

一种基于网格与R树的多级混合索引

赵楠

(哈尔滨理工大学 计算机科学与技术学院, 黑龙江 哈尔滨 150080)

摘要:结合网格索引和R树索引的特点,提出了一种基于网格与R树的多级混合索引。该方案首先将矩形地理空间进行粗网格划分建立多级网格索引,然后针对每个小网格建立基于R树的空间索引。详细讨论了该索引的结构、建立算法、删除算法以及应用该索引的检索算法,并进行了算法分析。与网格索引和R树索引相比,该索引以略大的空间开销换取了更高的查找性能。

关键词:空间数据对象;网格索引;R树索引;混合索引;空间索引结构

中图分类号:TP311.13

文献标识码:A

文章编号:1673-629X(2009)03-0091-04

A Hybrid Structure of Spatial Multilevel Index Based on Grids and R-Tree

ZHAO Nan

(College of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China)

Abstract: According to the characteristics of spatial index of grids and R-tree, a hybrid structure of spatial multilevel index is presented based on grid and R-tree. A rectangular region of geography is roughly partitioned in several times to multilevel sub-regions, in which spatial index of R-tree has been constructed. The building algorithm and the retrieval algorithm on the index were discussed and analyzed. Compared with grid and R-tree, the index has slightly space costs but most higher index performance.

Key words: spatial data objects; grid index; R-tree; hybrid index; spatial index structure

0 引言

空间索引是提高空间数据库性能的关键技术,它直接影响空间数据的存储效率以及空间检索的性能。按索引结构对被索引空间的划分是否是线性的,将空间索引分成线性的空间索引和非线性的空间索引。非线性空间索引按索引的逻辑组织方式分成网格空间索引和基于树的空间索引。当前,已经存在许多种非线性空间索引技术,这些索引技术包括R树^[1]、K-D-B树^[2]、Bang文件^[3]、R*树^[4,5]、R+树^[6]、四叉树^[7]、QR树^[8,9]、网格索引^[10]等。

基于固定网格划分的空间索引技术是将地理空间分割成 M 行、 N 列,可划分出 $M \times N$ 个小网格,每个网格区域为一个索引项,记录了所有完全或部分落在该矩形区域内的几何实体的标识和外接矩形,索引通过几何实体的外接矩形操作空间实体。固定网格空间

索引只需进行简单的地址计算就可以计算出与查询范围相重叠的所有索引项,索引速度快。为了获得一定的检索精度,网格需要划分得比较细,这就造成许多记录同一个实体索引信息的索引项可能落在多个网格矩形区域中,从而出现重复存储,造成空间索引整体性能的下降。

R树是B树在多维空间的扩展,是一种平衡的树结构。R树结构采用平行于数据空间轴的最小的边界矩形来近似复杂的空间对象,其主要优点是用一定数量的字节来表示一个复杂的对象。尽管这样会丢失很多的信息,但是空间物体的最小边界矩形保留了物体的最重要的几何特性,即空间物体的位置及其在整个坐标轴上的范围。传统的R树索引是一种在空间多维上的索引结构,实体的空间位置信息即最小外接矩形是建立R树索引的唯一依据。文献[11]指出这种面向对象的索引数据组织方式节省了存储空间,但是当索引的数据量增加时,树的深度以及索引空间的重叠会增加,从而索引的操作效率下降。

基于以上的分析,结合网格索引和R树索引的优点提出一种基于网格与R树的多级混合索引方案。

收稿日期:2008-06-16

基金项目:黑龙江省自然科学基金项目(F200601)

作者简介:赵楠(1981-),女,硕士研究生,研究方向为空间数据库;导师:郝忠孝,教授,博士生导师,研究方向为数据库理论、空值理论、无环数据库、主动数据库和时空数据库理论等。

该方案首先将矩形地理空间进行多级粗网格划分建立多级索引,然后针对每个不可再分的网格建立R树索引。实验证明,除了存储空间要略大以外,插入、删除、查询算法的性能要优于R树。

