空间数据库中约束 K 最接近对查询*)

刘小峰　刘云生　肖迎元
(华中科技大学计算机学院　武汉 430074)

摘　要　定义了满足空间约束的 K 最接近对查询，该查询检索两个数据集在给定约束区域中的 K 最接近对。在空间数据库中，对采用 R 树类型索引存储的数据集给出了三个查询处理算法。其中两阶段的 RJ 和 JR 算法采用了变换范围查询和最接近对查询执行顺序的策略。单阶段基于堆的 SPH 算法采用了最好优先的策略，并利用给出的裁减规则、更新规则和访问顺序规则来提高查询处理效率。实验表明 SPH 具有较好的适用性和性能。

关键词　空间数据库，R 树，最接近对查询，约束最接近对查询

Constrained K Closest Pairs Query in Spatial Databases

LIU Xiao-Feng　LIU Yun-Sheng　XIAO Yin-Yuan
(School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074)

Abstract　In this paper, constrained K closest pairs query is introduced, which retrieves the K closest pairs satisfying the given spatial constraint from two datasets. For data sets indexed by R-trees in spatial databases, three algorithms are presented for answering this kind of query. Among of them, two-phase RJ and JR algorithms adopt the strategy that changes the execution order of range and closest pairs queries, while single-phase heap-based SPH algorithm adopts best-first strategy and utilizes proposed pruning, updating and visit order heuristics to improve its efficiency. Experimental result shows that SPH has better applicability and performance than RJ and JR.

Keywords　Spatial databases, R-trees, Closest pairs query, Constrained closest pairs query

1　引言

最接近对查询(Closest Pair Query, CPQ)从两个数据集中检索相互距离最小的对象对，它在许多方面都有重要的应用。在空间数据库中，高效计算 CPQ 是空间数据库查询处理的重要内容。

按照查询结果的输出方式，空间数据库中 CPQ 查询处理方法可以分为两类：渐进和非渐进算法。Corral 等提出了非渐进的处理 K 最接近对查询(K Closest Pairs Query, K-CPQ)的分支-界定算法，该算法在 K 事先知道的情况下，直到算法停止时才输出查询结果[1]。Yang 等设计了一种 Rddn 树索引结构，这种树利用最近邻居信息裁减搜索空间，以便进一步提高空间数据库中 CPQ 和相关距离连接查询的处理效率，并给出了几个非渐进的查询处理算法[2]。Hjaltason 等的算法采用渐进方法解决距离连接查询，Shin 等对该算法利用 plane sweep 技术等进行了扩展[3,4]。

但是，所有这些方法都在整个数据空间中寻找最接近对。在某些情况下，用户可能关心某个特定空间区域内的最接近对。这种检索给定空间区域中的最接近对问题称为约束 K 最接近对查询(Constrained K Closest Pair Query, K-CCPQ)。这种查询的典型例子有寻找某个城区内最接近的商场和宾馆，获得某个地区内最接近的旅游胜地和城市等。在这里，最接近的商场和宾馆以及最接近的旅游胜地和城市只能是在给定区域(某城区和某地区)内。

目前，和约束最接近对查询相接近的研究工作有：约束最近邻居查询，对这种查询 Ferhatosmanoglu 等将搜索空间裁减和最近邻居查找集成到一个阶段完成查询处理[5]，但他们的方法不能解决约束最接近对查询处理；Jing Shan 等提出一种称为 SRCP 树的索引结构处理带空间约束的单个数据集中最接近对查询，显然这需要同时维护两个索引[6]。本文将在空间数据库、数据采用 R 树类型索引和欧拉空间环境中，讨论 K-CCPQ 及其查询处理，对于其他度量空间和基于数据划分的索引结构，我们的方法可以方便地扩展处理。

