

文章编号:1001-1595(2011)02-0249-07

顾及多细节层次的三维 R 树索引扩展方法

龚俊¹, 朱庆², 张叶廷², 李晓明², 周东波²

1. 江西师范大学 鄱阳湖湿地与流域研究教育部重点实验室, 江西 南昌 330022; 2. 武汉大学 测绘遥感信息工程国家重点实验室, 湖北 武汉 430079

An Efficient 3D R-tree Extension Method Concerned with Levels of Detail

GONG Jun¹, ZHU Qing², ZHANG Yeting², LI Xiaoming², ZHOU Dongbo²

1. Key Laboratory of Poyang Lake Wetland and Watershed Research, Jiangxi Normal University, Nanchang 330022, China; 2. State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China

Abstract: Aiming at the critical issues of LOD models organization, an extended dynamic 3D R-tree structure concerned with LODs is presented. In the node-choosing process, a globally-optimising approach is adopted which is first bottom-up then top-down, and that an improved clustering algorithm based on k -medoids is applied to the node-splitting process. This method makes more even node size, more regular node shape, and less node overlap. Furthermore, a method of integrating R-tree and LOD is put forward based on such kind of good R-tree structure. The simulated experimental analysis results approve that this approach improves 3D query performance relative to existing ones and succeeds in integrating LOD models.

Key words: levels of detail(LOD); R-tree; spatial index; 3D GIS

摘要:为提高细节层次模型的管理效率,提出一种扩展多细节层次功能的三维 R 树索引方法,基于全局优化和三维聚类分析建立具有均衡结构的动态三维 R 树索引,并扩展了三维 R 树索引结构,在中间节点层次引入细节层次模型信息。设计实现了先自下而上、后自上而下全局搜索的节点选择算法和基于 k -medoids 聚类算法的节点分裂算法,保证节点尺寸均匀、形状规则以及重叠减少。通过仿真试验,证明本文方法能很大程度地提升多细节层次三维城市模型数据库的空间查询效率。

关键词:多细节层次;R 树;空间索引;三维 GIS

中图分类号:P208 文献标识码:A

基金项目:国家 863 计划(2008AA121600);国家 973 计划(2010CB731801);国家自然科学基金(41001222)

1 引言

随着地球空间数据获取手段如摄影测量、激光扫描、地质勘探等技术的发展,三维模型数据的获取精度和建模速度不断提高,使得数据量激增、数据类型多样。同时,为表达场景真实性,纹理数据、CAD 模型也被广泛采用,多细节层次已经成为真三维空间数据的重要特征之一。OGC 标准 CityGML 关于三维城市模型的多细节层次(levels of detail, LOD)有了专门的定义^[1]。已有的 GIS 空间索引常将 LOD 模型作为空间目标的附属信息,导致全景描述时需要遍历全部或大部分目标才能获取合适的细节层次模型,查询效率很低,因此只能缩小空间搜索范围或者不予表达,导致尺度视觉信息丢失。

R 树索引具有动态更新、深度平衡的特点,能自然扩展至三维空间,被公认为是最有前途的

真三维空间索引方法^[2-3]。文献[4]对三维空间进行网格划分,将网格单元作为 R 树叶节点构建 R 树,将 R 树的中间层节点包围盒作为尺度表达的细节层次。文献[5]提出另外一种 R 树索引和 LOD 结合的途径,聚合地缘和形状相近的地物,其中考虑了高程因素。文献[6]将多比例尺综合融入 R 树索引方法中,改进节点选择子算法,一定程度上减少节点重叠现象。三种方法均试图采用 R 树的天然层次结构实现目标查询和细节层次查询的双重功能。然而,由于 R 树扩展至三维空间后,节点重叠现象和兄弟节点尺寸不均匀的问题十分突出,使得 R 树集成 LOD 模型的实用性很差。另外,应用 R 树节点包围盒作为低细节层次描述,忽略单个目标的 LOD 描述需求,也不能满足可视化精度要求。

