R 树家族的演变和发展

张明波[1),2)] 陆 锋[1)] 申排伟[1)] 程昌秀[1)]

[1)](中国科学院地理科学与资源研究所资源与环境信息系统国家重点实验室 北京 100101)

[2)](山东理工大学建工学院 淄博 255049)

摘 要 近年来,针对空间数据库索引的研究引起了人们越来越多的兴趣和关注.为了快速、有效地处理存储于空间数据库中的海量空间数据,专家学者提出了大量的基于磁盘的空间索引方法.其中,1984 年由 Guttman 提出的 R 树是目前最流行的动态空间索引结构,广泛应用于原型研究和商业应用中.其后,人们在此基础上针对不同空间运算提出了不同改进.经过 20 年的发展,不断产生的 R 树变体逐渐形成了一个枝繁叶茂的空间索引 R 树家族.该文回顾了 R 树及其各种主要变体;描述了基于 R 树的各种批量操作、空间查询处理算法、查询代价模型及查询优化过程;介绍了基于 R 树的并行处理、并发控制与锁定策略等方面的进展;并且分析了 R 树的未来研究方向.

关键词 空间数据库;空间索引;R 树;空间查询;代价模型

中图法分类号 TP311

The Evolvement and Progress of R-Tree Family

ZHANG Ming-Bo[1),2)] LU Feng[1)] SHEN Pai-Wei[1)] CHENG Chang-Xiu[1)]

[1)](*State Key Laboratory of Resources and Environmental Information System, Institute of Geographical Sciences and Natural Resources Research, Chinese Academy of Sciences, Beijing* 100101)

[2)](*School of Architecture and Engineering, Shandong University of Technology, Zibo* 255049)

Abstract In recent years, the research on spatial indexing is arousing more and more interests and attentions. A number of spatial indexing structures based on secondary memory have been proposed for handling the massive spatial data stored in the spatial database rapidly and efficiently. R-tree, proposed by Guttman in 1984, is the most popular dynamic spatial access method and has been widely used in many prototype researches and commercial applications. During the last two decades, more and more R-tree variations have come into being a prosperous R-tree family. In this paper, authors recall the R-tree and its main variations, study the bulk operation, spatial query processing, cost model and query optimizing technique based on R-tree, present the progress about the parallelism, concurrency control and locking strategy based on R-tree, and point out the directions for future work.

Keywords spatial database; spatial index; R-tree; spatial query; cost model

1 引 言

空间数据库是存储和管理空间数据的数据库系统.为了快速、有效地访问海量空间数据,专家学者提出了大量的空间索引方法.从空间数据库的观点看,空间索引结构可以分为处理点对象的点存取方法(PAM)和处理空间扩展对象(包括点、线、面、体)

收稿日期:2004-07-28;修改稿收到日期:2005-01-04.本课题得到国家自然科学基金(40201043)和国家"八六三"高技术研究发展计划项目基金(2003AA135070)资助.**张明波**,男,1971 年生,博士研究生,讲师,研究方向为空间数据库引擎、空间索引.E-mail:zhangmb@lreis.ac.cn.**陆 锋**,男,1970 年生,博士,研究员,研究方向为 GIS 空间数据模型、地理网络表达与分析方法.**申排伟**,男,1977 年生,博士研究生,研究方向为空间数据库技术.**程昌秀**,女,1973 年生,博士,副研究员,研究方向为 GIS 空间数据模型.

的空间存取方法(SAM)[1]. PAM 包括 Grid 文件[2]、Buddy 树[3]、K-D-B 树[4]、hB 树[5]、LSD 树[6]等. SAM 采用空间对象的近似进行索引,如常用的最小边界矩形(MBR). 依据空间对象的不同组织方式,SAM 分为对象映射、对象分割/复制和对象界定三类[1,7~9]. 对象映射是把非点对象映射为高维/低维空间中的点,然后利用点/一维存取方法索引映射后的数据集,如常见的空间填充曲线方法将高维空间中的空间对象线性映射到一维空间,用空间排列码(如 Morton 码、Hilbert 码、Gray 码等)作为地址码建立索引[10,11]. 对象分割/复制是把与子空间相交的数据对象分割成几个子对象,分别存储在互不重叠的子空间中,在子空间中复制对象本身或其标识符,如 R^+ 树[12]、Cell 树[13,14]、线性四叉树[15,16]等. 对象界定又称区域重叠技术,它允许子空间的相互重叠,一个数据对象唯一对应于一个子空间,如 R 树[17]、R* 树[18]等.

R 树是一种采用对象界定技术的高度平衡树,是 B 树在 k 维空间上的自然扩展. 作为目前最流行的动态空间索引结构之一[1,8,9],R 树广泛应用于原型研究和商用空间数据库系统中. R 树最初由 Guttman 于 1984 年提出,其后,人们在此基础上针对不同的空间操作需求提出了各种改进方案. 经过 20 年的发展,不断产生的 R 树变体逐渐形成了一个枝繁叶茂的空间索引 R 树家族. 从其覆盖的广度和深度来看,多维空间索引 R 树就像一维线性索引 B 树一样,是无处不在的①.

本文回顾了国内外 20 年来空间索引 R 树研究的重要成果,讨论了 R 树及其主要变体的结构特点,描述了 R 树构造过程中的批量操作方法,分析了基于 R 树的空间查询处理过程及基于 R 树的查询代价模型及查询优化过程,介绍了 R 树并行处理与并发控制的研究进展;并结合作者在该领域的研究工作和实验过程对这些研究成果进行了分析与评述,进而对当前 R 树的研究热点问题进行了讨论和展望.

2 R 树及其主要变体

经过搜集与整理 20 年来国际计算机与 GIS 领域对 R 树的研究文献,我们得出如图 1 所示的 R 树家族进化图,描述了 R 树家族自 1984 年提出以来的演变和发展历程.

R 树[17]是一种层次数据结构,它是 B 树在 k 维空间上的自然扩展,因此和 B 树一样,R 树是一种高度平衡树,在叶结点中包含指向实际数据对象的指针,并能保证至少 50%的有效空间存储利用率.

R 树是一种完全动态的空间索引数据结构,插入、删除和查询可以同时进行,并且不需要周期性的索引重组. R 树由中间结点和叶结点组成,叶结点存储的是实际空间对象的最小边界矩形(MBR),而不是实际的空间对象. R 树的其它特性可见文献[17].

R 树允许兄弟结点之间的相互重叠. 因此对于精确匹配查询,R 树不能保证唯一的搜索路径. 图 2 是一个二维 2 层 R 树的实例,左边是空间数据对象,右边是对应 R 树,其中 $M=3$(M 是结点最大容量).

为了避免 R 树由于兄弟结点的重叠而产生的多路径查询问题,1987 年,Sellis 等设计了 R^+ 树[12],以提高其检索性能. R^+ 树采用对象分割技术,避免了兄弟结点的重叠,要求跨越子空间的对象必须分割成两个或多个 MBR,即一个特定的对象可能包含于多个结点之中. R^+ 树解决了 R 树点查询中的多路径搜索问题,但同时也带来了其它问题,如冗余存储增加了树的高度,降低了域查询的性能;在构造 R^+ 树的过程中,结点 MBR 的增大会引起向上和向下的分裂,导致一系列复杂的连锁更新操作;在不利的情况下可能会造成死锁.

1990 年,Beckmann 等设计了 R* 树[18],指出区域重叠技术并不意味着更坏的平均检索性能,而插入过程才是提高检索性能的关键阶段;并且通过大量对不同分布数据的实验研究,找到了一系列相互影响的决定检索性能的参数,提出了一系列结点分裂优化准则,设计了结点强制重插技术. 这些研究成果提高了 R 树的空间利用率,减少了结点分裂次数,使得目录矩形(某一路径所有矩形的最小边界矩形)更近似于正方形,从而极大地改善了树结构,显著提高了树的查询性能,但同时也增加了 CPU 计算代价.

