

空间数据库中最小距离聚集查询及其算法^{*})

刘云生 刘小峰 肖迎元

(华中科技大学计算机学院 武汉 430074)

摘要 提出了一种新的距离查询形式—最小距离聚集查询,这种查询计算几个对象集中对象到一个中心对象集中对象的距离和,并返回最小的K个距离和。在空间数据库中,对基于R树索引的数据集给出了基于最近邻居的方法和阈值算法来回答查询。通过大量实验对两种算法进行了比较,结果显示阈值算法具有较好的性能。

关键词 空间数据库,查询处理,距离查询,距离聚集

Algorithms for Minimum Distances Aggregation Query in Spatial Databases

LIU Yun-Sheng LIU Xiao-Feng XIAO Yin-Yuan

(School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074)

Abstract In this paper, a new form of distance query, minimum distances aggregation query, is proposed. This kind of query computes the aggregations of distances from objects of several data sets to objects of a central data set, and return the K minimum distances aggregations. In spatial databases, nearest-neighbor-based method and threshold algorithm are presented for the data sets stored in the R-tree family. A performance study based on extensive experiments shows that threshold algorithm outperforms nearest-neighbor-based method.

Keywords Spatial databases, Query processing, Distance query, Distance aggregation

1 引言

最近邻居(Nearest Neighbor)和最接近对(Closest Pair)查询是空间数据库中两种重要的距离查询形式。最近邻居查询从一个集合P中检索和一个给定对象q最接近的 $K \geq 1$ 对象,这方面的研究工作有:N. Roussopoulos等利用R树索引的一些距离度量属性,提出了分支-限定算法,A. Papadopoulos等对算法进行了性能分析^[1,2];Ferhatosmanogl等扩展N. Roussopoulos的工作,解决了一个约束区域内的最近邻居查询^[3];A. Corral等用批最近邻居方法P中每个对象在集合Q中的最近邻居查询,即所有最近邻居(All Nearest Neighbor)查询^[4]。最接近对查询也是一种距离连接,它检索两个集合中的最接近的 $K \geq 1$ 个对象对。主要研究包括:G. Hjaltason等对于P和Q采用R树索引的情形提出了渐进式距离连接算法,按照对象对之间的距离顺序输出最接近对^[5];A. Corral等在N. Roussopoulos工作的基础之上,扩展R树索引有关距离度量,利用这些度量的性质裁减搜索空间和优化算法^[6]。

本文将讨论一种新的距离查询—最小距离聚集查询。对于某个距离度量函数,给定多维空间中的中心对象集Q和 $n \geq 1$ 个对象集 $R_1, R_2, \dots, R_n$,最小距离聚集查询输出 $K \geq 1$ 个 $n+1$ 元组 $(q, r_1, r_2, \dots, r_n) \in (Q \times R_1 \times R_2 \times \dots \times R_n)$,并且有 $r_1, r_2, \dots, r_n$ 到q的距离的和是最小的K个。最小距离聚集查询可以支持地理信息分析决策及相关服务、城市规划和聚类分析等方面的应用。例如:寻找最接近的酒店和购物广场对,为想购物的住宿者提供服务;城市规划中,考察交通枢纽(中心对象集Q)和周围的学校(R_1)、居民区(R_2)和商业区(R_3)分布情况等,寻找最接近的交通枢纽、学校、居民区和商业区;多个数据集相对于某个中心对象集的聚类情况等。

本文对于最小距离聚集查询的讨论将在空间数据库和欧拉空间中进行,并假设 $Q, R_1, R_2, \dots, R_n$ 利用R树类型结构存储^[7,8]。

2 问题提出

设d维数据空间为 R^d ,O为 R^d 中的有穷对象集,对象 $p, q \in O$ 之间的距离度量函数 $dist(p, q)$ 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 给定 E^d 中有穷对象集 $Q, R_1, R_2, \dots, R_n, (q, r_1, r_2, \dots, r_n) \in (Q \times R_1 \times R_2 \times \dots \times R_n)$,则 $(q, r_1, r_2, \dots, r_n)$ 的距离聚集为: $DA(q, r_1, r_2, \dots, r_n) = \sum_{k=1}^n dist(q, r_k)$ 。称n为聚集度, $(q, r_1, r_2, \dots, r_n)$ 为关于q的聚集元组。

