

基于 R 树的方向关系查询处理*

肖予钦⁺, 张 巨, 景 宁, 李 军

(国防科学技术大学 电子科学与工程学院, 湖南 长沙 410073)

Direction Relation Query Processing Using R-Trees

XIAO Yu-Qin⁺, ZHANG Ju, JING Ning, LI Jun

(School of Electronic Science and Engineering, National University of Defense Technology, Changsha 410073, China)

+Corresponding author: Phn: +86-731-4573480, Fax: +86-731-4575774, E-mail: xiao_yq@21cn.com, <http://www.nudt.edu.cn>

Received 2003-06-24; Accepted 2003-08-18

Xiao YQ, Zhang J, Jing N, Li J. Direction relation query processing using R-trees. *Journal of Software*, 2004,15(1):103~111.

<http://www.jos.org.cn/1000-9825/15/103.htm>

Abstract: Direction relations deal with order in space. Recently, direction relation query processing has gradually gained attention in geospatial databases applications, such as Spatial Data Mining (SDM) and GIS (geographic information system). The processing of direction relation queries needs spatial join operations. Until now, the research work on processing of spatial joins has primarily focused on topological and distance relations. There is little work on processing joins with direction predicates. This paper presents an efficient method for processing direction relation queries using R-trees. The quad-tuples model is defined to represent direction relations between MBRs (minimum bounding rectangles) of spatial objects. An algorithm of processing the filter step using R-trees is given and the refinement step is further decomposed into three different operations. The method presented can efficiently process direction relation queries between objects of any data types in a 2D space. Using both direction and distance constraints restricting the search space when traversing R-trees, this paper also presents an algorithm of direction relation query processing in SDM. Performance evaluation of the proposed method is conducted using real world datasets and the experiment results show that it performs well with respect to both I/O- and CPU-time.

Key words: direction relation; direction join; spatial database; spatial data mining; GIS (geographic information system)

摘 要: 方向关系描述了对象间的空间顺序关系. 近年来, 方向关系查询处理逐渐受到空间数据挖掘和地理信息系统等空间数据库应用领域研究者的关注. 方向关系查询处理需要执行方向连接操作, 目前有关空间连接的

* Supported by the National High-Tech Research and Development Plan of China under Grant Nos.2002AA131010, 2002AA134010 (国家高技术研究发展计划(863)); the Pre-Research Project of National University of Defense Technology of China under Grant No.JC02-04-18 (国防科学技术大学预研基金)

作者简介: 肖予钦(1975—),女,河南开封人,博士生,主要研究领域为空间数据库,空间数据挖掘,地理信息系统;张巨(1974—),男,博士生,主要研究领域为地理信息系统,数据库技术;景宁(1963—),男,博士,教授,博士生导师,主要研究领域为地理信息系统,数据库技术;李军(1973—),男,博士,副教授,主要研究领域为地理信息系统,数据库技术.

研究主要集中在拓扑关系和距离关系方面,而较少考虑方向关系.研究了基于 R 树的方向关系查询处理方法,定义了四元组模型表示对象 MBR 间的方向关系,提出了基于 R 树的处理方向关系查询过滤(filter)步骤的方法,并将提炼(refinement)步骤细化为 3 种不同的操作.所提出的方法能够高效处理任意对象间的方向关系查询.考虑到空间数据挖掘中方向关系查询通常是在满足一定距离约束条件的对象之间进行,还提出了一种同时利用方向和距离约束限制 R 树搜索空间的查询处理算法.实验证明,与不利用 R 树的方向关系查询处理方法相比,所提出的方法在 I/O 开销和 CPU 开销两方面都具有很高的性能.

关键词: 方向关系;方向连接;空间数据库;空间数据挖掘;地理信息系统

中图法分类号: TP311 文献标识码: A

对空间数据库进行空间查询,就是在空间数据库中找出满足给定空间条件的空间对象.空间关系作为空间查询的选取条件,在空间数据挖掘和地理信息系统等许多空间数据库应用领域中都起着重要的作用.空间关系查询处理需要执行空间连接操作,即通过空间关系将多个空间数据集关联起来进行分析.目前,有关空间连接的研究主要集中在相交、相邻等拓扑关系和距离关系方面,而较少考虑空间对象间的方向关系.

目前,已有很多处理拓扑和距离空间连接的研究成果.通过分别采用深度优先和宽度优先的策略同时遍历两棵 R 树,Brinkhoff 等人^[1]和 Huang 等人^[2]讨论了“相交”空间连接.Hjaltason 等人^[3]和 Shin 等人^[4]采用 R*树和优先队列讨论了具有一定范围约束的距离空间连接.还有一些研究者提出了不利用空间索引结构处理“相交”空间连接的算法,如文献[5,6]中的基于划分的方法以及文献[7,8]中的分布扫描方法.