1994 年,Kamel 和 Faloutsos 提出了 Hilbert R 树[19],以提高结点存储利用率,优化 R 树结构. 其主要思想是利用 Hilbert 分形曲线对 k 维空间数据进行一维线性排序,进而对树结点进行排序,借以获得面积、周长最小化的树结点. 此外,对于通过排序后得到的组织良好的兄弟结点集,实施类似于 B* 树的滞后分裂算法,从而获得较高的结点存储利用率.

① Manolopoulos Y. *et al*.. R-trees have grown everywhere. Technical Report. http://www.rtreeportal.org/. 2003

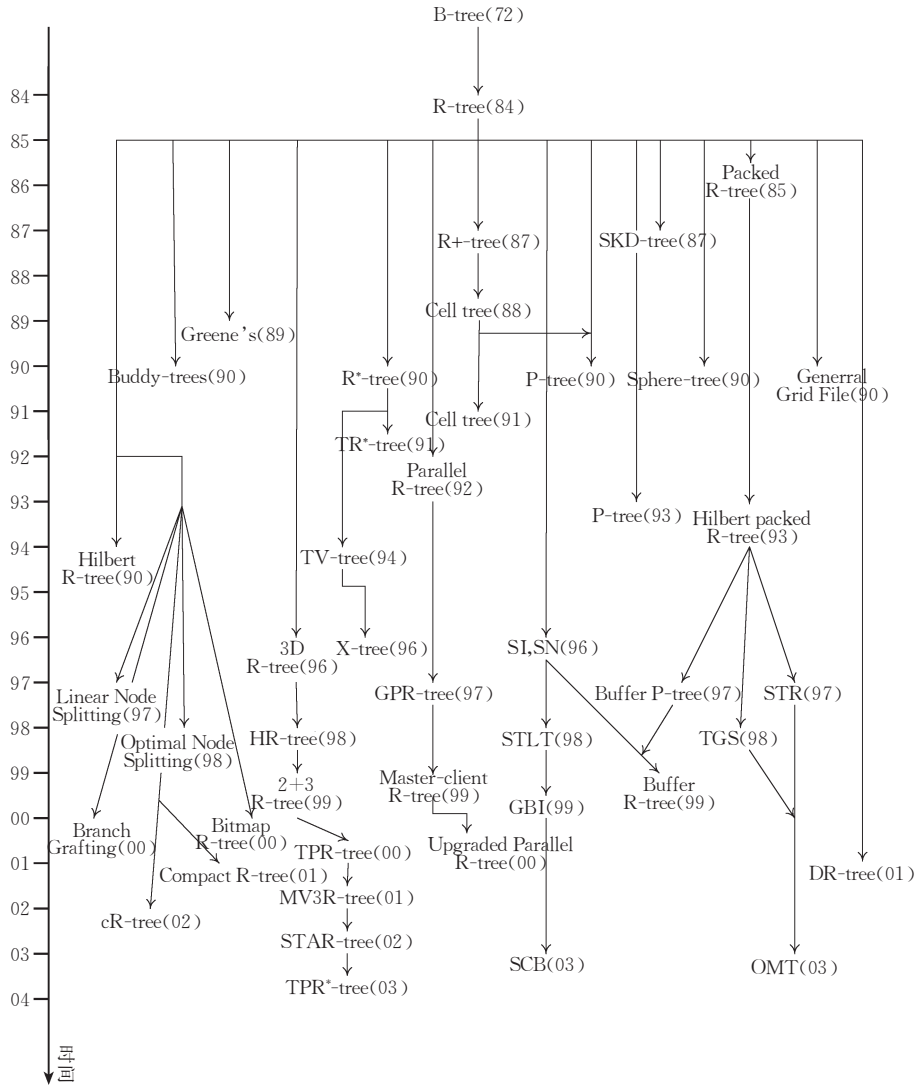

图 1　R 树家族演变和发展历史

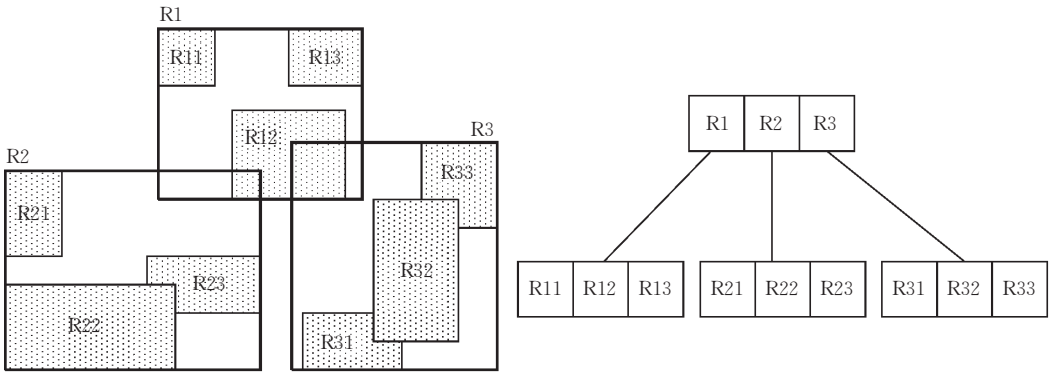

图 2　R 树示例

2001 年,Huang 等提出了 Compact R 树[20].由于其特殊的分裂算法,该变体可以达到几乎 100% 的存储利用率,并导致结点分裂的次数明显减少,构建开销低于 R 树,易于实现和维护.但其检索性能仅与 R 树相仿,不及 R^+ 树、R^* 树等变体优良.

2002 年,Brakatsoulas 等通过深入研究 R 树构建原理,提出了 cR 树[21],认为动态 R 树的创建本质上是一个典型的聚类问题,可以利用现有的成熟聚类算法来解决.他们采用通用的"k-means"聚类算法,把传统的两路分裂改进为由聚类技术支持的多路分裂.cR 树插入代价与 R 树相仿,而查询性能与 R^* 树相近,更适合于数据密集型环境且实现算法

简单,不需要强制重插等复杂技术,易于维护.

有关 R 树的数据结构和分裂算法方面的改进和变体还有还多,如在空间对象的近似表达方面,有利用最小边界球的球树(Sphere trees)[22],最小边界凸多边形的 CP 树[23],最小边界多边形的 Cell 树[24]、P 树[25];在结点分裂方面,文献[26～28]提出了不同的分裂算法;DR 树[29]是适合主存索引的变体;而位图 R 树[30]借鉴了位图索引的思想;其它的 R 树变体在文献[1,31]中有详细介绍.

3 R 树创建与维护的批量操作技术

传统的 R 树的构建是从空树开始,利用传统的插入算法逐个插入记录,直至生成整个 R 树,称为 OBO 方法(one-by-one). 由于要动态维护空间索引结构,该方法的插入代价非常高,特别对于海量空间数据而言,索引的创建过程将耗时巨大. 因此,面向应用需求,专家学者开始寻求高效的 R 树批量操作技术,在维持和提高查询性能的前提下,尽可能提高 R 树的静态加载和动态更新速度.

3.1 静态批量加载

R 树批量加载技术又称压缩技术. 在数据为已知且相对静态的情况下,对已知数据进行有效的预处理,提高数据加载速度,建立结构优化的 R 树,改善空间存储利用率,从而获得优良的检索性能,是压缩技术的初衷.