由距离聚集的定义可知,给定 $Q \times R_1 \times R_2 \times \dots \times R_n$ 上的聚集元组 $(q, r_1, r_2, \dots, r_n)$ 和 $(q, r'_1, r'_2, \dots, r'_n)$,对于每个 $1 \leq i \leq n$ 有 $dist(q, r_i) \leq dist(q, r'_i)$,则 $DA(q, r_1, r_2, \dots, r_n) \leq DA(q, r'_1, r'_2, \dots, r'_n)$,所以有下面的性质1:

性质1 距离聚集DA是单调的。

定义2 给定 E^d 中有穷对象集 $Q, R_1, R_2, \dots, R_n, K$ 最小距离聚集查询返回结果为: $K-DAQ(Q, R_1, R_2, \dots, R_n, K) = ((q_1, r_{11}, r_{21}, \dots, r_{n1}), (q_2, r_{12}, r_{22}, \dots, r_{n2}), \dots, (q_k, r_{1k}, r_{2k}, \dots, r_{nk}))$,且有:

(1) $(q_i, r_{1i}, r_{2i}, \dots, r_{ni}) \in (Q \times R_1 \times R_2 \times \dots \times R_n), 1 \leq i \leq K$;

(2) $|K-DAQ(Q, R_1, R_2, \dots, R_n, K)| = K$;

(3) $\forall (q, r_1, r_2, \dots, r_n) \in ((Q \times R_1 \times R_2 \times \dots \times R_n) - K-DAQ(Q, R_1, R_2, \dots, R_n, K))$:

$DA(q, r_1, r_2, \dots, r_n) \geq DA(q_k, r_{1k}, r_{2k}, \dots, r_{nk}) \geq \dots \geq$

^{*})基金项目:国防预研基金(413150403)资助。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。刘小峰 讲师,主要研究领域为空间数据库,实时数据库,时空数据库。

$DA(q_2, r_{12}, r_{22}, \dots, r_{n2}) \geq DA(q_1, r_{11}, r_{21}, \dots, r_{n1})$ 。

由定义 1 和定义 2 可知, 当 $n=1$ 时, K 最小距离聚集查询成为 K 最近对查询。因此, 最接近查询对是最小距离聚集查询的特例。

性质 2 Q, P 为 E^d 中有穷对象集, K 最近对 (K Closest Pairs) 查询为 K -CPQ(Q, P, K), 则 K -CPQ(Q, P, K) = K -DAQ(Q, P, K)。

使用文[9]中扩展 SQL 语法, K 最小距离聚集查询可以表示如下:

```
SELECT *
FROM Q, R1, R2, ..., Rn, sum(dist(Q, R1), ..., dist(Q, Rn)) AS d
ORDER BY d
[STOP AFTER <K>].
```

3 基于最近邻居的方法

寻找最小的 K 个距离聚集的一个简单和直接方法是, 对于 Q 中的每个对象 q , 求关于 q 的聚集元组中最小的 K 个距离聚集, 这样共有 $K|Q|$ 个聚集元组, 然后从这 $K|Q|$ 个元组获得最小的 K 个距离聚集, 产生最后的结果。定理 1 保证这种方法的正确。

定理 1 给定 E^d 中有穷对象集 $Q, R_1, R_2, \dots, R_n, M_q$ ($q \in Q$) 表示 $\{q\} \times R_1 \times R_2 \times \dots \times R_n$ 中最小的 K 个距离聚集元组集合, 所有这些 M_q 的并记作 M , 则 K -DAQ($Q, R_1, R_2, \dots, R_n, K$) $\subseteq M$ 。

