ 与 $n.t$ 都必须包含关键词列表 K 的所有关键词 $k_i \in K$ 。

例 1 假设两个虚拟的道路和城镇数据集 R 和 C 如图 1 所示,其中图 1(a)使用二维图形的形式表示 R 与 C 的空间信息,图 1(b)为 R 与 C 中各对象所具有的文本信息(这里,我们只列出其中的关键词)。现假定某用户提交了一个 SJKS 为“查询包含关键词 $\{c, d\}$ 且相互交叉的道路-城镇对”,则 $\langle R_1, C_1 \rangle \langle R_3, C_2 \rangle$ 为该查询的返回结果。

根据上述分析,本文的 SJKS 实际上是信息检索领域的关键词搜索和空间数据库的空间连接两种查询的结合,可广泛应用于服务推荐与决策支持系统。

目前,关键词搜索与空间连接在各自领域已有深入的研究^[3-6]。不过,对于两者的高效集成至今仍较少看见相关的文献报道。现有的系统大多采取一种关键词搜索与空间连接随意结合的解决方案。比如,首先使用空间索引结构(如 R-Tree^[9])进行空间连接查询得到满足空间连接条件的对象组合,然后利用倒排索引^[4]检查哪些空间对象组合包含指定的关键词。很明显,这种随机结合的解决方案存在效率低的缺

到稿日期:2008-11-11 返修日期:2009-02-03

陈德华(1976-),男,博士,讲师,主要研究方向为信息检索、空间数据库,E-mail: chendehua@dhu.edu.cn;刘良旭(1974-),男,博士,讲师,主要研究方向为移动数据库、空间数据库;乐嘉锦(1951-),男,教授,博士生导师,研究方向为数据库、软件工程。

点。因此,本文旨在研究一种关键词搜索与空间连接紧密集成的解决方案,以有效支持本文的 SJKS 查询,实现信息检索与空间数据库之间更深入的结合。

文献[10]中,作者首次将关键词搜索引入空间数据库中,解决了关键词搜索与最近邻居(Nearest Neighbor)查询的结合问题。作者通过一种全新的 IR²-Tree(Information Retrieval R-Tree)索引结构来实现关键词搜索与空间数据查询的结合。简单而言,IR²-Tree 实际上是 R-Tree 与 Signature 文件^[11,12]的结合体,即在每个 IR²-Tree 节点 v 上,除了存储常规 R-tree 信息之外,还附加存储以 v 为根的子树所含有的文本内容的 Signature。有关 IR²-Tree 将在本文的背景部分详细介绍。

受文献[10]思路的启发,本文针对 SJKS 查询所提出的解决方案的主要思想大致为:首先假设参与 SJKS 查询的两个空间对象集均建有 IR²-Tree 索引结构,在此基础上,我们运用文献[7]所提出的基于 R-tree 的空间连接技术思路,实现本文的基于 IR²-Tree 的 SJKS 查询处理算法(即 IR²-TreeSJKS 算法)。总而言之,本文的主要贡献如下:

- (1)定义了一种新型的基于关键词搜索的空间连接查询 SJKS;
- (2)在对空间对象集建立 IR²-Tree 索引的基础上,提出一种基于 IR²-Tree 的 SJKS 查询处理算法(IR²-TreeSJKS 算法);
- (3)使用合成数据集进行对比实验,实验表明本文提出的算法既可行又高效。

