基于空间索引的 DEM 数据质量检查*

陈小红[1]，潘 懋[1]，吴焕萍[1]，杨峻辉[2]，朱 雷[1]

(1. 北京大学 地球与空间科学学院，北京 100871；2. 江西理工大学，江西 赣州 341000)

摘 要：数字高程模型(DEM)数据主要来源之一是通过对现有地形图上的等高线等地貌信息进行数字化，生成计算机可识别的离散点或等高线。该方法对离散点或等高线数据质量要求较高。分别探讨了基于格网分块索引重复点检查和基于 R 树索引等高线相交检查两种方法。实验表明，空间索引的引入，大大提高了数据质量检查效率。

关键词：空间数据质量；离散点；等高线；DEM；空间索引

中图法分类号：TP335 **文献标识码**：A **文章编号**：1001-3695(2005)09-0028-03

DEM Data Quality Detecting Based on Spatial Index

CHEN Xiao-hong[1], PAN Mao[1], WU Huan-ping[1], YANG Jun-hui[2], ZHU Lei[1]

(1. School of Earth & Space Science, Peking University, Beijing 100871, China; 2. Jiangxi University of Science & Technology, Ganzhou Jiangxi 341000, China)

Abstract: To obtain data for generating Digital Elevation Model (DEM), one of the most important methods is to digitize physiognomic information such as contours on maps and get scattered points and contours which computer can recognize. This method requires high quality scattered points and contours. In this paper, two spatial index methods, GRID and R-tree, are introduced to detect the duplication of scattered points and the intersection of contours. The experiment shows that these two index methods greatly improve the efficiency of spatial data quality detecting.

Keywords: Spatial Data Quality; Scattered Points; Contour; DEM; Spatial Index

1 引言

目前，大规模生产数字高程模型(DEM)的方法主要有两种：①对现有地形图上的等高线等地貌信息进行数字化，生成计算机可识别的等高线或离散点，然后内插生成 DEM 数据。②以数字化的航摄影像、控制点为资料，采用数字摄影测量工作站自动生成 DEM。第一种方法因其充分利用已有的测绘成果速度快、成本低因而得到普遍采用[1]。

采用数字化方法，由于人为或其他原因，可能在图形、属性以及与其他要素关系等方面存在质量问题。等高线的质量问题主要有：①图形方面，如要素遗漏、位置跑线、等高线相交等；②属性方面，如等高线代码错误、高程值错误等；③与其他要素的关系方面，如等高线与相近的高程点矛盾，与河流、水体矛盾等[2]。离散点的质量问题主要有：①图形方面，如存在重复点；②属性方面，如高程值错误等。

对于上述 DEM 空间数据质量问题，如相交线段的判断，较为通用的方法是采用逐线扫描比较法。该方法随着线段数量的增加，比较的次数呈几何级数增加。因此，大数据量 DEM 空间数据质量检查是一个很值得研究的问题。本文讨论了基于空间索引的离散点和等高线数据质量检查，即采用格网分块索引对离散点重复点检查和基于 R 树索引对等高线相交检查。

收稿日期：2004-08-19；修返日期：2004-09-24

基金项目：国家“863”计划资助项目(2001AA135180)；教育部重点资助项目(99003)

2 空间索引技术

由于空间索引在整个空间数据库中的重要地位，一直是研究的重点[3]。空间索引大至可分为：格网分块索引、R 树索引以及二叉树索引系列。其中 R 树索引包括：R 树、R + 树、R^* 树、Hilbert R 树、SR 树等；较为典型的二叉树索引有：KD 树、K-D-B 树、LSD 树等。

格网分块索引的基本思想是根据离散点的空间密度分布将空间区域划分为若干正方形子块，并基于行、列值对子块 $B_{p,q}$ 进行索引[4,5]。R 树的基本思想是构造一棵由多层嵌套的外接矩形构成的平衡树，R 树索引中，R 树数据结构中每个非叶子节点代表一个矩形，该矩形为其子节点所代表矩形的最小包围矩形(MBR)。它的每一个非叶子节点包括其子节点和外包范围，同级节点的外包范围可能重叠[6]。