对于方向空间连接,目前的研究工作还较少.Gunther^[9]提出了利用 generalization trees 处理拓扑、距离和方向空间连接的通用方法,Becker,Hinrichs 和 Finke^[10]讨论了利用 grid file 处理空间连接,这些方法都利用了 I/O 代价较高的空间索引结构,因此算法的处理性能不高.不利用空间索引结构,Zhu,Su 和 Ibarra^[11]采用平面扫描技术和外优先搜索树提出了一种 I/O 高效的处理方向空间连接的算法,但是该算法的提出却基于以下前提假定,即所研究的空间对象全部是内部不相交的有界矩形,这一假定使得该方法不能够处理任意两个空间数据集上的方向空间连接,限制了它的实用性.由于空间连接的处理需要大量 I/O 访问和 CPU 计算,文献[5]指出,与其他索引结构相比,基于 R 树的空间连接处理方法具有较大的性能优势.因此,本文利用两个输入空间数据集上已有的 R 树索引来讨论方向空间连接的处理方法.

由于空间数据的复杂性,对空间数据库的查询处理一般分为两个步骤:(1) 过滤(filter)步骤,基于对象的 MBR 快速地找出符合查询条件的候选集合;(2) 提炼(refinement)步骤,对由第 1 步得到的候选集合,确切地检查该集合中的数据是否符合查询条件.大部分有关空间连接的研究工作都集中在如何处理 filter 步骤上,也有一些有关 refinement 步骤处理方面的研究工作,如文献[12].本文将同时考虑这两个步骤的处理方法.

1 方向关系的定义

在二维空间中,简单空间对象包括点、线和多边形 3 种类型.对这 3 种类型中的任意两个对象,都可以定义它们之间的方向关系.例如,可以定义点对象之间、点对象和线对象之间的方向关系.由于多边形对象的复杂程度最高,因此本文以多边形为例讨论对象间的方向关系.

方向关系是空间关系的重要组成部分,它描述了两个对象之间的空间顺序关系.在进行空间查询和空间分析时,对象间的方向关系大多是用“南”、“北”、“东”、“西”等方位术语进行定性描述.虽然这种定性描述比定量描述更简单而且更易于理解,但是它却带有特定的语义,这使得对方向关系查询处理的难度较大.目前,方向关系的描述模型主要有两种:基于锥形的模型和基于投影的模型^[13].基于锥形的模型利用对象间的方位角来确定它们的方向关系.方位角的计算是相对于点对象进行的,它是对方位的一种精确度量,而对于其他类型的空间对象,如两个多边形对象,其相互间方位角的计算则非常复杂.基于投影的模型通过分别考虑对象在 x 轴和 y 轴上投影之间的关系来确定它们的方向关系,这种模型比基于锥形的模型更具有实用性.本文采用基于投影的模型来描述方向关系.

1.1 MBR 之间的方向关系

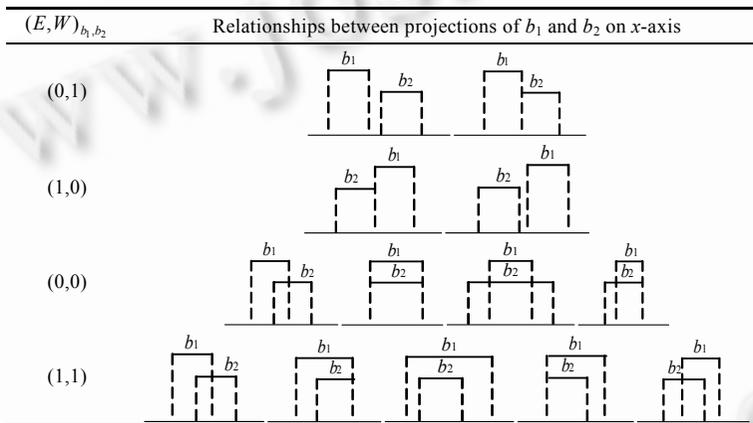
在空间数据库研究领域,为了提高查询处理的性能,通常利用 MBR 近似表示空间对象.假设坐标系统是参照方向框架定义的,即 North 方向和 y 轴方向一致,East 方向和 x 轴方向一致.若 b_1 和 b_2 分别表示对象 o_1 和 o_2 的 MBR,则 b_1 和 b_2 可由它们左下角的顶点 $(b_i.xl, b_i.yl)$ 和右上角的顶点 $(b_i.xu, b_i.yu)$ ($i=1,2$) 给出.这些顶点也称为相应 MBR 的代表点.设 b_1 是目标矩形, b_2 是参考矩形,根据代表点在 x 轴和 y 轴上投影之间的关系,可以定义 b_1 相对于 b_2 的方向关系谓词.我们采用 $(N, S)_{b_1, b_2}$ 和 $(E, W)_{b_1, b_2}$ 分别表示 North-South 方向和 East-West 方向上 b_1 相对于 b_2 的方向关系, $(N, S)_{b_1, b_2}$ 和 $(E, W)_{b_1, b_2}$ 定义如下:

$$(N, S)_{b_1, b_2} = \begin{cases} (1,0), & b_1.yl \geq b_2.yu, \\ (0,0), & (b_1.yl \geq b_2.yl) \wedge (b_1.yu \leq b_2.yu), \\ (0,1), & b_1.yu \leq b_2.yl, \\ (1,1), & \text{otherwise.} \end{cases} \quad (E, W)_{b_1, b_2} = \begin{cases} (1,0), & b_1.xl \geq b_2.xu, \\ (0,0), & (b_1.xl \geq b_2.xl) \wedge (b_1.xu \leq b_2.xu), \\ (0,1), & b_1.xu \leq b_2.xl, \\ (1,1), & \text{otherwise.} \end{cases}$$

实际上, $(E, W)_{b_1, b_2}$ 的取值取决于 b_1 和 b_2 在 x 轴上投影之间的关系.表 1 对 $(E, W)_{b_1, b_2}$ 的定义进行了直观的解释.同理,基于 b_1 和 b_2 在 y 轴上投影之间的关系,可以对 $(N, S)_{b_1, b_2}$ 的定义进行类似的解释.

