SAMBASE 空间查询 数据库

计算机科学1996Vol.23№5

SAMBASE 中的空间查询处理

47-49 TP316

易文根 韩 波 石树刚 郑振楣
(武汉大学计算机科学系 武汉430072)

摘 要 This paper introduces A-tree, a new method for indexing spatial data in SAMBASE. A-tree supports explicit queries, queries based on all kinds of topological relations and nearest neighbor queries, it proves that SAMBASE has powerful ability to process spatial query, and can meet the application request of most spatial database.

关键词 Spatial index, Spatial query, Spatial topological relation, Nearest neighbor query.

SAMBASE 是武汉大学计算机科学系数据库组研制开发的一个运行在 SUN 工作站 UNIX 环境下的集中式多用户的面向对象数据库系统，最近的方向集中于地理信息系统的结合，因而要求具备强大的空间查询处理能力。

A 树是在 SAMBASE 上实现的一种空间索引技术，它综合了 R 树与栅格文件索引，利用最小栅格块(最小基本块)来实现非精确查询。A 树的非精确性不同于四叉树在于：1)使用 A 树，无论用户所关心区域位于地图何处，选出的是能覆盖用户查询区的最少的基本块中的对象。2)四叉树建立时，全图范围必须已经确定，四叉树的深度由用户指定，当插入对象超出原图范围时，四叉树需要进行重构，对地图的修改极为不利；而 A 树索引采用类似 R 树由底向上生长的动态过程，全图范围可以随着对象的不断插入，从无到有逐步扩大，反映在 A 树上，是树的层次由底向上随着全图范围的扩大而向上生长。这种动态性给索引的建立和对象的插入带来了很大的灵活性。

对 A 树索引上的操作也是两种空间数据存取方法的结合，在对象插入或对象删除时，操作对象在索引中的定位技术类似于栅格文件索引，而查询应用时在 A 树中的下降则类似于 R 树的查询算法，因此其插入算法简单而查询仍然高效。

一、对精确查询的支持

在对空间对象进行空间索引时，由于其空间位置描述的复杂性(如点只需两个坐标值，而直线要四个坐标值，弧段和面则需要更多而且不确定)，索引中存储的往往是对于空间对象的一个抽象，其中最常用的是最小约束框(MBR，Minimal Bounding Rectangle)，即为能包含空间对象的与坐标轴平行的矩形(二维，三维则为立方体，依此类推)，所以在索引操作中所得出的结果是保守的，即除了包括所有满足查询条件的对象标识外，还可能包括一些并不满足查询条件的对象标识。因而在利用空间索引处理查询时，往往有以下两个步骤：

A Filter 步。使用空间索引来快速排除不可能满足查询的对象，这一步的结果是包含所有正确结果和可能的一些错误命中的一组候选值。

B Refine 步。检测 A 步的每一个候选值，剔除出并不满足查询条件的对象标识，得到查询操作的最终的精确结果。

B 步是典型的 CPU 操作，其运行时间依赖于 A 步的准确性，因为每个错误命中都需要检查。对一些存取方法来说，A 步的准确性可以在一定的程度内受到控制，如通过调整数据表示的冗余数可以在准确性和所花费的时间上进行折衷，因而可以根据特定的应用来调整各步中所花费的时间。比如说，如果数据和查询对象分别是点和框，那么每个候选值可以快速地验证，而数据和查询对象更复杂时，如任意的多边形，这种验证就非常昂贵，在 A 步上花费更多的时间以减少错误命中就更合适些。

在 A 树实现的最初，只考虑了模糊查询，精确度是建立 A 树时所给出的基本范围块，但实际应用中也有对精确度要求很高的环境，如编辑操作中经常用到的开窗删除对象。为了在数据库中实现精确的空间查询，必须对已有的 A 树和数据库的空间查询处理进行修改，这里主要介绍对 A 树的结构和查询操作的修改。前面提及，A 树的精确度是基本范围块的大小，在最初的设计中，只在 A 树的内部结点上存放此结点的所有子结点的 MBR 的最小覆盖

易文根 硕士生，主要研究方向为数据库。石树刚、郑振楣 教授，主要研究领域为数据库。

MBR，而在叶结点中只存放落在最小基本范围块中各对象的对象标识，为了支持精确查询，必要的修改有：

A 树结构的修改：叶结点中存放的数据项为(MBR，OID)，将落在最小基本范围块中各对象的对象标识和最小约束框一起存放。

A 树查询操作的修改：当查询算法下降到叶结点时，并不是简单地把叶结点中的所有 OID 全部返回，而是继续用叶结点中存放的 MBR 验证所要求的拓扑关系，相符合的才返回，其余的被剔除以减少错误命中。A 树经过这样的修改后，增加不多的存储花费和 CPU 花费就能支持精确的空间查询，由于错误命中的减少，减少了要读出的对象的数目，因而查询的效率并不一定降低。

