

文章编号:1671-8860(2007)02-0131-04

文献标志码:A

# 基于区间树索引的等高线提取算法

王 涛<sup>1</sup> 毋河海<sup>2</sup> 刘纪平<sup>1</sup>

(1 中国测绘科学研究院,北京市北太平路16号,100039)

(2 武汉大学资源与环境科学学院,武汉市珞喻路129号,430079)

**摘要:**重新设计了从高程格网中提取等高线过程中的遍历策略,以保证提取结果具有统一的方向;针对日益增长的高程格网数据量,提出了基于区间树索引来查找等高线起点的算法。

**关键词:**等高线;高程格网;提取算法;区间树

中图法分类号:P283.7; P231.5

格网DEM和等高线作为地形表达的两种重要方式,各有独特的优势,二者之间的相互转换是计算机制图和地理信息系统的基本功能之一。从格网DEM中内插提取等高线的算法较多<sup>[1-3]</sup>,但随着各种地形数据获取手段的改进,传统的从地形等高线内插生成格网DEM的状况发生了极大的改变,直接获取的格网DEM数据在数量和质量上都得到了极大的提高,从格网DEM中提取等高线将会成为算法的常用功能,因此迫切需要高效率的等高线提取算法。

## 1 格网DEM中的等高线提取

### 1.1 算法的基本过程

首先,在格网中确定待内插等高线的起点;然后,从起点出发确定下一个等高点并连接成等高线,直至边界或者回到出发点。

### 1.2 顾及方向的策略

在数字环境下,等高线的方向是表达地形的基本要素之一,各种基于等高线的数字地形分析算法要求等高线在全区域内具有固定一致的方向(左高右低或者左低右高),如地性线的自动提取<sup>[4]</sup>。以往等高线提取算法没有顾及这个问题,必须经过后续处理才能使得等高线具有统一的方向<sup>[5]</sup>。

为了使提取的等高线自动具有一致化的方

向,需要对现有算法中起点确定和遍历方向进行修正。假设高程网格左上角为原点,向右、向下分别为x轴、y轴正向,在网格的x轴方向上寻找闭合曲线起始单元边时,简化判断条件,将原始条件的 $(Z - Z_1) \times (Z - Z_2) < 0$  修改为 $Z > Z_1$ 且 $Z < Z_2$ ( $Z_1$ 在 $Z_2$ 左侧),在确定下一点时只考虑y轴正方向(向下)。如图1中,当提取高程为8的闭曲线时,只考虑7-10和6-11两条边,若选前者作为起点,其后续点将考虑6-11,而不是垂直边7-9。分析高程格网中等高线的特点可知,闭曲线至少经过一条左高右低的网格边,因此,这一简单的修改不但不会影响最终结果,而且保证了等高线在任意正负地貌表达上的左高右低。基于同样的原则,在边界上提取开曲线时可以作类似的修改以得到方向一致的结果<sup>[6]</sup>。该策略在简化判断条件、提高效率的同时,还为进一步引入索引结构

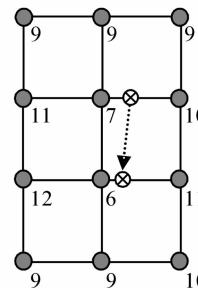


图1 有向等高线的提取

Fig. 1 Extraction of Directed Contour

收稿日期:2006-10-25。

项目来源:地理空间信息工程国家测绘局重点实验室开放研究基金资助项目(B2526,2534);辽宁工程技术大学地理空间信息技术与应用实验室开放研究基金资助项目(2006003)。

提供了可能。

## 2 基于区间树索引的算法

使用§1的简化条件确定起始点来提取某一特定高程等高线时,可能的起始网格边比以往算法可以降低一半左右,不过这种方法的平均运行效率仍旧是 $O(n)$ ,当等高线所经过的格网单元数相对很少或者分布零散时,效率还不高。

在高程维上,新的起始网格边是一个由左到右的增序区间。确定等高线起点的操作相当于在该起始边集合中找到一个合适的区间元素,这样就可以引入区间树索引来提高该步骤的效率<sup>[11]</sup>。