证明: 假设存在聚集元组 $(q, r_1, r_2, \dots, r_n) \in K$ -DAQ($Q, R_1, R_2, \dots, R_n, K$), 但 $(q, r_1, r_2, \dots, r_n) \notin M$, 则 $(q, r_1, r_2, \dots, r_n) \notin M_q$, 即 $(q, r_1, r_2, \dots, r_n)$ 最多只能为关于 q 的第 $K+1$ 个最小距离聚集元组, 所以 $(q, r_1, r_2, \dots, r_n) \notin K$ -DAQ($Q, R_1, R_2, \dots, R_n, K$), 矛盾。所以 K -DAQ($Q, R_1, R_2, \dots, R_n, K$) $\subseteq M$ 。证毕。

对于任何 $q \in Q, M_q$ 可以按照以下方法获得, 对每个 R_i ($1 \leq i \leq n$) 求 q 的 K 最近邻居, 然后由这 n 组 K 最近邻居求关于 q 的最小 K 个距离聚集元组。这相当于从 n 个表 (表大小为 K) 中每次各取一个元素, 计算和, 选取最小的 K 个和。从定理 2 可以知道, M_q 中聚集元组的分量属于 q 在 R_i 中 K 最近邻居。

定理 2 M_q 为关于 q 的最小 K 个距离聚集元组集合, $KNN(R_i, q)$ 为 q 在 R_i 中 K 最近邻居, $1 \leq i \leq n$, 则对于任意的 $(q, r_1, r_2, \dots, r_n) \in M_q, r_i \in KNN(R_i, q)$ 。

证明: 假设存在 $(q, r_1, r_2, \dots, r_n) \in M_q$ 但 $r_i \notin KNN(R_i, q)$ ($1 \leq i \leq n$), 即 r_i 至多是 q 在 R_i 中的第 $K+1$ 个最近邻居, 那么从 $KNN(R_i, q)$ 中任取一个元素代替 $(q, r_1, r_2, \dots, r_n)$ 中的 r_i 构成新的聚集元组, 该聚集元组的距离聚集小于 $(q, r_1, r_2, \dots, r_n)$, 这样可以找到 K 个比 $(q, r_1, r_2, \dots, r_n)$ 距离聚集小的元组, 因此 $(q, r_1, r_2, \dots, r_n) \notin M_q$, 矛盾。所以, 对于任意的 $(q, r_1, r_2, \dots, r_n) \in M_q, r_i \in KNN(R_i, q)$ 。证毕。

这种基于最近邻居计算 K -DAQ($Q, R_1, R_2, \dots, R_n, K$) 的方法, 称为 NNBM (NN-based method)。下面是 NNBM 算法:

```
NNBM(Q, R1, R2, ..., Rn, K)
用 K 个 (NIL, ∞) 初始化容量为 K 的最大堆 H;
for each object q ∈ Q do
for i = 1 to n do
li = getKNN(Ri, q); // 求 Ri 中 q 的 K 最近邻居
end for
利用 li (1 ≤ i ≤ n) 计算关于 q 的最小 K 个距离聚集元组集合 Mq;
for each 聚集元组 nr ∈ Mq do
if nr 的距离聚集 < H 堆顶元素距离聚集 then H 堆顶元素出堆; (nr, nr 的距离聚集) 入堆 H;
```

```
end if
end for
end for
return H; // End of NNBM
```

4 阈值算法

NNBM 由于对 Q 中数据循环处理, 导致效率不是很好。Fagin 等针对中间件查询提出了一种阈值处理方法^[10], 这种方法可以扩展以应用于最小距离聚集查询处理。

4.1 阈值思想

设有穷对象集 O 中对象具有 n 个属性 $A_1, A_2, \dots, A_n, A_i$ ($1 \leq i \leq n$) 在实数 R 上取值。给定对象 $r(a_1, a_2, \dots, a_n) \in O$ 和单调的聚集函数 $af: R^n \rightarrow R, af(a_1, a_2, \dots, a_n)$ 计算对象 r 的等级, 基于这个等级可以对 O 中对象进行排序。聚集函数的单调性保证如果对象 r 的各个属性的值不小于 s 的各个属性的值时, r 的等级不小于 s 的等级。