Table 1 Illustration of $(E, W)_{b_1, b_2}$

表 1 $(E, W)_{b_1, b_2}$ 定义的直观解释



综合考虑 $(N, S)_{b_1, b_2}$ 和 $(E, W)_{b_1, b_2}$, 将 $((N, S), (E, W))_{b_1, b_2}$ 记作 $(N, S, E, W)_{b_1, b_2}$, 我们称其为四元组模型. 基于四元组 $(N, S, E, W)_{b_1, b_2}$ 的不同取值, 我们定义 b_1 相对于 b_2 的方向关系谓词, 共有 16 种, 见表 2. 四元组 $(N, S, E, W)_{b_1, b_2}$ 的取值考虑了 $(N, S)_{b_1, b_2}$ 和 $(E, W)_{b_1, b_2}$ 的所有组合, 因此我们所定义的方向关系谓词是完备的并且是互斥的.

Table 2 Direction predicates between MBRs b_1 and b_2

表 2 MBR b_1 和 b_2 之间的方向关系谓词

Direction predicates	Values of $(N, S, E, W)_{b_1, b_2}$
NorthWest (b_1, b_2)	(1, 0, 0, 1)
NorthSame (b_1, b_2)	(1, 0, 0, 0)
NorthEast (b_1, b_2)	(1, 0, 1, 0)
NorthUnknown (b_1, b_2)	(1, 0, 1, 1)
SameWest (b_1, b_2)	(0, 0, 0, 1)
UnknownWest (b_1, b_2)	(1, 1, 0, 1)
SameSame (b_1, b_2)	(0, 0, 0, 0)
SameUnknown (b_1, b_2)	(0, 0, 1, 1)
SameEast (b_1, b_2)	(0, 0, 1, 0)
UnknownEast (b_1, b_2)	(1, 1, 1, 0)
SouthWest (b_1, b_2)	(0, 1, 0, 1)
SouthSame (b_1, b_2)	(0, 1, 0, 0)
SouthEast (b_1, b_2)	(0, 1, 1, 0)
SouthUnknown (b_1, b_2)	(0, 1, 1, 1)
UnknownSame (b_1, b_2)	(1, 1, 0, 0)
UnknownUnknown (b_1, b_2)	(1, 1, 1, 1)

虽然都是基于代表点信息定义方向关系谓词,但是我们的定义和文献[14]中的定义却有着本质的差别.我们所定义的方向关系谓词之间是互斥的,而在文献[14]中,两个 MBR 可以同时满足多种方向关系.另外,我们所定义的方向关系谓词和文献[11]中定义的方向关系谓词也是有区别的.首先,我们的方向关系谓词是基于代表点信息定义的,而文献[11]采用参考对象划分空间后所得到的无界区域来定义方向关系谓词.其次,我们考虑了两个 MBR 的所有可能的位置关系,而文献[11]仅考虑了两个矩形内部不相交的情况.

1.2 对象之间的方向关系

MBR 是空间对象的几何近似,依赖于特殊的外形,有时空间对象间的方向关系和相应 MBR 间的方向关系存在不一致性,因此采用 MBR 间的方向关系描述对象间的方向关系存在一定的局限.方向关系矩阵通过记录目标对象落在参考对象内对空间的不同划分来描述方向^[15-17].运用投影法,参考对象将所在空间划分为 9 个独立的部分:NorthWest(NW),North(N),NorthEast(NE),West(W),Same(O),East(E),SouthWest(SW),South(S)和 SouthEast(SE),如图 1(a)所示.式(1)给出了方向关系矩阵的表达式,它是一个 3×3 的矩阵,其元素保存了空间划分间的相邻关系及目标对象和参考对象间的相交情况.图 1(b)给出了一个示例,其中元素 0 表示不相交,元素 1 表示相交.若目标对象和参考对象的某个空间划分相交,则它们满足相应划分所定义的方向关系.只有一个非零元的方向关系矩阵又称为一元方向关系矩阵,它所描述的对象间的方向关系具有惟一性.具有多个非零元的方向关系矩阵又称为多元方向关系矩阵,它所描述的对象间的方向关系具有多值性.如在图 1(b)中,方向关系矩阵 $Dir(A,B)$ 有 7 个元素为 0,2 个元素为 1,故有 $NE(A,B)$ 和 $E(A,B)$ 同时成立.

