

受限空间连接查询及代价分析

杨泽雪^{1,2}, 郝忠孝^{1,3}

(1. 哈尔滨理工大学 计算机科学与技术学院, 150080 哈尔滨; 2. 黑龙江工程学院 计算机科学与技术系, 150050 哈尔滨;
3. 哈尔滨工业大学 计算机科学与技术学院, 150001 哈尔滨)

摘 要: 针对已有的空间连接查询算法无法解决限定范围内的空间连接查询问题, 提出了受限的空间连接查询, 在给定查询范围内找到满足某种空间谓词的空间对象, 给出直接解决方法和基于 R-树的受限空间连接查询算法. 基于 QR 树的优良特性, 提出一种基于 QR 树的受限空间连接查询算法, 该算法既避免了四叉树的较大存储代价, 又克服了 R 树的节点重复的弊端, 使得受限空间连接查询可以在多棵较小的 R 树上进行, 较好地解决了空间连接查询开销较大的问题. 对所提出的算法进行代价分析, 实验证明算法具有较高效率.

关键词: 空间连接查询; QR 树; 空间数据库; R 树; 受限空间连接查询

中图分类号: TP311.13

文献标志码: A

文章编号: 0367-6234(2012)11-118-05

Constrained spatial join queries and cost analysis

YANG Ze-xue^{1,2}, HAO Zhong-xiao^{1,3}

(1. College of Computer Science and Technology, Harbin University of Science and Technology, 150080 Harbin, China;
2. Dept. of Computer Science and Technology, Heilongjiang Institute of Technology, 150050 Harbin, China;
3. College of Computer Science and Technology, Harbin Institute of Technology, 150001 Harbin, China)

Abstract: Aimed at the problem that the existed spatial join algorithms can not solve the spatial join query within the constrained range, the constrained spatial join query is proposed which finds all the pairs of objects satisfying some spatial predicate within the given range. The directed solving methods and algorithms based on R-tree are given. Based on good property of QR-tree, a constrained spatial join algorithm is proposed which avoids larger storage cost of quadtree and overcomes the drawbacks of R-tree node overlapping. Thus the algorithm implements the constrained spatial join query on many small R-tree and the problem of expensive spatial join overhead is solved. The cost analysis for the proposed algorithms is given. Experiments show the algorithm has high efficiency.

Key words: spatial join query; QR-tree; spatial database; R-tree; constrained spatial join query

近年来空间连接查询成为空间数据库中的热点问题. 给定两个空间数据集, 空间连接查询找到两个数据集中所有满足某个空间谓词的对象. 空间谓词包括交叠、相交、方向和距离等, 本文主要研究相交的情况. 例如: 找到与某条河流相交的所有城市. 按照两个空间数据集是否有索引, 可以将

空间连接查询分为 3 种情况: 两个数据集都有索引^[1-7]; 一个数据集有索引^[8-9]; 两个数据集都没有索引.

对于受限的空间连接查询, 没有找到文献. 考虑到若对于两个数据集的空间连接只考虑某个限定的空间而不是整个空间, 提出了受限的空间连接查询. 例如: 找到某个国家与河流相交的城市. 而已有的算法都不能直接用于某个限定的空间, 根据已有的空间连接方法和范围查询方法, 给出受限空间连接的直接解决方法, 并给出基于 R-树的受限空间连接算法. 针对基于 R-树的空间连接方法和四叉树的空间连接方法的局限性, 提

收稿日期: 2011-08-22.

基金项目: 国家自然科学基金资助项目(60673136); 黑龙江省自然科学基金资助项目(F201134).

作者简介: 杨泽雪(1978—), 女, 博士, 讲师;

郝忠孝(1940—), 男, 教授, 博士生导师.

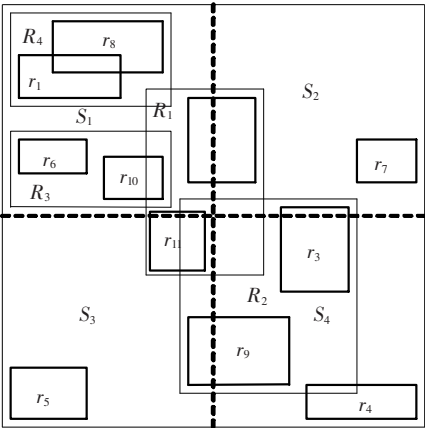
通信作者: 杨泽雪, yzx1978@126.com.

