DOI:10.13203/j.whugis2003.03.010

第28卷 第3期 武汉大学学报·信息科学版 Vol.28 No.3
2003年6月 Geomatics and Information Science of Wuhan University June 2003

文章编号:1671-8860(2003)03-0306-05 文献标识码:A

大型 GIS 空间数据库的有效索引结构 QR-树*

郭 菁[1] 郭 薇[2] 胡志勇[2]

(1 武汉大学动力与机械学院,武汉市珞珈山,430072)
(2 武汉大学计算机学院,武汉市珞喻路129号,430079)

摘 要:在分析R-树索引问题的基础上,提出了一种面向大型GIS空间数据库的QR-树索引新方法。

关键词:空间索引;QR-树;GIS
中图法分类号:TP311;P208

空间数据一般可用多属性的一条记录来描述。但传统的数据库系统不能有效地支持空间数据处理,因为:①空间数据的数据量大,结构和关系复杂;②空间目标不规则,目标之间的关系复杂(相交关系、相邻关系和包含关系),空间操作与计算比传统的关系运算代价高;③不能像传统数据库那样用某种命令对空间目标进行分类、排序。因此,空间数据处理必须采用特定的索引技术。

1 R-树的空间索引结构分析

1.1 R-树的数据结构

R-树[1]是基于最小包围矩形MBR(最小的包围矩形)的索引方法。它是B-树在多维空间的扩展,是一种平衡的树结构(所有的叶结点出现在同一层上),且保证空间利用率至少在50%以上。其数据结构如下:

叶子结点:(数,水平,<OI,MBR>,<OI,MBR>,…,<OI,MBR>)

非叶结点:(数,水平,<CP,MBR>,<CP,MBR>,…,<CP,MBR>)

R-树的叶子结点存储了一组结构为(OI,MBR)的数据项,其中OI为空间目标标识,MBR为该目标在 k 维空间中的最小包围矩形;非叶结点则存储了一组结构为(CP,MBR)的索引项,其中CP为指向子树根结点的指针,MBR代表其子树索引空间,为包围其子树根结点中所有索引项或数据项的MBR的最小包围矩形。"数"指示结点中用到的索引项或数据项个数(即该结点的"孩子"个数),"水平"指示该结点在树中的层数(0表示叶结点)。

插入一个空间目标,从R-树的根结点开始,找到使索引空间(MBR)的"面积"增量最小、增量相同情况下"面积"最小的索引项,再对该索引项所对应的子树重复以上的查找操作直至叶结点。如果叶结点不"满",则直接插入该目标,然后向上依次调整其父结点对应索引项的MBR,直至对根结点实施完调整操作;否则需对该叶结点进行分裂(切分)操作,产生一个新叶结点,然后插入到其父结点。如父结点又"满",还需对其进行分裂操作。

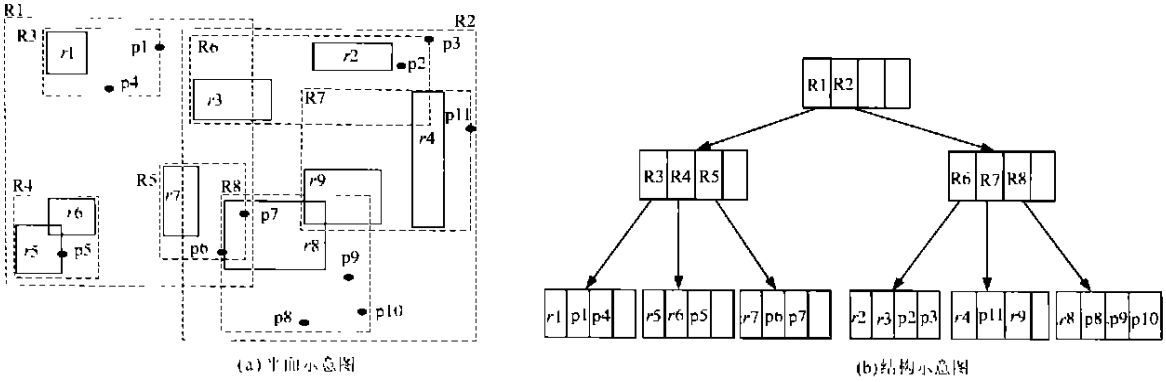
如要查询所有与一指定区域相交的空间目标,则需从根结点开始,遍历所有MBR与查询区域相交的索引项所指的子树直至叶结点。如果叶结点数据项中的MBR与查询区域相交,则根据OI找到对应的空间目标,并计算其是否相交。