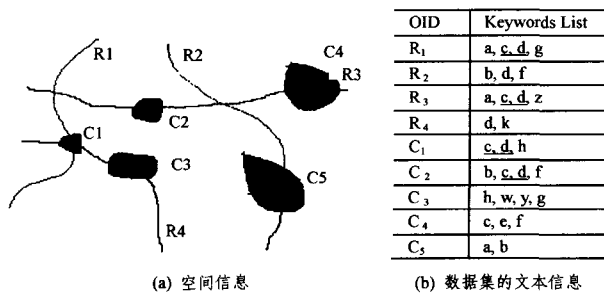


图1 两个虚拟数据集

1 IR²-Tree 背景知识

如前所述,本文的 SJKS 查询处理算法使用了一种新近提出的多维数据索引结构 IR²-Tree。为此,本节给出 IR²-Tree 的相关背景知识。

IR²-Tree 是 Ian De Felipe^[10]等人所提出的一种空间数据索引结构。它以 R-Tree 为基础,通过在每个 R-tree 节点 v 中添加该节点所具有的关键词信息而成。具体而言,在每个 IR²-Tree 节点 v 上,不仅存储 R-tree 的常规信息,而且存储以 v 为根的子树所含有的文本内容的 Signature。IR²-Tree 与 R-Tree 一样,也是一棵高度平衡树。每个 IR²-Tree 叶子节点都具有格式为 $\langle oid, mbr, Sig \rangle$ 的项,其中 oid 是指向数据库中空间对象的指针, mbr 表示该空间对象的最小外包矩形 MBR (Minimum Bounding Rectangle), Sig 为 oid 所指向的空间对象的文本内容 Signature。该 Signature 是利用信息检索领域中较为通用的 Signature 文件技术对节点文本内容进行哈希计算^[13]或者重叠编码(Superimposed Coding)^[14]而得到的。

而每个 IR²-Tree 非叶子节点则包含格式为 $\langle addr, mbr, Sig \rangle$, 其中 $addr$ 与 mbr 跟上述的定义一致, Sig 则为该节点的 Signature。与叶子节点不同,一个非叶子节点的 Signature 是通过对该节点的所有项的 Signature 进行重叠运算(即布尔 OR 运算)得到的。换言之,一个非叶子节点的 Signature 等价于以该节点为根的子树所拥有的文本内容 Signature。图 2 所示为基于图 1 中城镇数据集 C 建立的一棵 IR²-Tree。

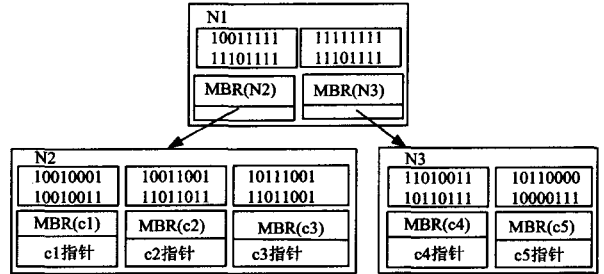


图2 IR²-Tree 结构

为了便于下面的叙述,我们给出 IR²-Tree 结构的一些描述标记。假设有一棵 IR²-Tree R ,各个标记的含义如下:

- $H(R)$ 表示树 R 的高度;
- $N(R_i)$ 是树 R 的第 i 层节点数目,其中 $0 \leq i < H(R)$;
- R_i^j 是树 R 第 i 层的第 j 个节点,其中 $0 \leq i < H(R), 0 \leq j < N(R_i)$;
- $N(R_i^j)$ 表示节点 R_i^j 所具有的项数;

$E_k(R_i^j)$ 表示节点 R_i^j 的第 k 项,其中 $0 \leq k < N(R_i^j)$ 。另外, $E_k(R_i^j)$ 具有 3 个部分的信息:(1) $E_k(R_i^j)$. oid 为该项的指针,即非叶子节点的 $addr$ 或者叶子节点的 oid ;(2) $E_k(R_i^j)$. mbr 该项的最小外包矩形 MBR;(3) $E_k(R_i^j)$. sig 则是该项的 Signature。

以图 2 的 IR²-Tree 为例,该树的各标记值分别为:该树的高度 $H(R)=2$;根节点层(第 0 层)的节点数 $N(R_0)=1$;第 1 层的节点数 $N(R_1)=2$;第 1 层的第 1 个节点 $R_1^1=N_2$;节点 N_2 的第 1 项为 $E_1(R_1^1)$,其中 $E_1(R_1^1)$. $oid=c_1, E_1(R_1^1)$. $mbr=MBR(c_1), E_1(R_1^1)$. $sig=1000101100000010$ 。