1 多级空间索引的表示

在应用中,用户提出的查询既有精确查询,也有非精确查询。对于精确查询,如果只分成 $M \times N$ 个小块往往达不到查询要求。例如,查询点实体与线实体是否相交。如果点实体和线实体存储在同一个块中,但这并不能说明它们是否相交,只能表明二者比较接近而已。为了达到精确查询的要求,除非 M 和 N 足够大,以至小块不可再分。显然,当 M 和 N 过大时,空间和时间效率将都变得较低。所以为了提高效率,采用多级网格划分策略,使小块仍可再分。块的划分可分为若干等级,但等级过多,就会带来存储空间过多的开销以及降低时间效率(为此对于不同的实际应用,需要不同的索引级数,这可以由用户指定,或按照某种条件进行优选)。

第一级网格划分将整个空间划分成 m_1 行 n_1 列的块,每个块又可进行第二级划分,其中每个块都可划分成任意行和列的下一级块,划分块数可以不同。每个块是否进行下一级划分视实际需要而定。可以采用动态链表存储该结构。通过指针指向下一级划分,如果当前块不再进行划分则该块的多级网格索引建立完毕,然后为每一块建立R树索引,进而建立混合索引。

图1列举了多级网格中部分地物的最小外接矩形的分布情况,首先将整个空间进行 2×2 一级划分,分

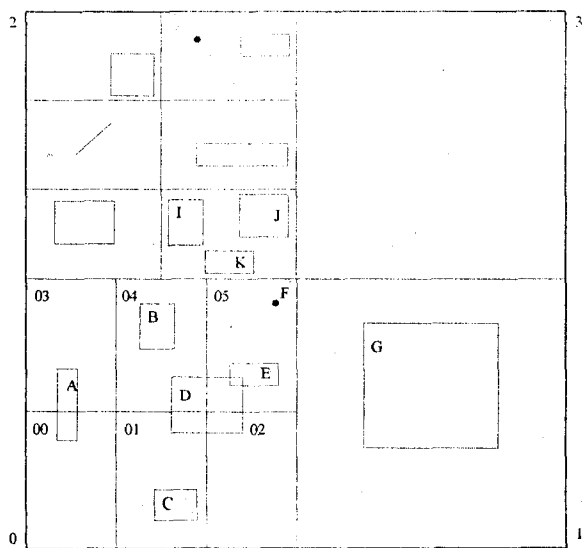


图1 多级网格地物分布图

为四个一级块,块3中未包含任何地物或地物的一部分,因此无需进行二级划分;块1中完整包含了面积较大的地物G,直接在该块中建立以G为根结点的R树

索引;块0与块2中包含较多地物,若直接在区块中建立R树索引,R树中包含的结点数量比较多,因此可对块0与块2进行二级划分,图中对块0进行 2×3 划分,对块2进行 3×2 划分。F、E完全落入05号块中,则形成一棵2个对象的MBR组成的R树;B与C分别落入04和01号块中,分别形成只有根结点的R树;D跨越01、02、04、05四个块,不再存储在R树中,直接存储在跨块地物列表中。另外,图中可以看到I、J、K同处于一个二级块中,可建立一棵3个对象的MBR组成的R树,由于实际应用中,每个块可能包含几百乃至几千个对象,因此亦可选择对其进行再次划分。

2 索引机制和数据结构

此多级混合索引结构是结合多级网格索引和R树索引而提出的一种多级混合索引结构。即将整个索引空间划分成多级子索引空间,然后对最后一级的子索引空间采用R树进行索引。其思想是将一棵“大”的R树分解成多棵“小”的R树,从而减少索引空间重叠,同时降低了R树的平均深度,提高查找性能。

将矩形地理空间按多级索引结构分成若干块,每个块作为一个桶,对最小外接矩形完全落入同一个桶中的若干对象建立R树索引,桶中存储指向R树根结点的指针,跨块对象索引信息直接存储在对应的桶中。

不妨设 $Buck[i]$ 中存放跨块地物索引信息的集合为 $S_Buck[i]$, $Buck[i]$ 所对应的R树为 $Buck[i].RTree$ 。

(1) 对任意的点对象 D_OBJ_x ,对象的唯一标识为 Oid_x ,坐标为 $Point_x$,若 $D_OBJ_x \in Buck[i]$,则有 $\langle Oid_x, Point_x \rangle \in Buck[i].RTree$ 。