O 中对象根据 A_i ($1 \leq i \leq n$) 的值可以形成一个升序排列的表 L_i, L_i 中元素格式为 (O 中对象, 对象的 A_i 属性值)。对 L_i 的访问有两种方式: 顺序访问和随机访问。在对 L_i 顺序访问时, 用 t_i 表示最近访问的属性值, 定义阈值 $t = af(t_1, t_2, \dots, t_n)$, 用一个容量为 $K \geq 1$ 的最大堆 H 保存已经发现的最小 K 个等级对象和它们的等级值。

阈值方法具体步骤如下:

(1) 初始化 t_i 和 t 为 0, 用 K 个 (NIL, ∞) 初始化最大堆 H ;

(2) 当 t 小于 H 堆顶元素的等级时, 依次顺序访问表 L_i ($1 \leq i \leq n$), 设 L_i 当前顺序访问对象为 r , 随机访问其他表获得 r 的其他属性值, 计算 r 的等级, 更新 t_i 和 t 。如果 r 的等级值小于堆顶元素的等级值, 则堆顶元素出堆, 对象 r 和相应的等级值入堆;

(3) 当 t 大于或等于 H 堆顶元素的等级时, 结束, H 为最后的输出结果, 否则继续步骤 (2)。

以下是阈值方法的一个例子, O 对象中有两个属性, 对象 $o(a_1, a_2) \in O$, 聚集函数 $af(a_1, a_2) = a_1 + a_2$ 计算 o 的等级, $o_1(2.1, 1.2), o_2(0.8, 0.7), o_3(3.4, 1.6), o_4(0.3, 0.8), o_5(1.3, 2.3)$ 为 O 中对象, 寻找最小 2 个等级及其对象。

表 L_1 为: $(o_4, 0.3), (o_2, 0.8), (o_5, 1.3), (o_1, 2.1), (o_3, 3.4)$; 表 L_2 为: $(o_2, 0.7), (o_4, 0.8), (o_1, 1.2), (o_3, 1.6), (o_5, 2.3)$ 。

表 1 是阈值方法计算过程。访问对象列中的下划线数字表示顺序访问表得到的对象属性。先顺序访问 L_1 得到 o_4 的第一个属性 0.3, 然后随机访问 L_2 得到 o_4 第二个属性 0.8, o_4 的等级为 1.1, H 堆顶元素出堆, $(o_4, 1.1)$ 入堆, $t_1 = 0.3, t = 0.3 + 0 = 0.3$ 。接着顺序访问 L_2 得 o_2 的第二个属性 0.7, 随机访问 L_1 获得 o_2 的第一个属性 0.8, o_2 的等级为 1.5, 更新堆 H, t_2 和 t 。当 t 更新为 1.5, t 等于堆顶元素等级, 停止, 得到最后的结果。

表 1 阈值方法例子

t_1	t_2	t	堆 H	访问对象
0	0	0	(NIL, ∞)(NIL, ∞)	$o_4: \underline{0.3} + 0.8 = 1.1$
0.3	0	0.3	(NIL, ∞)($o_4, 1.1$)	$o_2: 0.8 + \underline{0.7} = 1.5$
0.3	0.7	1.0	($o_2, 1.5$)($o_4, 1.1$)	$o_2: \underline{0.8} + 0.7 = 1.5$
0.8	0.7	1.5	($o_2, 1.5$)($o_4, 1.1$)	停止

引理 1 对于单调的聚集函数 af , 阈值方法正确地获得

最小 K 个等级的对象。

证明:当堆 H 未满,方法第(2)步向 H 中填入 K 个元素,显然,这 K 个元素的等级是已经看到对象中的最小 K 个;当堆满时,步骤(2)保证入堆的元素的等级值是已经看到的对象中最小的 K 个等级。因此,步骤(2)每次执行都能保证 H 中的元素的等级值是已经看到对象中的最小的 K 个等级。当阈值 t 大等于堆顶元素等级值时,方法结束。此时,对于所有未看到的对象,它的每个属性值 $a_i \geq t_i$,由聚集函数的单调知道它们的等级 $af(a_1, a_2, \dots, a_n) \geq t (=af(t_1, t_2, \dots, t_n)) \geq$ 堆顶元素等级,所以堆中对象的等级为最小的 K 个等级。证毕。