2 基于 IR²-Tree 的 SJKS 查询处理算法

本节中,我们论述本文的基于 IR²-Tree 的 SJKS 查询处理算法,即 IR²-Tree SJKS 算法。

我们假设数据库中的两个空间对象集分别拥有 IR²-Tree 索引结构 R 和 S 。此时,假定某用户提交了一个基于关键词搜索的空间连接查询 $SJKS(K\{k_1, \dots, k_m\}, P_f)$, IR²-Tree SJKS 算法则首先对树 R 的根节点 R_0 和树 S 的根节点 S_0 分别进行 Signature 匹配和空间关系判定两种操作。其中,Signature 匹配操作检查 SJKS 查询中关键词 K 的 Signature 与 R_0 . sig 和 S_0 . sig 是否相匹配;而空间关系判定操作则是判定 R_0 与 S_0 之间的空间关系是否满足空间连接谓词 P_f 。上述两个操作的结果为一组 2-元组 $\langle E_k(R_{ij}), oid, E_k(S_{ij}), oid \rangle$, 它们被临时存储在第 0 层的中间连接索引 $IJ[0]$ 中。显然, $IJ[0]$ 中的每个元组 $\langle E_k(R_{ij}), oid, E_k(S_{ij}), oid \rangle$ 需要满足如下的条件:(1) $E_k(R_{ij})$. mbr 与 $E_k(S_{ij})$. mbr 满足 P_f 条件;(2) $E_k(R_{ij})$. sig 和 $E_k(S_{ij})$. sig 均与 K . Signature 相匹配。接着,对于 $IJ[0]$ 中的每个元组 $\langle E_k(R_{ij}), oid, E_k(S_{ij}), oid \rangle$, IR²-Tree SJKS 算法分别从树 R 和树 S 中取出 $E_k(R_{ij})$ 和 E_k

(S_{ij}) 所指向两个节点。针对两个节点中的各个项, IR^2 -TreeSJKS 算法又逐一地执行空间关系判定计算和 Signature 匹配操作。此步骤将在第 1 层的中间连接索引 $IJI[1]$ 中存储一组形式为 $\langle E_k(R_{ij}), oid, E_k(S_{ij}), oid \rangle$ 的 2-元组。当 $IJI[0]$ 的所有元组都完成连接计算, $IJI[0]$ 被释放掉, 此时算法继续对 $IJI[1]$ 的元组进行组合连接计算。算法按照这种方式逐层遍历两棵 IR^2 -Tree, 直至得到树 R 与树 S 叶子节点的中间连接索引 $IJI[H(R)-1]$ 。那么, $IJI[H(R)-1]$ 中的各元组 $\langle E_k(R_{H(R)-j}), mbr, E_k(S_{H(R)-j}), mbr \rangle$ 即为 IR^2 -Tree SJKS 查询的候选集。假设树 R 与树 S 的高度不相同, 并且算法遍历至其中一棵树(比如树 R) 的叶子层次, 此时 IR^2 -Tree SJKS 算法则停留在树 R 的叶子层次, 而继续往下遍历树 S 直至达到树 S 的叶子节点。上述的步骤完成了 IR^2 -Tree SJKS 算法的过滤阶段(Filter Step)。此时, IR^2 -Tree SJKS 算法进入求精阶段, 该阶段的主要任务是验证过滤阶段所得到的候选节点组合是否满足 SJKS 查询的条件, 即每个节点组合是否满足空间连接谓词 P_f , 且该组合的描述信息是否包含关键词 $\{k_1, \dots, k_m\}$ 。

图 3 所示为 IR^2 -Tree SJKS 算法的伪代码描述。这里需要指出, 图 3 中的 IR^2 -Tree SJKS 算法只考虑输入的两棵 IR^2 -Tree 高度相同(即 $H(R) = H(S)$) 的情形。 IR^2 -Tree SJKS 算法接受 3 个输入参数, 分别为树 R 、树 S 与基于关键词的空间连接查询 $SJKS(K\{k_1, \dots, k_m\}, P_f)$ 。 IR^2 -Tree SJKS 算法的输出为候选节点组合 $IJI[H(R)-1]$ 。

