

# 路网中基于 RQOP 树的移动对象索引技术研究<sup>①</sup>

张彦满, 余建桥

西南大学 计算机与信息科学学院, 重庆 400715

**摘要:** 基于位置的信息服务需要高效的索引方法来管理移动对象. 针对 PMR QUAD 树索引路网空间时不平衡、部分路段重复存储且索引结构可调整性差的问题, 用 RQOP 树对路网空间按照路段的空间分布进行划分, 使树的高度尽可能低, 改进基于路网的动态组合索引结构. 对照实验表明, 基于 RQOP 树的索引结构提高了查询效率.

**关键词:** RQOP 树; 路网; 移动对象; 索引结构

**中图分类号:** TP311

**文献标志码:** A

移动对象数据库已被广泛应用于交通管理等基于位置的信息服务领域, 移动对象数据的动态特性和查询的实时要求, 使得移动对象索引技术成为研究热点<sup>[1]</sup>. 现实生活中, 移动对象的运动往往限制在一定的拓扑网络中, 称之为路网<sup>[2]</sup>. 当前国内外学者对移动对象索引技术的研究, 往往是借鉴空间数据库和时态数据库现有的索引方法, 通过对其添加一定辅助结构或进行算法改进来满足特定的应用需求<sup>[3-4]</sup>. FNR 树<sup>[5]</sup>在基于路网的移动对象索引方面整体性能表现较好, 文献<sup>[6]</sup>在 FNR 树的基础上提出了一种基于路网的动态组合索引, 该索引结构顶部为 PMR QUAD 树, 中部为 R 树森林, 底部为 HASH 结构. 由于 PMR QUAD 树是通过对路网空间进行四等分的分割来建立索引, 这就要求索引建立前须预知整个路网空间, 空间划分可调整性差, 建立索引后, 当新增路段出现在原路网空间范围外时, 必须重建整个索引. 同时, 由于现实中路段分布不均匀, 这时所产生 PMR QUAD 树高度就不平衡, 从而影响查询效率. 另外, PMR QUAD 树中把路段存储在空间划分后每个与其相交的子空间中, 这就造成了部分路段重复存储, 增加了存储空间和查询代价.

RQOP 树<sup>[7]323</sup>是一种结合了四叉树和 R 树的空间数据索引结构, 按照数据的分布情况对数据空间进行分割, 使树的高度尽可能低, 同时使中间结点间的交叠尽可能小. 针对 PMR QUAD 树索引路网空间时存在的问题, 用 RQOP 树对路网空间按照路段的空间分布进行划分, 进而改进动态组合索引结构, 提高查询性能.

## 1 动态组合索引结构和 RQOP 树

### 1.1 动态组合索引结构

基于路网的移动对象动态组合索引结构使用 PMR QUAD 树基于空间将路网空间平均分割成多级子索引空间, 并且使得树的叶结点包含路段数量在规定阈值内, 各路段被逐个插入与该路段相交的所有子空间块中. PMR QUAD 树叶结点的每一条记录(对应于一条路段)指向一棵 R 树, R 树用来对该路段按时间段(根据时间阈值划分时间段)建立索引, 并管理各个时间段中经过该路段的移动对象. HASH 结构由 HASH 表和动态链表构成, HASH 表中存储了移动对象标志, 每一个移动对象都指向一个单链表. 链表中

① 收稿日期: 2010-11-12

作者简介: 张彦满(1984-), 男, 甘肃庆阳人, 硕士研究生, 主要从事移动对象数据库技术的研究.

通信作者: 余建桥, 教授, 博士.

的结点存放着按时间顺序存储的移动对象时间、速度和位置信息及两个指针,一个指针定位该结点中位置信息在 R 树中的记录位置,另一指针指向该链表的下一个结点.

### 1.2 RQOP 树

RQOP 树通过建立数据 MBR(最小外包矩形)间的序关系对数据空间进行分割,假设任一数据的 MBR 为  $R_n$ ,  $R_n$  左上角点为  $(a_x^n, a_y^n)$ , 右下角点为  $(b_x^n, b_y^n)$ , 有如下定义<sup>[7]324</sup>:

**定义 1** 设  $R_i, R_j (i \neq j)$  是两条不相同路段的 MBR, 若  $a_x^i > a_x^j$ , 或当  $a_x^i = a_x^j$  时,  $a_y^i > a_y^j$ , 或当  $a_x^i = a_x^j$  且  $a_y^i = a_y^j$  时,  $b_x^i > b_x^j$ , 或当  $a_x^i = a_x^j$  且  $a_y^i = a_y^j$ ,  $b_x^i = b_x^j$  时,  $b_y^i < b_y^j$ , 则称  $R_i$  按  $x$  坐标大于  $R_j$ , 记作  $R_i^x > R_j$ .