4.2 阈值算法

现在可以对最小距离聚集问题 $K\text{-DAQ}(Q, R_1, R_2, \dots, R_n, K)$ 换个角度来看。 $Q \times R_i (1 \leq i \leq n)$ 中的二元组 (q, r_i) 可以按照 q 和 r_i 之间的距离形成一个升序表 L_i , L_i 表中元素格式为 $((q, r_i), \text{dist}(q, r_i))$, L_i 中第一个元组是 Q 和 R_i 之间的最接近对,第二个元组是 Q 和 R_i 之间次最接近对,依次类推。 L_i 表中元素可以通过渐进式距离连接获得,并按照距离升序访问,不需要事先计算保存。 $\text{dist}(q, r_i)$ 看作 q 的第 i 个属性值, q 共有 n 个属性,聚集函数 af 为距离聚集 DA , DA 计算 q 的等级。这样, K 最小距离聚集查询就转换成求最小 K 个等级的问题。

下面是 $K\text{-DAQ}(Q, R_1, R_2, \dots, R_n, K)$ 的阈值算法 (Threshold Algorithm, TM):

$\text{TM}(Q, R_1, R_2, \dots, R_n, K)$

初始化 t_i 和 t 为 0, 用 K 个 (NIL, ∞) 初始化最大堆 H ;

while $t < H$ 堆顶聚集元组的距离聚集 do

for $i = 1$ to n do

$(q, r_i) = \text{getNextCP}(Q, R_i)$;

//渐进式距离连接,顺序访问表 L_i

$t_i = \text{dist}(q, r_i)$; $t = \sum t_i$; // 更新 t_i 和 t

求 q 在其他数据集 $Q_j (j \neq i)$ 中的最近邻居 r_j ; // 随机访问其他表

计算形成的聚集元组 at 的距离聚集 d ;

if $d < H$ 堆顶元素的距离 then

H 堆顶元素出堆; (at, d) 入堆 H ;

end if

end for

end while

return H ; // End of TM

由引理 1 和性质 1 (距离聚集 DA 的单调性) 可以知道 TM 算法是正确的, 即:

定理 3 TM 正确地找到最小 K 个距离聚集。

实验结果与结论 实验从以下几个方面对算法 NNBM 和 TM 进行分析: (1) 查询响应时间随最小距离聚集数 K 、距离聚集度 n 和数据大小 D 的变化; (2) 磁盘节点访问次数随最小距离聚集数 K 、距离聚集度 n 和数据大小 D 的变化。这里数据大小 D 是 $K\text{-DAQ}(Q, R_1, R_2, \dots, R_n, K)$ 中每个数据集的大小。实验数据采用一致分布的合成数据, 数据类型为二维点类型。空间索引采用文[7]中的 R 树, R 树节点对应页面的大小为 1K 字节, 采用无缓冲区机制。实验程序用 C++ 实现, 由 Visual C++ 6 编译, 在 Pentium 1.7G, 256M 内存的计算机上运行。

图 1 是两个算法查询响应时间随最小距离聚集数 K 、距离聚集度 n 和数据大小 D 的变化。图 2 是磁盘节点访问次数随最小距离聚集数 K 、距离聚集度 n 和数据大小 D 的变化。