2　约束最接近对查询和 R 树

设 R 为实数集，d 维数据空间为 R^d，R^d 中两点 $p(p_1, p_2, \cdots, p_d)$，$q(q_1, q_2, \cdots, q_d)$ 之间的距离度量函数 $\mathrm{dist}(p,q)$ 为：$L_2=\sqrt{\sum_{i=1}^{d}(p_i-q_i)^2}$，则 d 维欧拉空间为 $E^d=(R^d, L_2)$。

定义 1　P、Q 为 E^d 中有限对象集，CR 为空间区域，则满足约束 CR 的 $K(1\leqslant K\leqslant |P|\cdot|Q|)$最接近对查询 K-CCPQ (P,Q,CR)寻找 $P\times Q$ 上的 K 个不同对象对：

1) $\text{K-CCPQ}(R,Q,CR)=\{(p_1,q_1),(p_2,q_2),\cdots,(p_K,q_K)\}$，$\text{K-CCPQ}(R,Q,CR)\subseteq P\times Q$；

2) $(p_i,q_i)\neq(p_j,q_j)$，$i\neq j$，$1\leqslant i,j\leqslant K$；

3) $\forall(pi,qi)\in\text{K-CCPQ}(R,Q,CR)$，$1\leqslant i\leqslant K$；$\mathrm{conatains}(CR,p_i)\wedge\mathrm{conatains}(CR,q_i)$；

4) $\forall(p,q)\in P\times Q$，$\mathrm{contains}(CR,p)\wedge\mathrm{contains}(CR,q)$，$(p,q)\notin\text{K-CCPQ}(P,Q,CR,K)$，$(p_i,q_i)\in\text{K-CCPQ}(P,Q,CR,K)$；$\mathrm{dist}(p,q)\geqslant\mathrm{dist}(p_i,q_i)$，$1\leqslant i\leqslant K$。

*)基金项目：国家自然科学基金(60073045)资助。刘小峰　讲师，博士生，主要研究领域为空间数据库，实时数据库，时空数据库。刘云生　教授，博士生导师，主要研究领域为现代数据库理论与技术，实时信息处理，软件方法学与支撑环境。

其中，contains 为判断空间区域是否包括对象的谓词。

在文[1]中，设 M_P 和 M_Q 为 R 树中两 MBR，不考虑空间约束，M_P 和 M_Q 中对象之间最小可能距离定义为 MINMINDIST(M_P,M_Q)，MINMAXDIST(M_P,M_Q)是 M_P 和 M_Q 中对象形成至少一个对象对的距离的上界。但是，由于约束区域的引入，这两个距离度量函数需要扩展和重新定义。

定义 2 M_P 和 M_Q 为两 MBR，CR 为约束空间区域，则满足约束 CR 的 M_P 和 M_Q 对象之间最小可能距离定义为：

$$\text{C-MINMINDIST}(M_P,M_Q,CR)=\begin{cases}\text{Mindist}(M_P\cap CR,M_Q\cap CR), & M_P\cap CR\neq\varnothing\wedge M_Q\cap CR\neq\varnothing\\ \infty, & \text{否则。}\end{cases}$$

其中，Mindist 求两空间区域之间最小距离的函数，$M_P\cap CR$ 和 $M_Q\cap CR$ 分别为两 MBR 和 CR 的相交得到的空间区域。

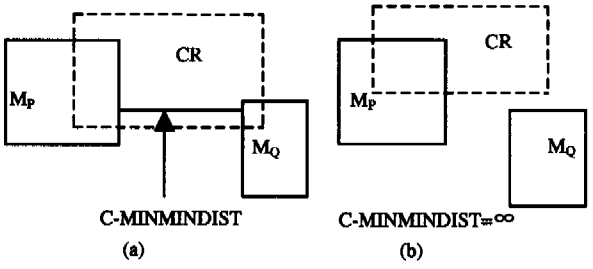

图 1　C-MINMINDIST 例子

图 1 是 C-MINMINDIST 的两个例子。由 C-MINMINDIST 的定义可以得到定理 1。

定理 1 M_P 和 M_Q 为两 MBR，CR 为约束空间区域，M_P 和 M_Q 分别包含 M 和 N 个子 MBR，记作{M_{P1},M_{P2},…,M_{PM}}和{M_{Q1},M_{Q2},…,M_{QN}}，则有：