作为一维空间上的索引,区间树是一种有序二叉树,它可以提高二维平面上矩形相交探测的效率<sup>[3]</sup>,每个节点上记录多个区间元素,其上的每个区间左右端点分处该节点关键值左右,而其左子树上节点所含区间元素的左右端点值都小于父节点关键值,右子树上节点所含区间元素相反。区间树索引的对象也即节点元素是数值区间,因此在构建和查询等操作上与普通的二叉树有较大的差异。区间树的构建和查询效率强烈依赖于各级节点的关键字(分割值)的设定,即便是相同的关键字,如果给定的顺序不同,对树结构也有较大影响,因此在构建区间树的预处理过程中,应当选择合适关键字,以充分顾及树的平衡程度、深度和各节点中元素个数等三方面因素的均衡。

### 2.1 区间树索引的构建

最简单而快速的构建区间树索引方法是按照高程范围逐级取平均值作为关键字,但是这种方法显然不可能使得上述3个因素在树结构中得到均衡。最好的方法是在构建树之前对所有的可能起始单元边按照左端点(或者右端点)高程进行排序,然后逐层确定单元边数目的中位数,将该位置的高程作为各级节点的关键字,但是对于大数据量的高程格网而言,精确的排序过程将会使得构建过程显得效率低下。本文采用了直方图累积值的分段来逼近这一结果,并据此得到一系列关键字,然后构建一棵空树,最后将可能的起始单元边填充进去。具体的构建过程如下。

1) 直方图统计。对可能的 $x$ 单元边左端点高程(或者右端点高程,或者左右端点高程平均值)进行直方图统计,得到各高程的分布情况,记录数组HgtHistogram[]。实验中用的是起始单元边左右端点高程平均值。

2) 计算关键字序列。给定一个节点负载最

大值 $M$ ,顺序对HgtHistogram[]进行累积,当累积和达到 $M$ 时,记录对应的高程值作为一个分割值,将累积值置零,继续累积直至HgtHistogram[]的最后一个非零元素,得到用作树节点关键字的分割值序列Separator[]。如果HgtHistogram[]的总累积量没有超过 $M$ ,则说明可能的起始单元数量较少,此时用所统计的高程值作为分割值序列即可。

3) 构建空树。分割值序列Separator[]是递增的,可以把它视为一个已经构建起来的二叉搜索树中序遍历的结果,那么现在要解决的是它的反问题,构建树的过程是利用二分的思想对Separator[]数组进行递归地生成各级树节点,并记录节点之间的父子关系,具体实现与线化简的Douglas-Peucker算法非常相似。

4) 填充节点元素。假设 $I$ 是可能的 $x$ 单元边集合 $\{i_0, i_1, \dots, i_N\}$ ,它包含的每个元素都是一个区间,即 $i_k = [b_k, e_k]$ ,这里 $b_k < e_k$ ,且 $b_k \in \mathbf{R}, e_k \in \mathbf{R}$ , $\mathbf{R}$ 为实数集。当 $i_k$ 插入到树中时,如果 $b_k$ 大于当前树节点关键字,则取节点的右子节点作比较,反之取左子节点;如果 $b_k$ 小于某节点关键字而 $e_k$ 大于该关键字,将 $i_k$ 记入该节点,每个单元边只存放一个节点。

5) 节点内排序。标准区间树中每个节点上的元素集合存放左端增序和右端减序两个序列,由于可能的起始单元边数量通常较大,因此本文的具体实现中只对各单元边按其左端点进行增序排列,这可以节约一半的空间,当然时间效率会稍低。

图2是标准区间树示例, $I = \{a, b, c, d, e, f, g, h, i\}$ ,首先取5为关键字构造根节点,并且记录包含5的区间元素 $d$ 和 $e$ ,然后对它们按照左端点增序排列得到 $(d, e)$ ,按照右端点降序排列得到 $(e, d)$ ,对剩余的元素递归执行上述操作。对于 $N$ 个元素的 $I$ ,区间树的构建需花费 $O(N \log N)$ 时间并需要 $O(N)$ 的空间<sup>[3,11]</sup>,本文没有记录降序列表,所需空间降低一半。