3.1.1 packed R 树

第一个基于 R 树的压缩算法由 Roussopoulos 等在 1985 年提出[32],用以建立能够达到 100%空间利用率的压缩 R 树. 其基本思想是根据空间对象的最小边界矩形的一个角点的 x 坐标或 y 坐标对空间对象进行排序,然后用这些有序的空间对象逐个压满树的叶结点,自下而上,一次一层,递归生成最终的压缩 R 树. 由算法可知,在压缩 R 树中的每一层中,除了最后一个结点可能不满外,其它所有结点都是满的,因此,可以获得几乎 100% 的空间利用率. 但是,该算法仅在点数据的点查询方面优于线性分裂或平方分裂的 R 树和 R* 树,而对于域查询和空间扩展数据(如矩形等)性能较差.

3.1.2 Hilbert packed R 树

Kamel 和 Faloutsos 在 1993 年提出了一种基于分形曲线构建静态 R 树的压缩算法[33],更确切地讲,是利用分形 Hilbert 曲线对已知空间数据对象进行更好的一维排序,以获得优良的压缩效果. 他们设计了不同的变体进行了大量的实验,最终"2D-C"(利用空间对象 MBR 中心点的 Hilbert 码进行排序)变体性能最优. 该变体在查询性能上不仅优于 Roussopoulos 等的 packed R 树,而且优于当时 R 树的所有动态版本,如 R^+ 树、R* 树等,且特别适合实际的、不规则分布的数据.

3.1.3 Bercken's Buffer R 树

与基于排序的批量加载方法不同,Bercken 在 1997 年提出的批量加载方法是一种基于缓冲的技术[34],也可以认为是基于种子树方法[35]的推广,它并不局限于 R 树,而是适用于所有基于树的空间索引结构. 该方法借鉴了 buffer 树[36]的思想,通过 Lazy buffer 技术,利用一个有效的临时数据结构,一次一层地建立整个索引结构,避免了数据排序预处理过程.

Bercken 通过性能分析得出该算法的 R 树插入代价达到了外排序的下界,但是并没有进行实验证明和性能比较.

3.1.4 STR 压缩算法

1997 年,Leutenegger 等提出了一种 STR(Sort-Tile-Recursive)压缩算法[37],可称为递归网格排序算法. 该算法易于实现,比前几种压缩算法的数据适用范围更广,且可以适应不同的缓冲区大小. 其基本思想如下:假设在 2 维空间中有 N 个矩形,M 是 R 树结点的最大容量,用矩形中心点的 x 坐标对数据矩形排序,用 $S=\sqrt{N/M}$ 个垂直切片切割数据空间,使得每个切片包含 S 个结点和 $S \cdot M$ 个矩形(最后一个切片会少于 $S \cdot M$ 个矩形);然后在每一个垂直切片中,用矩形中心点的 y 坐标对数据矩形排序,每 M 个矩形一组依次压入结点. 自下而上,一次一层,递归生成整个 R 树. 该算法可以很容易地推广到高维空间.

Leutenegger 等的实验结果表明:没有一种最优算法适合于所有的数据类型. 总体来讲,与以前提出的最好的算法(Hilbert packed R 树)相比,对于均匀分布的数据或者中等程度畸变的点数据和区域数据,该算法在点查询和区域查询上会减少 50%的磁盘获取,然而对于高度畸变的点数据和区域数据性能大致相同.

Leutenegger 等的另一贡献是在 STR 实现中加入了 LRU(最近最少使用算法)缓冲管理,并提出一个分析模型来预测缓冲管理下的磁盘存取次数,用以评价压缩算法的质量.

3.1.5 TGS 压缩算法

1998 年,Garcia 等提出了 TGS(Top-down Greedy-

Split)算法[38],称为自上而下地贪婪分裂算法,其特点为自上而下地递归构建 R 树. TGS 递归应用如下基本分裂过程:对于数据空间中的 N 个矩形,TGS 沿着某一坐标轴把所有数据垂直分割为两个子集. 该分割要求满足以下两个条件:(1)使得用户自定义代价函数 $f(r1,r2)$的值最小,其中 $r1$,$r2$ 分别是两个子集的 MBR;(2)每个子集有 $i \cdot S$ 个矩形,其中 S 是该层每个子树的最大矩形个数,i 小于该层结点的个数. 该过程递归应用于两个子集,直至生成整个 R 树.

作者实验结果表明,TGS 算法的检索性能要优于前述的 STR 和 Hilbert 算法,且特别适合于区域数据和不规则分布的数据集. 但是由于其自适应预处理过程需要多次对数据对象进行重新组织,导致数据加载时间远大于其它方法.

3.1.6　Arge's Buffered R 树

Arge 等提出了不同于 Bercken 的 buffer 技术[39,40]. 两者的共同之处在于都借鉴了 buffer 树[36]的思想和 Lazy buffer 技术,都充分利用了操作系统的可用主存和页面大小. 不同的是,Arge 等的方法不仅可以进行静态批量加载,而且可以进行动态批量更新,是一种在线处理的有效方法. 另外,与所有其它的批量操作方法相比,它能够支持并发的批量操作. 但是,在数据批量加载方面,该文仅仅进行了理论分析,并没有进行实验性能比较.

3.1.7　OMT 算法

OMT (Overlap Minimizing Top-down Bulk Loading)批量加载技术[41]可以看作 STR[37]方法的一个倒装,与 STR 不同的是,OMT 首先通过简单的计算获得目标 R 树的拓扑关系;然后自上而下地分割空间,聚簇数据,创建整个 R 树. 其主要思想是尽可能减少上层结点间的重叠区域,从而减少查询遍历路径,提高查询性能. 但是,该文没有提供针对 OMT 的理论分析和实验性能比较. 直观看来,该技术构建 R 树的代价相当大,而且由于没有提出解决上层结点间重叠区域问题的更为有效的方法,与 STR 算法相比,查询性能的提高也值得怀疑.

3.2　动态批量插入

对于动态 R 树来讲,利用 OBO 生成的 R 树结构应该是最好的,检索性能也应该是最高的,因为它在 R 树构建过程中利用结点分裂算法实现了树结构的局部重组. 因此,动态批量插入要解决的问题是如何把新数据集批量插入到现有的空间索引结构中,目的在于提高插入效率,而不是提高查询性能. R 树动态批量插入技术的研究起步较晚.

3.2.1　SI 和 SN 算法

1996 年,Kamel 等基于新数据通常总是以数据集的形式出现的事实,首次提出了动态 R 树的批量加载问题[42]. 作者设计了 SI(排序插入)和 SN(排序结点插入)两种插入算法,SI 首先利用空间对象 MBR 中心点的 Hilbert 码对新数据集进行一维排序,然后借助于结点缓冲区的帮助,依次把数据对象逐个插入到现有的 R 树中. 而 SN 首先利用空间对象 MBR 中心点的 Hilbert 码对新数据集进行一维排序,然后将其依次分组构建成为新的 70%满的叶结点,最后把构建好的叶结点逐个插入到现有 R 树中的叶子层. 实验结果表明,SI 和 SN 算法在维持 R 树索引查询性能基本不变的情况下,极大地减小了插入代价,其中以 SN 更为显著. 然而,SN 更适合大数据集的批量加载,因为对于新的小数据集,经过多次的插入之后,R 树的查询性能将显著下降. 因此,SI 和 SN 方法应该根据新数据集的大小交替使用.

3.2.2　STLT 技术

1998 年,Chen 等针对新的不规则分布的数据集的动态插入问题进行了研究,提出了 STLT (Small-Tree-Large-Tree)技术[43],后来将应用该技术的 R 树称为归并 R 树(Merging R-tree)[44]. STLT 首先把新数据集独立构建为一颗小 R 树,然后在原有的大 R 树中为其定位,最后把小 R 树插入到大 R 树中. 实验结果表明,STLT 技术特别适合于不规则分布数据和大 R 树与小 R 树数据量比率较大的情况,此时的插入代价远小于 OBO 的传统插入技术,且查询性能也在多数情况下优于用 OBO 技术形成的 R 树. 但 STLT 存在数据分布的局限.