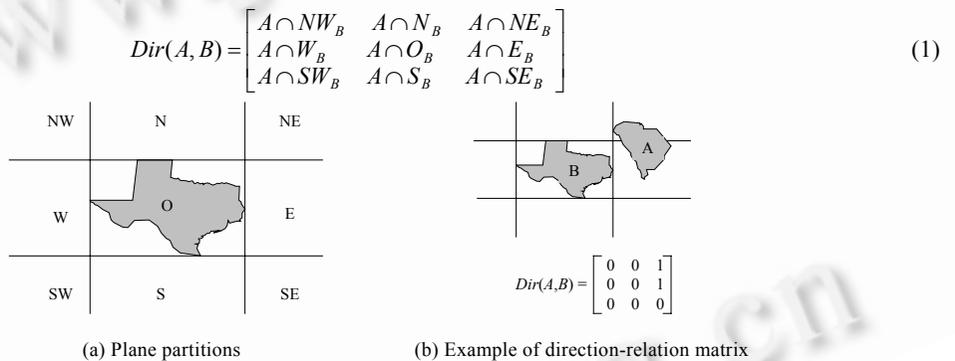


Fig.1 Direction relations between objects

图 1 对象之间的方向关系

在具体实现中,本文采用位段[NW|N|NE|W|O|E|SW|S|SE]来存储方向关系矩阵,即矩阵中每个元素都用 1 位表示,这样可以极大地减少存储空间.如[1|0|0|0|0|0|0|0|0]表示目标对象在参考对象的西北方向,[0|0|0|0|0|1|0|0|1]表示目标对象在参考对象的东和东南方向.对于方向关系矩阵中每个元素取值的判断,采用计算几何中多边形相交的算法就可以解决.

2 方向关系查询处理

2.1 方向连接

R 树采用 MBR 近似表示空间对象.一般情况下,MBR 和其所包含的实际对象是不同的,MBR 之间的关系并不能完全代表实际对象之间的关系.因此,方向关系查询由以下两个步骤组成:

- (1) 首先基于 MBR 执行一个 filter 操作,快速排除不可能满足查询条件的对象,得到查询结果的候选集;
- (2) 然后执行 refinement 操作,对由 filter 操作产生的候选集中的对象执行进一步计算,检测并删除不满足查询条件的候选对象以得到真正的查询结果.

方向关系查询处理需要执行方向空间连接操作.方向连接就是连接条件是方向关系谓词的连接^[11].假设有旅店和海滩两组空间对象,则查询“找出所有在海滩西北方向的旅店”就需要执行一个基于旅店和海滩之间方

向关系的连接操作.

假设有两组空间对象 $R=\{r_i, i \in I\}$ 和 $S=\{s_j, j \in J\}$, $Id(r_i)$ 和 $Mbr(r_i)$ 分别表示 r_i 的标识符和最小包围矩形. 对于给定的方向关系谓词, 以 N 为例, R 和 S 的方向连接表示如下:

(1) MBR 方向连接: 找出所有满足 $Direction(Mbr(r_i), Mbr(s_j))$ 的 $(Id(r_i), Id(s_j))$, $Direction$ 的取值将在第 2.2 节给出;

(2) ID 方向连接: 找出所有满足 $N(r_i, s_j)$ 的 $(Id(r_i), Id(s_j))$.

由于 MBR 间方向关系的计算代价要比对象间方向关系的计算代价低很多, 因此 MBR 方向连接可以作为 ID 方向连接的 filter 步骤. 以下我们讨论如何利用 R 树处理 MBR 方向连接.

2.2 基于 R 树的 MBR 方向连接处理

与处理拓扑、距离空间连接时一样, 我们要用到 R 树的一个重要特性, 即 R 树中某个节点所表示的空间区域总包含在其父节点所表示的空间区域中. 设 r 和 s 分别是 R 树索引 R 和 S 的非叶节点, 则 r 和 s 对空间所作的西北、东北、西南和东南划分包含在它们子节点对空间所作的相应划分中, 图 2 以西北划分为例进行了解释, 例如, $NW_{s_i} \supseteq NW_s$ ($i=1, \dots, 3$). 这一性质可总结为以下引理 1.

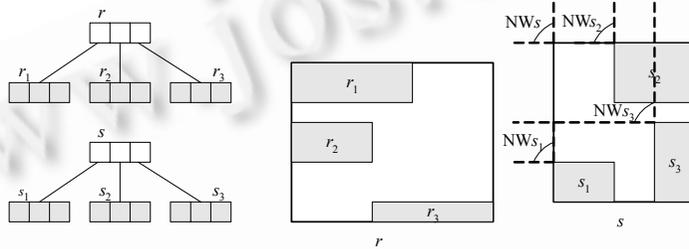


Fig.2 R-Tree indexes and the direction tiles around tree nodes

图 2 R 树及其节点对空间的划分

引理 1. 给定 R 树索引 R 和 S , 若 $r \in R$ 和 $s \in S$ 均不是根节点, 则有

$$\text{partition}_n \supseteq \text{partition}_{\text{parent}(n)}, \text{partition} \in \{\text{NW}, \text{NE}, \text{SW}, \text{SE}\}, n=r, s,$$

且对于方向关系谓词 $Direction \in \{\text{NorthWest}, \text{NorthEast}, \text{SouthWest}, \text{SouthEast}\}$, 有

$$Direction(\text{parent}(r), \text{parent}(s)) \Rightarrow Direction(r, s),$$

其中, $\text{parent}(n)$ 表示 n 的父节点.

证明: 由以上论述可得. □

引理 1 表明, 如果父节点间满足 NorthWest, NorthEast, SouthWest 和 SouthEast, 则它们的子节点间也满足相应的方向关系. 对于父节点间的其他一些方向关系谓词, 也能够推导出相应子节点间的方向关系.