1) C-MINMINDIST(M_{Pi},M_{Qj},CR)≥C-MINMINDIST(M_P,M_Q,CR)，$1\leqslant i\leqslant M$，$1\leqslant j\leqslant N$；

2) C-MINMINDIST(M_P,M_{Qj},CR)≥C-MINMINDIST(M_P,M_Q,CR)，$1\leqslant j\leqslant N$；

3) C-MINMINDIST(M_{Pi},M_Q,CR)≥C-MINMINDIST(M_P,M_Q,CR)，$1\leqslant i\leqslant M$。

定理 1 保证，当某对 MBR 之间的 C-MINMINDIST 大于某个值 T 时，那么它们的子 MBR 之间的 C-MINMINDIST 一定大于 T。

定义 3 M_P 和 M_Q 为两 MBR，CR 为约束空间区域，满足约束条件下，M_P 和 M_Q 对象之间至少存在一个对象对的距离的上界为 C-MINMAXDIST(M_P,M_Q,CR)，F_P 和 F_Q 分别为 M_P 和 M_Q 中被 $M_P\cap CR$ 和 $M_Q\cap CR$ 完整包含的面的集合。则有：

$$\text{C-MINMAXDIST}(M_P,M_Q,CR)=\begin{cases}\min\{\text{Maxdist}(f_i,f_j):\ f_i\in F_P,f_j\in F_Q\}, & F_P\neq\varnothing\wedge F_Q\neq\varnothing\\ \infty,\text{否则。}\end{cases}$$

其中，Maxdist 求两空间区域之间最大距离的函数，min 求一个集合中的最小者。

图 2 为 C-MINMAXDIST 的例子。由 C-MINMINDIST 和 C-MINMAXDIST 的定义可知，当 $M_P\cap CR$ 和 $M_Q\cap CR$ 包含有对象时，这些对象之间的距离不小于 C-MINMINDIST(M_P,M_Q,CR)，并且，这些对象之间存在距离不大于 C-MINMAXDIST(M_P,M_Q,CR)的对象对。即有下面的定理 2 和定理 3。

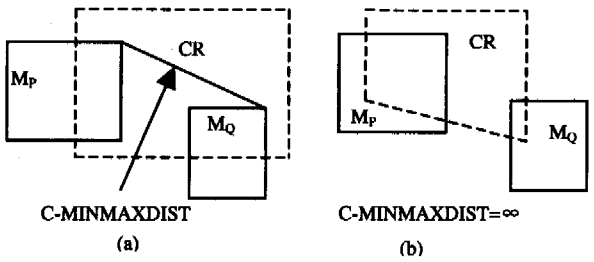

图 2　C-MINMAXDIST 例子

定理 2 M_P 和 M_Q 为两 MBR，CR 为约束空间区域，若 $M_P\cap CR$ 和 $M_Q\cap CR$ 分别包含非空对象集 O_1 和 O_2，则有：$\exists(o_1,o_2)\in O_1\times O_2$，C-MINMINDIST($M_P$,$M_Q$,$CR$)≤dist($o_1$,$o_2$)。

定理 3 M_P 和 M_Q 为两 MBR，CR 为约束空间区域，若 $M_P\cap CR$ 和 $M_Q\cap CR$ 分别包含非空对象集 O_1 和 O_2，则有：$(o_1,o_2)\in O_1\times O_2$，dist($o_1$,$o_2$)≤C-MINMAXDIST($M_P$,$M_Q$,$CR$)。

3　两阶段算法

K-CCPQ 可以看作最接近对查询和范围查询的组合，因此，一种简单和直接的方法是将这两种查询处理方法顺序执行来处理 K-CCPQ。根据最接近对查询和范围查询执行的次序不同可以有两种算法。