算法 IR^2 -TreeSJKS($R, S, SJKS$)

```

//R, S 为两棵 IR2-Tree, 且 R 与 S 的高度相同, 即 H(R) = H(S)
//IJI[i] 为第 i 层的中间连接索引
1: IJI[i] =  $\phi$ ; //所有的 IJI 都置为空
2: W = Signature(SJKS, K); //求 SJKS 查询关键词列表的 Signature
3: IJI[0] = Node_Pair_Join( $R_0, S_0, W, SJKS, P_f$ ); //两个根节点
的连接
4: i = 0;
5: while(i < H(R) - 1)
6:   for each(  $E_k(R_j), oid, E_k(S_j), oid$ ) IJI[i]
7:     IJI[i+1] = IJI[i+1]  $\cup$  Node_Pair_Join( $E_k(R_j), oid, E_k(S_j), oid, W, SJKS, P_f$ );
8:   i = i + 1; //往下层
9:   return IJI[H(R) - 1];

```

图 3 IR^2 -TreeSJKS 算法

在图 3 所示的 IR^2 -Tree SJKS 算法, $Node_Pair_Join()$ 是一个关键的过程。该过程完成节点间的空间关系判定与 Signature 匹配两种操作。它的输入参数包括分别取自树 R 和树 S 的两个节点、查询的 Signature W 及空间连接谓词 P_f 。它的输出结果为那些满足 SJKS 查询条件的节点 oid 组合。

过程 $Node_Pair_Join(r, s, W, P_f)$

```

//r, s 分别为两棵 IR2-Tree 树的节点
1: for each entry  $E_i(r)$  of r
2:   for each entry  $E_j(s)$  of s
3:     if(Spatial_relationship( $E_i(r).mbr, E_j(s).mbr$ ) satisfying
        $P_f$ ) and( $E_i(r).sig$  match  $W$ ) and( $E_j(s).sig$  match  $W$ )
4:       return  $\langle E_i(r).oid, E_j(s).oid \rangle$ ;
5:     else
6:       return null;

```

图 4 $Node_Pair_Join$ 过程

3 实验

本节通过实验验证 IR^2 -Tree SJKS 算法的性能, 并给出实验结果。其中 3.1 节给出本文的实验设置; 3.2 节给出本文的实验结果及分析。

3.1 实验设置

为了进行对比实验, 本文实现了一个基准(Baseline)算法。该基准算法在执行 SJKS 查询时采用一种空间连接, 与关键词搜索分阶段处理的解决方案, 其主要思路为: 第一阶段执行空间连接即在两棵 R -Tree 上进行空间连接查询, 以得到满足 P_f 的对象组合; 第二阶段则执行关键词搜索, 即利用倒排索引分别检查两个对象文本描述属性是否包含查询指定的关键词。

实验用的数据结构(IR^2 -Tree 和 R -Tree)及算法(IR^2 -Tree SJKS 算法和 Baseline 算法)均采用 Java 语言实现。另外, IR^2 -Tree 和 R -Tree 均采用磁盘存储的方式, 每个磁盘块大小为 4,096kB。实验使用磁盘访问次数和查询执行时间作为衡量指标。实验环境为一台 PC 机, 其软硬件配置: CPU 为赛扬 1.8GHz; 内存容量为 1GB; 硬盘容量为 80GB; 操作系统为微软公司的 Windows XP 专业版。

实验用的两个合成数据集(DS1 和 DS2)都是利用自制数据集生成器生成的。两个数据集都是以文本形式存储(用 Tab 键作为分隔符), 其中每个空间对象占用一行。两个数据集的细节如表 1 所列。

表 1 数据集细节

数据集	大小 (MB)	空间对象数	每个对象的平均不同单词数	数据集的总单词数	每个对象的平均磁盘块
DS1	45MB	100,000	50	约 500	3
DS2	40MB	100,000	20	约 300	2