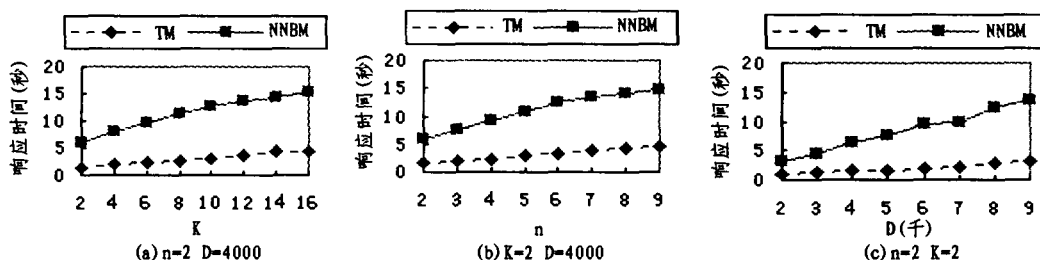


图 1 查询响应时间

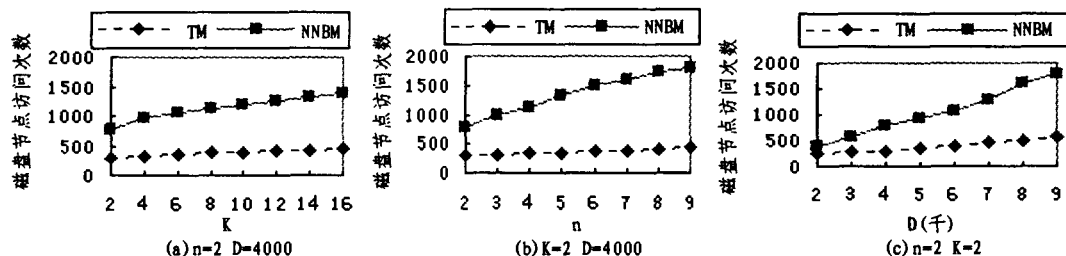


图 2 磁盘节点访问次数

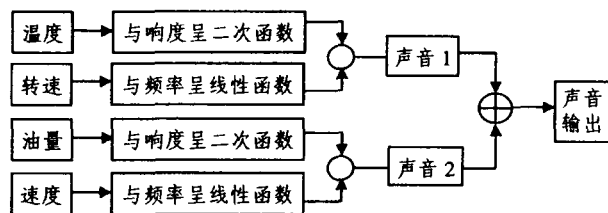
根据实验可以得出以下结论: (1) TM 比 NNBM 无论是在 CPU 还是 I/O 方面具有更好的性能; (2) TM 查询响应时间和磁盘节点访问次数随着 D 、 n 和 K 的增大变化不大, 具有较好的伸缩性能。

产生以上结果的原因是: (1) NNBM 需要对 Q 中每个对

象 q 求 R_i 中 q 的 K 最近邻居, 然后计算 M_q , 从众多的中间结果 M_q 中选择最小的 K 个距离聚集, 这将产生大量的 I/O 和计算。 (2) TM 中计算和磁盘访问主要代价是顺序访问表, 即渐进式距离连接函数 $\text{getNextCP}(Q, R_i)$, 但是 TM 具有逐步

(下转第 122 页)

户可以选择不同的映射函数、参数范围等设定,以适应用户的听觉感知特点。



通道 1 映射设置如下:

①温度映射到声音的幅度:输入参数最小值 0,最大值 105,输出参数类型选择幅度,映射函数表达式为 $0.01 * \exp(0.5 * x)$; $A = 0.01 * \exp(0.5 * T)$; A 为声音 1 的幅度, T 为温度。

②转速映射到频率:输入参数最小值 0,最大值 7000 (rpm),输出参数类型频率。映射函数表达式 $x+50$; $F = R + 50$; F 为声音 1 的频率, R 为转速。

通道 2 映射设置如下:

①油量映射到声音的幅度:输入参数最小值 0,最大值 4,输出参数类型选择幅度,映射函数表达式为 $0.01 * \exp(1.2 * x)$; $A = 0.01 * \exp(1.2 * G)$; A 为声音 2 的幅度, G 为油量。

②速度映射到频率:输入参数序号 1,最小值 0,最大值 200,输出参数类型频率,映射函数表达式 $20 * x + 50$ 。 $F = 20 * V + 50$; F 为声音 2 的频率, V 为速度。