引理 2. 给定 R 树索引 R 和 S , 若 $r \in R$ 和 $s \in S$ 均不是根节点, 则:

1) 对于 $Direction \in \{\text{NorthSame}, \text{NorthUnknown}\}$, 有

$$Direction(\text{parent}(r), \text{parent}(s)) \Rightarrow \text{NorthWest}(r, s) \vee \text{NorthSame}(r, s) \vee \text{NorthEast}(r, s) \vee \text{NorthUnknown}(r, s).$$

2) 对于 $Direction \in \{\text{SameWest}, \text{UnknownWest}\}$, 有

$$Direction(\text{parent}(r), \text{parent}(s)) \Rightarrow \text{NorthWest}(r, s) \vee \text{SameWest}(r, s) \vee \text{SouthWest}(r, s) \vee \text{UnknownWest}(r, s).$$

3) 对于 $Direction \in \{\text{SameEast}, \text{UnknownEast}\}$, 有

$$Direction(\text{parent}(r), \text{parent}(s)) \Rightarrow \text{NorthEast}(r, s) \vee \text{SameEast}(r, s) \vee \text{SouthEast}(r, s) \vee \text{UnknownEast}(r, s).$$

4) 对于 $Direction \in \{\text{SouthSame}, \text{SouthUnknown}\}$, 有

$$Direction(\text{parent}(r), \text{parent}(s)) \Rightarrow \text{SouthWest}(r, s) \vee \text{SouthSame}(r, s) \vee \text{SouthEast}(r, s) \vee \text{SouthUnknown}(r, s).$$

证明: 由第 1.1 节给出的 MBR 间方向关系的定义和 R 树的特性可得. □

从根节点开始遍历 R 树索引时, 需要确定中间节点应该满足的方向关系. 例如, 要找出所有在 S 中对象西北方向的 R 中的对象, 需要访问所有满足 NorthWest, NorthUnknown, UnknownWest 或 UnknownUnknown 的 MBR

对.而对于中间节点,根据引理 1 和引理 2,只有那些满足 NorthWest, NorthSame, NorthUnknown, SameWest, UnknownWest, SameSame, SameUnknown, UnknownSame 或 UnknownUnknown 的节点对,才有可能包含这样的 MBR 对.因此,在遍历 R 树时,可以通过排除不满足上述方向关系的中间节点对来限制搜索空间.例如,对图 2 中的对象执行以下查询:找出所有在 S 中对象西北(NW)方向的 R 中的对象.对于 s_1 ,有 $SouthWest(r_3, s_1)$ 成立,根据引理 1 可知, r_3 和 s_1 的子节点之间不可能满足上述方向关系,因此可以直接排除 r_3 ;对于 s_2 ,有 $SouthWest(r_2, s_2)$ 和 $SouthWest(r_3, s_2)$ 成立, r_2 和 r_3 均被排除;对于 s_3, r_3 满足 $SouthWest(r_3, s_3)$,故可直接排除 r_3 .如果 r 和 s 均为 R 树 R 和 S 的叶节点,即 $r_i(i=1, \dots, 3)$ 和 $s_j(j=1, \dots, 3)$ 表示对象的 MBR,则由于 r_2 满足 $SameWest(r_2, s_3)$, r_2 也可被排除.表 3 给出了对于不同的连接条件,在遍历 R 树时需要访问的 MBR 对和中间节点对应该满足的约束条件.

Table 3 Processing of direction joins

表 3 方向连接处理

Join conditions $\theta(R \triangleright \triangleleft \rho S)$	Constraints for the MBR pairs to be searched	Constraints for the intermediate node pairs to be searched
NW	NorthWest, NorthUnknown, UnknownWest, UnknownUnknown	NorthWest, NorthSame, NorthUnknown, SameWest, UnknownWest, SameSame, SameUnknown, UnknownSame, UnknownUnknown
N	NorthSame, NorthUnknown, UnknownSame, UnknownUnknown	NorthSame, NorthUnknown, SameSame, SameUnknown, UnknownSame, UnknownUnknown
NE	NorthEast, NorthUnknown, UnknownEast, UnknownUnknown	NorthSame, NorthEast, NorthUnknown, SameEast, UnknownEast, SameSame, SameUnknown, UnknownSame, UnknownUnknown
W	SameWest, UnknownWest, SameUnknown, UnknownUnknown	SameWest, UnknownWest, SameSame, SameUnknown, UnknownSame, UnknownUnknown
O	SameSame, SameUnknown, UnknownSame, UnknownUnknown	SameSame, SameUnknown, UnknownSame, UnknownUnknown
E	SameEast, UnknownEast, SameUnknown, UnknownUnknown	SameEast, UnknownEast, SameSame, SameUnknown, UnknownSame, UnknownUnknown
SW	UnknownWest, SouthWest, SouthUnknown, UnknownUnknown	SameWest, UnknownWest, SouthWest, SouthSame, SouthUnknown, SameSame, SameUnknown, UnknownSame, UnknownUnknown
S	SouthSame, SouthUnknown, UnknownSame, UnknownUnknown	SouthSame, SouthUnknown, SameSame, SameUnknown, UnknownSame, UnknownUnknown
SE	UnknownEast, SouthEast, SouthUnknown, UnknownUnknown	SameEast, UnknownEast, SouthSame, SouthEast, SouthUnknown, SameSame, SameUnknown, UnknownSame, UnknownUnknown

2.3 ID方向连接处理

假设利用以上 MBR 方向连接的处理方法得到一组 MBR 对,这些 MBR 对所表示的对象偶称为候选对象偶,所有候选对象偶构成了查询结果的候选集.ID 方向连接就是要对候选集中的元素进行检测并删除不满足查询条件的候选对象偶,以得到真正的查询结果.