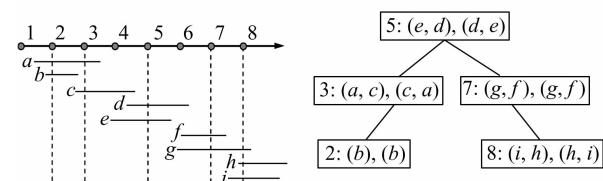


图2 区间树的构建以及查询示例

Fig. 2 Construction and Query of Interval Tree

### 2.2 区间树的查询

在区间树中查询包含给定数值 $x$ 的所有区

间元素的过程与二叉树的相关操作过程相似。首先将根节点作为当前节点,在该节点记录的增序列表中使用二分法查找输出包含  $x$  的区间元素。如果  $x$  大于(小于)该节点的关键字,将当前节点的右(左)孩子节点赋为当前节点;如果当前节点有效则递归执行以上操作,若为空则查询完毕。当区间树比较平衡时,它的查询效率为  $O(\log N + k)$ ,其中  $N$  为区间元素总数,  $k$  是包含查询点的区间元素个数。在图 2 的示例中,当查找包含 4 的区间元素时,4 小于根节点关键字 5,在根节点左端点序列中顺序检测各元素,得到  $d$ ,然后转至左孩子节点;4 大于该节点关键字 3,则在其右端点序列中查找,得到  $c$ ,之后转至其右孩子节点,由于该节点无效,操作结束。

### 2.3 在区间树中等高线起点的确定

当需要内插给定高程等高线时,对于开曲线仍需要遍历高程格网的 4 个边界;对于闭曲线,根据给定高程值,利用前文的查询算法到区间树中确定可能的起始单元边。由于这里构建的区间树

节点上只存放了左端点的增序列表,因此无论高程值比当前节点关键字大还是小,都需要从该列表中查找出包含该值的所有区间,并将每个区间作为起始单元边向  $y$  轴正向追踪直至回到起点。当高程值与当前节点关键字相等时无需深入到左右子树。

## 3 实验与分析

本文使用 Visual Basic6.0 实现了常规无索引算法和基于区间树索引的算法,实验使用了 3 个格网高程数据,表 1 为相应的数据特征。

表 2 为没有索引的算法和不同  $M$  值的区间树索引算法的效率比较,在 3 个数据中分别用 50 m 和 100 m 的等高距提取等高线。区间树方法对应的数字中,反斜杠之前为程序第一次耗费时间,它包括区间树的构建和等高线提取,之后为 3 次之平均时间(仅为等高线提取)。可以看出,在建立索引之后,等高线的提取效率得到了很大提高。

表 1 3 个数据集的基本情况及等高距为 50 m 和 100 m 时使用无索引方法提取的结果

Tab. 1 Characteristics of Three Testing Datasets and Contour Lines of 50 m and 100 m

	格网大小	高程范围/m	50 m 等高线		100 m 等高线	
			层	点数	层	点数
数据 A	1 201×1 201	1 732~5 757	81 层	4 202 根 1 386 482 点	40 层	2 111 根 693 133 点
数据 B	1 201×1 201	791~1 568	16 层	13 781 根 959 326 点	8 层	6 789 根 478 911 点
数据 C	2 402×3 603	16~3 059	61 层	40 437 根 3 990 461 点	30 层	18 352 根 1 977 028 点

注:由于实验中在交叉网格处选择方向时使用了全局最短连接,没有结合固定方向,因此使用带索引的算法所提取的结果可能会稍有不同<sup>[6]</sup>。

表 2 无索引和区间树索引算法效率/s

Tab. 2 Efficiency of Algorithms Without Index and with Interval Tree Index

	数据 A		数据 B		数据 C	
	100 m	50 m	100 m	50 m	100 m	50 m
无索引	1.51	3.0	0.41	0.79	6.58	13.37
区 M=10	1.98/0.46	2.53/0.92	2.17/0.24	2.47/0.48	20.65/1.71	21.98/3.57
间 M=1 000	1.91/0.48	2.65/0.95	2.12/0.25	2.39/0.51	19.78/1.89	21.57/3.80
树 M=5 000	1.84/0.51	2.18/0.98	2.15/0.26	2.40/0.54	19.11/2.00	19.59/4.14

当一个高程格网的区间树建立完成后,提取任意等高距时都可以使用它,不过,表 2 中为了检验区间树建立花费的时间是否稳定,在提取不同等高距的等高线时都重新构建树结构。虽然表 2 中记录的时间不是严格获得的,但是基本反映了相对的大小关系。从表 2 中的数据可以看出,高程格网规模越大,构建区间树所要的时间就越长,而当  $M$  取值越小时,这个时间也越长。如果区间树已经构建好,提取等高线的效率比没有索引时显著提高,并且随着  $M$  的降低该效率会有所提高。需要注意的是,本文方法中  $M$  值并不代表每个节点上负载元素的数量,这里没有对  $M$  的最优取值进行估计。当然如果允许更长的预处理时