出了基于 QR 树的受限空间连接方法.

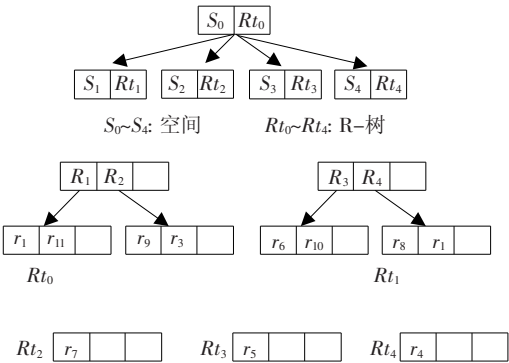
1 QR 树^[10]

QR 树是结合四叉树和 R 树而提出的一种空间索引结构. 设 $d > 0, n = \sum_{l=0}^{d-1} (2^k)^l$, 其中: k 为空间的维数; d 为四叉树深度, 则 QR 树由一棵深度为 d 的四叉树 Q_t 和 n 棵 R 树组成. Q_t 共有 n 个结点, 依次为: $Q_{t_0}, Q_{t_1}, \dots, Q_{t_{n-1}}$. Q_t 将整个数据空间 S 划分成 n 个 d 级子空间, 依次记为: $S_0, S_1, \dots, S_{n-1}$ (其中 $S_0 = S$). 每一级的所有子空间两两不相交, 且一起构成整个索引空间 S .

图 1 是二维空间 QR 树的一个例子. 在该例中, QR 树由一棵深度为 2 的四叉树和 5 棵 R 树组成. 整个空间被划分为 2 级共 5 个子空间: S_0, S_1, S_2, S_3, S_4 ($S_0 = S_1 \cup S_2 \cup S_3 \cup S_4$), $R_{t_0}, R_{t_1}, R_{t_2}, R_{t_3}, R_{t_4}$ 这 5 棵 R 树分别与之相关联.



(a) 空间划分



(b) QR-树结构

图 1 二维空间 QR-树

2 基于 R-树的受限空间连接查询

2.1 受限空间连接查询的直接解决方法

给定两个数据集和一个查询范围, 受限的空间连接(CSJ)查询是指找到与给定查询范围相交的且满足某空间谓词的数据对象的集合.

受限的空间连接查询是空间连接查询和空间范围查询的结合. 虽然已有很多方法解决了空间连接查询和范围查询, 但已有的方法并不能直接实现受限的空间连接查询. 对于受限的空间连接查询, 直接的解决方法就是顺序执行两个算法, 按照两个算法的执行顺序分为两种方法, 一种是先进行范围查询, 然后再进行连接查询, 此方法表示为 FRLJ. 另一种方法是先进行连接查询, 然后在进行范围查询, 此方法表示为 FJLR.

假设两个数据集都由 R-树索引, 对应 R-树索引分别为 A 和 B , 给定受限空间为 W . 利用基于 R-树的空间连接算法和基于 R-树的空间范围查询算法^[11], 实现 FRLJ 的过程如下:

- 1) 对 A, B 在 W 上分别执行范围查询算法, 获取 A 中与 W 相交的结果集 R_1 和 B 中与 W 相交的结果集 R_2 ;
- 2) 对 R_1 和 R_2 执行基于 R-树的空间连接算法, 得到最终结果.

实现 FJLR 的过程如下:

- 1) 对 A, B 执行基于 R 树的空间连接算法, 获取两个数据集中相交的对象对的集合 R ;
- 2) 对 R 执行范围查询算法, 判断 R 中每一对对象对应 MBR 是否与 W 相交, 若相交, 输出结果, 否则继续其他结果的判断.

分析两种方法发现, 如果受限范围较小, FRLJ 的效果较好, 因为执行范围查询可以剪掉许多不会成为结果的节点, 但随着受限范围的增大, FRLJ 的优势逐渐减小; 而对于 FJLR 来说, 首先执行连接查询, 对于两个数据集的数量都不是很大时, 效果较好, 但随着数据量的增大, 连接代价变得很昂贵, 因此效率不高.