二、支持的拓扑关系的扩充

在 A 树的最初实现时，大多数空间存取方法都只致力于两种拓扑关系即 disjoint 和 not_disjoint，及在距离信息上的检索，这并不是因为其它的空间拓扑关系在实际应用中不重要，大部分原因在于空间拓扑关系定义上的欠缺。文[1]中提出的拓扑关系有八种：disjoint(P,Q)，meet(P,Q)，overlap(P,Q)，covers(P,Q)，covered_by(Q,P)，contains(P,Q)，inside(P,Q)，equal(P,Q)。其中后面七种即为 not_disjoint 的细分，研究实际应用中可能遇到的情况，A 树选择支持以下的拓扑关系：disjoint，与原分类无异；covered_by，包括 inside，equal 和 covered_by；overlap，包括 meet 和 overlap；covers，包括 contains，equal 和 covers。

为了支持这些拓扑关系，A 树需要存取操作的扩充。在原 A 树仅处理 not_disjoint 的查询过程中修改在非叶子结点下降过程中对分支的删除的判断以及叶子结点中对于空间对象的 MBR 是否符合要查询的拓扑关系的判定。假定有以下几个函数：(其中 p，q 均为 MBR)

disjoint(p，q)，p 和 q 不相交则返回1，否则返回0；

overlap(p，q)，p 和 q 重叠则返回1，否则返回0；

covers(p，q)，p 完全覆盖 q 则返回1，否则返回0；

若要查询的 MBR 为 Q_MBR，则各拓扑关系在 A 树上的查询操作为：

overlap：在非叶结点中，对各索引项的 MBR 计算 disjoint(MBR，Q_MBR)，为1则删除此分支，为0则沿此分支下降，重复直至到达叶子结点。在叶子结点中，用各对象的 MBR 计算 overlap(MBR，Q_MBR)，为1则返回对象标识，为0则丢弃。

covers：在非叶结点中，对各索引项的 MBR 计算 disjoint(MBR，Q_MBR)，为1则删除此分支，为0则沿此分支下降，重复直至到达叶子结点。在叶子结点中，用各对象的 MBR 计算 covers(MBR，Q_MBR)，为1则返回对象标识，为0则丢弃。

covered_by：在非叶结点中，对各索引项的 MBR 计算 disjoint(MBR，Q_MBR)，为1则删除此分支，为0则沿此分支下降，重复直至到达叶子结点。在叶子结点中，用各对象的 MBR 计算 covers(Q_MBR，MBR)，为1则返回对象标识，为0则丢弃。

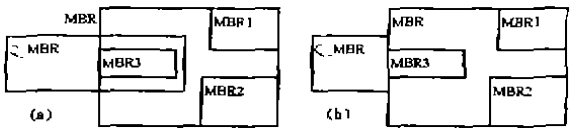

图1 (a)非叶结点所用拓扑关系 not(disjoint)；实际查询的拓扑关系 covered_by。

(b)当 not(disjoint)关系细分为 meet 时，此分支就无需检索。

由于非叶结点中各索引项的 MBR 是此索引项所指子树的所有 MBR_i 的最小覆盖，它也包含一些子树中实际 MBR_i 以外的空间，而 MBR 通常又比 MBR 小得多，因此在非叶结点中要查询的分支的判定所用到的拓扑关系与实际查询的拓扑关系有所不同，如图1(a)所示。如果把拓扑关系细分(如分成前面所讲的八种)，则能更精确地删除不需检索的分支，如图1(b)中的查询，not_disjoint 中的一种，meet 就是在查询 coverd_by 时子结点可以在搜索时删除的一种拓扑关系。

三、最近邻查询

最近邻查询(Nearest Neighbor (NN)Queries)是 GIS 中经常遇到的一种查询，比如编辑操作中用鼠标选取编辑对象就是 NN 查询，NN 查询的有效处理要求所用的空间数据结构能够利用空间对象的近似表示将查询范围集中在潜在的相邻对象上。利用 R 树执行 NN 查询是最新的研究成果[5]，通过 MBR 对对象的近似性我们可以快速地排除与给定点相距较远的对象，缩小搜索范围，从而加速 NN 查询的执行。改造文[5]中提出的 NN 查询算法应用到 A 树上为：

```
PROCEDURE NearestNeighborSearch(Node,Point,Near-
          est,k)
//在 Node 中查找离 Point 最近的 k 个对象，距离的初值为
//正无穷大，Nearest 为指向 k 个保持升序的最近对象的对
//象标识和相应距离的指针
  If Node 为叶结点  Then
    对于 Node 中各对象的 MBR，计算与 Point 的距离；
    如果有比 Nearest[k]中距离小的值，插入此对象到
    Nearest 中并保持升序。
  Else
    对于 Node 中每一索引项中的 MBR，计算与 Point 的
    距离 Distance；
    将它们排序后放于队列 branchlist 中；
    用 Node，Point，Nearest 删去 branchlist 尾中不可能含
    k 个 Nearest 对象的分支
    对 branchlist 的每个分支，递归调用 NearestNeigh-
```

borSearch，求出最新 Nearest；
并用最新 Nearest 再对 branchlist 进行删除。
Endif

这个算法能保证在全图的任意一点出发都能得出最近的 K 个对象，对于一些实际应用（如查找离武汉最近的几个城市），这是行之有效而且是必需的。但对另一些应用，要求的是在一定范围内的 NN 对象，如果在指定范围内并无相邻对象，则不管在此之外是否有相邻对象都返回空值，如在编辑中点取对象的操作，返回离鼠标一定范围内的最近对象，如果没有则点取操作失败，这是符合实际要求的，这样的 NN 实现高效实际，设计算法如下：