第一种两阶段算法先执行渐进式距离连接，在连接得到每个结果后检查它们是否位于约束范围内，若是，则输出，否则丢掉。这种 Join+Range 方法简称 JR。但是，当约束区域 CR 中对象对之间距离较大时，JR 算法会对较多无用数据空间(不满足空间约束的数据)进行搜索。利用满足约束的最接近对之间的最大可能距离可以限制搜索无用空间。

定理 4 设约束空间 CR 中相距最远的两点之间的距离为 Maxd(CR)，则对于 $\forall(p,q)\in$ K-CCPQ(R,Q,CR)，有 dist(p,q)≤Maxd(CR)。

根据定理 4，在渐进式连接输出对象对(p,q)时，当 dist(p,q)>Maxd(CR)时，算法可以停止。因为，下一个最接近对之间的距离一定也大于 Maxd(CR)。下面是 JR 算法：

$JR(P,Q,CR,K)$

/* P,Q 为 R 树节点，CR 为约束空间区域，K 为最接近对数 */

1. 渐进式距离连接 P 和 Q 获得下一个最接近对(p,q)；

2. 如果 dist(p,q)> Maxd(CR)，停止。如果 p 和 q 位于 CR 内，输出(p,q)；

3. 如果输出结果数为 K，停止。否则，转步骤 1 继续。

颠倒连接和范围查询的顺序，可以得到第二种方法 Range+Join 方法，简称为 RJ。RJ 算法先对约束范围区域执行范围查询，然后从满足约束的对象对中寻找最接近对。下面是 RJ 算法：

$RJ(P,Q,CR,K)$

/* P,Q 为 R 树节点，CR 为约束空间区域，K 为最接近对数 */

1. 对数据集 P 和 Q 执行范围查询获得位于 CR 内的对象集 PCR 和 QCR；

2. 对集合 PCR 和 QCR 进行距离连接，输出具有最小 K 距离的对象对。

JR 和 RJ 两阶段算法具有各自的应用特点。当 CR 范围

比较小时，采用 Range+Join 方法，即 RJ 算法比较合适；当约束空间区域 CR 的范围和整个数据空间大小可比较时，Join+Range 方法，即 RJ 算法比较有效。

4 单阶段基于堆的算法

单阶段基于堆的算法(Single-phase Heap-based Algorithm，SPH)基于分支-界定技术，并利用一个容量为 K 的最大堆 KH 记录目前已经发现的约束 K 最接近对。设 T 为目前已经发现的满足约束第 K 最接近对之间距离，T 的初值为 ∞。SPH 按照下列原则从 R 树的树根开始遍历：(1)当遇到叶子节点时，计算目前发现的约束最接近对，更新 KH 和 T；(2)当遇到中间节点时，只访问它的那些可能包含最终结果的子节点。

根据定理 2，可以得到下面的裁减规则。SPH 应用规则 1 避免访问不可能包含最终结果的 R 树节点。

规则 1(裁减规则) M_P 和 M_Q 为两 MBR，CR 为约束空间区域，若 C-MINMINDIST(M_P，M_Q，CR)$>T$，则相对于(M_P，M_Q)的搜索路径可以抛弃。

当 $K=1$ 时，根据定理 3，规则 2 可以加快 T 的减小。

规则 2(更新规则) 当 $K=1$ 时，M_P 和 M_Q 为两 MBR，CR 为约束空间区域，M_P 和 M_Q 分别包含子 MBR 集$\{M_{Pi}: 1\leqslant i\leqslant M\}$和$\{M_{Qj}: 1\leqslant j\leqslant N\}$，若{C-MINMAXDIST($M_{Pi}$，$M_{Qj}$，$C_R$)}中的最小者小于 T，则 T 更新为这个最小者。