为此,本文提出基于全局优化和三维聚类分析的动态三维 R 树空间索引方法,并基于此顾及

LOD 管理扩展了三维 R 树索引结构。

2 方法原理和算法描述

动态 R 树生成算法也称为 R 树动态插入算法,将目标的索引元组依次插入到 R 树结构中,节点选择子过程挑选最优叶节点用于插入新目标,节点分裂子过程对插入目标后的上溢节点实施分裂操作,确保节点不上溢。该方法的实施步骤分为三个步骤。

(1) 节点选择。利用先自下而上、后自上而下的节点选择算法搜索叶节点以插入目标元组。该算法能够在全树范围内寻找最优的叶节点,避免节点重叠导致的选择失误问题,而且启用综合考虑节点重叠、覆盖和形状等因子的评价标准,保证插入目标后节点形状合理。

(2) 节点分裂。有条件地采用二变三的节点分裂算法,将上溢节点和重叠最严重的兄弟节点重组为三个小节点,其中采用三维空间聚类算法,启用综合考虑节点重叠、覆盖和形状等因子的分裂评价标准,使得分裂后兄弟节点形状规则、尺寸均匀和重叠减少。

(3) 顾及多细节层次功能。基于步骤 1 和步骤 2 生成的良好树形,对 R 树结构实施扩展以实现多细节层次功能,R 树中间层允许包含目标索引元组,采取科学合理的原则将重要目标自动地分配至 R 树的上层节点中。

2.1 全局优化的节点选择子算法

传统的节点选择方法采用自上而下的选择方式,即从根节点开始,选择插入目标后评价标准最优的子节点,逐级向下,直至到达叶节点。这种方式表面上很合理,实际上若以节点重叠或者节点覆盖增长量为评价标准,往往无法找到最佳的叶节点,因为如果目标落于兄弟节点重叠范围之内,目标插入任意一个重叠节点均不会影响评价标准,而节点重叠现象却是 R 树不可避免的问题^[2,7]。图 1 是新目标落于节点重叠区域,即传统方法失效的情况,当目标落于两个中间节点的重叠区域时,由于三维目标的插入操作不会改变任何一个节点的形状,因此无法判断最佳的插入节点。

最优的节点选择结果,应该是插入目标后从已选叶节点上溯根节点路径上各层节点的影响最小,如重叠体积、覆盖体积增长最小^[8]。根据 R 树的特性,当节点范围包含新目标时,插入操作不

会影响节点范围变化,也不会影响其父节点范围变化,因此寻找包含目标的节点最深层非常关键,进而寻找该层中包含目标的节点集合。集合中所有节点之子节点不再包含目标,从中选择一个子节点,要求插入目标后子节点的评价标准最优,然后依次向下寻找最佳子节点,直至叶节点,即为选择结果。

选择合适的判断标准至关重要,直接决定 R 树的树形质量。一般而言,选择覆盖范围和重叠范围作为判断标准比较常见。然而,本索引方法顾及多细节层次功能,要求各层节点的形状规则和尺寸均匀,因此,节点形状也是重要判断标准,要求节点形状趋近于立方体,即三个坐标轴方向上长度尽可能相等。本方法选择综合考虑覆盖体积、重叠体积和形状等三个因子,由于覆盖体积和重叠体积的量纲是长度单位的三次方,因此要选择一个相同量纲的评价因子来代替形状,本文提出一个三维柯西值的定义作为形状的评价因子。

三维柯西值:三维节点在三个坐标轴方向上的长度 X, Y, Z 的平均值三次方。

选择该评价因子的原因是存在如下柯西定理:

$$\forall X, Y, Z > 0, \exists \left(\frac{X+Y+Z}{3} \right)^3 \geq X \times Y \times Z$$

当且仅当 $X=Y=Z$ 时,等式成立。

假设 $X \times Y \times Z$ 的值一定时,当 $X=Y=Z$ 时, X, Y, Z 的平均值三次方最小,即在三维节点体积一定的条件下,当节点为立方体时,三维柯西值最小。因此,将三维柯西值作为节点形状的评价标准,有助于控制节点的形状,即其值越小,节点形状越趋近立方体。