**定义 2** 设  $R_i, R_j (i \neq j)$  是两条不相同路段的 MBR, 若  $a_y^i > a_y^j$ , 或当  $a_y^i = a_y^j$  时,  $a_x^i > a_x^j$ , 或当  $a_y^i = a_y^j$  且  $a_x^i = a_x^j$  时,  $b_y^i > b_y^j$ , 或当  $a_y^i = a_y^j$  且  $a_x^i = a_x^j$ ,  $b_y^i = b_y^j$  时,  $b_x^i < b_x^j$ , 则称  $R_i$  按  $y$  坐标大于  $R_j$ , 记作  $R_i^y > R_j$ .

**定义 3** 一棵 RQOP 树定义为一棵满足下列条件的四叉树:

- 1) 若根结点不是叶结点, 则它至少有两棵子树;
- 2) 所有中间结点 node 至多有 4 棵子树、至少有 2 棵子树, 并且若孩子结点存在, 则必满足(e、s、w、n 分别代表东、南、西、北 4 个相对方位):

$node \rightarrow child(ne) \rightarrow R^y > node \rightarrow child(se) \rightarrow R$ ;  $node \rightarrow child(nw) \rightarrow R^y > node \rightarrow child(sw) \rightarrow R$ ;

$node \rightarrow child(ne) \rightarrow R^x > node \rightarrow child(nw) \rightarrow R$ ;  $node \rightarrow child(se) \rightarrow R^x > node \rightarrow child(sw) \rightarrow R$ ;

- 3) 中间结点是 RQOP 树.

## 2 路网中基于 RQOP 树的移动对象索引

### 2.1 路网中基于 RQOP 树的移动对象索引结构

路网中基于 RQOP 树的移动对象索引由 RQOP 树、R 树森林和 HASH 结构组成, 如图 1 所示:

RQOP 树按照定义 1 和定义 2 对路段 MBR 进行排序, 得到路段空间分布情况, 进而依次对路网空间进行多次划分, 每次划分都使其子空间尽量包括相同数量 MBR. RQOP 树的叶子结点存储路段信息(路段标志、位置等)和一个指向与其对应 R 树的指针. R 树根结点存储了指向与其对应路段属于同一条道路(一条道路可以分为多条路段)且前后相邻路段的指针. 非叶子结点数据项结构

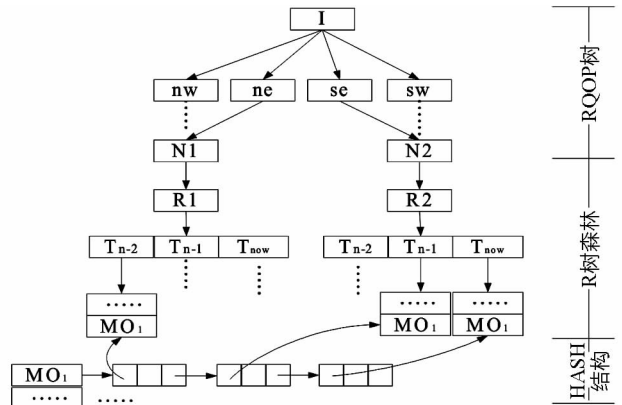


图 1 路网中基于 RQOP 树的移动对象索引结构

为  $(T_s, T_e, P_{child})$ ,  $P_{child}$  为指向孩子结点的指针,  $T_s$  和  $T_e$  表示该结点所管理的起始和终止时间. 叶子结点的数据项为  $(MOID, T_e, T_l, F, P_{last}, P_i)$ ,  $MOID$  是移动对象标志,  $T_e$  为此移动对象进入该路段的时刻,  $T_l$  为此移动对象离开该路段的时刻,  $F$  标记移动对象是否在原路段, 当  $F=0$  时,  $P_{last}$  指向移动对象上一个位置结点,  $P_i$  指向存储该路段上的所有移动对象的动态数组. HASH 结构由 HASH 表和动态链表构成, HASH 表中存储了 MOID, 每一个移动对象都指向一个单向动态链表. 链表中的结点按时间顺序存储着移动对象信息(时间、速度和位置等)和两个指针, 一个指针定位与该结点中位置信息对应的 R 树结点, 另一个指向该链表的下一个结点.

### 2.2 索引路网空间的 RQOP 树生成算法

RQOP 树的生成就是对路网空间中路段建立索引的过程. 对于给定  $n$  条路段, 首先, 求出包含这  $n$  条

路段的 MBR, 作为这棵树的根结点  $I$  ( $I$  是记录结点中路段 MBR 的数组). 然后, 把  $I$  里的 MBR 分别根据定义 1 和定义 2 进行排序, 对  $I$  分别建立基于  $x$  轴和  $y$  轴的序列, 然后把  $I$  划分到 4 个子空间中, 使得每个子空间中包含几乎相等个数的路段, 依次对每个子空间按照上述规则递归划分, 直到结点包含的路段数量小于 4 为止. 生成的 RQOP 树中, 非叶子结点与其兄弟结点包含路段数量之差最大为 2, 叶结点都分布在树的最下面两层, 且树的高度为  $\log_4 n$  ( $n$  表示路段总数), 每条路段只存储一次.