(2) 对任意的线对象 L_OBJ_x ,对象的唯一标识为 Oid_x ,最小外接矩形为 L_MBR_x ,不妨设 L_MBR_x 所占的桶号集合为 $\{l_1, l_2, \dots, l_m\}$,若 $m=1$,则 $\langle Oid_x, L_MBR_x \rangle \in Buck[i].RTree$,若 $m>1$,则对 $\forall i \in \{l_1, l_2, \dots, l_m\}$ 有 $\langle Oid_x, L_MBR_x \rangle \in S_Buck[i]$ 。

(3) 对任意的面对象 R_OBJ_x ,对象的唯一标识为 Oid_x ,最小外接矩形为 R_MBR_x ,不妨设 R_MBR_x 所占的桶号集合为 $\{r_1, r_2, \dots, r_m\}$,若 $m=1$,则 $\langle Oid_x, R_MBR_x \rangle \in Buck[i].RTree$,若 $m>1$,则对 $\forall i \in \{r_1, r_2, \dots, r_m\}$ 有 $\langle Oid_x, R_MBR_x \rangle \in S_Buck[i]$ 。

此索引的数据结构由若干个桶数组、一个包含跨块地物索引信息的单链表和一个包含R树根结点索引信息的指针组成。依据图1建立的索引结构如图2所示,一级块0经过二级划分后得到了6个二级块,存储于一级块0的指针指向的区域中,若区块中不包含

跨块地物,则指向单链表的指针为空,若区块中不完全包含任何地物,则该块对应的R树的指针域为空;若区块完全包含若干地物,则该区块中对应R树的指针域指向地物组成的R树根结点。各级区块结点中除存储了区块的标识外,还存储了区块的左下角和右上角坐标信息,如图1中整个区域的左下角位于坐标轴原点,右上角的纵横坐标分别为60和60,一级区块0的左下角纵横坐标分别为0和0,右上角的纵横坐标分别为30和30,则坐标信息表示为(0,0,30,30)。

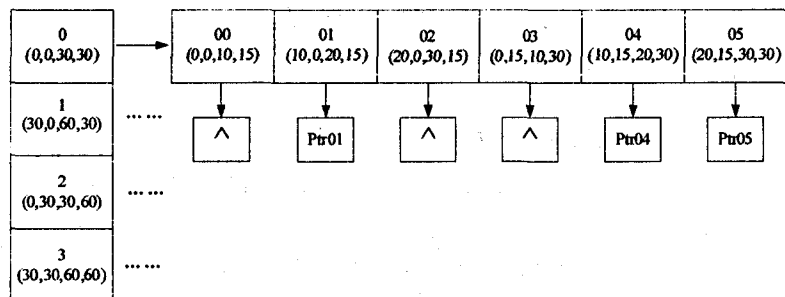


图2 多级混合索引结构

为了便于管理跨块地物对象,建立跨块地物及其对应桶的关系表,表中记录跨块地物的唯一标识Oid、地物跨越桶的数量Num和地物所跨越的桶编号BulkNo_i,如图3所示。在对跨块对象进行操作时,可以根据该表直接找到对象及其相应的桶,提高操作效率。

D	4	01	02	04	05
---	---	----	----	----	----

图3 跨块地物关系表结构

3 主要操作和算法

3.1 插入算法

插入过程涉及点对象的坐标插入过程和线、面对象的外接矩形插入过程。

(1) 点对象索引项的插入算法。

输入:点对象的坐标 point(x, y) 和对象的唯一标识Oid。

输出:带有点状标识及其特征信息的索引。

① 在动态链表中查找点 point(x, y) 相应块号 i 。

② 根据 i 找到相应的桶 Bulk[i], 如果指向R树根结点的指针为空,则建立R树,新结点作为R树的根结点,并将新的R树链接到桶中。转到③。

如果桶中指向R树根结点的指针不为空,如果叶子结点还能容纳索引项,则新结点的索引项直接插到叶子结点中。如果叶子结点溢出,而产生结点分裂,采用文献[1]中的分裂算法。如果产生了结点分裂,则结点分裂后,采用文献[1]中的R树调整算法。如果结点分裂向上传播导致根结点分裂,则生成新的根结点,修改指向R树根结点的指针,使其指向新的根结点。

