

文章编号: 1671-8860(2007)04-0347-03

文献标志码: A

# 三维 GIS 中基于空间拓扑约束条件的 R 树研究

陈 鹏<sup>1</sup> 孟令奎<sup>1</sup> 宋 杨<sup>1</sup>

(1 武汉大学遥感信息工程学院,武汉市珞喻路 129 号, 430079)

**摘要:** 在研究了 R 树的索引机制后,结合三维空间实体对象,提出了一种带有三维空间拓扑约束的 R 树结构,并依据空间拓扑关系制定了构建索引时的四条规则,在规则的约束下,建立该索引时能有效地减少 R 树的区域重叠,提高了三维空间的查询效率。

**关键词:** 空间索引; R 树; 三维数据模型; 三维空间拓扑关系; GIS

中图法分类号:P208

目前,对于二维的 GIS 系统,主要的空间索引技术可分为树索引、网格索引、线性映射索引等类型。对于三维空间数据的索引,由于三维数据模型不同,已有的三维空间索引差别也很大,如基于八叉树三维空间索引<sup>[1]</sup>、基于改进的静态 R 树三维索引<sup>[2]</sup>。前一种索引虽然结构简单,实现容易,但建立索引时必须预知空间对象的分布范围,不能满足空间数据的动态要求;后一种索引因采用的仍是 2.5 维的数据,不能应用于真正三维的数据环境,而其静态的 R 树结构又恰恰失去了 R 树的动态调整优势。本文结合三维空间实体对象,提出了一种基于空间拓扑约束的 R 树索引结构。

## 1 带拓扑约束条件的 R 树空间索引结构

### 1.1 三维空间实体间的空间拓扑关系

参考国际化组织所提出的空间数据的抽象数据类型(ADT)<sup>[3]</sup>,所有的三维 GIS 终端的空间数据可以归结为 8 种:三维空间点、多边形线段、圆的弧段、由弧段或多边形线段组成的线段、具有一个外边界的多边形、具有一个外边界和一个或多个内边界的多边形、具有一个外边界的体对象、具有一个外边界和一个或多个内边界的体对象。在这些基本的空间几何类型上,可定义不同的操作函数,封装成各种基本的空间对象。任意复杂的空间对象都可由这些基本的空间几何类型的聚合

与继承得到。

上述空间几何类型在三维欧氏空间  $R^3$  中可抽象为  $k$ -单纯形 ( $0 \leq k \leq 3$ )<sup>[4]</sup>。在三维空间中,任意两个  $k$ -单纯形  $A$  和  $B$  的边界 ( $\partial A, \partial B$ )、内部 ( $A^\circ, B^\circ$ ) 和余 ( $A^{-1}, B^{-1}$ ) 之间的交集构成的九元组可参见文献[5]。同时,文献[5]还定义了相邻、包含、相交、部分覆盖、相离、相等 6 种基本拓扑关系类型。这六种空间关系可涵盖三维空间对象  $A, B$  之间所有的拓扑关系。笔者认为,可以通过这六种空间拓扑关系对三维空间的每个三维实体进行空间划分,其中,具有相邻、包含、部分覆盖、相交这四种拓扑关系的空间对象具有一个共同的性质,即  $A \cap B \neq \emptyset$  ( $A, B$  分别为  $R^3$  空间的两个空间对象),具有以上四种空间类型的空间实体因具有相互接触的性质可以作为一个空间划分。相等关系的空间对象虽然也具有  $A \cap B \neq \emptyset$  的性质,但具有这样性质的空间对象在实际的系统中往往表现为同一三维对象的完全不同的属性,因而笔者认为,具有相等关系的所有空间对象应视为一个空间对象。对于相离关系,由于  $A \cap B = \emptyset$ ,故可容易理解为不同的空间划分。这样,每个空间对象就具有 6 种空间关系,具有外接立方体以及与其相交的空间对象列表,具体到每一个对象,再根据情况继续细分。

### 1.2 R 树与三维空间关系的结合

在设定上述空间拓扑关系与空间对象后,就可以利用空间拓扑关系作为约束条件来构建三维空间中的 R 树索引。R 树允许的外包范围重叠

对 R 树的性能影响非常大,无约束的重叠会导致 R 树的索引次数与存储次数大量增加,在高维空间中,情况会更严重。传统的提高 R 树的性能有优化范围矩形区域、重叠区域、空白区域和提高节点利用率四个方面。当 R 树索引上升到三维空间时,提高索引性能也需要从上述几个方面着手。本文中,笔者采用空间拓扑关系来进行约束,从而减少兄弟节点间的范围重叠程度,提高节点利用率。