### 2.3 索引路网空间的 RQOP 树结点插入算法

RQOP 树结点插入就是把一路段添加到已有  $n$  条路段的索引中, 新增路段可以在原路网空间以外. 路段插入时, 从根结点的孩子结点开始, 将子结点 MBR 依次和新增路段 MBR 按定义 1 和定义 2 比较, 找到覆盖该路段 MBR 的子结点, 依次递归, 直到子结点为叶子结点时, 把路段增添到该叶子结点. 若 head 为已知一棵 RQOP 树的头结点的指针,  $R_{\text{new}}$  为待插入路段的 MBR,  $R$  为包含所有子结点空间的 MBR,  $R_{\text{new}}$  插入算法描述如下:

```

Node-Insertion( $n$ , head,  $R_{\text{new}}$ )           // RQOP 树结点插入算法
1: node = head;
2: if  $R_{\text{new}}^x > \text{node} \rightarrow \text{child}(\text{sw}) \rightarrow R$  并且  $\text{node} \rightarrow \text{child}(\text{sw}) \rightarrow R^y > R_{\text{new}}^y$  then
3: call Node-Insertion( $n$ ,  $\text{node} \rightarrow \text{child}(\text{sw})$ ,  $R_{\text{new}}$ );
4: else if  $R_{\text{new}}^x > \text{node} \rightarrow \text{child}(\text{nw}) \rightarrow R$  并且  $R_{\text{new}}^y > \text{node} \rightarrow \text{child}(\text{sw}) \rightarrow R$  then
5: call Node-Insertion( $n$ ,  $\text{node} \rightarrow \text{child}(\text{nw})$ ,  $R_{\text{new}}$ );
6: else if  $R_{\text{new}}^x > \text{node} \rightarrow \text{child}(\text{se}) \rightarrow R$  并且  $\text{node} \rightarrow \text{child}(\text{se}) \rightarrow R > R_{\text{new}}^y$  then
7: call Node-Insertion( $n$ ,  $\text{node} \rightarrow \text{child}(\text{se})$ ,  $R_{\text{new}}$ );
8: else call Node-Insertion( $n$ ,  $\text{node} \rightarrow \text{child}(\text{ne})$ ,  $R_{\text{new}}$ );
9:  $\text{node} \rightarrow R = \text{MBR containing } R_{\text{new}} \text{ and } \text{node} \rightarrow R$ ;           //修改结点 node 的 MBR
10:  $n = n + 1$ ;           //修改路段条数

```

## 3 性能分析

为了验证基于 RQOP 树的移动对象索引结构性能, 将其与动态组合索引结构性能进行实验比较, 并对其更新性能和查询性能进行分析. 实验环境为: Intel 处理器 2.8 GHz, 2.00 GB 内存, Windows xp 操作系统. 实验数据为 Brinkhoff 基于路网的移动对象数据产生器<sup>[8]</sup>生成的仿真数据集. 通过产生的 50~2 000 个仿真移动对象, 分别对两种索引结构的道路查询、轨迹查询和窗口查询性能比较, 结果分别如图 2、图 3、图 4 所示.