对方向关系而言,MBR 间的方向关系能够推导出对象间的某些方向关系.即使两个 MBR 互相相交,这一性质也是成立的,这一点不同于拓扑关系.例如,给定两个空间对象 o_1, o_2 及它们的 MBR b_1, b_2 ,有以下结论成立:

$$NorthWest(b_1, b_2) \Rightarrow NW(o_1, o_2),$$

$$NorthSame(b_1, b_2) \Rightarrow N(o_1, o_2),$$

$$SameSame(b_1, b_2) \Rightarrow O(o_1, o_2),$$

$$NorthUnknown(b_1, b_2) \Rightarrow (NW(o_1, o_2) \wedge N(o_1, o_2)) \vee (N(o_1, o_2) \wedge NE(o_1, o_2)) \vee (NW(o_1, o_2) \wedge N(o_1, o_2) \wedge NE(o_1, o_2)),$$

$$UnknownWest(b_1, b_2) \Rightarrow (NW(o_1, o_2) \wedge W(o_1, o_2)) \vee (W(o_1, o_2) \wedge SW(o_1, o_2)) \vee (NW(o_1, o_2) \wedge W(o_1, o_2) \wedge SW(o_1, o_2)),$$

$$SameUnknown(b_1, b_2) \Rightarrow (W(o_1, o_2) \wedge O(o_1, o_2)) \vee (O(o_1, o_2) \wedge E(o_1, o_2)) \vee (W(o_1, o_2) \wedge O(o_1, o_2) \wedge E(o_1, o_2)),$$

$$UnknownSame(b_1, b_2) \Rightarrow (N(o_1, o_2) \wedge O(o_1, o_2)) \vee (O(o_1, o_2) \wedge S(o_1, o_2)) \vee (N(o_1, o_2) \wedge O(o_1, o_2) \wedge S(o_1, o_2)).$$

根据以上讨论,我们可以将第 1.1 节中定义的 MBR 间的方向关系谓词分为以下 3 类:

(1) 在 x 轴和 y 轴方向上均能确定对象间方向关系的谓词:NorthWest, NorthSame, NorthEast, SameWest, SameSame, SameEast, SouthWest, SouthSame 和 SouthEast.

(2) 只能在 x 轴方向或 y 轴方向上确定对象间方向关系的谓词:NorthUnknown, UnknownWest, SameUnknown, UnknownSame, UnknownEast 和 SouthUnknown.

(3) 在 x 轴和 y 轴方向上均不能确定对象间方向关系的谓词:UnknownUnknown.

对于满足不同种类方向关系谓词的 MBR 对,可以在处理 ID 方向连接时执行不同的 refinement 操作来得到相应对象之间的方向关系:

对于第(1)类方向关系谓词,通过判断 MBR 之间的方向关系就能够得到实际对象之间的方向关系,因此不需要执行 refinement 操作;

对于第(2)类方向关系谓词,MBR 之间的方向关系只能够推导出实际对象间在某一坐标轴方向上的方向关系,而对于另一坐标轴方向上的方向关系,需要执行 refinement 操作;

对于第(3)类方向关系谓词,由于在两个坐标轴方向上均不能推导出实际对象间的方向关系,因此在两个坐标轴方向上都需要执行 refinement 操作.

根据 MBR 间方向关系所属种类的不同,以上我们提出了执行不同的 refinement 操作来得到对象间的方向关系.采用这种处理方法最重要的思想是,尽可能利用 MBR 间方向关系的信息推导出对象间的方向关系.由于在执行 refinement 操作时,需要访问实际的对象数据且需要进行大量的计算,因此所提出的方法可以减少 I/O 费用和计算费用,提高了查询处理的性能.

3 空间数据挖掘中的方向关系查询处理

空间数据挖掘(spatial data mining,简称 SDM)技术主要用来提取空间数据库中非显式存在的知识、空间关系或其他有意义的模式.随着数据获取技术的飞速发展,人类采集存储的空间数据呈指数级数增长,SDM 已经成为帮助人们理解海量数据、获取有价值信息的重要工具.在 SDM 研究中,分析常常围绕主题对象展开,通过分析主题对象及其邻近区域内相关对象之间的关系来发现数据中隐含的信息.因此 SDM 中的方向关系查询通常是在满足一定距离范围约束的对象之间进行.例如,在分析旅店客房价格和旅店相对于海滩分布位置之间的关系时,就需要执行以下查询:找出所有距离海滩 1km 以内并且在海滩西北方向的旅店.