从树中删除一个空间目标,首先需找到存放该目标的叶结点,如果删除该目标后叶结点"下溢",则需重新插入该叶结点中剩余的所有数据项,然后释放此叶结点的存储空间。同样,如果叶结点的删除导致其父结点的"下溢",还需删除其父结点,然后插入剩余的所有索引项于树中的同一层。图1举例说明了R-树的结构。

1.2 R-树的索引结构存在的问题

R-树的查找效率主要取决于R-树的深度及子树索引空间的重叠程度。R-树的深度越低,查找过程中访问的结点数就越少;子树索引空间的

* 收稿日期:2003-03-20
项目来源:国家自然科学基金资助项目(60173045/F020204)。



(a) 平面示意图　　(b) 结构示意图

图 1　R 树的结构

Fig. 1　Structure of R-tree

重叠越少, 查找过程中遍历的子树就越少, 访问的结点数也就越少。

由于 R-树的深度只与索引空间目标的数量有关, 所以, R-树构造的关键就是尽量减少子树索引空间的重叠。在 R-树的构造算法中, Guttman 将索引空间的"面积"作为惟一的度量, 将空间目标总是插入使索引空间"面积"增量最小的子树中, 以期尽量缩小索引空间的覆盖"面积", 减少索引空间的重叠程度及缩短查找路径。但是, 由于构造算法的动态性, R-树不可避免地会产生大量的索引空间重叠和"死空间(不包含空间目标的索引空间)", 从而导致效率下降。

许多索引构成, 如 R^*-树[2]、Ployhedral 树(Jagadish, 1990)和 X-树[3], 被设计在中间的结点中, 减少重叠的区域, 多样的搜寻路径仍然是 R-树的一个争议。R^+-树的非叶结点不允许索引空间重叠, 但空间目标需要重复存储, 这会浪费存储空间, 增加树的深度, 导致插入、删除算法的复杂化。R^*-树在构造算法中不仅考虑了索引空间的覆盖"面积", 而且考虑了索引空间的重叠, 对结点的分裂算法进行了改进, 并采用了"强制重新插入"的方法, 使树的结构得到了优化, 但仍然不能有效地降低索引空间的重叠程度, 尤其是在数据量较大、空间维数增加时。Berchtold 等作了详细的试验比较, 发现索引空间的重叠随索引空间维数的增加而剧增, 从而导致 R^*-树的性能下降。

综上所述, 基于 R-树的主要问题是索引空间的重叠, 且重叠程度随数据量或空间维数的增加而剧增。

2　QR-树

当空间目标的数量增加, 重叠在 R-树的中间结点快速地增加。因此, 首先利用四叉树(quadtree)将空间划分成一些子空间, 然后在各子空间内使用许多 R-树索引, 改良空间目标的重叠。这种结构称为 QR-树。

2.1　描述

实际上, QR-树可以看作是一棵特殊的四叉树($2k$ 叉树, k 为空间维数), 这棵四叉树的每个结点均指向一棵与其对应索引空间相关联的 R-树。如图 2, 用深度为 2 的四叉树将整个索引空间(IS0)划分成 4 个子索引空间(IS1, IS2, IS3, IS4), 四叉树的每个结点均与一索引子空间(ISi)和一棵 R-树(RTi)相关联。另外, QR-树也可以看作是一群 R-树的集合, 这些 R-树分别索引不同空间的目标实体。两种特殊情况是: ①当四叉树的深度为 1 时, QR-树即为 R-树; ②当所有 R-树的深度均为 1 或 0 时, QR-树即为空树。

在构造 QR-树时, 将空间目标插入到哪棵 R-树, 遵循如下规则: 对于空间目标 r, 设其 MBR 为 b, 包围 b 的最小索引空间为 ISi, 则 r 应插入到与 ISi 相对应的 R-树 RTi。如 $r2$ 插入与 IS0 相对应的 RT0, $r1$ 插入与 IS1 相对应的 RT1。