2.2 基于 R 树的受限空间连接查询算法

基于上述分析, 将范围查询和空间连接查询结合在一起, 提出基于 R-树的受限空间连接算法 RCSJ. 该算法的基本思想是: 对两棵 R 树进行同步遍历, 在遍历过程中, 设定节点是否与限定范围 W 相交作为条件, 按照节点的类型可分为两个节点都为叶节点、一个为叶节点另一个为中间节点和两个都为中间节点 3 种情况分别进行处理.

3 基于 QR-树的受限空间连接查询

利用 QR 树将四叉树和 R-树相结合的良好特性, 提出一种基于 QR 树的受限空间连接算法, 在连接查询的过程中判断节点是否与限定空间相交, 通过一次的树遍历可得到查询结果. 算法的基本思想是: 利用四叉树将空间进行划分, 然后在

每个子空间按照数据进行划分构成 R-树,利用 R-树实现受限空间连接. 因此按照空间划分和数据划分两个方面实现受限空间连接. 给定两个数据集和对应的两棵 QR 树, 首先将两棵 QR 树的四叉树节点进行匹配, 然后将匹配的节点对所对应的 R-树的 MBR 以及受限空间进行相交判断, 如果相交, 则调用对应的 R-树进行连接.

3.1 节点匹配算法

考虑到 QR 树的构成, 将节点分配到包含该节点的子空间中的最小者, 则上一层子空间节点可能与下一层子空间的节点相交. 所以要进行不同层空间的节点匹配. 算法如下.

算法 1 $NM(N_A^l, N_B^l, l)$

输入: N_A^l, N_B^l ;

输出: 匹配的节点对集合 R .

begin

for $i = 0$ to $4^l - 1$ do

$R = (N_{Ai}^l, N_{Bi}^l)$;

for $j = 1$ to l do

for $m = 0$ to $4^{(l-j)} - 1$ do

if $S_{Ai}^l \subset S_{Bm}^{(l-j)}$;

then

$R = R \cup (N_{Ai}^l, N_{Bm}^{(l-j)})$;

for $i = 0$ to $4^l - 1$ do

for $j = 1$ to l do

for $m = 0$ to $4^{(l-j)} - 1$ do

if $S_{Bi}^l \subset S_{Am}^{(l-j)}$;

then

$R = R \cup (N_{Bi}^l, N_{Am}^{(l-j)})$;

return R ;

end.

3.2 连接算法

算法思想如下.

1) 从两棵 QR 树的根开始, 对两棵 QR 树进行同步遍历.

2) 对于每一对节点, 对与节点相关联的子空间和受限空间进行相交判断, 如果相交, 则调用相应的 R-树进行连接, 否则对该节点及其子节点的判断结束.

3) 针对两个节点的类型, 分成 3 种情况, 对每种情况分别进行处理.

算法 2 $QRCSJ(N_A, N_B, W)$

输入: 两棵 QR 树 A, B 的节点 N_A, N_B , 限定空间 W ;

输出: 相交的数据对组成的结果集 CRL .

begin

$CRL = \Phi$;

$IS_A =$ 与 N_A 相关联的子空间;

$IS_B =$ 与 N_B 相关联的子空间;

if $IS_A \cap IS_B = \Phi$ then return CRL ;

else

if $IS_A \cap IS_B \cap W = \Phi$ then return CRL ;

else

if $N_A \cdot MBR \cap W \neq \Phi$ and $N_B \cdot MBR \cap W \neq \Phi$ and $N_A \cdot MBR \cap N_B \cdot MBR \neq \Phi$ then

$CRL = CRL \cup RSJ(N_A \cdot PR \cap W, N_B \cdot PR \cap W)$ // 调用 R 树连接算法;

if N_A, N_B 是叶子节点 then

return CRL ;

else if N_A 是叶子节点, N_B 是中间节点 then

for N_B 的每个子节点 N_{Bi} do

$CRL = CRL \cup QRCSJ(N_A, N_{Bi}, W)$;

else

for N_A 和 N_B 的子节点集 N_A^l 和 N_B^l do

$R = NM(N_A^l, N_B^l)$;

for 每个 $(N_A^i, N_B^j) \in R$ do

if $i \neq j$ then

if $N_A^i \cdot MBR \cap W \neq \Phi$ and

$N_B^j \cdot MBR \cap W \neq \Phi$ and

$N_A^i \cdot MBR \cap N_B^j \cdot MBR \neq \Phi$ then

$CRL = CRL \cup RSJ(N_A^i \cdot PR \cap W, N_B^j \cdot PR \cap W)$;

else

$CRL = CRL \cup QRCSJ(N_A^i, N_B^j, W)$;

return CRL ;

end.