由于同级节点的外包范围存在重叠，使得 R 树的查询效率降低。一般通过优化范围矩形区域、重叠区域、空白区域和节点利用率这四个参数来提高 R 树的性能[7]。

3 基于格网分块索引的重复点检查

3.1 分块索引的建立

分块索引的建立可以分以下四步进行：

(1) 统计检查域内的离散点数目 P_n，根据最小外包矩形 MBR 求出该区域的面积 S，由此求出离散点的数据密度 $D = S / P_n$。

(2) 根据 MBR 内数据密度 D 以及子块内所要求具有的离散点平均数目 P_m 求出子块的宽度 *BlockSize* 以及网格分块的行列(P, Q) 数目, 如式(1) 所示, 其中 $\alpha = \sqrt{p_m}$。可以看出, 它的大小决定了子块的宽度和网格分块的行列数。

$$BlockSize = \sqrt{D \cdot P_m} = \alpha\sqrt{D}$$
$$P = \lfloor yRange / BlockSize \rfloor + 1 \qquad (1)$$
$$Q = \lfloor xRange / BlockSize \rfloor + 1$$

(3) 新建一个 $P \times Q$ 大小的数组 pBlock, 初始值为 - 1, 它记录落入每个子块中离散点的最小 ID 号; 再新建一个 P_n 大小的数组 pNext, 用来记录其他离散点的 ID 号。两者的数据类型均为整型。

(4) 遍历 MBR 内离散点, 判别离散点所在的子块 $B_{p,q}$, 并将离散点的 ID 号记录在数组 pBlock 和数组 pNext 中去。对于子块边界上的点, 规定每一子块包括落在其左边界和下边界上的点, 但不包括落在其右边界和上边界上的点。算法描述如下:

```
for i = Pn - 1, Pn - 2, … , 1, 0
{    p = x[ i] / BlockSize;
     q = y[ i] / BlockSize;
     Index = pBlock[ p, q] ;
     if ( Index = = - 1)  pNext[ i] = i;
     else pNext[ i] = Index;
     pBlock[ p, q] = i;
}
```

图 1 简单示意了采用上述算法进行分块索引后, pBlock 和 pNext 所记录的离散点 ID 号的情况。

图 1 中 DEM 的数据范围内的六例个离散点被分成了 2 × 2 个正方形子块。Piegl 等提到在每一个子块内可以使用动态链表[4], 但缺点是每次插入节点时均需动态申请内存, 索引使用完后, 也只能逐个释放节点空间, 因此耗时较大。而该方法采用一次性申请内存, 并通过两个数组来维护索引关系, 最后再一次性释放, 效率得到了很大的提高。

3.2　重复点的检查过程

网格分块索引建立后, 重复点的检查即已限定在某一子块中进行。在检查某一子块时, 取子块内的第一点作为参考点, 将其与子块内其他点按给定限差进行比较, 找出重复点。遍历子块内所有离散点的搜索算法描述如下:

```
NextID <-- pBlock[ p, q] ;
if NextID = -1 Stop //没有重复点
do{ … // 其他操作
     CurrentID = NextID ;
     NextID = pNext[ CurrentID] ;
}
while NextID = CurrentID
```

4　基于 R 树索引的等高线相交检查

4.1　R 树索引的建立

R 树是一种基于对象划分的空间索引数据结构, 描述为: ①R 树是 n 叉树, n 称为 R 树的扇(fan) ; ②每个节点对应一个矩形; ③叶子节点上包含了小于等于 n 的对象, 其对应的矩形为所有对象的外包矩形; ④非叶子节点的矩形为所有子节点矩形的 MBR。R 树节点的相关数据结构如下:

```
struct NODE
{    int nCount;
     int nLevel;
     BRANCH branch[ nMax] ;
}
struct BRANCH
{    RECT rect;
     struct NODE * child;
}
```

其中 RECT 为空间对象的 MBR; *n*Level 为节点所在的层次, 其值为 0 则表示是叶子节点; *n*Max 为 R 树扇的最大数目。

建立 R 树索引的过程同时是构建 R 树并优化的过程, 步骤如下:

(1) 计算待检区域内等高线的 MBR。

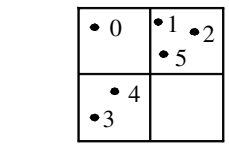
(2) 建立根节点 Root。

(3) 插入节点。在节点插入时, 如果不需要分裂节点, 则在插入后节点的外包范围增加最小的分支节点处直接插入记录即可; 如果需要分裂节点, 则选择两个分离的空间对象作为新的节点。节点插入算法具体描述如下:

```
if ( pNode- > nLevel = = nLevel)
{    AddBranch( ) ;  //直接加入节点
}
else if ( pNode- > nLevel  >  nLevel)
{    SplitNode ( ) ;  //分裂节点
}
```