直觉上，C-MINMINDIST 较小的 MBR 之间更有可能包含最后的结果，因此，SPH 对 R 树的遍历按照 C-MINMINDIST 升序进行。

规则 3(访问顺序规则) R 树的遍历按照 C-MINMINDIST 升序进行。

应用规则 3 意味着 SPH 采用一种最好优先的搜索策略，具体实现上，SPH 利用一个最小堆 MH 保存待搜索处理的 MBR 对(基于 MBR 之间的 C-MINMINDIST 排序)，具有最小 C-MINMINDIST 的 MBR 对位于 MH 堆顶。下面是相同高度 R 树的 SPH 算法，对于不同高度的 R 树，利用文[1]中的 fixed-at-root 或者 fixed-on-leaves 技术可以容易地进行扩展处理。

```
SPH(P,Q,CR,K)
/* P,Q为R树节点,CR为约束空间区域,K为最接近对数 */
1. 初始化MH和KH,(P,Q)入堆MH,T←∞;
2. while MH不空
3.     MH堆顶元素(NP,NQ)出堆;
4.     if C-MINMINDIST(NP,NQ,CR)> T
5.         return KH;
6.     if NP和NQ是内部节点
7.         if K=1 利用规则2更新T;
8.         计算NP和NQ中所有可能MBR对之间的C-MINMINDIST;
9.         将那些C-MINMINDIST不大于T的MBR对插入MH;// 规则1
10.    else
11.        for each object p∈NP,q∈NP
12.        计算p和q之间的距离;
13.        if这个距离小于T
14.            更新KH和T;
15. return KH;
```

5 实验

本节设计了两个实验，第一个实验通过调节约束区域的相对大小(固定 K)考察三个算法的适用范围；第二个实验考察 SPH 算法的性能随 K(固定约束区域大小)的变化。

实验数据来自某城市的真实二维地理数据。该数据集大小为 5367，包括点、线和多边形等类型空间实体。数据存储在页面大小为 $2K$ 的 R*树中，不使用缓冲区。约束区域的形状采用矩形，位置随机选择产生，它占整个数据空间大小比率为其相对大小。所有程序采用 C++实现并用 MS Visual C++6 编译。程序运行在 Intel P4 2.4G，512M RAM，Windows2000 Professional 机器上。

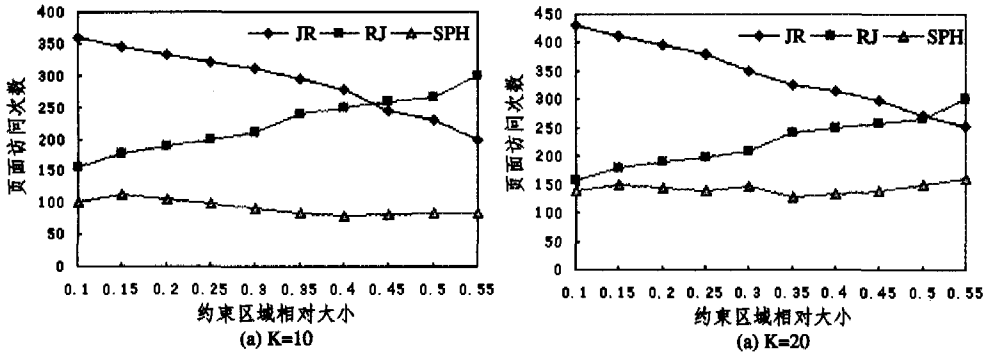

图 3 页面访问次数随约束区域的变化

图 3(a)，3(b)是三个算法的页面访问次数随约束区域大小的变化。可以看出：

(1)最接近对数 $K=10$ 和 $K=20$ 时页面访问次数变化趋势大致相同；

(2)SPH 算法具有较广的适用范围，当约束区域相对大小从 0.10 变化到 0.55，其磁盘页面访问次数变化不大；

(3)随着约束区域的相对大小增大，JR 算法页面访问次数减少，而 RJ 算法增加。这是因为，JR 先渐进式连接后检查是否满足约束，当约束区域增大时，连接得到的中间结果满足约束的可能性增大。而 RJ 算法先进行范围查询，这将随着范围的增大而产生更多的磁盘读写。

表 1 SPH 算法 CPU 和 I/O 随 K 的变化($CR=0.2$)