通过大量实验测试发现,赋予三个因子相同权重,生成 R 树树形最为合理。因此,综合考虑节点重叠、节点覆盖和节点形状等三个因子,将评价标准设为, $Metric = \frac{1}{3}Overlap + \frac{1}{3}Overlay + \frac{1}{3}Shape$, 其中,假设节点的边长为 X, Y, Z , 则 $Overlap$ 是目标插入后节点与兄弟节点的重叠体积增加值; $Overlay$ 是插入后节点体积增加值; $Shape$ 是插入后节点三维柯西值的增加值。

节点选择环节的算法流程如下:

算法入口, R 树, 待插目标 T 。

算法出口, 选择的叶节点 L 。

(1) 假设 R 树当前层数是 N , 根节点为第 1

层,叶节点为第 N 层。从第 N 层开始,查找该层中完全包含 T 的节点集合。如果集合不为空,则令 $I=N$,进入(2);如果集合为空,则在第 $N-1$ 层查找。如此递推,直至根节点层,令根节点加入集合。如果上溯至第 J 层时集合不为空,则令 $I=J$,进入(2)。

(2) 在第 I 层的节点集合不为空,将节点集合中所有节点的子节点作为一个集合 C 。

(3) 将 T 分别插入集合 C 中的节点,计算节点插入目标元素后的综合评价指标变化值 $Metric$,选择其中综合评价指标变化值最小的节点 M 。

(4) 以 M 为新的根节点,将 T 分别插入 M 中的子节点,选择评价标准值 $Metric$ 最小的子节点 M' ,令 M' 为新的根节点,依次向下,直至叶节点层。

(5) 选择评价标准值 $Metric$ 最小的叶节点即为最终节点选择的结果,退出。

2.2 基于三维聚类分析的节点分裂子算法

R 树节点的扇出属性(fanout),即节点中子元组数目具有最小值和最大值,目标插入操作导致超过最大值时,上溢的节点必须分裂为小节点。节点分裂,可以视为一种约束条件下的三维空间聚类过程,因为分裂后的小节点也要满足扇出属性的限值要求。传统的方法是,将上溢节点一分为二,分裂原则是拆分后节点的覆盖或重叠范围的增加值最小。为了空间索引易于同多细节层次方法集成,分裂后节点形状趋近于立方体,覆盖范围、空间重叠和节点形状综合最优。

本算法有条件地采用二分为三的分裂原则,即当一个节点上溢时,搜寻与其重叠最严重的兄弟节点,将两个节点中的子元素分为三个小节点,增加了减少重叠和优化节点形状的机会,见图 2。如果没有与之重叠的兄弟节点,仍然采用一分为二的分裂方式。同时,为了使分裂后小节点趋近于立方体,需要为分裂制定合适的评价指标,分裂结果的评价指标最优即为最佳结果。前文采用的三维柯西值作为节点形状评价指标,有助于节点形状趋近于立方体,保持较好的空间聚集特性,节点分裂评价指标依然设为 $Metric = \frac{1}{3}Overlap + \frac{1}{3}Overlay + \frac{1}{3}Shape$,这里重叠、覆盖和形状因子被赋予同等权重,其作用就是保证三者综合最

优。在后续实验分析中,本文将通过大量实验结果来对比分析本算法的优越性。本算法采用改进的 k -medoids 算法来实现节点集合的分裂操作。

节点分裂环节的算法流程如下:

算法入口,上溢的叶节点 L_1 (包含刚插入的目标元组)。

算法出口,调整后的 R 树。

(1) 寻找同一父节点中与 L_1 重叠范围最大的兄弟节点 L_2 。

(2) 如果不存在重叠的兄弟节点 L_2 ,则将 L_1 中所有子元素作为集合 C_1 ,进入(2),准备将其分为两个子集合;如果存在重叠的兄弟节点 L_2 ,则将二者中所有子元素作为集合 C_1 ,进入(5),准备将其分为三个子集合。

(3) 从集合 C_1 中选出两个子元素,保证其构成范围的最小包围盒的空间覆盖范围和形状评价指标之和即($Overlay+Shape$)之值最大,分别作为两个子集合的种子元素。从剩余元素集合中选出一个元素,保证其分别插入两个子集合后,综合评价指标变化值 $Metric$ 最大。将该元素插入到综合评价指标变化值 $Metric$ 较小的子集合中。