3.2 实验结果

实验中, 我们将 IR^2 -Tree 的节点 Signature 长度固定为 160 字节, SJKS 查询的关键词数目分别取为 1, 2, 3 和 4。如图 5 所示, 实验结果表明, 随着关键词数目的增加, IR^2 -Tree SJKS 算法的执行时间和磁盘访问次数均略有增长, 这是因为关键词数目的增加导致算法需要处理更多的空间对象。而对于 Baseline 算法而言, 关键词数目与执行时间和磁盘访问次数之间并没有明显的影响趋势。

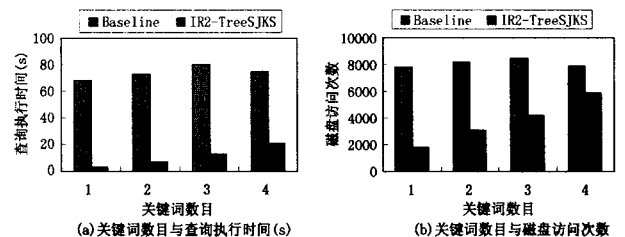


图 5 不同关键词数目的实验结果

结束语 本文根据实际应用的新需求, 探讨了一种新型的具备关键词搜索功能的空间连接(SJKS)查询, 旨在实现信息检索与空间数据库两大领域之间的紧密结合。本文借鉴基于 R 树的连接算法^[7]的思路, 使用 IR^2 -Tree 作为访问空间数据集的索引结构, 在此基础上提出一种基于 IR^2 -Tree 的 SJKS 查询处理算法, 简称 IR^2 -Tree SJKS 算法。另外, 本文

(下转第 214 页)

提出的静音检测算法相对于 G. 729B 静音检测使得 MCU 的混音路数明显减少,从而减少了 MCU 的运算量,并允许更多的客户端参加语音讨论,如图 5 所示。

结束语 本文提出了结合新的静音检测技术的集中式混音系统。在新的静音检测算法中,采用变分辨率频谱 MFCC 参数及两个感觉参数作为语音特征,为了将纯静音、气流产生的伪发音与正常语音区别开,使用多门限过零率检测首先对纯静音进行预判,然后用支持向量机对语音特征进行分类。相比于 G. 729B 中的静音检测技术和基于 MFCC+SVM 静音检测技术,在噪声比较大的情况下仍能达到比较高的语音识别率。

对于混音器的重要组成部分,本文采用 SAW 混音算法对各路解压音频进行混音,听觉测试比较优秀,并且通过对算法结构进行优化,获得了比较低的混音延时,即使是对 20 路音频混音,其运算延时仍很低,可以满足实时传输的要求。将新的静音检测技术与 SAW 混音技术使用在视频会议系统后,MCU 的混音计算量大大小于采用 G. 729B 静音检测技术的视频会议系统 MCU,而且设备条件也允许更多的客户端连接服务器参加语音讨论。

参 考 文 献

- [1] Nemer E, Goubran R, Mahmoud S. Robust voice activity detection using higher-order statistics in the LPC residual domain [J]. IEEE Transactions on Speech and Audio Processing, 2001, 9:217-231
- [2] Junqua J C, Reaves B, Mak B. A study of endpoint detection algorithms in adverse conditions: Incidence on a DTW and HMM recognize[C]// Proc. Eurospeech'91. 1991;371-1374
- [3] Sangwan A, Chiranth M C, Jamadagni H S, et al. VAD techniques for real-time speech transmission on the Internet[C]// IEEE International Conference on High-Speed Networks and Multimedia Communications. 2002;46-50
- [4] Guo Guodong, Li S Z. Content - Based Audio Classification and Retrieval by Support Vector Machines[J]. IEEE Trans. on Neu-