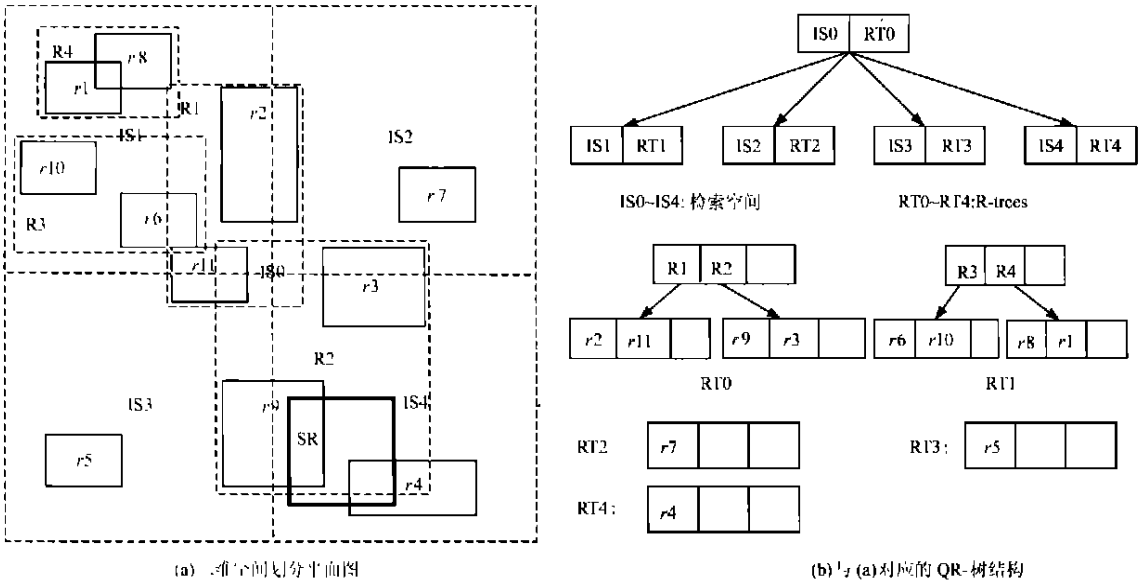
QR-树的数据结构也是 R-树和四叉树的数据结构的一个组合。R-树结点的结构如前所述; 四叉树采用线性的存储结构, 其结点结构可描述为〈MBR, PR〉, MBR 存储对应的 R-树的索引空间, PR 为与 MBR 关联的 R-树。

2.2　算法

设 R-树的插入、删除、查找算法分别为:

R-INSERT(R, 或): R 为 R-树的根结点, "或"为要插入的空间目标。该算法将"或"插入到 R-树的相应叶结点。

R-DELETE(R, 或): R 为 R-树的根结点, "或"为要删除的空间目标。该算法将"或"从 R-树的



(a) 二维空间划分平面图 (b) 与(a)对应的 QR-树结构

图 2 QR-树结构示意图

Fig. 2 Structure of QR-tree

相应叶结点中删除。

R-SEARCH(R, W): R 为 R-树的根结点, W 为查找窗口。该算法递归地遍历 R-树并返回所有与 W 相交的空间目标。

又设求包围空间目标 ORi 的 MBR 的最小子索引空间的算法为 GET-SPACE(ORi), 求与子空间 ISi 相关联的 R-树的算法为 GET-RTREE(ISi), 求与四叉树结点 Qi 相对应的 R-树的算法为 GET-RTREE(Qi), 则 QR-树的插入、删除、查找算法如下。

插入算法 QR-INSERT(OR):

ISx = GET-SPACE(OR);

Rx = GET-RTREE(ISx);

R-INSERT(Rx, OR)。

删除算法 QR-DELETE(OR):

ISx= GET-SPACE(OR);

Rx= GET-RTREE(ISx);

R-DELETE(Rx, OR)。

查找算法 QR-SEARCH (Q, W)(Q 指向四叉树结点):

Rx = GET-RTREE(Q);

IF MBR(Rx) OVERLAP W

THEN R-SEARCH(Rx, W);

FOR Qx IS SON (Q) DO

QR-SEARCH(Qx, W)。

从上面的算法描述可知, QR-树的插入、删除算法除了要先求出空间目标对应的 R-树外, 其余与 R-树完全相同。查找算法则相对复杂, 给定查找矩形区域 SR, 查找所有与 SR 重叠的空间目标, 必须对所有与 SR 相交的子空间所关联的 R-树进行查找操作, 需要遍历所有的四叉树结点, 在多棵"小"R-树中进行查找操作。

3 QR-树的性能分析

QR-树的存储空间开销与其四叉树的深度成正比, 比一般的 R-树大。QR-树中的一些 R-树可能包含很少的数据矩形。如在图 2 中, RT2、RT3 和 RT4 都只包含一个数据长方形, 这就浪费了一些存储空间。如此, QR-树的存储开销比 R-树更大, QR-树的代价通常比 R-树高。但随着四叉树的深度增加, QR-树在存储开销方面的优势越来越小。