SDM 中方向关系查询可以描述如下:

输入:两个空间数据集 R 和 S ,方向关系谓词 θ 以及距离约束条件 ε ;

输出:找出所有满足方向关系 θ 和距离约束条件 ε 的 R 和 S 中的对象偶,即给出所有 $(Id(r),Id(s))$,其中 $r \in R, s \in S$ 满足 $\theta(r,s)$ 和 $distance(r,s) \leq \varepsilon$.

由于上述查询中包含方向和距离两种连接条件,需要同时执行方向和距离空间连接.处理这两种连接的基本思想仍然是利用 R 树的性质:R 树中某个节点所表示的空间区域总包含在其父节点所表示的空间区域中. ε -距离邻近问题可以通过下述方法转化为拓扑相交问题.将每个矩形 b 的长度和宽度在两个边界方向上都增加 ε ,即将 $b.xl$ 和 $b.yl$ 的值减少 ε ,将 $b.xu$ 和 $b.yu$ 的值增加 ε .这样,只有和变换后矩形的交集相交的子节点矩形才会满足距离约束 ε .只有这些节点对应的子树才有必要进行进一步的处理.图 3 给出了一个示例,其中 r_2 和 s_3 可以直接被排除.对于方向连接条件 θ ,在遍历 R 树时,可以按照表 3 给出的约束条件选取相应的中间节点对和 MBR 对.这样,通过综合利用方向和距离连接条件限制搜索空间,我们得到了一个处理 SDM 中方向关系查询的算法.假设 R 和 S 对应的 R 树索引的高度相等,下面我们给出算法的伪代码.

DirectionDistanceJoin (R, S : R-tree node, θ : direction predicate, ε : distance constraint)

```
FOR (all  $E_s \in S$  and  $E_s.rect \cap Ext(R) \neq \emptyset$ ) DO
  FOR (all  $E_R \in R$  and  $E_R.rect \cap Ext(S) \neq \emptyset$ ) DO
    IF ( $E_R.rect \cap Ext(E_s) \neq \emptyset$  and  $Check1(E_R, E_s)$ ) THEN
      IF ( $R$  is a leaf node and  $Check2(E_R, E_s)$ ) THEN      //  $S$  is also a leaf node //
        Output( $E_R, E_s$ )
      ELSE
```

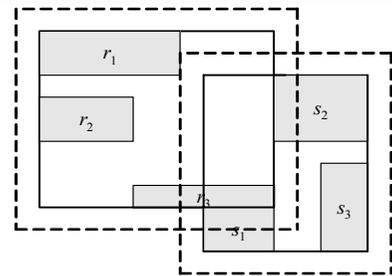


Fig.3 Transformation of ε -distance join to intersection join

图 3 ε -距离邻近问题的转化

```

    ReadPage( $E_R.ref$ ); ReadPage( $E_S.ref$ );
    DirectionDistanceJoin( $E_R.ref, E_S.ref$ )
  END
END
END
END

```

function *Ext*(R): returns the ε -extended rectangle of R ;

function *Check1*(E_R, E_S): returns true if E_R and E_S satisfy the θ -direction relations between intermediate nodes and false if not;

function *Check2*(E_R, E_S): returns true if E_R and E_S satisfy the θ -direction relations between MBRs and false if not.

4 实验设计与性能评估

前面我们提出了基于 R 树的方向关系查询处理方法,本节将评估该方法的性能.由于算法执行时间由 I/O 时间和 CPU 时间共同决定,因此实验采用查询处理所需的 I/O 访问次数和比较次数作为性能度量标准.

实验中的测试代码用 C 语言编写,测试环境是 Celeron 1.3G CPU,256M SDRAM,100M PCI 总线和 5 400 RPM IDE 硬盘,运行的操作系统是 Windows2000 Server.实验数据取自 US Bureau of the Census 发布的 TIGER/Line 文件^[18],它是测试空间连接处理算法性能的标准数据集.我们选取美国 District of Columbia 地区的道路(*tgr11001ka*)和街区(*tgr11001blk00*)作为两个输入数据集,前者包含 15 141 个线对象,后者包含 5 665 个面对象.我们为每个数据集创建了页面大小为 1 KBytes 的 R*树,叶节点存储的是对象的 MBR,表 4 给出了两棵 R*树的基本参数.

Table 4 Parameters of R*-trees

表 4 R*树的基本参数

R*-trees (page size 1 KBytes)	Max number of entries fit in one node	Height	Number of directory pages	Number of data pages
<i>lka</i>	51	3	13	431
<i>blk</i>	51	3	6	170

我们比较了利用 R 树(R-tree)的和利用 R 树(no R-tree)的方向关系查询处理方法的性能,表 5 给出了两种方法处理查询所需的 I/O 访问次数和需要进行比较的次数,列(%)表示利用 R 树处理查询的 I/O 访问次数和比较次数相对于不利用 R 树的 I/O 访问次数和比较次数的百分比.