(4) 按照(3)的方法,逐一将剩余元素插入到合适的子集合中。保证子集合中的元素数目满足 R 树节点的下限,即当剩余的元素数目刚好满足某个子集合的下限值时,将它们全部赋给该子集合。进入(12)。

(5) 从集合 C_1 中搜索两个子元素,保证其构成范围的最小包围盒的空间覆盖范围和形状评价指标之和即($Overlay+Shape$)之值最大,分别作为三个子集合中两个子集合的种子元素。从剩余元素中选出一个元素,保证插入上述某个子集合后综合评价指标变化值 $Metric$ 最小,则将该元素插入到该子集合中。

(6) 通过(5)中的方法,使得上述两个子集合中元素数目正好达到 R 树节点下限值。

(7) 剩余元素集合 $\{O_1, O_2, \dots, O_i, \dots, O_n\}$, 元素 O_i 的最小包围体为 $(x_{i_{\min}}, y_{i_{\min}}, z_{i_{\min}}, x_{i_{\max}}, y_{i_{\max}}, z_{i_{\max}})$, 其中, $x_{i_{\min}}, y_{i_{\min}}, z_{i_{\min}}$ 分别表示最小包围体的最小 x 坐标、最小 y 坐标和最小 z 坐标, $x_{i_{\max}}, y_{i_{\max}}, z_{i_{\max}}$ 分别表示最小包围体的最大 x 坐标、最大 y 坐标和最大 z 坐标。利用质心公式计算剩余元素的重心

$$\left\{ \begin{array}{l} X = \frac{\sum_{i=1}^n \frac{(x_{i_{\min}} + x_{i_{\max}})}{2} Area(O_i)}{\sum_{i=1}^n Area(O_i)} \\ Y = \frac{\sum_{i=1}^n \frac{(y_{i_{\min}} + y_{i_{\max}})}{2} Area(O_i)}{\sum_{i=1}^n Area(O_i)} \\ Z = \frac{\sum_{i=1}^n \frac{(z_{i_{\min}} + z_{i_{\max}})}{2} Area(O_i)}{\sum_{i=1}^n Area(O_i)} \end{array} \right.$$

式中, $Area(O_i) = (x_{i_{\max}} - x_{i_{\min}})(y_{i_{\max}} - y_{i_{\min}})(z_{i_{\max}} - z_{i_{\min}})$ 。如果剩余元素数目大于 R 树节点下限值, 从剩余元素集合中选出距离质心最远的元素, 并予以剔除, 返回(7); 如果等于 R 树节点下限值, 从剩余元素集合中选出距离质心最近的元素, 作为第三个子集合的种子元素, 进入(8)。

(8) 从没有加入任何一个分组的剩余元素集合中选出一个元素, 该元素插入第三个子集合后评价指标增加值为 W_1 , 插入另外两个子集合后两个综合评价指标变化值 $Metric$ 之较小值为 W_2 , 保证 $(W_2 - W_1)$ 最大。将该元素插入第三个子集合。

(9) 按照(8)的方法, 从剩余元素集合中依次选出合适元素插入第三个子集合, 使得第三个子集合中的元素数目达到 R 树下限值。

(10) 从剩余元素中选出一个元素, 分别插入三个子集合, 选择其中综合评价指标变化值 $Metric$ 最小之子集合, 将该元素插入该子集合中。

(11) 按照(10)的方法, 将剩余元素全部分配到子集合中。

(12) 如果分裂操作导致父节点也上溢, 则令父节点为 L_1 , 进入(1); 如果不上溢, 则进入(13)。

(13) 退出算法流程。

2.3 顾及多细节层次的 R 树扩展方法

多细节层次是空间目标在三维空间中的多尺度描述需要, 单个空间目标具有多份 LOD 模型并作为其附属信息是传统 LOD 模型管理方式^[9]。同时, 三维空间尺度变化中, 忽略次要目标, 保留重要目标, 是来源于制图综合思想的另一种细节层次描述方法^[6]。基于前述算法生成的良好树形结构, 本文提出一种顾及多细节层次的三维 R 树索引扩展方法, 将两种细节层次描述方式融入 R 树索引中, 实现细节层次查询任务。