	4	8	12	16	20	24	28	32	36	40
页面访问次数	98	109	118	125	131	139	143	156	159	160
响应时间(毫秒)	903	1039	1090	1112	1108	1115	1198	1203	1245	1252

表 1 是约束区域相对大小为 0.2 时 SPH 算法 CPU 和 I/O 随 K 的变化。可以看出，随着 K 的增大，无论是查询响应时间还是页面访问次数的增加不是很明显。因此，SPH 具有较好的性能。

(下转第 165 页)

图1描述了算法的参考价值性质。随着近似参数的增大,参考价值基本上呈现下降趋势,虽然中间可能有波动,波动是由于算法的不确定性决定的。当近似参数一样的时候,随着概率参数的增大,算法的参考价值会变小。图2展示了算法的准确度性质。同样,准确度随着近似参数的增大而缩小。同样的近似参数,准确度随着概率参数的减小而保持着上扬趋势。由此可见,近似参数和概率参数很大程度上决定了算法的参考价值和准确度。

其次,本文对本文算法和采样算法进行了比较。为了具有可比性,通过调整采样率,使得采样算法利用的空间等于本文算法的利用空间。

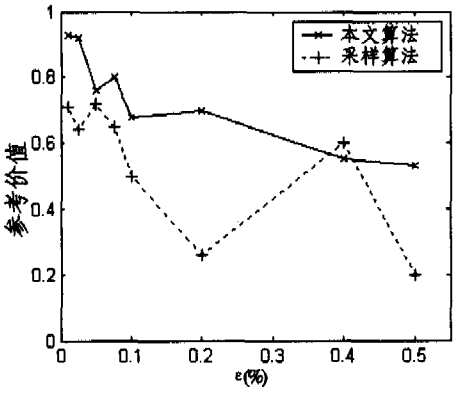

图3 本文算法和采样算法的参考价值比较

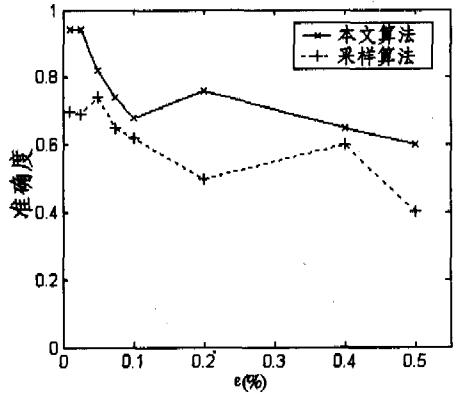

图4 本文算法和采样算法的准确度比较

由图3和图4可见,无论是参考价值和准确度,本文算法都要比采样算法好。随着近似参数的增大,采样算法的采样率降低,因此采样算法的参考价值和准确度都会降低。但是,由于随机性,其呈现较大的波动。虽然在局部采样算法可能比本文算法好,但是总体趋势上,本算法比采样算法更优秀、更实用。

总结 变化热门元素的检测问题,虽然在许多文献里提到,但真正解决此问题的很少。本文构造了一种新型的概要结构,基于此结构提出了一个算法,解决了该问题。算法以一定的概率(可以任意大)输出满足条件的元素,而需要的空间却远远小于数据流的尺寸。一旦为两个窗口的数据流建立ε近似概要数据结构,就可以查询绝对变化率不小于任意β(只要β大于ε)的变化热门元素。但本算法的缺点是可能会输出不属于数据流的元素。这个缺点也是下一步要解决的难点。与本文工作相关的数据流的变化检测模型也是亟需要研究的问题。数据流的研究尚处于初级阶段,还有许多工作需要做。

参考文献

1 Fischer M, Salzberg S. Finding a Majority among n Votes: Solution to Problem 81-5. Journal of Algorithms, 1982, 3(4): 376~379

2 Misra J, Gries D. Finding Repeated Elements. Sci Comput Programming, 1982, 2(2): 143~152

3 Demaine E D, L_opez-Ortiz A, Munro J I. Frequency Estimation of Internet Packet Streams with Limited Space. In: Proc. of the 10th Annual European Symp. on Algorithms, Rome, 2002