- ral Networks, 2003, 14(1):209-215
- [5] Stegmann J, Schroeder G. Robust Voice Activity Detection Based on the Wavelet Transform[C]// Proc. IEEE Workshop on Speech Coding. September 1997;99-110
- [6] Lin Chien-chang, Chen Shi-huang, Truong T K, et al. Audio Classification and Categorization Based on Wavelets and Support Vector Machine[J]. IEEE Transactions on Speech and Audio Processing, 2005, 13(5):644-651
- [7] ETSI; Draft Recommendation prETS 300 724; GSM Enhanced Full Rate(EFR) speech codec. 1996
- [8] ITU-T; Draft Recommendation G. 729, Annex B; Voice Activity Detection, 1996
- [9] Rabiner L, Juang B H. Fundamentals of Speech Recognition. Englewood Cliffs[M]. NJ; Prentice-Hall, 1993
- [10] Agustin JG, Hussein AW. Audio mixing for interactive multimedia communications [C] // JCIS' 98. Research Triangle, NC, 1998; 217-220
- [11] Fan Xing, Gu Wei-kang. Research on fast real-time adaptive audio mixing in multimedia conference[J]. Journal of Zhejiang University Science, 2005, 6a(6):507-512
- [12] Venkat RP, Harrick MV, Srinivas R. Communication architectures and algorithms for media mixing in multimedia conferences [J]. IEEE/ACM Trans. on Networking, 1993, 1(1):20-30
- [13] Cortes C, Vapnik C. Support Vector Networks [J]. Machine Learning, 1995, 20:273-297
- [14] Cvetkovic Z, Vetterli M. Discrete-time Wavelet Extrema Representation Design and Consistent Reconstruction [J]. IEEE Trans. SP, 1995, 143:681-693
- [15] Daubechies I. Ten Lectures on Wavelets. SIAM, Philadelphia, 1992
- [16] Thomas Parsons W. Voice and Speech Processing[M]. McGraw-Hill Book Company, 1986
- [17] Platt J C. A Fast Algorithm for Training Support Vector Machines[R]. MSR-TR-98-14. April 1998

(上接第 152 页)

通过合成数据集的对比实验,验证了 IR²-Tree SJKS 算法的性能优越性。

参 考 文 献

- [1] Yiu Man Lung, Dai Xiangyuan, Mamoulis N, et al. Top-k Spatial Preference Queries[C]//Proceedings of ICDE. 2007;1076-1085
- [2] Shekhar S, Chawla S. Spatial Databases: A Tour [M]. Prentice Hall, 2003
- [3] Grossman D A, Frieder O. Information Retrieval: Algorithms and Heuristics [M]. Springer, 2006
- [4] Manning C D, Raghavan P, Schütze H. Introduction to Information Retrieval[M]. Cambridge University Press, 2007
- [5] Brinkhoff T, Kriegel H-P, Seeger B. Efficient Processing of Spatial Joins Using R-trees[C]// Proceedings of SIGMOD. 1993; 237-246
- [6] Lo M-L, Ravishankar C V. Spatial Joins Using Seeded Trees[C] //Proceedings of SIGMOD. 1994;209-220
- [7] Huang Yun - Wu, Jing Ning. Spatial Joins Using R - trees ; Brea-

- dth-First Traversal with Global Optimizations [C] // Proceedings of VLDB. 1997;396-405
- [8] Lo M-L, Ravishankar C V. Spatial Hash-Joins[C]//Proceedings of SIGMOD. 1996;247-258
- [9] Guttman A. R - Trees : a dynamic index structure for spatial Searching[C]//Proceedings of SIGMOD. 1984;47-57
- [10] De Felipe I, Hristidis V, Risse N. Keyword Search on Spatial Databases[C]//Proceeding of ICDE. 2008;656-665
- [11] Faloutsos C, Christodoulakis S. Signature Files : An Access Method for Documents and Its Analytical Performance Evaluation [J]. ACM Trans. Inf. Syst, 1984, 2(4):267-288
- [12] Lee Dik Lun, Kim Young Man, Patel G. Efficient Signature File Methods for Text Retrieval [J]. IEEE Transactions on Knowledge and Data Engineering, 1995, 7(3):423-435
- [13] Larson P A. A method for speeding up text retrieval[C]//Proceedings of SIGMOD. 1983;117-123
- [14] Christodoulakis S, Faloutsos C. Design Considerations for a Message File Server [J]. IEEE Trans. on Software Engineering, 1984, 10(2):281-210