1 对鼠标点(x,y)，生成最小误差框 Box=(x−dx,y−dy,x+dx,y+dy)；

2 将 A 树中与 Box 有 not_disjoint 关系的对象用精确查询得出；

3 计算2中得出的各对象与(x,y)的距离，返回值为最小的对象；如果2返回空则返回空值，标识搜索失败。

说明：通过对 dx,dy 的动态调整可以折衷搜索时间和成功率，增大 dx,dy 能使成功率增大而搜索时间增加，因为面积增大而使2中返回的对象增加，减小 dx,dy 则效果相反。

以上两种 NN 问题的解决方案都有一定的适用范围，相对而言，后者灵活一些，但它是以用户的干预为代价的，即用户必需调整 dx 和 dy 以折衷搜索时间和成功率，而且 dx 和 dy 的值不能过大，否则将使要搜索的对象非常多而效率明显下降。

四、SAMBASE 支持空间操作的语言扩充

前面提到的各种空间操作极大地丰富了 SAM 处理空间查询的能力，为支持这些空间操作，需要对 SAM 的查询语言进行扩充。在 A 树最初对拓扑关系为 not_disjoint 的模糊查询的支持时，对标准 SQL 语言中 SELECT 语句进行的扩充为：

```
SELECT …… ……
FROM …… ……
[WHERE(条件)]
[INSIDE(x,y,dx,dy)];
```

其中(x,y,dx,dy)定义了一个二维空间上的 MBR，我们称为 BOX，为了支持扩充的空间操作能力，对语言的继续扩充有：

JOINT(x,y,dx,dy)：与 BOX 相交的对象的精确查询。

COVERS(x,y,dx,dy)：覆盖 BOX 的对象的精确查询。

COVERED_BY(x,y,dx,dy)：被 BOX 覆盖的对象的精确查询。

UNLIMITED_NN(x,y,k)：查询离点(x,y)最近的 K 个对象。

LIMITED_NN(x,y,dx,dy)：查询 BOX 内离点(x,y)最近的一个对象，由于在对象编辑时往往要求的是最近的一个对象，这里只给出这种形式，要扩充到 K 个对象也非常容易。

有了这些谓词的扩展，SAM 处理空间查询的能力就相当强大了。为实现这些谓词，需要在语法分析、语义检查、查询优化、查询处理上进行一系列的修改扩充。比如说，这些谓词可以用析取或者合取连接起来，此时，空间索引可以使用的次数不止一次，我们可以多次利用空间索引搜索满足谓词的对象标识，在结果的集合中求交或求并来得到最终所要的对象标识集。然而，这对于合取联结的谓词往往不是最佳选择，事实上，我们通常选择一个选择性小的空间谓词利用空间索引快速得到候选集，而将其它空间谓词在此候选集上进行计算而得到最终的结果。具体来说，就是在前三个谓词中选择 BOX 最小的谓词，而后两个谓词总是优先级最高的。由于利用空间索引处理空间查询的步骤总有 *FILTER* 和 *REFINE* 两步，因此系统必须有将这些空间查询谓词翻译成系统能不利用空间索引就能执行的操作，因而这样处理并不需要系统额外的处理能力，如果谓词中有选择性足够小的，这样的处理总是最优的。

结束语 我们已经针对 SAMBASE 中 A 树原有实现的不足提出了大量的改进，A 树自身的性能得到了极大的提高，SAMBASE 支持的空间操作也丰富得多，可以满足大部分空间数据库应用的需要。随着空间数据库的发展，要求执行的空间操作也越来越多，现在研究较多的是空间联接，比如基于 NN 的联接操作，在 SAMBASE 上利用 A 树执行空间联接也是今后要研究的主要课题之一。

参考文献

[1]D. Papadias 等，Topological Relations in the World of Minimun Bounding Rectangles：A study with R-trees，In：Proc. of the 1995 ACM SIGMOD Int. Conf.

[2]李立，空间索引技术及其在 SAMBASE 中的实现，交通与计算机，1995年第三期

[3]A. *Guttman*，R-Trees：*A Dynamic Index Structure for* Spatial Searching，In：Proc. of ACM SIGMOD Int. Conf.，1984

[4]严乐安、易文根、王延彬等，嵌套索引在 SAMBASE 中的实现，同[2]

[5]N. Roussopoulos 等，Nearest Neighbor Quries，同[1]