3.2.3　GBI 技术

STLT 的作者为了摆脱仅适合不规则分布数据集的局限性,对该技术进行了改进,提出一种通用动态 R 树批量插入策略——GBI(Generalized Bulk-Insertion)[45],使其能适合任意分布的数据集. GBI 首先利用聚类算法将新数据集分割为一系列的组群和离群值,然后为每一个组群构建一颗小 R 树,并为其在原有大 R 树中定位插入位置;最后批量插入所有的小 R 树和离群值. 实验结果表明,GBI 适合各种分布的数据集,尤其是随机分布数据和包含少数自然组群的实际数据集,其插入代价要比已有批量插入技术低 200%,且性能优良.

3.2.4　Arge's Buffered R 树

该 R 树变体已在静态批量插入部分做过介绍.

该变体不仅适合做静态批量加载,还可以进行动态批量插入、删除和查询,而且支持并发[39,40].但是,从概念上来讲,其动态批量插入算法等同于重复插入算法;而且在批量更新方面,作者仅仅与 SN 方法进行了性能比较,认为该 R 树变体具有更快的更新时间和更好的查询性能.

3.2.5 SCB 技术

2003 年,Lee 等充分考虑了已有的目标 R 树结构对于待输入数据集的空间聚簇分类的影响,提出了一种 SCB(Bulk Insertion by Seeded Clustering)方法[46].SCB 利用种子聚簇(Seeded Clustering)技术,减少目标 R 树和输入 R 树之间的结点重叠面积,优化树结构,提高查询性能.该方法首先通过复制目标 R 树的最上面的 k 层索引结构以构建种子树(seed tree),然后利用种子树来指导输入数据集的聚簇分类,再为每一个组群构建相应的小 R 树,最后批量插入所有的新构建的小 R 树和离群值.另外,作者还提出了一种再压缩(repacking)技术,该技术在插入过程中用来最小化目标 R 树和输入 R 树的重叠面积.实验结果表明,SCB 方法在插入和查询代价上都优于传统的 OBO 方法和 GBI 方法,而且其查询性能不会随着不断的数据插入而恶化[45].SCB 是唯一一种在查询性能上优于传统 OBO 方法的技术,之前的 STLT 方法虽也声称如此,但它仅针对不规则分布的数据集.

4 空间查询处理

由于空间对象和空间操作的复杂性,使得空间数据库中的空间操作既是 I/O 密集型又是 CPU 密集型的.因此,充分利用空间索引有效地进行空间检索,是空间数据库的一项关键技术[47].空间索引 R 树流行的一个重要原因就是能够有效支持多种空间操作.近年来,许多专家学者针对基于 R 树的空间查询算法进行了大量的研究.

空间查询主要包括精确匹配查询、点查询、窗口查询、域查询、拓扑查询、方位查询、最近邻查询和空间连接等.空间查询处理通常采用如图 3 所示的两步查询处理过程[48,49].基于 R 树的空间查询算法基本集中在过滤步骤.

由于点可以看作退化的矩形,而窗口查询也可以看作域查询的一个特例,因此点查询和窗口查询都可以归入域查询.基于 R 树的域查询算法在文献[17]中有详细论述,且少有改进,仅在文献[50]中提出了同时处理多个域查询以提高性能的方法;拓扑查询和方位查询分别在文献[51]和文献[52]中有所论述;大部分基于 R 树的空间查询处理研究集中在空间连接和最近邻查询.

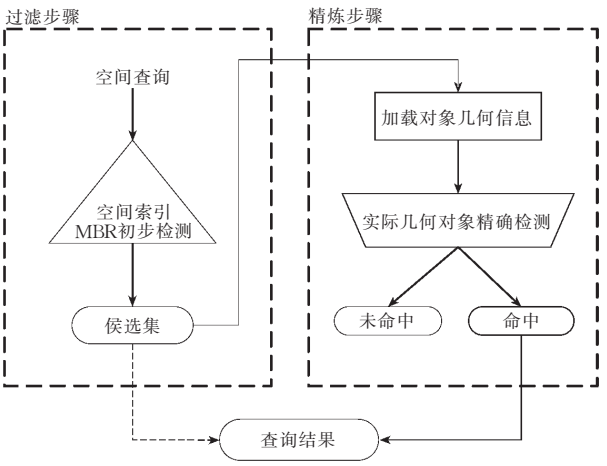

图 3 两步空间查询处理流程

4.1 空间连接

在空间数据库中,空间连接是非常重要的运算符.当两个关系(表)R 和 S 基于一个空间谓词 θ 进行连接时,称该连接为空间连接[47].空间连接主要用来根据空间属性合并两个或多个数据集的空间对象.GIS 中的重要的运算符'地图叠加'(map overlay)是空间连接的一个变体.空间谓词 θ 主要有 intersect(相交)、contains(包含)、is_enclosed_by(被包围)、distance(距离)、adjacent(邻接)、meets(接触)、overlap(交叠)等.

空间连接查询与域查询和最近邻查询等单向扫描查询不同,它是双向扫描(两路空间连接)或多向扫描(多路空间连接)的.基于 R 树的空间连接算法研究通常都假设参与连接的关系(表)事先已建立 R 树索引,且往往针对两步空间查询处理中的过滤步骤.

4.1.1 基于深度优先遍历的 R 树空间连接算法

1993 年,Brinkhoff 等提出了基于 R 树的空间连接算法[53].该算法是一种深度优先遍历过程,其出发点是非叶结点的矩形能够容纳所有子结点中矩形的 MBR.因此,若两个目录矩形不相交,则其子树中所有数据矩形必然也不相交;如果目录矩形相交,则其子树中某些数据矩形有可能相交.

为了减少 CPU 计算代价,作者提出了限制搜索空间及其空间排序和平面搜索的优化措施;为了减少 I/O 代价,作者提出了基于空间临近性的局部优化策略,包括锁定局部平面搜索排序和局部 Z 排

序,用以控制结点读入内存缓冲区的次序.

1995 年,Martynov 对该算法进行了改进,用基于网格的启发式优化代替平面搜索,提高了空间连接的执行效率[54].

4.1.2　基于广度优先遍历的 R 树空间连接算法

针对深度优先遍历算法只能进行局部优化的缺陷,Huang 等在 1997 年提出了一种基于全局优化的广度优先遍历算法[55].该方法以广度优先的方式同时遍历两棵 R 树,同时逐层处理连接计算.在每一层,创建中间连接索引并实施全局优化策略,如排序、内存管理和缓冲区管理等,用以提高下一层的空间连接计算性能.该算法与深度优先遍历算法的实验比较表明,在正确选择全局优化选项的前提下,该算法性能提高了 50%.

基于 R 树的空间连接仍是一个活跃的研究领域,其它研究成果包括多路空间连接[56~60]、增量距离连接和半连接[61,62]以及最近邻对查询[61,63,64]等.

4.2　最近邻查询

基于 R 树的最近邻查询算法最早由 Roussopoulos 等于 1995 年提出[65].作者使用分枝界定的 R 树遍历算法,提出了两个度量标准用于 R 树的排序和剪枝,一个是 MINDIST,称为乐观距离;另一个是 MINMAXDIST,称为悲观距离;分别作为查询点到树结点 MBR 中最近邻对象的距离下界和上界.基于这两个度量标准,作者提出了三个启发式规则来过滤不包含最近邻居的结点,从而减少结点访问个数,减少磁盘 I/O,进而有效地提高查询性能.1998 年,Cheung 等进一步改进该算法,移除前两个既不能增强剪枝效果又具有高计算复杂度的剪枝规则,减少了 CPU 计算代价[66].

其它基于 R 树的最近邻算法有增量最近邻查询(INN)[67]、逆最近邻查询[68,69]、条件最近邻查询[70]等.