插入算法采用文献[1]中的方法。

③ 算法结束。

(2) 线、面对象索引项的插入。

输入:对象的最小外接矩形 mbr(min_ x , min_ y , max_ x , max_ y) 和对象的唯一标识Oid。

输出:带有线、面对象标识及其特征信息的索引。

① 计算对象外接矩形左下角和右上角所对应相对块 i 和 j , 左上角和右下角所对应的相对块 k 和 l 。

② 如果 $i = j$ 或 $k = l$, 那么矩形完全落入同一个块中,则采用与点对象索引项的插入算法类似的算法。转到④。

③ 如果 $i \neq j$ 或 $k \neq l$, 则矩形落入多个块中,块的集合为 $\{i, i+1, \dots, l, l+1, \dots, k, k+1, \dots, j\}$, 记为 Set_ B , 每个块均对应一个桶,则桶可以表示为 Bulk[x]($x \in \text{Set}_B$)。将新结点的索引项分别插入 Bulk[x]($x \in \text{Set}_B$), 同时将对象的Oid和其涉及到的桶编号写入跨块地物及其对应的桶关系表中。

④ 算法结束。

3.2 删除算法

删除算法包括点对象的索引信息的删除和线、面对象的索引信息的删除。

(1) 点对象索引项的删除。

输入:待删除点对象的坐标 point(x, y) 和对象的唯一标识Oid。

输出:无。

① 在动态链表中查找点 point(x, y) 的相应块号 i 。

② 根据 i 找到相应的桶 Bulk[i], 从 Bulk[i] 中取出指向R树的指针,如果指针为空,则转③, 否则采用文献[1]中R树结点的删除算法,从R树中删除对象的索引项。

③ 算法结束。

(2) 线、面对象索引项的删除。

输入:待删除对象的索引项信息:最小外接矩形 mbr(min_ x , min_ y , max_ x , max_ y), 对象的唯一标识Oid。

输出:无。

① 计算对象外接矩形左下角和右上角所对应的相对块号 i 和 j 。

② 如果 $i = j$, 那么外接矩形完全落入同一个块中,则采用与点对象索引项的删除算法类似的算法。转到④。

③ 如果 $i \neq j$, 那么对象的外接矩形落入多个块

中,则根据对象的唯一标识 Oid 从跨块地物及其对应的桶关系表中检索出对象所跨越的桶号,将存储在各桶中的对象索引项信息删除。

④算法结束。

3.3 查询算法

查询算法比较复杂,这里只讨论基于窗口的检索。给定一个查询窗口 S , 设查询窗口所覆盖的块号集合为 Set_B , 则查询窗口所对应的桶号集合为 $Set_Bulk(i)(i \in Set_B)$ 。

检索算法描述如下:

输入: 查询窗口 $S(x_1, y_1, x_2, y_2)$, 其中 (x_1, y_1) , (x_2, y_2) 分别为查询窗口的左下角坐标和查询窗口的右上角坐标。

输出: 落入窗口中地物对象索引项的集合 Set_R 。

①置 $Set_R = \emptyset$, 找出查询窗口 S 所对应的桶号集合 $Set_Bulk(i)(i \in Set_B)$ 。

②如果 $Set_Bulk(i) = \emptyset$, 转 ⑧。

③任取 $Bulk[x] \in Set_Bulk(i)$, 如果 $S \cap Bulk[x].Rtree.MBR \neq \emptyset$, 采用文献[1]中的区域查询算法找出 R 树叶节点中所有落入查询窗口 S 中的地物对象的索引项, 并将它们并入 Set_R 。

④找出桶 $Bulk[x]$ 对应的跨块地物索引项的集合 $S_Bulk[x]$ 。

⑤如果 $S_Bulk[x] = \emptyset$, 则 $Set_Bulk(i) \leftarrow Set_Bulk(i) - S_Bulk[x]$, 转 ②。