Table 5 Number of I/O and number of comparisons

表 5 I/O 访问次数和比较次数

Join conditions θ ($lka \triangleright \triangleleft blk$)	Number of disk I/O			Number of comparisons		
	R-tree	No R-tree	%	R-tree	No R-tree	%
NW	29 236	33 264	87.9	56 217 806	343 095 060	16.4
N	5 941	33 264	17.9	23 546 457	343 095 060	6.9
NE	18 733	33 264	56.3	45 170 526	343 095 060	13.2
W	7 960	33 264	23.9	32 721 995	343 095 060	9.5
O	2 313	33 264	7.0	8 811 093	343 095 060	2.6
E	7 919	33 264	23.8	31 635 620	343 095 060	9.2
SW	20 467	33 264	61.5	60 158 187	343 095 060	17.5
S	6 371	33 264	19.2	24 005 779	343 095 060	7.0
SE	32 945	33 264	99.0	84 830 188	343 095 060	24.7

图 4 比较了两种方法处理方向关系查询所需的执行时间,实验结果表明,所提出的基于 R 树的方向关系查询处理方法的性能比不利用 R 树的方法的性能高很多.综上,我们可以得出结论,利用 R 树处理方向关系查询不仅是可行的,而且能起到很好的性能优化作用.

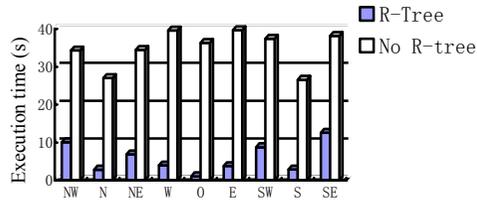


Fig.4 Comparisons of execution time

图 4 执行时间比较

References:

- [1] Brinkhoff T, Kriegel HP, Seeger B. Efficient processing of spatial joins using R-trees. *SIGMOD Record*, 1993,22(2):237~246.
- [2] Huang YW, Jing N, Rundensteiner EA. Spatial joins using R-trees: Breadth-First traversal with global optimizations. In: Jarke M, *et al.* eds. *Proc. of the VLDB'97*. San Francisco: Morgan Kaufmann Publishers, 1997. 396~405.
- [3] Hjalton GR, Samet H. Incremental distance join algorithms for spatial databases. In: Haas LM, Tiwary A, eds. *Proc. of the SIGMOD'98*. New York: ACM Press, 1998. 237~248.
- [4] Shin H, Moon B, Lee S. Adaptive multi-stage distance join processing. *SIGMOD Record*, 2000,29(2):343~354.
- [5] Patel JM, DeWitt DJ. Partition based spatial-merge join. *SIGMOD Record*, 1996,25(2):259~270.
- [6] Lo M-L, Ravishankar CV. Spatial hash-joins. *SIGMOD Record*, 1996,25(2):247~258.
- [7] Arge L, Procopiuc O, Ramaswamy S, Suel T, Vitter JS. Scalable sweeping-based spatial join. In: Gupta A, *et al.*, eds. *Proc. of the VLDB'98*. San Francisco: Morgan Kaufmann Publishers, 1998. 570~581.
- [8] Vitter JS. External memory algorithms. In: *Proc. of the ACM-PODS'99*. New York: ACM Press, 1998. 119~128.
- [9] Gunther O. Efficient computation of spatial joins. In: *Proc. of the ICDE'93*. Los Alamitos: IEEE Computer Society Press, 1993. 50~59.
- [10] Becker L, Hinriches K, Finke U. A new algorithm for computing joins with grid files. In: *Proc. of the ICDE'93*. Los Alamitos: IEEE Computer Society Press, 1993. 190~197.
- [11] Zhu H, Su J, Ibarra OH. On multi-way spatial joins with direction predicates. In: Jensen CS, *et al.*, eds. *Proc. of the SSTD 2001*. Lecture Notes in Computer Science 2121, Berlin: Springer-Verlag, 2001. 217~235.
- [12] Brinkhoff T, Kriegel HP, Schneider R, Seeger B. Multi-Step processing of spatial joins. *SIGMOD Record*, 1994,23(2):197~208.
- [13] Liu X, Shekhar S, Chawla S. Consistency checking for Euclidean spatial constraints: A dimension graph approach. Technical Report, TR00-039, Minneapolis: University of Minnesota, 2000.
- [14] Papadias D, Theodoridis Y, Sellis T. The retrieval of direction relations using r-trees. In: Karagiannis D, ed. *Proc. of the DEXA'94*. Lecture Notes in Computer Science 856, Berlin: Springer-Verlag, 1994. 173~182.
- [15] Goyal RK, Egenhofer MJ. Similarity of cardinal directions. In: Jensen CS, *et al.*, eds. *Proc. of the SSTD 2001*. Lecture Notes in Computer Science 2121, Berlin: Springer-Verlag, 2001. 36~55.
- [16] Cao H, Chen J, Du DS. Qualitative extension description for cardinal directions of spatial objects. *Acta Geodetica et Cartographica Sinica*, 2001,30(2):162~167 (in Chinese with English abstract).
- [17] Feng YC, Chen L, Cao ZS. Direction relation model of spatial database. *Computer Engineering and Applications*, 2001,20:115~117 (in Chinese with English abstract).
- [18] US Bureau of the Census. Census 2000 TIGER/Line files. 2000. <http://www.census.gov/>

附中文参考文献:

- [16] 曹蕊,陈军,杜道生.空间目标方向关系的定性扩展描述.测绘学报,2001,30(2):162~167.
- [17] 冯玉才,陈琳,曹忠升.空间数据库的方向关系模型.计算机工程与应用,2001,20:115~117.