5　代价模型与查询优化

如前所述,R 树变体众多,因此如何评价或预测它们的性能就显得至关重要.除了通过实验来比较 R 树及其变体的性能之外,还需要从理论上研究其代价模型,包括插入、删除和查询代价模型.代价模型的重要性体现在:(1)能够更好地理解在不同类型和大小的输入数据集下的空间索引行为;(2)能够对各种空间索引的性能进行客观比较;(3)在空间数据库中,基于代价的查询优化需要利用代价模型来评估复杂空间查询的代价,以选择最优查询计划[71].

鉴于空间查询的重要性和通用性,且空间索引的性能通常通过域查询的能力来反映,因此关于 R 树及其变体的代价模型的研究大部分集中在域查询方面.

Faloutsos 等最先尝试分析 R 树域查询性能[72].他们提出的模型假设数据是一致分布的,并且树的每个结点都被填满(packed 树).虽然该模型有其局限性,但是以后几乎所有的代价模型都是以此为基础发展起来的.式(1)是最著名的计算 R 树高度的公式:

$$h = \log_f \frac{N}{C} \tag{1}$$

其中,N 是数据记录总数,C 是叶节点容量,f(fanout)是结点平均容量.

1993 年,Kamel 和 Pagel 分别独立地对式(1)进行了扩展,提出了式(2)[33,73].假设 n 维空间中有一 n 维的 R 树,该 R 树有 k 个结点,结点 s_j 的边 $(s_{j1},s_{j2},\cdots,s_{jn})$ 为已知,则对于任一查询窗口 $q=(q_1,q_2,\cdots,q_n)$,可以得到查询的平均磁盘访问次数 $DA(q)$.

$$DA(q) = \sum_{j=1}^{k}\left\{\prod_{i=1}^{n}(s_{j,i}+q_i)\right\} \tag{2}$$

该模型的贡献之一是揭示了除面积之外 R 树结点周长最小化的重要性,有助于更好地理解 R* 树.但该模型要求 R 树必须预先建立,且假设数据和查询窗口是均匀分布.

1995 年,Faloutsos 和 Kamel 对文献[33]进行了扩展[74],利用点数据集的分形维属性 d,使代价模型能够预测磁盘访问次数.该模型是第一个能够处理非均匀分布数据的代价模型,但仅针对点数据集.同样,Bleussi 和 Faloutsos 也利用分形维作出了选择性评估[75].

几乎同时,Pagel 等也对文献[73]进行了扩展[76],作者提出了一个优化算法用来计算静态 R 树性能的下界,从而能够在不同空间索引结构之间进行绝对性能比较.利用该算法,作者计算出当时最好的静态 R 树和动态 R 树分别是 Hilbert packed R 树[33]和 R* 树[18].一年之后,作者又对结点矩形的面积、周长和数量三要素进行了进一步研究,推导出了各种域查询的查询代价模型,并且得出一个重要结论:窗口查询性能在一般情况下可以代表其它各种域查询的性能[77].

1996 年,Theodoridis 和 Sellis 提出了一个新的

代价模型[71],该模型仅利用了数据集本身的特性,即数据的个数和数据密度,因此,该模型可以在 R 树创建之前就可以评估其查询性能.根据实验结果,对于均匀分布和非均匀分布的数据集,该模型的相对误差在 10%～15%之间.

2000 年,Jin 等采用与文献[71]类似的数据密度的思想,提出了自己的选择性评估和查询代价模型[78,79].不同之处在于,Jin 等维护的是一个作为辅助数据结构的数据密度文件,基于数据密度文件来实施累积密度方案.根据作者的实验结果,该模型的选择性评估和域查询代价评估的相对误差不超过 5%.

有别于数据密度的思想,1999 年,Proietti 等研究了 R 树结点中 MBR 的分布,发现它们遵循所依赖数据的分布[80].因此,作者利用 MBR 的分布,得出了不同以往的域查询代价模型,实验结果表明其域查询 I/O 次数的最大平均相对误差小于 30%.

至此,所有上述代价模型都是以结点访问次数作为查询性能的基准,都忽略了缓冲区对于查询代价评估的影响.直至 2000 年,Leutenegger 等才将基于 LRU 算法的缓冲区尺寸引入到 R 树查询代价模型中[81].通过研究三种著名的 R 树变体的加载性能,作者得出一个重要结论:如果忽略缓冲区的影响,只把节点访问次数作为性能评价基准,将会得出错误的结果.因此,在 R 树的性能研究中,应该把基于缓冲区的磁盘访问次数作为主要衡量基准.

综上所述,基于 R 树的域查询代价模型研究已经相当深入,而最近邻查询和空间连接的代价模型和选择性评估也取得了一定的发展[82,83];由于大多数的代价模型和选择性评估采用的是参数化技术,即事先对数据集的特性进行假设,如假设数据集遵循均匀分布等,然后用近似公式对其进行评估,因此受到较大的限制.目前,部分学者提出利用采样技术和直方图技术,从给定的数据集中提取充分的信息对数据集进行选择性评估的方法.该方法限制性小、更有针对性且更为精确,一经提出就得到了广泛的关注[84,85].

6 基于 R 树的并行处理与并发控制

6.1 R 树并行处理

并行是数据库的主要发展趋势之一,而且由于空间操作既是 CPU 密集型又是 I/O 密集型的,扩展空间查询语言又比传统的 SQL 有更多的基本操作,使得空间数据库系统在并行方面的需求与传统的关系数据库有所不同.

针对传统串行 R 树算法的并行改进主要基于多磁盘系统和多处理器系统.在多磁盘系统下,主要是利用 I/O 并行提高系统吞吐量,而难点在于如何在多个可用磁盘上合理分解 R 树结构,以期在查询处理过程中,在多个磁盘间尽可能平均分配工作量.在多处理器系统下,需同时利用 I/O 并行和 CPU 并行来提高系统性能,其难点在于如何更好地分解查询以便在多个处理器之间实现负载平衡.

早在 1992 年,Kamel 等就针对单处理器、多磁盘的硬件结构研究了基于 R 树的并行查询机制[86].作者采取多元 R 树(Multiplexed R 树)软件结构,应用邻近指数(proximity index)准则进行优化,对于均匀分布数据获得了优于其它方法的并行域查询性能.随后,专家学者们研究了多磁盘环境下的 R 树最近邻并行查询算法[87]、多计算机系统下的 R 树并行算法[88]、共享虚拟内存结构下的并行空间连接算法[89]、多磁盘环境下的 R 树最近邻并行查询算法[90,91];出现了 GPR 树[92]、Master-client R 树[93]、Upgraded Parallel R 树[94]等并行 R 树变体.

6.2 并发控制与锁定技术

在多用户环境中,并发控制技术是必不可少的.在空间数据库环境下基于 R 树的并发机制仍是一个尚待研究的领域[47].当多个用户同时对一个索引结构进行查询、插入、删除等操作时,并发机制必须保证数据一致性和各个操作的正确性.成熟的 B 树并发控制技术并不完全适用于 R 树,主要原因是 R 树的键值是多维的 MBR,其上没有线性顺序,而且兄弟结点间的 MBR 可能交叠.

1993 年,Ng 等首次对 R 树的并发访问机制进行了研究[95],提出了三种不同的锁定方法用于并发控制,分别为单一锁、修改锁和耦合锁.实验结果表明,耦合锁具有最好的检索性能.但是,该方法修改了原始 R 树的结构,限制了它的通用性.

一个提高 B 树并发度的策略是确保在给定时间内只有一个结点被锁定[96],该策略随后在 1994 年被 Ng 和 Kornacker 等修改用于 R 树[97,98],并且分别提出了 R-link 方法.两者不同的是,Kornacker 应用了更为高级的逻辑序列码技术(Logical Sequence Numbers,LSN)来对兄弟结点排序.实验结果显示,R-link 方法比耦合锁机制有更好的吞吐量和更快的响应速度.