采用 R 树的深度平衡层次结构扩展多细节层次功能, 与传统方法不同的是, 中间节点还以管理子节点的方式管理重要目标的索引元组, 图 3 形象地描述了本方法的原理。父节点从其子节点所管理的目标集合中选择数个重要目标, 目标数目限定为子节点的数目, 即从每个子节点中挑选一个最重要目标, 挑选依据可以根据应用需要灵活选择, 如目标包围盒的高度、体积或者投影面积等。依此类推, 树层次越高的节点管理重要性越大的目标索引元组。为了方便后续实现数据自适应调度, 每个节点可以记录所管理目标模型的数据量^[8-9]。对于单个目标具有多份细节层次模型的情况, 目标索引元组包含细节层次模型的索引信息, 即各个细节层次模型的标识号和作用范围。

以下给出本方法的节点和目标索引元组以及 LOD 索引信息的数据结构(伪 C++ 码描述)。

```
// * * * * * 单个目标中单个 LOD 模型的索引信息 * * * * * //
Structure LODINFO
{
    Integer iLODModelID;           // LOD 模型的标识号
    Double dNearDist;              // 该 LOD 模型最近作用范围
    Double dFarDist;              // 该 LOD 模型最远作用范围
}

// * * * * * R 树目标索引元组描述 * * * * * //
Structure RtreeTuple
{
    Integer iObjectID;            // 目标标识号
    Interval3D Interval;         // 该目标索引元组的最小包围盒
    Bool bHasLOD;                // 是否具有 LOD 模型信息
    Integer iNumLODLevels;       // LOD 模型层数
    Array<LODINFO> LODInfos;     // 目标中 LOD 索引信息数组
```

```

}
// * * * * * R 树节点描述 * * * * *
Structure RTreeNode
{
Interval3D Interval;           // 节点的最小包围盒
Bool bIsLeaf;                 // 是否是叶节点标志
Integer iNumChildren;        // 子节点的数目,也是目标索引元组的数目
Array<RTreeNode> pChildNodes; // 节点中子节点数组
Array<RtreeTuple> pTuples;    // 节点中目标索引元组数组,本方法允许中间节点管理目标索引元组
}

```

本文方法在中间节点中增加目标索引元组,其数目和子节点数目相同,因此单个节点数据量增加不足一倍,对整个 R 树的数据量影响有限,规范化的数据结构也易于转换为数据库模型。实际应用中,为 R 树的各层次设置一个细节层次描述范围。从 R 树根节点开始向下遍历,如果视点距离某 R 树节点的最小距离大于该节点层次描述范围的最远距离,则不描述该节点中包含的目标;如果最小距离小于节点层次描述范围的最远距离且大于描述范围的最近距离,则绘制该节点中包含的目标,如果目标具有 LOD 模型,则根据当前视距选择合适的 LOD 模型;如果最小距离小于节点层次描述范围的最近距离,则进入其子节点层继续上述判断过程,直至叶节点。通过改变 R 树各层次的细节层次描述范围,可以调整数据调度和可视化的负载量,因此本集成方法也易于同自适应数据调度和可视化方法相结合。

3 实验方法与测试结果

3.1 三维 R 树索引结构的测试方法和实验结果

为了测试节点分裂算法的有效性,本文开发了针对该算法的测试程序,采用多线程方法分解展现分裂过程中的各个步骤。其中主线程负责分裂过程的可视化功能,辅助线程负责控制分裂算法进度,分步展现节点分裂的中间过程有助于检验分裂结果的正确性和分析错误原因。图 4 是某个节点分裂过程的分解图。

同时,为定量证明本文方法生成树形结构的合理性,开发了树形结构测试平台。采用武汉大学测绘校区的三维模型数据进行测试,该数据样本包括 6 060 个目标,建立空间索引时没有分地物类处理,目标差异很大,有利于说明本方法的稳健性。图 5 是本方法建立的三维 R 树索引结构(扇出属性设为 4 和 10),仅显示树中三个层次的节点。