在 R-树中, 搜寻整个的数据空间, 而在 QR-树中, 通常只搜寻整个空间的一部分。如在图 2 中, 寻找所有与 SR 相交的数据矩形, 仅仅只涉及 RT0 和 RT4 的索引空间。如此, QR-树的搜寻表现通常比 R-树好。QR-树的插入和删除只涉及部分数据空间的"矮小"R-树, 因此 QR-树的插入和删除性能优于 R-树。

为了评估 QR-树的性能, 笔者编制了 R-树和 QR-树的试验程序, 并通过均匀分布的随机数据和实际地图数据, 对 R-树及四叉树深度不同的 QR-树分别从空间开销和访问磁盘页次数两个方面进行了性能测试(试验中, 1 页= 1 024bytes)。对于二维随机数据, 测试结果如图 3 所示(Level-i 表示深度为 i 的 QR-树)。

对于如图 4 所示的二维实际地图数据, 其测试结果如图 5 所示。从试验结果可以发现:

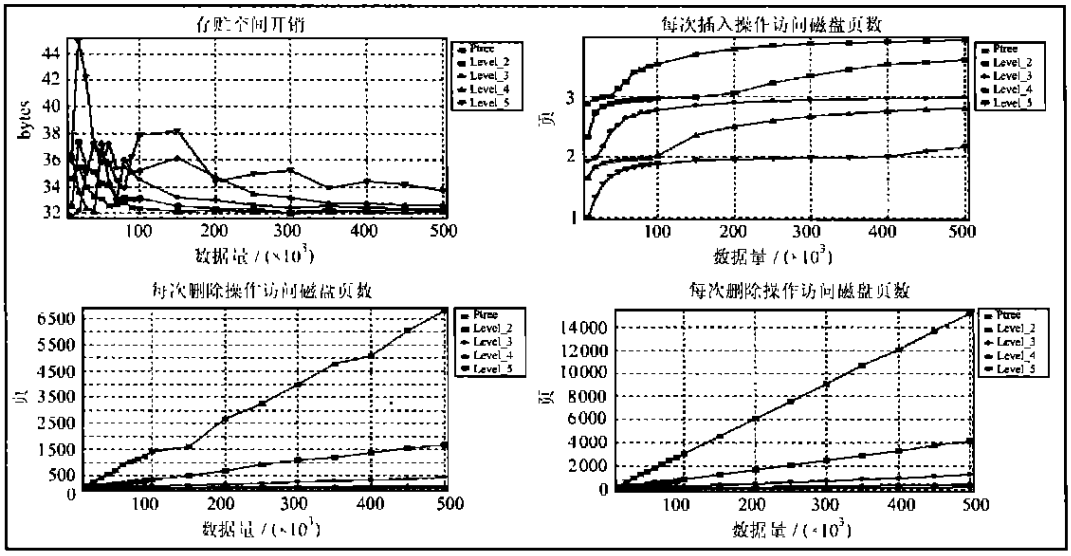


图 3　二维随机数据测试结果曲线

Fig. 3　Random Two-Dimensional Data

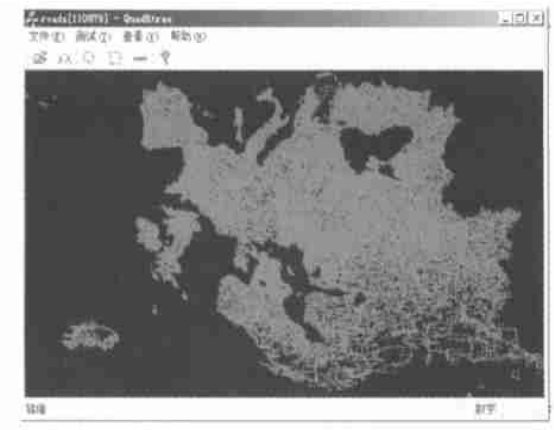


图 4　欧洲道路分布(数据矩形数为 110 979 个)

Fig. 4　Roads of Europe(Number of MBR: 110 979)

1) QR-树在插入、删除和查找效率(尤其是查找效率)方面优于 R-树。QR-树的表现与四叉树的深度成正比。四叉树的深度越大, 性能越好。

2)在大部分情形下, R-树的存储开销比 QR-树小(尤其是那些四叉树深度比较高的情况), QR-树的存储代价与四叉树的深度成正比。

3)尽管 QR-树总是要求较多的存储开销, 但只要选择适当的四叉树深度, 就能够换取更高的查找性能, 且索引目标数越多, QR-树的整体优势越明显。这一点对于大型的 GIS 空间数据库系统来说, 是很有实用价值的。