间,最好的方法是在构建树结构之前对可能的起始单元边排序,这必然会使得各节点负载量也一样均衡,提高后续查询的效率。

表 1 所示的 3 个数据集合中,数据 B 与数据 A 在大小上相同,而前者格网点高程的跨度、提取等高线的总点数远小于第一个数据集,但是,所提取出的等高线条数远大于第一个,这就说明该数据中起始单元边比例相对较高。表 2 的数据也说明了为该类型的高程格网构建区间树要比地形单元完整的第一个数据集花费更多的时间,相对于无索引算法而言,效率提高的程度也不如其他类型数据的大。

## 4 结语

本文基于简单的左高右低原则提出调整处理遍历方向和选择带方向的起始单元边方案,使提取出的等高线具有一致化的方向,为进一步构建等高线空间邻近关系提供了基础信息。更为重要的是,该原则为选择合适的单元边构建索引来加速起始单元边查找的算法实现提供了可能。

对于大数据量的格网高程模型而言,提高执行的时间效率是非常重要的。本文提出的方法显然是典型的空间换时间的策略,当原始格网矩阵很大时,存储索引也不可避免地会消耗很大空间,常规的区间树还要为左右端点存储两个有序列表,因此需要寻求其他的策略来降低这一要求。

## 参考文献

- [1] �毋河海,龚健雅. 地理信息系统(GIS)空间数据结构与处理技术[M]. 北京:测绘出版社, 1997
- [2] 徐庆荣. 计算机地图制图原理[M]. 武汉:武汉测绘科技大学出版社, 1997
- [3] 普雷帕拉塔,沙莫斯. 计算几何导论[M]. 庄心谷,译. 北京:科学出版社, 1990
- [4] 艾廷华,祝国瑞,张根寿. 基于 Delaunay 三角网模型的等高线地形特征提取及谷地树结构化组织[J]. 遥感学报, 2003, 7(4): 292-298
- [5] 王涛,毋河海. 等高线拓扑关系的构建以及应用[J]. 武汉大学学报·信息科学版, 2004, 29(5):

438-442

- [6] 王涛,毋河海. 一种从格网 DEM 中提取等高线的算法[C]. 2004 年地图学与地理信息系统年会, 福州, 2004
- [7] 王涛,毋河海. SRTM 高程数据中空缺单元的内插填补[C]. 2004 年地图学与地理信息系统年会, 福州, 2004
- [8] Paul B. CONREC: a Contouring Subroutine [J]. Byte: The Small Systems Journal, 1987, 12(6): 143-150
- [9] McCullagh M. Terrain and Surface Modeling Systems: Theory and Practice [J]. Photogrammetric Record, 1988, 12(72): 747-779
- [10] Watson D F. Contouring: a Guide to the Analysis and Display of Spatial Data[M]. Oxford: Pergamon Press, 1992
- [11] van Kreveld M. Efficient Methods for Isoline Extraction from a TIN [J]. International Journal of GIS, 1996, 10: 523-540
- [12] Jones N L, Kennard M J, Zundel A K, et al. Fast Algorithm for Generating Sorted Contour Strings [J]. Computers and Geosciences, 2000, 26: 831-837
- [13] Hennig T, Kretsch J. The Shuttle Radar Topography Mission[C]. Proceedings of DEM 2001, Manano, 2001

第一作者简介:王涛,博士,从事地图学与地理信息系统研究。

E-mail: mapwang@tom.com

## An Algorithm for Extracting Contour Lines Based on Interval Tree from Grid DEM

WANG Tao<sup>1</sup> WU Hehai<sup>2</sup> LIU Jiping<sup>1</sup>

(1 Chinese Academy of Surveying and Mapping, 16 Beitaiping Road, Beijing 100039, China)

(2 School of Resource and Environment Science, Wuhan University, 129 Luoyu Road, Wuhan 430079, China)

**Abstract:** There are two sub-task of automatic contouring from grid DEM. The first is choosing a starting cell and the next is the threading process. A new algorithm is proposed based on interval tree to determine starting points of contour lines. The result improves the algorithm's efficiency remarkably. Further, a new strategy is designed to constrain the direction of threading and the resulting contour bears more meaningful information.

**Key words:** contour; grid DEM; threading; interval tree