(4) 分裂节点。将第(2) 步中所插入的新节点作为根节点, 插入余下的节点。新插入的节点将采用如下策略进行分裂节点以优化 R 树: ①在插入后外包范围增加最小的节点处加入节点; ②在插入后外包范围更小的节点处加入节点; ③在插入后记录数更少的节点处加入节点。

图 2 简单示意了采用上述算法进行索引并分裂节点后, 等高线 ID 在 R 树的分布情况。节点分裂后, 图中等高线的两个矩形区域 R_1 和 R_2, 分别对应 R 树的两个子节点。



pBlock[0,0]=3　pNext[0]=0
pBlock[0,1]=-1　pNext[1]=2
pBlock[1,0]=0　pNext[2]=3
pBlock[1,1]=1　pNext[3]=4
pNext[4]=4

图 1　分块索引建立示例

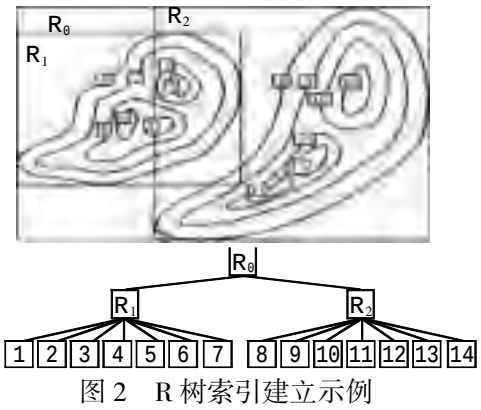


图 2　R 树索引建立示例

4.2　等高线相交检查过程

R 树索引建立后, 采用如下算法即可对所有等高线相交进行有效的判断。

(1) 计算待判断等高线的最小外包矩形 CoutourRect。

(2) 以 CoutourRect 作为搜索 R 树的 MBR, 得到包含在 CoutourRect 内或与 CoutourRect 相交的所有等高线, 将等高线 ID 记录在数组 pIndex 中。搜索 R 树算法描述如下:

```
Search ( &root, &CoutourRect)
{ if( pNode- > nLevel > 0)  //非叶子节点
  { for i = 0, 1 … nMax
    { if Overlap ( CoutourRect, pNode- > branch [ i] . rect)  Search
( &branch[ i] . child, &CoutourRect) ;
    }
  }
else if( pNode- > nLevel = = 0)  //叶子节点
{ for i = 0 , 1 … nMax
```

```
        { if Overlap( CoutourRect, pNode-> branch[ i]. rect) return pNode-> branch[ i]. child;
        }
    }
    }
```

(3) 待判断等高线与保存在数组 pIndex 中的等高线分别相交判断。

(4) 重复上述过程,直至所有等高线检查完毕。

判断线段相交传统的方法是通过直接解二元一次方程组求得,Frankling 提出了利用栅格化方法缩小搜索范围,减少盲目判断,提高运算速度的方法[8]。这两种方法各有弊端,前者运算量大,速度慢;后者算法复杂。有的作者提出了采用混合积求交的方法判断线段相交[9]。本文采用包络矩形法,先判断等高线的包络矩形是否相交。该方法简单且减少判断范围。

5 实验分析

本文提出的基于格网索引的重复点算法和基于 R 树索引的等高线相交检查算法,在VC++编程中实现,并与逐点(线)比较的扫描算法进行时间效率的比较,见图 3 和图 4。网格索引试验的时间包括建立索引的时间和扫描检查重复点的时间,R 树索引试验的时间包括建立索引和等高线相交判断的时间。实验环境的微机为 PⅢ550Hz、128MB 内存配置,操作系统为 Windows2000(SP3)。

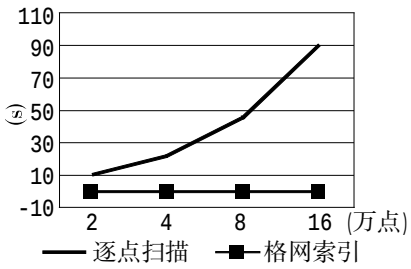


图 3 基于网络索引时间效率

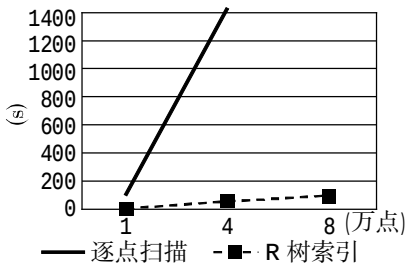


图 4 基于 R 树索引时间效率

从图 3 中可以看出,基于网格索引的重复点的算法的时间效率是非常明显的。当离散点的数量达到 16 万个点时,基于网格算法的时间在秒级,约 0.78s;而逐点扫描算法的时间上升到接近分钟级,约 89s。