以叶节点的重叠体积、覆盖体积和节点形状为

统计数据(节点形状值为本文中的三维柯西值),同经典 R 树和 R * 树算法进行比对分析^[2,7]。实验结果见表 1,表中数据是所有叶节点的统计数据之和,可以清楚看到本文方法在重叠体积、覆盖体积和节点形状等三个方面均远远优于其他两种方法。

表 1 树形结构统计数据
Tab. 1 Statistics of tree structure /m³

方法名称	重叠体积	覆盖体积	节点形状值
Guttman 方法	14 088 749.29	75 518 738.73	847 279 196.60
R * 树	10 936 084.46	77 192 880.91	707 731 094.87
本文方法	2 502 644.16	12 721 218.63	122 310 440.17

3.2 三维空间查询效率的测试方法和实验结果

为了测试三维空间查询效率,将本文方法与经典 R 树和 R * 树算法进行比对分析。考虑到空间数据库实现 R 树索引时,每个 R 树索引节点对应一条记录或者一个数据页,查询过程中访问节点即访问数据页,公认是费时操作。一个查询过程中访问的 R 树节点越少,查询效率越高。因此,性能测试以节点访问数目作为查询性能评价指标,设置 R 树扇出参数为 4 和 10。将三种算法分别应用于 5 份随机数据集,数据集中的目标数目分别为 1 000、2 000、5 000、10 000 以及 20 000 个。为保证查询的可靠性,采用 1 000 个随机三维空间查询范围实施范围查询,取每次 R 树节点访问数目的平均值作为实验数据,测试结果见图 6。不难看出,本方法优于经典 R 树和 R * 树算法。

3.3 扩展多细节层次功能的测试方法和实验结果

为验证本文方法扩展细节层次功能的有效性,本文开发测试平台,利用交互式人机界面实时移动视点位置,记录各个位置所访问的目标数目和观察绘制效果。图 7 是在三个不同视距条件下的场景绘制效果,整个场景总共有 6 060 个目标,采用目标包围盒替代目标,(a)中绘制 169 个目标,(b)中绘制 562 个目标,(c)中绘制 1 545 个目标。