1997 年,Chen 等提出了不同于 R-link 的算

法[99].该算法使用三种类型的锁:读锁、写锁和排它锁,应用广度优先检索和耦合锁技术来定位查询对象.其优势在于不需要修改原始 R 树的结构,因此易于和 R 树家族中的各种变体集成.

R 树中的幻像保护问题由 Chakrabarti 等提出[100],作者设计了三种算法来平衡计算复杂度、并发度和锁定开销,结果是各有利弊.进而,1998 年,作者首次提出了基于动态粒度锁的多维数据结构幻像问题的解决方法[101].

7　讨论与展望

综上所述,就其研究的广度和深度以及商业实用化的进程来看,R 树是无处不在的,是目前最流行,也是最成功的多维索引方法.

但是,正如 Gaede 等所述[1],面对如此众多的 R 树新变体和新算法,即使是该领域的专家也难以判定其优劣性.没有哪一种 R 树变体在学术和商业应用上获得一致的认可和肯定.R^* 树是学术界公认的各方面性能优良的变体,但在商业应用上却远不如算法简单、易于实现和维护的原创 R 树广泛.R^+ 树解决了兄弟结点区域重叠问题,检索性能优良,但高算法复杂度使其难以与并发、并行算法整合,在实际应用中少有实现.本文仅旨在客观评述和介绍 R 树及其各种变体和相关算法的优缺点和研究进展,以期为在具体应用中正确地选择 R 树的变体和算法提供依据.

20 年来,对于 R 树数据结构和分裂算法的研究已经相当成熟,许多专家学者已经转向研究基于 R 树的各种相关算法.以下是当前 R 树家族的研究热点问题.

(1)批量操作技术.R 树生成中的静态批量加载和动态批量插入仍是当前的研究热点之一,而作为重要空间操作的动态批量修改和删除方面的算法研究较少,亟待加强.

(2)空间连接和最近邻查询算法.就目前商业空间数据库系统应用来看,海量空间数据的分析和操作效率难以让用户满意,仍需研究更为高效的算法.然而不论是空间查询还是空间分析过程,都与空间索引结构和基于空间索引的基本空间操作密切相关.另外,方位查询算法也是值得研究的领域之一,对于 GIS 应用系统尤为重要.

(3)基于采样和直方图技术的选择性评估和代价模型研究.该研究刚刚起步,但该技术决定了最优查询方案的选取,直接关系到系统的检索性能,对于空间数据库的查询优化非常重要.

(4)并行处理和并发控制.它们都是数据库前沿领域的研究热点,能够全面提升系统性能,保证数据的一致性和完整性,关系到一种空间索引或变体是否能够成功与实际应用整合.

(5)时空高维索引.随着维度的增加,R 树结点间重叠区域的快速增长使其性能急剧恶化,称为"维度咒语".为解决大量高维应用中的数据检索问题,如多媒体数据库、时空数据库、移动对象、弹道监控等,需要研究基于 R 树的高效时空高维索引技术.

(6)统一的测试平台.Gaede 等为了更好地比较各种索引方法的性能,揭示理论模型中掩盖的问题和缺陷,提出建立标准的测试平台,提供平台独立的方法以实现各种各样的索引方法.但到目前为止,仍然没有令人满意的统一测试平台对层出不穷的索引方法及其变体进行令人信服的性能测试和比较,需要在以后的研究中解决.

参　考　文　献

1　Gaede V., Gunther O.. Multidimensional access methods. ACM Computing Surveys, 1998, 30(2): 170～231

2　Nievergelt J., Hinterberger H., Sevcik K. C.. The grid file: An adaptable, symmetric multikey file structure. ACM Transactions on Database Systems, 1984, 9(1): 38～71

3　Seeger B., Kriegel Hans-Peter. The Buddy-Tree: An efficient and robust access method for spatial data base systems. In: Proceedings of the 16th VLDB, Brisbane, Australia, 1990, 590～601

4　Robinson J. T.. The K-D-B-Tree: A search structure for large multidimensional dynamic indexes. In: Proceedings of ACM SIGMOD, Ann Arbor, 1981, 10～18

5　Lomet D. B., Salzberg B.. The hB-Tree: A multiattribute indexing method with good guaranteed performance. ACM Transactions on Database Systems, 1990, 15(4): 625～658

6　Henrich A., Six Hans-Werner, Widmayer P.. The LSD tree: Spatial access to multidimensional point and nonpoint objects. In: Proceedings of the 15th VLDB, Amsterdam, Netherlands, 1989, 45～53

7　Sellis T. K., Roussopoulos N., Faloutsos C.. Multidimensional access methods: Trees have grown everywhere. In: Proceedings of the 23rd VLDB, Athens, Greens, 1997, 13～14

8　Ahn H. K., Mamoulis N., Wong H. M.. A survey on multidimensional access methods. UU-CS, Utrecht, The Netherlands: Technical Report 2001-14, 2001

9　Lu Hongjun, Ooi Beng Chin. Spatial indexing: Past and future. IEEE Data Engineering Bulletin, 1993, 16(3): 16～21

10　Faloutsos C., Roseman S.. Fractals for secondary key retrieval. In: Proceedings of the 8th PODS, Philadelphia, Pennsylva-

nia, 1989, 247～252

11 Lu Feng, Zhou Cheng-Hu. A GIS spatial indexing approach based on Hilbert ordering code. Journal of Computer-Aided Design & Computer Graphics, 2001, 13(5): 424～429(in Chinese)
(陆 锋,周成虎.一种基于 Hilbert 排列码的 GIS 空间索引方法.计算机辅助设计与图形学学报, 2001, 13(5): 424～429)

12 Sellis T. K., Roussopoulos N., Faloutsos C.. The R+-tree: A dynamic index for multi-dimensional objects. In: Proceedings of the 13th VLDB, Brighton, England, 1987, 507～518

13 Günther O.. Efficient Structures for Geometric Data Management. New York: Springer-Verlag, 1988

14 Günther O.. The design of the cell tree: An object-oriented index structure for geometric databases. In: Proceedings of ICDE, Los Angeles, CA, 1989, 598～605

15 Gargantini I.. An effective way to represent quadtrees. Communications of the ACM, 1982, 25(12): 905～910

16 Samet H.. The Design and Analysis of Spatial Data Structures. Reading, MA: Addison-Wesley, 1990

17 Guttman A.. R-trees: A dynamic index structure for spatial searching. In: Proceedings of ACM SIGMOD, Boston, MA, 1984, 47～57

18 Beckmann N., Kriegel H. P., Schneider R., Seeger B.. The R*-tree: An efficient and robust access method for points and rectangles. In: Proceedings of SIGMOD, Atlantic City, New Jersey, 1990, 322～331

19 Kamel I., Faloutsos C.. Hilbert R-tree: An improved R-tree using fractals. In: Proceedings of the 20th VLDB, Santiago, Chile, 1994, 500～509

20 Huang P. W., Lin P. L., Lin H. Y.. Optimizing storage utilization in R-tree dynamic index structure for spatial databases. Journal of Systems and Software, 2001, 55(3): 291～299

21 Brakatsoulas S., Pfoser D., Theodoridis Y.. Revisiting R-tree construction principles. In: Proceedings of the 6th ADBIS, Bratislava, Slovakia, 2002, 149～162

22 Oosterom P., Johannes P., van M.. Reactive Data Structures for Geographic Information Systems. Oxford, New York: Oxford University Press, 1993

23 Shi Wen-Zhong *et al*.. A spatial indexing method for GIS. ACTA Geodaetica et Cartographica Sinica, 2001, 30(2): 156～161(in Chinese)
(史文中等.一种面向地理信息系统的空间索引方法.测绘学报, 2001, 30(2): 156～161)

24 Guenther O.. The cell tree: An object oriented index structure for geometric databases. In: Proceedings of the 5th IEEE ICDE, Los Angeles, CA, 1989, 598～605