4 Karp R M, Shenker S, Papadimitriou C H. A Simple Algorithm for Finding Frequent Elements in Streams and Bags. ACM Trans. on Database Systems, 2003, 28(1): 51~55

5 Manku G S, Motwani R. Approximate Frequency Counts over Data Streams. Ramakrishnan. In: Proc. of the 28th Intl Conf. on Very Large Data Bases, Hong Kong, 2002

6 Golab L, DeHaan D, Demaine E, et al. Identifying Frequent Items in Sliding Windows over On-line Packet Streams. SIGCOMM Internet Measurement Conference, Miami, 2003

7 Arasu A, Manku G S. Approximate Counts and Quantiles over Sliding Window. In: Proc. of ACM Symp. on Principles of Database Systems, Paris, 2004

8 Cormode G, Muthukrishnan S. What's New: Finding Significant Differences in Network Data Streams: [technique report]. www.cs.rutgers.edu/~muthu/: S. Muthukrishnan, 2003

9 Alon N, Matias Y, Szegedy M. The space complexity of approximating the frequency moments. Journal of Computer and System Sciences, 1999, 58(1): 137~147

10 Feigenbaum J, Kannan S, Strauss M, et al. An approximate L1-difference algorithm for massive data streams. In: Proc. of the 40th Annual Symposium on Foundations of Computer Science, New York, 1999

11 Indyk P. Stable distributions, pseudorandom generators, embeddings and data stream computation. In: Proc. of the 41th Symposium on Foundations of Computer Science, Redondo Beach, 2000

12 Gilbert A, Kotidis Y, Muthukrishnan S, et al. How to summarize the universe: Dynamic maintenance of quantiles. In: Proc. of the 28th Intl Conf. on Very Large Data Bases, Hong Kong, 2002

(上接第158页)

结论和未来的工作 本文在空间数据库和数据采用R树索引的环境中解决了约束最接近对查询K-CCPQ。通过变换连接和范围查询的次序,给出了两个两阶段查询处理算法*RJ*和*JR*。同时,在定义的距离函数的基础上给出了裁减、更新和访问顺序规则,设计了基于堆的单阶段SPH算法。最后对一个真实的数据集进行了实验比较和分析,实验结果表明SPH算法具有较好的性能和较大的适用范围。设计约束最接近对查询的代价模型将是下一步的研究工作。

参考文献

1 Corral Y, Manolopoulos Y, Theodoridis, Vassilakopoulos M. Closest Pair Queries in Spatial Databases. In: Proc. of ACM SIGMOD Conf. 2000. 189~200

2 Yang, Lin K I. An Index Structure for Improving Nearest Closest Pairs and Related Join Queries in Spatial Databases. In: Proc. of Intl. Database Engineering and Applications Symposium (IDEAS'02), 2002. 140~149

3 Hjaltason G R, Samet H. Incremental Distance Join Algorithms for Spatial Databases. In: Proc. of ACM SIGMOD Conf. 1998. 237~248

4 Shin H, Moon B, Lee S. Adaptive Multi-Stage Distance Join Processing. In: Proc. of ACM SIGMOD Conf. 2000. 343~354

5 Ferhatosmanoglu H, Stanoi I, Agrawal D, Abbadi A E. Constrained Nearest Neighbor Queries. In: 7th International Symposium on Spatial and Temporal Databases(SSTD '01), 2001. 257~278

6 Shan Jing, Zhang Donghui, Salzberg B. On Spatial-Range Closest-Pair Query. In: Proc. of 8th Symposium on Spatial and Temporal Databases (SSTD'03), 2003. 252~269

7 Guttman. R-trees: A Dynamic Index Structure for Spatial Searching. SIGMOD, 1984

8 Beckmann N, Kriegel H P, Schneider R, Seeger B. The R*-tree: an Efficient and Robust Access Method for Points and Rectangles. SIGMOD, 1990