即使目标绘制数目相差很大,视觉感官的尺度差异性使得并没有造成明显的视觉误差,验证了本文方法在细节层次查询和多尺度描述方面的作用。

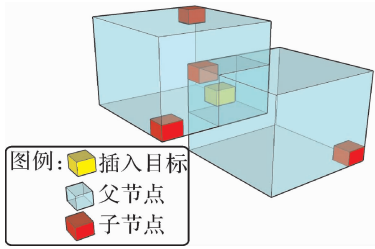


图1 传统节点选择方法失效的案例

Fig. 1 The case in which old node-choosing method cannot work correctly

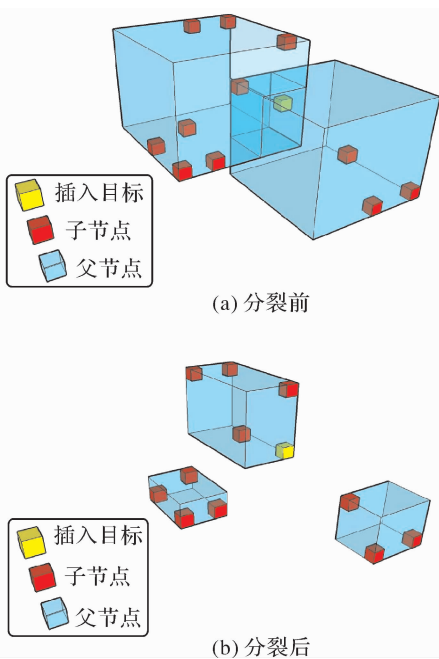


图2 二分为三的分裂方式

Fig. 2 The 2-to-3 splitting mode

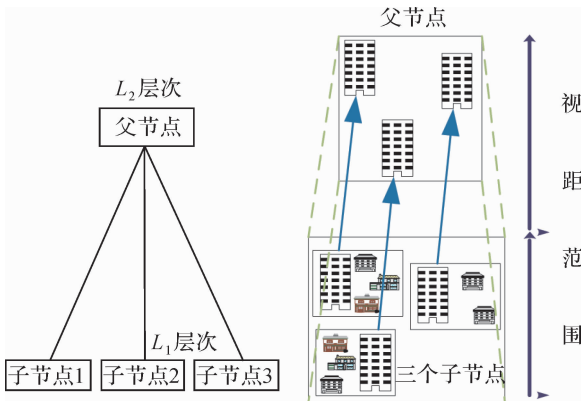


图3 R树结构和LOD模型的结合方法

Fig. 3 Integration of R-tree and LOD models

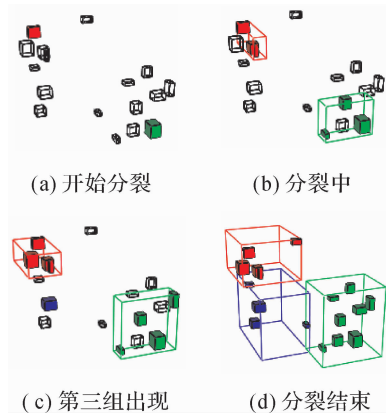


图4 某个节点分裂的分解过程

Fig. 4 Procedure illustration of 2-to-3 node-splitting

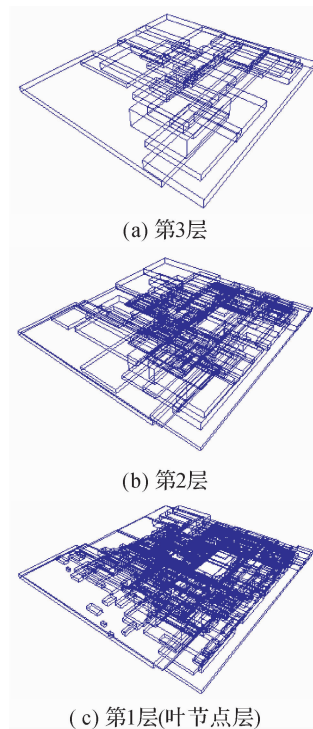


图5 本文方法生成的三维R树索引结构

Fig. 5 3D R-tree structure generated by this approach

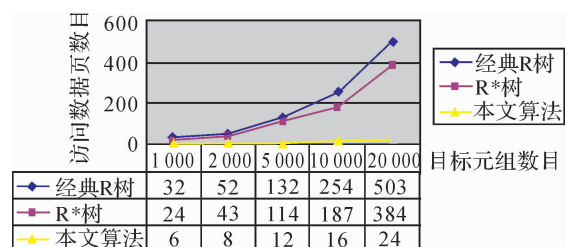


图6 三维R树算法查询效率对比

Fig. 6 Comparison of query performance of 3 different methods

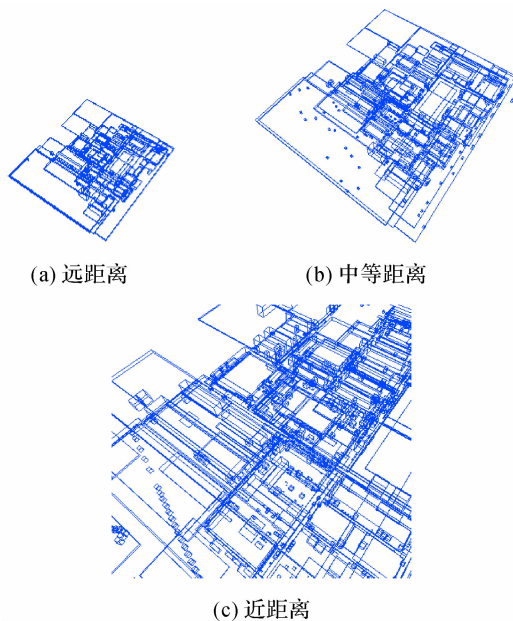


图7 基于R树索引结构的细节层次描述

Fig.7 LOD representation based on R-tree structure

4 结论

为解决R树索引扩展至真三维空间后节点重叠现象严重、形状不尽合理等问题,本文针对性地提出了全局优化的节点选择算法和基于三维聚类分析的节点分裂算法。通过多线程测试平台,步步还原节点分裂步骤,有效地检验了算法流程的正确性。与经典R树和R*树索引方法进行对比分析,实验结果表明本方法的R树树形结构合理,节点重叠现象大幅度减少,节点形状规则,明显改善三维空间查询性能。基于该R树的良好树形结构进行了扩展,提出了顾及LOD功能的三维R树索引扩展方法,实现目标查询和细节层次查询的双重任务。