25 Jagadish H. V.. Spatial search with polyhedra. In: Proceedings of the 6th IEEE ICDE, Orlando, FL, 1990, 311～319

26 Ang C. H., Tan T. C.. New linear node splitting algorithm for R-trees. In: Proceedings of the 5th SSD, Berlin, Germany, 1997, 339～349

27 Garcia Y., Lopez M., Leutenegger S.. On optimal node splitting for R-trees. In: Proceedings of the 24th VLDB, New York, NY, 1998, 334～344

28 Schrek T., Chen Z.. Branch grafting method for R-tree implementation. Journal of Systems and Software, 2000, 53(1): 83～93

29 Lee Y. J., Chung C. W.. The DR-Tree: A main memory data structure for complex multidimensional objects. Geoinformatica, 2001, 5(2): 181～207

30 Ang C. H., Tan T. C.. Bitmap R-trees. The National University of Singapore. Singapore: Technical Report TRBS/99, 1999

31 Böhm C., Berchtold S., Keim D. A.. Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. ACM Computing Surveys, 2001, 33(3): 322～373

32 Roussopoulos N., Leifker D.. Direct spatial search on pictorial databases using packed R-trees. In: Proceedings of ACM SIGMOD, Austin, TX, 1985, 17～31

33 Kamel I., Faloutsos C.. On packing R-trees. In: Proceedings of CIKM, Washington, DC, USA 1993, 490～499

34 van den Bercken J., Seeger B., Widmayer P.. A generic approach to bulk loading multidimensional index structures. In: Proceedings of the 23rd VLDB, Athens, Greece, 1997, 406～415

35 Lo M. L., Ravishankar C. V.. Generating seeded trees from data sets. In: Goos G. ed.. Lecture Notes in Computer Science 951. London: Springer-Verlag, 1995, 328～347

36 Arge L.. Efficient external-memory data structures and applications[Ph. D. dissertation]. University of Aarhus, Danmark, 1996

37 Leutenegger S., Edgington J. M., Lopez M. A.. STR: A simple and efficient algorithm for R-tree packing. In: Proceedings of the 13th IEEE ICDE, Birmingham, England, 1997, 497～506

38 Garcia Y., Lopez M., Leutenegger S.. A greedy algorithm for bulk loading R-trees. In: Proceedings of the 6th ACM-GIS, Washington, DC, 1998, 163～164

39 Arge L., Hinrichs K., Vahrenhold J., Vitter J. S.. Efficient bulk operations on dynamic R-trees(Extended Abstract). In: Proceedings of ALENEX, Baltimore, MD, USA, 1999, 328～348

40 Arge L., Hinrichs K. H., Vahrenhold J., Vitter J. S.. Efficient bulk operations on dynamic R-trees. Algorithmica, 2002, 33(1): 104～128

41 Lee T., Sukho. OMT: Overlap minimizing top-down bulk loading algorithm for R-tree. Advanced Information Systems Engineering, 2003, 7: 69～72

42 Kamel I., Khalil M., Kouramajian V.. Bulk insertion in dynamic R-trees. In: Proceedings of SDH, Delft, the Netherlands, 1996, 3B. 31～3B. 42

43 Chen Li, Choubey R., Rundensteiner E. A.. Bulk-Insertions into R-trees using the Small-Tree-Large-Tree approach. In: Proceedings of ACM-GIS, Washington, DC, USA, 1998, 161～162

44 Chen Li, Choubey R., Rundensteiner E. A.. Merging R-trees: Efficient strategies for local bulk insertion. Geoinformatica, 2002, 6(1): 7～34

45 Choubey R., Chen Li, Rundensteiner E. A.. GBI: A generalized R-tree bulk-insertion strategy. In: Proceedings of SSD, Hong Kong, China, 1999, 91～108

46 Lee Taewon, Moon Bongki, Lee Sukho. Bulk insertion for R-tree by seeded clustering. In: Proceedings of DEXA, Prague, Czech Republic, 2003, 129～138

47 Shekhar S., Chawla S.. Spatial Databases: A Tour. New Jersey: Prentice Hall, 2002

48 Brinkhoff T., Horn H., Kriegel H. P., Schneider R.. A storage and access architecture for efficient query processing in spatial database systems. In: Proceedings of the 3rd SSD Symposium, Singapore, 1993, 357～376

49 Brinkhoff T., Kriegel H. P., Schneider R., Seeger B.. Multi-Step processing of spatial joins. In: Proceedings of ACM SIGMOD, Minneapolis, MN, 1994, 197～208

50 Papadopoulos A. N., Manolopoulos Y.. Multiple range query optimization in spatial databases. In: Proceedings of the 2nd ADBIS, Poznan, Poland, 1998, 71～82

51 Papadias D., Theodoridis Y., Sellis T., Egenhofer M.. Topological relations in the world of minimum bounding rectangles: A study with R-trees. In: Proceedings of ACM SIGMOD, San Jose, CA, 1995, 92～103

52 Papadias D., Theodoridis Y., Sellis T.. The retrieval of direction relations using R-trees. In: Proceedings of the 5th DEXA, Athens, Greece, 1994, 173～182

53 Brinkhoff T., Kriegel H. P., Seeger B.. Efficient processing of spatial joins using R-trees. In: Proceedings of ACM SIGMOD, Washington DC, 1993, 237～246

54 Martynov M.. Spatial joins and R-trees. In: Proceedings of the 3rd ADBIS, Moscow, Russia, 1995, 295～304

55 Huang Y. W., Jing N., Rundensteiner E.. Spatial joins using R-trees: Breadth-First traversal with global optimizations. In: Proceedings of the 23rd VLDB, Athens, Greece, 1997, 396～405

56 Lo Ming Ling, Ravishankar C. V.. Spatial hash-joins. In: Proceedings of SIGMOD, Montreal, Quebec, Canada, 1996, 247～258

57 Park Ho-Hyun, Cha Guang-Ho, Chung Chin-Wan. Multi-way spatial joins using R-trees: Methodology and performance evaluation. In: Proceedings of SSD, Hong Kong, China, 1999, 229～250

58 Papadias D., Mamoulis N., Theodoridis Y.. Processing and optimization of multiway spatial joins using R-trees. In: Proceedings of the 18th ACM PODS Symposium, Philadelphia, PA, 1999, 44～55

59 Mamoulis N., Papadias D.. Multiway spatial joins. ACM Transactions on Database Systems, 2001, 26(4): 424～475

60 Mamoulis N., Papadias D.. Slot index spatial join. IEEE Transactions on Knowledge and Data Engineering, 2003, 15(1): 211～231

61 Hjaltason G., Samet H.. Incremental distance join algorithms for spatial databases. In: Proceedings of ACM SIGMOD, Seattle, WA, 1998, 237～248

62 Shin H., Moon B., Lee S.. Adaptive multi-stage distance join processing. In: Proceedings of ACM SIGMOD, Dallas, TX, 2000, 343～354

63 Corral A., Manolopoulos Y., Theodoridis Y., Vassilakopoulos M.. Closest pair queries in spatial databases. In: Proceedings of ACM SIGMOD, Dallas, TX, 2000, 189～200

64 Corral A., Vassilakopoulos M., Manolopoulos Y.. The impact of buffering on closest pairs queries using R-trees. In: Proceedings of the 5th ADBIS, Vilnius, Lithuania, 2001, 41～54

65 Roussopoulos N., Kelley S., Vincent F.. Nearest neighbor queries. In: Proceedings of ACM SIGMOD, San Jose, CA, 1995, 71～79

66 Cheung K. L., Fu A.. Enhanced nearest neighbour search on the R-tree. ACM SIGMOD Record, 1998, 27(3): 16～21

67 Hjaltason G., Samet H.. Distance browsing in spatial databases. ACM Transactions on Database Systems, 1999, 24(2): 265～318