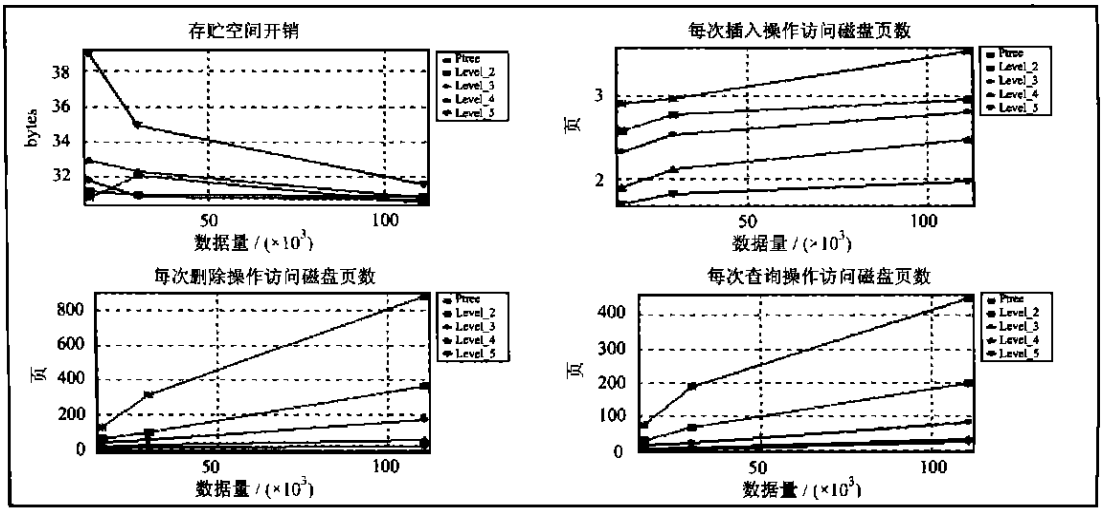


图 5　二维实际地图数据测试结果

Fig. 5　Real Two-Dimensional Map Data

4 结 语

四叉树和 R-树在 GIS 系统软件及空间数据库系统中(如 Arc/Info8、Informix、Oracle 等)均有应用,但当索引数据量增加时,它们都不可避免地出现数据存贮量及索引空间重叠区域增大,从而导致查询效率降低的问题。为此,本文提出了一种基于 R-树与四叉树空间层次划分的空间索引结构 QR-树。试验证明,与 R\|树相比,QR-树可以在略大的空间开销的前提下,换取更高的查找性能,且索引目标数越多,QR-树的整体性能优势越明显。

参 考 文 献

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

2 Beckmann N, Kriegel H P, Schneider R, et al. The R*-tree: An Efficient and Robust Access Method for Points and Rectangles. ACM SIGMOD, Atlantic, USA, 1990

3 Berchtold S, Keim D A, Kriegel H P. The X-tree: An Index Structure for High-Dimensional Data. The 22nd Int. Conf. on VLDB, Mumbai(Bombay), India, 1996

4 胡志勇,郭 薇. 空间数据库索引研究. 计算机研究与发展, 2000(增刊): 164~170

5 Sellis T, Roussopoulos N, Faloutsos C. The R+-tree: A Dynamic Index for Multi-dimensional Objects. The 13th Int. Conf. on Very Large Databases, Brighton, U.K., 1987

第一作者简介:郭菁,讲师,博士生。主要研究方向为数据库系统、空间数据索引。

E-mail: ylguo@wuhee.edu.cn.

QR-tree: An Efficient Spatial Indexing Structure for GIS with Very Large Spatial Database

GUO Jing[1] *GUO Wei*[2] *HU Zhiyong*[2]

(1 School of Power and Mechanical Engineering, Wuhan University, Luojia Hill, Wuhan, China, 430072)

(2 School of Computer, Wuhan University, 129 Luoyu Road, Wuhan, China, 430079)

Abstract: Spatial indexing techniques can efficiently improve the storage and query efficiency. IN the large spatial database of GIS, the traditional spatial indexing structures, the R-tree has the problem that with the growing dimensionality and data number, the searching function will decrease greatly. On the basis of the analysis of R-tree, this paper puts forward a new spatial indexing structure for GIS with very large spatial database-the QR-tree.

Key words: spatial index; QR-tree; GIS

About the first author: GUO Jing, lecturer, Ph. D candidate. She majors in database manage syatem, spatial index.

E-mail: ylguo@wuhee.edu.cn.

(责任编辑: 平子)