4 代价分析

4.1 基于 R 树的受限空间连接查询代价分析

设受限空间为 $W = (q_1, q_2, \dots, q_d)$, 设两棵 R 树分别为 A, B , 其对应数据个数分别为 M_A, M_B , 树的高度分别为 h_A, h_B , 设 $h_A \geq h_B$, 则 $h_A = 1 + \lceil \log_{cM} \frac{M_B}{cM} \rceil, h_B = 1 + \lceil \log_{cM} \frac{M_B}{cM} \rceil$, 其中: c 为平均结点容量百分比; M 为结点 (包含叶结点和非叶结点) 存储的最大数据项数量; cM 为扇出度. 由文献[12]可知, 两棵 R-树 A, B 进行空间连接, 总的连接代价为

$$C_R(A, B) = \sum_{j=1}^{h_A-1} [NA(A, B, j) + NA(B, A, j')].$$

其中

$$j' = \begin{cases} 1, 1 \leq j \leq h_A - h_B; \\ j - (h_A - h_B), h_A - h_B + 1 \leq j \leq h_A - 1. \end{cases}$$

$$NA(A, B, j) = N_{A,j} N_{B,j} \prod_{t=1}^k \min\{1, (s_{A,j,t} + s_{B,j',t})\}.$$

因为进行受限空间连接要计算每一层节点与 W 相交且彼此相交的数据个数, 因此要计算每一层 R -树节点与 W 相交的概率, 为此有

$$NA(A, B, j) = N_{A,j} N_{B,j'} \prod_{t=1}^k \min\{1, (s_{A,j,t} + s_{B,j',t})\} \prod_{t=1}^k (s_{A,j,t} + q_t) \prod_{t=1}^k (s_{B,j',t} + q_t).$$

所以总的连接代价为

$$C_R(A, B) = O\left(\sum_{j=1}^{h_A-1} \left(N_{A,j} N_{B,j'} \prod_{t=1}^k \min\{1, (s_{A,j,t} + s_{B,j',t})\} \prod_{t=1}^k (s_{A,j,t} + q_t) \prod_{t=1}^k (s_{B,j',t} + q_t)\right)\right).$$

4.2 基于 QR 树的受限空间连接查询代价分析

设两棵 QR 树分别为 $\langle A, R_A \rangle, \langle B, R_B \rangle$, 其对应数据个数分别为 M_A, M_B , 树的高度分别为 h_A, h_B , 设 $h_A \leq h_B$, 两棵 QR 树划分子空间的个数分别为 n_A, n_B , 则对应 R -树个数也为 n_A, n_B , 其中

$$n_A = \sum_{l=0}^{h_A-1} (2^k)^l, n_B = \sum_{l=0}^{h_B-1} (2^k)^l.$$

假设数据均匀分布, 每棵 R -树对应的数据个数为 $M_A/n_A, M_B/n_B$, 设两棵 R -树 RAu, RBv 的高度分别为 h_{RAu} 和 h_{RBv} , 设 $h_{RAu} \geq h_{RBv}$, 由上述分析可知, 两棵 R -树进行空间连接代价为

$$NA(RAu, RBv) = \sum_{j=1}^{h_{RAu}-1} [NA(RAu, RBv, j) + NA(RBv, RAu, j')].$$

其中

$$j' = \begin{cases} 1, 1 \leq j \leq h_{RAu} - h_{RBv}; \\ j - (h_{RAu} - h_{RBv}), h_{RAu} - h_{RBv} + 1 \leq j \leq h_{RAu} - 1. \end{cases}$$

$$NA(RAu, RBv, j) = N_{RAu,j} N_{RBv,j} \prod_{t=1}^k \min\{1, (s_{RAu,j,t} + s_{RBv,j',t})\} \prod_{t=1}^k (s_{RAu,j,t} + q_t) \prod_{t=1}^k (s_{RBv,j',t} + q_t).$$

式中: $N_{RAu,j}$ 为 RAu 在第 j 层的结点数; $s_{RAu,j,t}$ 为 RAu 在第 j 层的结点的平均范围。

对于 QR-树的受限空间连接代价, 主要计算进行 R -树连接的次数, 按照两个 QR-树节点是否同层, 分别考虑:

1) 两个 QR 树节点位于同一层, 此时对两棵 QR-树同一层所有对应的节点进行 R -树连接, 此时考虑的是最大代价. 则进行 R -树连接的次数是每一层节点个数的总和, 即为 $\sum_{l=0}^{h_A-1} (2^k)^l$, 所以

同层节点连接代价为 $\sum_{l=0}^{h_A-1} (2^k)^l \cdot NA(RAu, RBv)$;

2) 两个 QR 树节点不同层, 此时每棵 QR 树低层的节点分别要和另一棵 QR 树高层的节点进行连接, 总的连接次数为 $\sum_{s=l}^{h_A-1} \sum_{l=s}^{h_A-1} (2^k)^l + \sum_{s=l}^{h_B-1} \sum_{l=s}^{h_B-1} (2^k)^l$, 因此不同层节点连接代价为 $(\sum_{s=l}^{h_A-1} \sum_{l=s}^{h_A-1} (2^k)^l + \sum_{s=l}^{h_B-1} \sum_{l=s}^{h_B-1} (2^k)^l) \cdot NA(RAu, RBv)$.

综合以上两种情况, 基于 QR-树的受限连接查询总代价为

$$C_{QR}(A, B) = \left(\sum_{l=0}^{h_A-1} (2^k)^l + \sum_{s=l}^{h_A-1} \sum_{l=s}^{h_A-1} (2^k)^l + \sum_{s=l}^{h_B-1} \sum_{l=s}^{h_B-1} (2^k)^l \right) \cdot NA(RAu, RBv).$$

4.3 两种查询代价比较

要比较 QRCSJ 和 RCSJ 的查询代价, 通过计算比值 $ratio = C_{QR}(A, B)/C_R(A, B)$ 的值来衡量, 如果 $ratio < 1$, 说明 QRCSJ 要比 RCSJ 好.

$$\text{其中 } C_R(A, B) = 2 \sum_{j=1}^{h_A-1} \left(N_{A,j} N_{B,j'} \prod_{t=1}^k \min\{1, (s_{A,j,t} + s_{B,j',t})\} \prod_{t=1}^k (s_{A,j,t} + q_t) \prod_{t=1}^k (s_{B,j',t} + q_t) \right).$$

因为

$$C_{QR}(A, B) = \left(\sum_{l=0}^{h_A-1} (2^k)^l + \sum_{s=l}^{h_A-1} \sum_{l=s}^{h_A-1} (2^k)^l + \sum_{s=l}^{h_B-1} \sum_{l=s}^{h_B-1} (2^k)^l \right) \cdot NA(RAu, RBv) \approx \left(\sum_{l=0}^{h_A-1} (2^k)^l + \sum_{s=l}^{h_A-1} \sum_{l=s}^{h_A-1} (2^k)^l + \sum_{s=l}^{h_B-1} \sum_{l=s}^{h_B-1} (2^k)^l \right) \frac{1}{n_A n_B}.$$

$$C_R(A, B) = \left(\sum_{l=0}^{h_A-1} (2^k)^l + \sum_{s=l}^{h_A-1} \sum_{l=s}^{h_A-1} (2^k)^l + \sum_{s=l}^{h_B-1} \sum_{l=s}^{h_B-1} (2^k)^l \right) / \left(\sum_{l=0}^{h_A-1} (2^k)^l \cdot \sum_{l=0}^{h_B-1} (2^k)^l \right) \cdot C_R(A, B).$$

$$\text{而 } \left(\sum_{l=0}^{h_A-1} (2^k)^l + \sum_{s=l}^{h_A-1} \sum_{l=s}^{h_A-1} (2^k)^l + \sum_{s=l}^{h_B-1} \sum_{l=s}^{h_B-1} (2^k)^l \right) / \left(\sum_{l=0}^{h_A-1} (2^k)^l \cdot \sum_{l=0}^{h_B-1} (2^k)^l \right) < 1$$

即 $ratio < 1$

因此 $C_{QR}(A, B)$ 明显小于 $C_R(A, B)$.