68 Korn F., Muthukrishnan S.. Influence sets based on reverse nearest neighbor queries. In: Proceedings of SIGMOD, Dallas, Texas, USA, 2000, 201～212

69 Stanoi I., Agrawal D., Abbadi A.. Reverse nearest neighbor queries for dynamic datasets. In: Proceedings of the 5th DMKD Workshop, Dallas, TX, 2000, 44～53

70 Ferhatosmanoglu H., Stanoi I., Agrawal D., Abbadi A.. Constrained nearest neighbor queries. In: Proceedings of the 7th SSTD Symposium, Redondo Beach, CA, 2001, 257～278

71 Theodoridis Y., Sellis T. K.. A model for the prediction of R-tree performance. In: Proceedings of PODS, Montreal, Canada, 1996, 161～171

72 Faloutsos C., Sellis T. K., Roussopoulos N.. Analysis of object oriented spatial access methods. In: Proceedings of SIGMOD, San Francisco, California, 1987, 426～439

73 Pagel B. U., Six H. W., Toben H., Widmayer P.. Towards an analysis of range query performance in spatial data structures. In: Proceedings of PODS, Washington, DC, USA, 1993, 214～221

74 Faloutsos C., Kamel I.. Beyond uniformity and independence: Analysis of R-trees using the concept of fractal dimension. In: Proceedings of PODS, Minneapolis, Minnesota, 1994, 4～13

75 Belussi A., Faloutsos C.. Estimating the selectivity of spatial queries using the correlation fractal dimension. In: Proceedings of the 21th VLDB, Zurich, Switzerland, 1995, 299～310

76 Pagel B. U., Six H. W., Winter M.. Window query optimal clustering of spatial objects. In: Proceedings of PODS, 1995, San Jose, California, 86～94

77 Pagel B. U., Six H. W.. Are window queries representative for arbitrary range queries?. In: Proceedings of PODS, Montreal, Canada, 1996, 150～160

78 Jin J., An N.. Sivasubramaniam A.. Analyzing range queries on spatial data. In: Proceedings of ICDE, San Diego, California, USA, 2000, 525～534

79 An N., Jin J., Sivasubramaniam A.. Toward an accurate analysis of range queries on spatial data. IEEE Transactions on Knowledge and Data Engineering, 2003, 15(2): 305～323

80 Proietti G., Faloutsos C.. I/O complexity for range queries on region data stored using an R-tree. In: Proceedings of ICDE, Sydney, Australia, 1999, 628～635

81 Leutenegger S. T., Lopez M. A.. The effect of buffering on the performance of R-trees. In: Proceedings of ICDE, Orlando, Florida, USA, 1998, 164～171

82 Huang Y. W., Jing N., Rundensteiner E.. A cost model for

estimating the performance of spatial joins using R-trees. In: Proceedings of SSDBM, Olympia, WA, 1997, 30～38

83 Papadopoulos A. N., Manolopoulos Y.. Performance of nearest neighbor queries in R-trees. In: Proceedings of ICDT, Delphi, Greece, 1997, 394～408

84 An N., Yang Zhen-Yu, Sivasubramaniam A.. Selectivity estimation for spatial joins. In: Proceedings of ICDE, Heidelberg, Germany, 2001, 368～375

85 Sun Chengyu, Agrawal D., Abbadi A. E.. Selectivity estimation for spatial joins with geometric selections. In: Proceedings of EDBT, Prague, Czech Republic, 2002, 609～626

86 Kamel I., Faloutsos C.. Parallel R-trees. In: Proceedings of SIGMOD, San Diego, California, 1992, 195～204

87 Papadopoulos A., Manolopoulos Y.. Similarity query processing using disk arrays. In: Proceedings of SIGMOD, Seattle, Washington, USA, 1998, 225～236

88 Koudas N., Faloutsos C., Kamel I.. Declustering spatial databases on a multi-computer architecture. In: Proceedings of EDBT, Avignon, France, 1996, 592～614

89 Brinkhoff T., Kriegel Hans-Peter, Seeger B.. Parallel processing of spatial joins using R-trees. In: Proceedings of ICDE, New Orleans, Louisiana, 1996, 258～265

90 Papadopoulos A., Manolopoulos Y.. Parallel processing of nearest neighbor queries in declustered spatial data. In: Proceedings of ACM-GIS, Rockville, MD, 1996, 35～43

91 Papadopoulos A., Manolopoulos Y.. Nearest neighbor queries in shared-nothing environments. Geoinformatica, 1997, 1(4): 369～392

92 Fu X., Wang D., Zheng W.. GPR-tree: A global parallel index structure for multiattribute declustering on cluster of workstations. In: Proceedings of APDC'97, Shanghai, China, 1997, 300～306

93 Schnitzer B., Leutenegger S. T.. Master-client R-trees: A new parallel R-tree architecture. In: Proceedings of SSDBM, Cleveland, Ohio, USA, 1999, 68～77

94 Lai Shu-Hua, Zhu Feng-Hua, Sun Yong-Qiang. A design of parallel R-tree on cluster of workstations. In: Proceedings of DNIS, Aizu, Japan, 2000, 119～133

95 Ng V., Kameda T.. Concurrent accesses to R-trees. In: Proceedings of SSD, Singapore, 1993, 142～161

96 Lehman P. L., Yao S. B.. Efficient locking for concurrent operations on B-trees. ACM Transactions on Database Systems, 1981, 6(4): 650～670

97 Ng V., Kameda T.. The R-link tree: A recoverable index structure for spatial data. In: Proceedings of DEXA, Athens, Greece, 1994, 163～172

98 Kornacker M., Banks D.. High-concurrency locking in R-trees. In: Proceedings of VLDB, Zurich, Switzerland, 1995, 134～145

99 Chen J. K., Huang Yin-Fu, Chin Yeh-Hao. A study of concurrent operations on R-trees. Information Sciences, 1997, 98 (1～4): 263～300

100 Chakrabarti K., Mehrotra S.. Concurrency control in R-trees. In: Proceedings of the 1st Symposium on the Federated Lab on Interactive and Advanced Display, Aberdeen, MD, 1997

101 Chakrabarti K., Mehrotra S.. Dynamic granular locking approach to phantom protection in R-trees. In: Proceedings of ICDE, Orlando, USA, 1998, 446～454

ZHANG Ming-Bo, born in 1971, Ph. D. candidate, lecturer. His main research interests include spatial database engine and spatial index.

LU Feng, born in 1970, Ph. D., professor. His main research interests include GIS spatial data modeling, representation and analysis of geographical network.

SHEN Pai-Wei, born in 1977, Ph. D. candidate. His research interests include spatial databases.

CHENG Chang-Xiu, born in 1973, Ph. D., associate professor. Her main research interests include GIS spatial data modeling.

Background

This work is supported in part by the National High Technology Research and Development Program (863 Program) under grant No. 2003AA135070, in part by the National Natural Science Foundation of China under grant No. 40201043. One of the common aims of the two projects is to study and accomplish the effective spatial index structure on massive spatial data. Spatial indexing is a key technique of spatial database providing rapid and efficient processing on various spatial operations. But, the classical one-dimensional database indexing structures, such as B-tree, are not appropriate to multi-dimensional spatial searching. In recent years, a great variety of multi-dimensional access methods have been proposed. One of the most influential access methods in this area is the R-tree structure proposed by Guttman in 1984.

This work, as the research base of author' projects, summarizes and comments on the evolvement and the progress of the R-tree family, including: (1) R-tree and it's primary variety. (2) Various bulk operation techniques based on R-tree, such as static bulk loading, dynamic bulk inserting. (3) Spatial query processing algorithm based on R-tree, such as range query, spatial join. (4) Spatial query cost model and query optimization based on R-tree. (5) The parallelism, concurrency control and locking strategy based on R-tree.