从图 4 中可看出,基于 R 树索引的线段等高线相交检查的时间效率是非常明显的。当等高线的数量达到 4 万条时,逐线扫描算法的时间已经很难容忍,时间为 1500.5s 约合 25min;而基于 R 树索引的算法的时间维持在秒级,约 21.5s。

6 结束语

本文讨论了基于格网分块索引对离散点重复点检查和基于 R 树索引的等高线相交检查,并对效率进行试验分析。结果表明,格网分块索引和 R 树索引的引入,大大提高了重复点检查和等高线相交检查的时间效率,尤其在大数据量的条件下,效果尤为显著。文中提出的两种方法在 Visual C++ 6.0 编译环境下调试通过,并作为海量 DEM 生成系统的数据质量检查模块得到了成功的应用。

参考文献:

[1] 刘建军,王东华,商瑶玲.数字化等高线质量的软件自动检查方法[J].测绘科学,2001,26(4):36-39.

[2] 华慧,刘大杰.地图数字化空间数据的质量控制[J].城市勘测,1997,(4):9-13.

[3] 顾军,吴长彬.常用空间索引技术的分析[J].微型电脑应用,2001,17(12):40-43.

[4] Les A Piegl, Wayne Tiller. Algorithm for Finding All k Nearest Neighbors[J]. Compuer-Aided Design, 2002, 34(2): 167-172.

[5] Robert J, Renka. Multivariate Interpolation of Large Sets of Scattered Data[J]. ACM Transactions on Mathematical Software, 1988, 14(2):139-148.

[6] A Guttman. R-tree: Dynamic Index Stucture for Spatial Searching [C]. Proc. ACM-SIGMOD. International Conference on Management of Data, Boston, 1984, 47-57.

[7] Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, *et al*. The R* -tree: An Efficient and Robust Access Method for Points and Rectangles[C]. Proc. ACM SIGMOD Conf. on Management of Data, 1990.322-331.

[8] 郭仁忠.空间分析[M].武汉:武汉测绘科技大学出版社,2000.207-209.

[9] 李永红,华一新.一种快速判断线段相交的方法[J].测绘通报,2003,(7):30-31.

作者简介:

陈小红(1976-),男,硕士生,研究方向为 GIS 和数据库应用开发;潘懋(1954-),男,教授,博士生导师,研究方向为 GIS 和信息地质;吴焕萍(1977-),男,博士,研究方向为 GIS;杨峻辉(1980-),男,学士,研究方向为信息管理;朱雷(1973-),男,博士,研究方向为 GIS.

(上接第 27 页)

4 总结

本文阐述了 KVM 在基于构件的智能手机操作系统 ElaME 上的移植过程,分析了工作原理和移植中的实现方式。随着 Java 技术的发展,将会有更多 Java 应用程序运行在智能手机上,这一切都依赖于虚拟机 KVM。由于 KVM 是利用 CAR 构件和 ElaME 交互的,因而性能上得到了很大的保证。相信今后通过完善系统构件的功能,能够更好地支持 KVM 的运行。

参考文献:

[1] hen Rong. The Application of Middleware Technology in Embedded OS[R]. Hangzhou: Workshop on Embedded System, In Conjunction with the ICYCS (6th), 2001. 1-3.

[2] CLDC Reference Implementation version 1.1 [EB/OL]. http://Java.sun.com/products/cldc, 2004-06-01.

[3] MIDP Reference Implementation version 2.0 FCS [EB/OL]. http://Java.sun.com/products/midp, 2004-06-01.

[4] 潘爱民.COM 原理与应用[M].北京:清华大学出版社,1999. 9-19.

[5] 探矽工作室.深入嵌入式 Java 虚拟机[M].北京:中国铁道出版社,2003.283-329.

作者简介:

叶磊(1980-),男,湖北武汉人,硕士,主要研究方向为操作系统、计算机网络;陈榕(1957-),男,北京人,上海科泰世纪技术有限公司首席科学家,博士,主要研究方向为因特网操作系统、中间件技术、CAR 技术;赵岳松(1946-),男,上海人,副教授,主要研究方向为分布式操作系统、计算机网络。