5 结 果

实验是在 Pentium IV 2.8 GHz CPU, 1 GB 内存, Windows XP 平台上用 Visual C++ 6.0 实现的, 磁盘页大小为 1 024 Bytes. 测试数据是由 Spa-

tial Data Generator(<http://www.rtreeportal.org>)产生的标准分布的数据,数据尺寸为12、13 K. 对数据进行实验,图2对受限范围的大小对各个算法的影响进行了实验. 图2中, x 坐标轴表示受限范围的占整个数据空间的百分比,FRLJ、FJLR、RCSJ和QRCSJ表示上述的4种受限空间连接算法,其中QR-树的高度为3. 由图2可知,基于QR树的受限空间连接查询明显优于其他几个算法. 随着查询范围的增大,FRLJ的查询时间逐渐增加,这是因为随着查询范围的增大,先做范围查询的优势越来越小,当查询范围变为整个数据区域的时候,FRLJ的效率是很低的. FJLR的查询时间逐渐变少,这是因为随着查询范围的增大,由连接查询得到的中间结果可能成为最终结果的可能性逐渐增大.

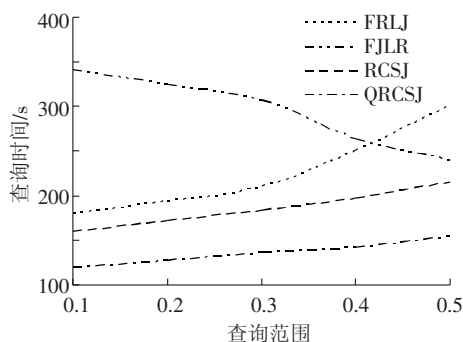


图2 算法执行时间比较

6 结 论

1) 提出受限的空间连接查询,给出直接的解决方法和基于R-树的受限空间连接算法. 基于QR树的优良特性,给出了基于QR树的受限空间连接算法.

2) 通过代价分析得到基于QR树的受限空间连接算法与基于R树的算法代价比值小于1,证明所提算法优于其他的方法.

参考文献:

[1] BRINKHOFF T, KRIEGEL H P, SEEGER B. Efficient processing of spatial joins using R-trees[C]//Proceedings of the ACM SIGMOD International Conference on Management of Data. Washington, DC: ACM, 1993: 237-246.

[2] HUANG Yunwu, JING Ning, RUNDENSTEINER E A. Spatial joins using R-trees: breadth-first traversal with global optimizations[C]//Proceedings of the 23rd International Conference on Very Large Data Bases. San Francisco, CA: Morgan Kaufmann Publishers Inc, 1997: 396-405.

[3] BRINKHOFF T, KRIEGEL H P, SEEGER B. Parallel processing of spatial joins using R-trees[C]//Proceedings of the 12th International Conference on Data Engineering. New Orleans: IEEE, 1996: 258-265.

[4] CHEN Hue-ling, CHANG Ye-in. Spatial joins based on NA-trees[J]. Journal Information Processing Letters, 2009, 109(13): 713-718.

[5] HOEL E G, SAMET H. Benchmarking spatial join operations with spatial output[C]//Proceedings of the 21th International Conference on Very Large Data Bases. San Francisco: Morgan Kaufmann Publishers Inc, 1995: 606-618.

[6] LIMA A A B, ESPERANCA C, MATTOSO M. A parallel spatial join framework using PMR-quadtrees[C]//Proceedings of the 11th International Workshop on Database and Expert Systems Applications. Los Alamitos: IEEE, 2000: 889-893.

[7] 郝忠孝. 时空数据库-查询与推理[M]. 北京: 科学出版社, 2010: 220-230.

[8] GURRET C, RIGAUX P. The sort/sweep algorithm: a new method for R-tree based spatial joins[C]//Proceedings of the 12th International Conference on Statistical and Scientific Database Management. Washington, DC: IEEE Computer Society, 2000: 153-165.

[9] JACOX E H, SAMET H. Spatial join techniques[J]. ACM Transactions on Database Systems, 2007, 32(1): 1-44.

[10] FU Yuchen, HU Zhiyong, WEI Guo, et al. QR-tree: a hybrid spatial index structure[C]//Proceedings of International Conference on Machine Learning and Cybernetics. Xi-an: Institute of Electrical and Electronics Engineers Inc, 2003: 459-463.

[11] GUTTMAN A. R-trees: a dynamic index structure for spatial searching[J]. ACM SIGMOD, 1984, 14(2): 47-57.

[12] THEODORIDIS Y, STEFANAKIS E, SELLIS T. Efficient cost models for spatial queries using R-trees[J]. IEEE TKDE, 2000, 12(1): 19-32.

(编辑 张 红)