在三维空间中,考虑到查询算法的简便,R 树的外包范围选择的是六面体。由于实际系统中的三维空间对象很复杂,在空间关系上有着相互接触的空间对象,其外包六面体不可避免地会产生重叠,甚至在空间关系上相离的空间对象,其外包范围也会覆盖,由此,在构建 R 树索引时,需确立如下约束规则。

1) 对于其拓扑关系存在  $A \cap B \neq \emptyset$  的空间对象,其公共三维尺度比率  $l$ (指空间对象在三维空间中不同坐标轴伸展度的比值)超过阈值  $R$  的,每一空间对象作为一个节点。

2) 对于其拓扑关系存在  $A \cap B \neq \emptyset$  的空间对象,且其共同的三维尺度比率  $l$  在阈值  $R$  的范围内,这些空间对象采用相同的外包六面体,即共用一个父节点(如图 1(a)所示),并采用链表的方式连接每个空间对象,以便遍历。

3) 对于单一空间对象,其三维尺度比率  $l$  超过阈值  $R$  的,在同一维度上( $x$ 、 $y$  或  $z$ )拆分成不同的空间对象(如图 1(c)所示),作为不同的独立节点。而比率  $l$  在阈值内的,则不作处理(如图 1(b)所示)。



图 1 三维空间对象外包六面体

Fig. 1 External Connection Cube of 3D Object

4) 特殊空间对象采用独立索引机制。这条规则主要是针对如地面、河流、道路等大型面状或线面状空间对象,因为这类地物覆盖的范围较广(本实验系统不包括这些特殊的空间对象)。

参照上述规则建立的 R 树索引,其树的平衡可能会被破坏,这是因为当遇到相互连接的多个空间地物(规则 2 特例)时,会产生超大节点,检索这样的节点时,性能也会有所下降,但由于出现该种情况的节点并不多见(通常这样的大型空间对象会因为三维尺度比率超过阈值而被拆分为多个小对象),故仍然可以接受。该规则的好处是能有效地减少节点间相互重叠的区域,并提高节点的

利用率,从而从整体上提高索引的效率。

### 1.3 带拓扑约束条件的 R 树的查找、插入、删除、分裂算法

#### 1.3.1 查找算法

设搜索区域为  $S$ ,从根节点  $R_{\text{Node}}$  开始,若  $R_{\text{Node}}$  不是叶节点,则判断该节点的各单元与搜索区域  $S$  的空间关系,若相交,则以该单元所指的节点为子树的根节点,重复上述操作。若  $R_{\text{Node}}$  为叶节点,则判断该节点的各空间对象与搜索区域空间的位置关系,如其空间对象所在的外包六面体与搜索区域相交,则返回(结果找到,查询结束)。

#### 1.3.2 插入算法

计算插入对象与周围空间对象的拓扑关系时,参照前面的 4 条规则,分为如下几种情况:

- ① 计算新插入物体的三维尺度比率  $l$ 。若符合规则 3,则将该物体拆分为具有相同属性值的不同空间对象,每一个对象作为一个单独节点插入。
- ② 若相交,找到第一个相交的空间对象,计算其外包六面体,并衡量三维尺度比率。如果符合阈值条件,则计算第二个与之相交的空间对象,循环下去,如果全部满足阈值条件,则整体作为一个独立节点插入,节点内部参照规则 2 的方式组织数据结构。如三维尺度比率超过阈值,则作为单独节点插入叶节点  $L_{\text{Node}}$  中。
- ③ 若分离,则作为单独节点插入。首先选择合适的子树(计算体积增加最小的子树) $L_{\text{Node}}$  存放空间对象,如果  $L_{\text{Node}}$  中有空位(每一个节点都有一定的存放空间来容纳各个单元,存放空间的数量由树的阶来控制,当存放空间没有被用完时,就存在空位),则在  $L_{\text{Node}}$  中记录新对象的外包六面体与 ID;如果没有空位,则  $L_{\text{Node}}$  需要进行单元拆分(保持 R 树的动态平衡),分别记录在节点  $L_{\text{Node}}$  与新分裂的节点  $LL_{\text{Node}}$  中,然后再调整树结构。

#### 1.3.3 删除算法

先运用搜索算法查找到要删除的空间对象,删除该对象,然后调整树结构。

树结构的调整有两种情况:① 因节点增加而导致的树调整。当插入新节点导致子树节点拆分时,则需要对索引树进行调整。设临时节点  $\text{Node}$  为叶节点  $L_{\text{Node}}$ , $N_{\text{Node}}$  为因新对象插入而分裂成的新节点。调整  $\text{Node}$  的父节点  $P_{\text{Node}}$  中的对应单元,使其包含  $\text{Node}$  的所有单元。在  $P_{\text{Node}}$  中寻找空单元,指向新节点  $N_{\text{Node}}$ ,使其包含  $N_{\text{Node}}$  的所有单元。如果父节点  $P_{\text{Node}}$  中无空位,则将父节点  $P_{\text{Node}}$  再进行拆分,重复上述步骤,直至树调整完成。树的调整可导致树的层数增加。② 因节点删减而导致的树调整。其过程相当于因节点增加导致树调整的逆过程。该调整可使树的子节点合