4.3 AudiMeters 实验评价

可听化应用设计能否成功,最大的不确定因素就是数据映射。因为映射不单纯是一个信息编码问题,还涉及到用户的听觉感知的心理学问题,即复杂的人因(Human Factors)问题。成功的可听化设计不能仅靠设计者主观决定,需要进行严格的可用性测试。图 3 所示为通过反复试验后确定的初步

设计方案之一,这一方案已经通过了非正式的实验评估,在实验室环境下证明是可接受的^[7]。系统运行表明,在两通道四维线索的声场中,用户能优先辨别速度和转速的变化。同时,不同声音线索之间的干扰是存在的,但是用户可以通过关闭对于某些仪表参数的映射来消除或减轻干扰。

结束语 AudiMeters 作为一个实验平台进行数据映射探索和试验,目前其技术层面的工作基本完成,可以进行实时的演示,并可以设定不同的数据映射方案和参数设定。下一步的工作是实现可用性测试的数据采集和处理接口,并进行严格的可用性测试和工效学研究。

参 考 文 献

- 1 Kramer G, et al. Sonification Report: Status of the Field and Research Agenda. See <http://www.icad.org/websiteV2.0/References/nsf.html>.
- 2 张琼,宋哲,石教英.可听化技术综述[J].计算机辅助设计与图形学学报,1999,11(2):189~192
- 3 Brewster S A. Providing a Structured Method for Integrating Non-speech Audio into Human-computer Interfaces[D]; [PhD Thesis]. University of York, UK, 1994
- 4 Bly S. Multivariate Data Mappings. In: Kramer G ed. Auditory Display: Sonification, Audification and Auditory interfaces Reading, MA. USA: Addison-Wesley Publishing Company, 1994. 405~416
- 5 Mansur D L, Blattner M M, Joy K I. Sound-Graphs: A Numerical Data Analysis Method for the Blind. J. Med. Sys., 1985, 9: 163~174
- 6 Yeung E S. Pattern Recognition by Audio Representation of Multivariate Analytical Data. Anal. Chem. [J], 1980, 52: 1120~1123
- 7 徐义东. 听觉显示开发平台的设计及应用[D]; [浙江大学硕士论文]. 2004. 3

(上接第 110 页)

收敛的性质,当 K 、 n 和 D 增长时, $getNextCP(Q, R_i)$ 调用次数增长并不明显,因此具有较好的伸缩性能。

由于最小距离聚集查询可以看作一种特殊的多路空间连接^[11],下一步的研究工作将把多路连接方面的研究成果扩展到最小距离聚集查询处理以进一步提高查询处理效率。

参 考 文 献

- 1 Roussopoulos N, Kelley S, Vincent F. Nearest Neighbor Queries. In: Proc. ACM SIGMOD, 1995. 71~79
- 2 Papadopoulos A N, Manolopoulos Y. Performance of Nearest Neighbor Queries in R Trees. In: Proc. 6th Int. Conf. on Database Theory (ICDT'97), 1997. 394~408
- 3 Ferhatosmanoglu H, Stanoi I, Agrawal D, Abbadi A. Constrained Nearest Neighbor Queries. SSTD, 2001
- 4 Mamoulis N, Papadias D, Tao Y. All-Nearest-Neighbors Queries in Spatial Databases. In: IEEE Conf. on Scientific and Statistical

Database Management (SSDBM), 2004

- 5 Hjalason G, Samet H. Incremental Distance Join Algorithms for Spatial Databases. SIGMOD, 1998
- 6 Corral A, Manolopoulos Y, Theodoridis Y, Vassilakopoulos M. Closest Pair Queries in Spatial Databases. SIGMOD, 2000
- 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
- 9 Carey M J, Kossmann D. On saying "enough already!" in SQL. In: Proc. of ACM SIGMOD, 1997. 219~230
- 10 Fagin, Lotem R, Naor A. Optimal Aggregation Algorithms for Middleware. PODS, 2001
- 11 Mamoulis P N, Theodoridis Y. Processing and Optimization of Multi-way Spatial Joins Using R-trees. In: Proc. 18th ACM PODS Conf. 1999. 44~55