下一步研究工作将聚焦到将本方法应用于室内建筑场景、三维地质环境等复杂应用中,优化改进本方法的鲁棒性和适用性^[10-11],同时研究其在分布式并行数据库架构中的应用^[12-13]。

参考文献:

- [1] Open Geospatial Consortium. OpenGIS CityGML Encoding Standard [EB/OL]. 2008-08-20 [2009-12-09]. <http://www.opengeospatial.org/standards/citygml>.
- [2] GUTTMAN A. R-trees: A Dynamic Index Structure for Spatial Searching[J]. ACM SIGMOD Record, 1984, 14(2): 47-54.
- [3] ZHU Q, GONG J, ZHANG Y. An Efficient 3D R-tree Spatial Index Method for Virtual Geographic Environments

- [J]. ISPRS Journal of Photogrammetry and Remote Sensing, 2007, 62(3): 217-224.
- [4] KOFLE M. R-trees for Visualizing and Organizing Large 3D GIS Databases [D]. Graz: Graz University of Technology, 1998.
- [5] ZLATANOVA S. 3D GIS for Urban Development [D]. Enschede: ITC, 2000.
- [6] LI Jun, JING Ning, SUN Maoyin. A Mechanism of Implementing Visualization with Level of Detail at Multi-scale [J]. Journal of Software, 2002, 13(10): 2037-2043. (李军, 景宁, 孙茂印. 多比例尺下细节层次可视化的实现机制[J]. 软件学报, 2002, 13(10): 2037-2043.)
- [7] BECHMANN N, KRIEGEL H P, SCHNEIDER R, et al. The R*-tree: An Efficient and Robust Access Method for Points and Rectangles [C] // Proceedings ACM SIGMOD Conference. New York: ACM, 1990: 322-331.
- [8] GONG J, KE S, BAO S. A Brand-new Node-choosing Algorithm in R-tree Spatial Index [C] // Proceedings of SPIE. Bellingham: SPIE, 2007: 1-10.
- [9] ZHENG Kun, LIU Xiuguo, YANG Hui. Research on Spatial Index Structure of LOD-OR Tree in 3D GIS [J]. Bulletin of Surveying and Mapping, 2005, (5): 72-74. (郑坤, 刘修国, 杨慧. 三维GIS中LOD-OR树空间索引结构的研究[J]. 测绘通报, 2005, (5): 72-74.)
- [10] GUO Wei, GUO Jing, HU Zhiyong. Spatial Database Index Techniques [M]. Shanghai: Shanghai Jiao Tong University Press, 2006. (郭薇, 郭菁, 胡志勇. 空间数据库索引技术 [M]. 上海: 上海交通大学出版社, 2006.)
- [11] MANOLOPOULOS Y, NANOPOULOS A, PAPAPOULOS A N, et al. R-trees: Theory and Applications [M]. Berlin: Springer-Verlag, 2005.
- [12] ZHANG Mingbo, LU Feng, SHEN Paiwei, et al. The Evolvement and Progress of R-tree Family [J]. Chinese Journal of Computers, 2005, 28(3): 289-230. (张明波, 陆锋, 申排伟, 等. R树家族的演变和发展[J]. 计算机学报, 2005, 28(3): 289-230.)
- [13] ZHOU Yan. Spatial Data Declustering Method and Parallel Spatial Index [D]. Wuhan: Wuhan University, 2008. (周艳. 空间数据分布方法与并行空间索引 [D]. 武汉: 武汉大学, 2008.)

(责任编辑:丛树平)

收稿日期: 2009-11-18

修回日期: 2010-02-26

第一作者简介: 龚俊(1978—),男,博士,副教授,硕士生导师,研究方向为多维地理信息系统和网络地理信息系统的理论与应用。

First author: GONG Jun(1978—), male, PhD, associate professor, post graduate adviser, majors in theory and application of multi-dimensional GIS and Web GIS.

E-mail: gongjunbox@gmail.com