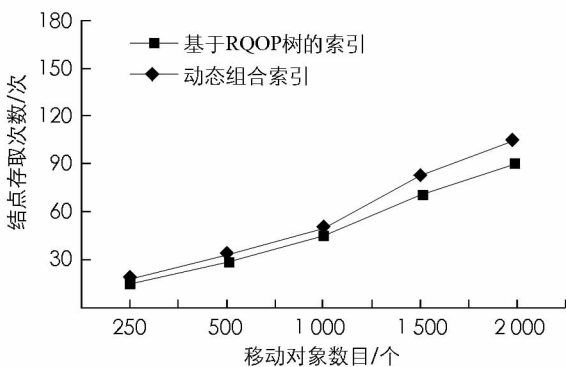


图 2 道路查询性能比较

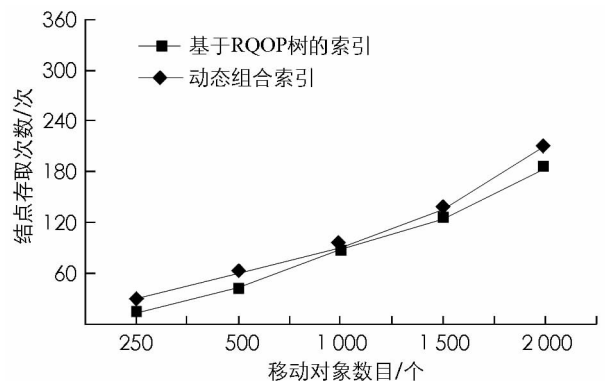


图 3 轨迹查询性能比较

由于 RQOP 树在生成和结点插入时要进行排序, 而 PMR QUAD 树只是进行简单的结点分裂与合并, 所以 RQOP 树更新代价相对增加, 但由于路网更新的操作很少进行, 对实际应用影响不大. 同时 RQOP 树可以在原路网空间外增加路段而不用重构整个索引, 提高了索引结构的可调整性且路段不会重复存储.

图 2、图 3、图 4 分别为两种索引在道路查询、轨迹查询、窗口查询中的性能比较结果。道路查询和窗口查询时都要先对 PMR QUAD 树基于自顶向下的遍历。PMR QUAD 树基于空间对路网分割,往往由于道路分布不均匀造成树的深度不平衡,增大路段查询代价。RQOP 树按路段分布对路网空间进行划分,尽量降低了树的高度,且路段没有重复存储,所有叶结点都位于树的最下面两层,从而降低了路段查询的代价,这就使得基于 RQOP 树的索引提高了道路查询、轨迹查询和窗口查询的效率。

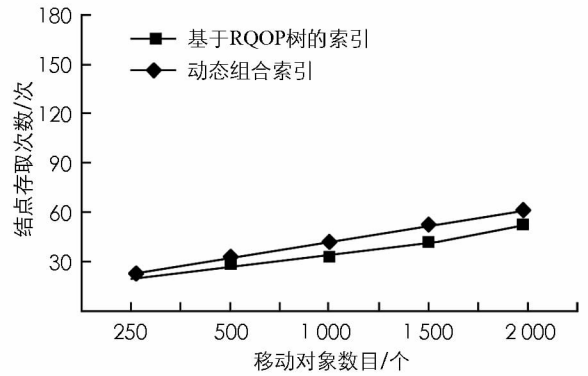


图 4 窗口查询性能比较

## 4 结束语

使用 RQOP 树建立路网索引,改进了基于路网的动态组合索引结构。RQOP 树对路网空间的分割是按照空间所包含路段的分布情况进行的,使树的高度尽可能的低,避免了路段重复存储,提高了索引的可调整性,从而提高了查询性能,下一步将研究该索引结构中移动对象未来位置的预测。

### 参考文献:

- [1] 肖 晖,李清泉. 移动对象数据库索引研究综述 [J]. 计算机应用, 2010, 30(4): 1064—1067.
- [2] 宋广军,郝忠孝,王丽杰. 一种基于受限网络的移动对象索引 [J]. 计算机科学, 2009, 36(12): 138—150.
- [3] GÜTING R H, BEHR T, ALMEIDA V T, et al. Modeling and Querying Moving Objects in Networks [J]. The VLDB-Journal, 2006, 15(2): 165—190.
- [4] CHEN Ji-dong, HU Zhi-zhi, MENG Xiao-feng, et al. Indexing the Past, Present and Future Positions of Moving Objects in Urban Traffic Networks [J]. Journal of Computer Research and Development, 2007, 44(6): 1008—1014.
- [5] FRENTZOS E. Indexing Objects Moving on Fixed Networks [C] // SSTD 2003: Proceedings of the 8th International Symposium on Advances in Spatial and Temporal Databases. Berlin: Springer-Verlag, 2003: 289—305.
- [6] 刘润涛,郝忠孝. R 树和四叉树的空间索引结构: RQOP 树 [J]. 哈尔滨工业大学学报, 2010, 42(2): 323—327.
- [7] 曾 倩,金 敏. 基于道路分布的移动对象动态组合索引方法 [J]. 计算机应用, 2008, 28(12): 3251—3253.
- [8] BRINKHOFF T. Generating Network-based Moving Objects [C]//SSDBM 00: Proceeding of the 12th International Conference on Scientific and Statistical Database Management. Washington: IEEE Computer Society, 2000: 253—255.

# Indexing of Moving Objects Based on RQOP Tree in Networks

ZHANG Yan-man, YU Jian-qiao

College of Computer and Information Science, Southwest University, Chongqing 400715, China

**Abstract:** Location-based service requires an efficient indexing method to manage moving objects. In PMR QUAD tree, the imbalance, the repeated storage and the poor adjustability of the indexing structure require improvement. This paper uses RQOP tree to divide the networks, according to the distribution of roads. It can make the height of the tree as low as possible so as to improve the dynamic compound indexing structure. The experiment shows that the query speed of the improved index structure is increased.

**Key words:** RQOP tree; networks; moving objects; index structure