⑥任取某一跨块地物的索引项 $\langle Oid_j, MBR_j \rangle \in S_Bulk[x]$, 如果 $S \cap MBR_j \neq \emptyset$, 则 $Set_R \leftarrow Set_R \cup \{\langle Oid_j, MBR_j \rangle\}$ 。

⑦ $Set_Bulk[x] \leftarrow Set_Bulk[x] - \{\langle Oid_j, MBR_j \rangle\}$, 转 ⑤。

⑧输出集合 Set_R , 算法结束。

4 算法分析

空间复杂性: 此索引采用的技术是将矩形地理空间先进行多级网格划分, 从而使得每个格中涉及到的跨块对象数量大大减少, 降低了重复存储。完全包含于每一网格中的对象, 采用基于对象分割的 R 树存储方式, 进一步减少了存储开销。多级网格索引由于需要达到一定的检索精度, 必须进行细分, 带来的直接影响是数据的冗余度增高, 重复存储增加, 在空间上需要较大的开销。 R 树索引是基于面向对象的存储方法, 存储开销最经济。所以此索引结构在存储开销上要远远好于多级网格索引, 比 R 树索引略大。

时间复杂性: 多级网格索引只需进行简单的地址

运算和少量的磁盘访问, 就能够快速定位到所要检索的对象, 检索速度最快, 但是由于重复存储, 对象索引的插、删操作比较耗时, R 树检索速度和树的深度成正比, R 树的每个结点对应一个磁盘页, R 树的高度每增加一层, 检索时对磁盘的访问至少增加一次。此多级混合索引中, 每个小 R 树涉及到结点数量要比大 R 树少得多, 因此其磁盘访问次数要比单纯的大 R 树少, 检索速度也相应快很多。

综上所述此索引具有较好的空间复杂性和较好的时间复杂性, 特别是当数据量比较大时, 具有一定的实用价值。

5 结束语

介绍了一种基于网格和 R 树的多级混合索引结构, 给出了其数据结构和算法描述, 并通过分析得出, 与网格索引和 R 树索引相比, 此多级混合索引可以在略大的空间开销的前提下, 换取更高的查找性能。

参考文献:

- [1] Guttman A. R-Trees: A Dynamic Index Structure for Spatial Searching[C]//Proc of ACM SIGMOD. Boston: ACM Press, 1984: 47-57.
- [2] Robinson J T. The K-D-B-tree: A Search Structure for Large Multidimensional Dynamic Indexes[C]//Proc of ACM SIGMOD. Boston: ACM Press, 1981: 10-18.
- [3] Freeston M. The BANG file: a new kind of grid file[C]//Proc of ACM SIGMOD. Boston: ACM Press, 1987: 260-269.
- [4] Beckmann N. The R^* -tree: An Efficient and Robust Access Method for Points and Rectangles[C]//Proc of ACM SIGMOD. Boston: ACM Press, 1990: 322-331.
- [5] 过志峰, 王宇翔, 杨崇俊. 空间数据索引与查询技术研究及其应用[J]. 计算机工程与应用, 2002, 38(23): 176-178.
- [6] Sellis T, Roussopoulos N, Faloutsos C. The R^+ -Tree: A dynamic index for multidimensional objects[C]//In: Proc 13th VLDB Conf. Brighton, England: Morgan Kaufmann Publishers Inc, 1987: 507-518.
- [7] Samet H. The Quad tree and Related Hierarchical Data Structures[J]. ACM Comput Surv, 1984, 16(2): 187-260.
- [8] 胡志勇. 空间数据库索引技术研究[D]. 武汉: 武汉大学, 2001.
- [9] 郭菁, 郭薇, 胡志勇. 大型 GIS 空间数据库的有效索引结构 QR-树[J]. 武汉大学学报: 信息科学版, 2003, 28(3): 306-310.
- [10] 肖伟器, 冯玉才, 缪勇武. 空间对象数据库网格索引机制[J]. 计算机学报, 1994, 17(10): 45-51.
- [11] 岳小平, 鞠时光, 李芷. 空间数据索引技术[J]. 计算机应用研究, 2002, 19(2): 32-34.