并,同时减少树的层数。

### 1.3.4 分裂算法

*R* 树的分裂算法在传统的二维空间中表现为如何将一个矩形集适当地分成更小的矩形集。本文采用 Guttman 给出的复杂度为  $M^2$  的平方耗费算法<sup>[6]</sup>,并在该算法的基础上作了部分修改,以适应前面的 4 条规则。

## 2 实验结果分析

本实验系统是基于 JOGL 开发的,实验系统的数据集管理采用文件与数据库混合的方式,即空间数据采用文件的方式存储(实验中的空间数据文件采用 3DS 文件存储),属性数据采用传统的关系数据库存储,空间数据与属性数据采用惟一 ID 的方式联系起来。为减轻系统压力,三维对象采用同一材质,没有使用纹理贴图。

对于实验证,由于没有现成的三维索引程序代码,故只能与传统的二维 *R* 树索引作对比,对比中,采用相同数目的空间对象比较两种索引的性能参数。从图 2 可以看到,笔者提出的带拓扑约束的三维 *R* 树索引与传统的二维 *R* 树索引相比,在存储量、构树时间以及查询时间上只是略有增加,而没有因空间维数的增加导致性能的大幅下降。这说明以拓扑约束规则建立的 *R* 树空间索引抑制了三维空间中索引区域的重叠,提高了节点的利用率,故该三维索引结构是有效的。

## 3 结语

由于现实三维空间中还没有一种三维数据结构能概括所有的三维空间对象类型,所以本文介绍的空间对象不可能适应所有的三维模型,如对于矿山这样的三维空间实体,其不同的属性空间对象相互纠缠,可导致本文提出的规则失效。另

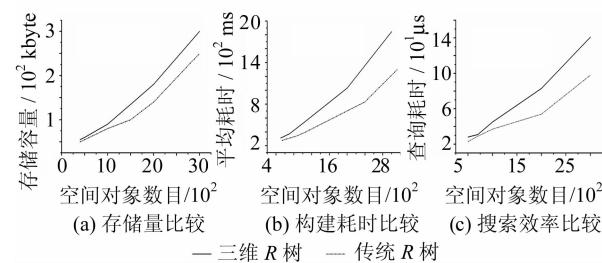


图 2 两种索引的比较

Fig. 2 Comparison of Two Indexes

外,对于具有海量数据性质的三维 GIS 系统,整体建立空间索引时必然导致构建索引时间的增加,效率下降。进一步的研究工作还有:①利用混合的三维数据模型来完善空间索引的拓扑约束规则。②对于海量数据的三维 GIS 系统,需要寻求一种空间划分的方法来减少空间索引规模,以解决因索引的无限膨胀而导致的效率低下问题。

## 参 考 文 献

- [1] 惠文化,郭新成. 三维 GIS 中的八叉树空间索引研究[J]. 测绘通报, 2003(1): 25-27
- [2] 宋扬,潘懋,朱雷. 三维 GIS 中 *R* 树索引研究[J]. 计算机工程与应用, 2004(14): 9-10
- [3] Specification and Description Language ICE Website. Web ProForum Tutorials[OL]. <http://www.iec.org/online/tutorials/sdl>, 2007
- [4] Armstrong M A. 基础拓扑学[M]. 孙以丰译. 北京:北京大学出版社,1983
- [5] 陈军,郭薇. 三维空间实体间拓扑关系的矩阵描述[J]. 武汉测绘科技大学学报, 1998,23(4):360-362
- [6] Guttman A. *R*-tree: a Dynamic Index Structure for Spatial Searching [M]. New York: ACM Press, 1984:45-57

第一作者简介:陈鹏,博士生。主要从事三维 GIS 方向的研究。  
E-mail: hellocp@163.com

## R-tree Structure Appended with Spatial Topology Restrictions in 3D GIS

CHEN Peng<sup>1</sup> MENG Lingkui<sup>1</sup> SONG Yang<sup>1</sup>

(1) School of Remote Sensing and Information Engineering, Wuhan University, 129 Luoyu Road, Wuhan 430079, China)

**Abstract:** After detailedly researching on the mechanism of *R*-trees and combining with the 3D entity object, a bran-new *R*-tree structure appended with topology restrictions in 3D space is presented, four rules are defined which enable to decrease the overlap of *R*-trees' area when constructing spatial indexes and enhance the query efficiency assuredly.

**Key words:** spatial index; *R*-tree; 3D data model; 3D spatial topology relationship; GIS