

基于相邻关系的等高线树建立算法研究

郭沛沛,李成名,殷 勇,丁圣陶

(中国测绘科学研究院,北京 100830)

[摘要] 等高线树在提取地形线、地貌综合等方面具有重要的应用意义。常用的等高线树表达的是等高线的层次关系,直接表达等高线的包含和邻接关系,建树过程复杂,对数据要求较高,需要对数据进行预处理。本文将图幅内等高线按照是否闭合分为两类,分别对两类等高线群进行建树,最后将两个独立的等高线树合并,形成完整的等高线树。最后生成的等高线树表示了等高线之间的相邻关系,再借助其高程值就可以判断出等高线之间的包含邻接关系,算法容易理解,程序实现简单。

[关键词] 等高线树;相邻关系;高程值;闭合等高线;不闭合等高线

[中图分类号] P208 **[文献标识码]** A **[文章编号]** 1009-1742(2013)05-0042-05

1 前言

等高线之间的空间关系通常采用等高线树来表示。等高线树表达的是等高线之间的层次关系,表现为等高线之间隶属/包含的父子关系和邻接/并列的兄弟关系等。等高线树在地表形态组成分析和地貌综合等方面应用很广泛^[1]。现有的等高线树中父子节点直接表示等高线之间的包含关系,兄弟节点直接表示邻接关系。在建立过程中,最重要的一步是进行等高线闭合方向的正确判断。但是现有判断方法得到的结果往往存在二义性,有时得到的结果和实际情况会有差异。

基于相邻关系的等高线树中当前节点和它的子节点、父节点之间具有邻接关系。借助等高线的高程信息就能判断出相邻等高线之间的包含关系。等高线树的生成确立了等高线空间关系,为提取地性结构线提供了便利的条件和基础。以提取谷底线为例,从最高高程开始,向下寻找谷底点时的搜索范围因为有了等高线树才大大缩小,只需在与当前等高线相邻的低高程等高线上寻找。如果

仅靠距离限制,那么有可能追踪到邻近山头所涉及的等高线上。使用等高线树可使搜索的正确性及速度大大提高^[2]。

2 算法概述

首先对图幅内所有等高线进行识别,分为闭合等高线和不闭合等高线两类。对于闭合等高线,直接使用现有方法建立闭合等高线树 Close Tree;对于每一条不闭合等高线,它必定和图幅边有两个交点。求出所有交点,按照沿图幅边逆时针方向排序,记录它们的序号。假设存在两条等高线 L_1 、 L_2 ,它们的起点序号分别为 S_1 、 S_2 ,终点分别为 E_1 、 E_2 。如果 $S_1 = S_2 - 1$ 并且 $E_1 = E_2 + 1$,则 L_1 和 L_2 相邻,且在 L_2 一侧只和 L_2 相邻;如果 $S_1 = S_2 - 1$ 或者 $E_1 = E_2 + 1$,则 L_1 和 L_2 相邻,但不唯一和 L_2 相邻。依此建立不闭合等高线树结构 Open Tree。最后将两个树进行合并,得到整个等高线图幅的关系(见图1)。如果不同高程的两条相邻等高线具有包含关系,则相同高程的两条相邻等高线具有并列/邻接关系。

[收稿日期] 2013-03-21

[基金项目] 国家测绘地理信息局科技基金资助项目(A1118)

[作者简介] 郭沛沛(1986—),女,河北邢台市人,博士研究生,主要从事地图制图与自动综合研究;E-mail: guopeipei925@163.com

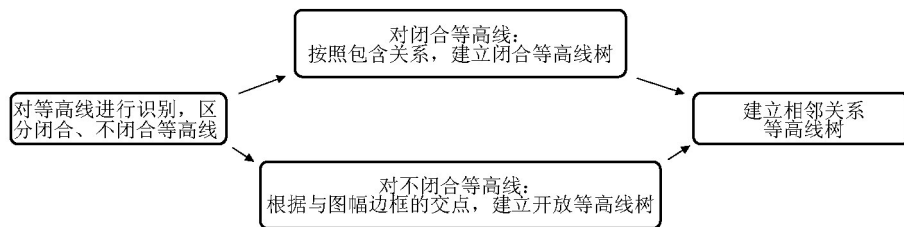


图1 相邻关系等高线树的建立流程

Fig.1 Process of building the contour neighbor Tree

3 算法具体步骤

3.1 等高线识别分类

读取图幅内所有等高线,判断等高线是否闭合。如果闭合,将其ID放入数组 closeIDs[]中;如果不闭合,计算它和图幅边的交点,放入到数组 intersectNodes[]中。对 intersectNodes[]中的交点,按照交点沿着图幅边到图幅左下角点的距离升序排列。

3.2 闭合等高线建树

取 closeIDs[]中存储的所有闭合等高线,按照等高线形成的闭合面的面积由小到大排序。取面积最小且没有处理过的等高线 L_i ,判断其余闭合面面积较大的等高线 L_j 和它的包含关系:可以首先取 L_i 上任意一点 P_i ,判断其余等高线 L_j 是否包含 P_i 。如果 L_j 不包含 P_i ,则 L_j 肯定不包含 L_i ;如果 L_j 包含 P_i ,则需要进一步判断 L_j 是否包含 L_i 上全部点。确定等高线之间的包含关系后按照下面方法,建立树结构(见图2)。

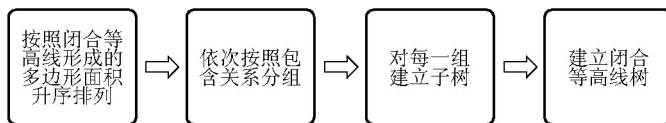


图2 闭合等高线树建立过程

Fig.2 Process of building the Close Tree

1)按面积排序。按照等高线形成的闭合面的面积升序排列,得到等高线序列。

2)按包含关系分组。取闭合面积最小的等高线上任一点,判断其余等高线对它的包含关系,得到一组等高线,并且标记这组等高线,它们不再作为面积最小等高线进行判断。再取面积最小且没有标记的等高线,同样方法得到一组等高线……最后得到若干组包含关系的等高线群 ContainContourGroups[[], [], []...]

3)建立子树。对 ContainContourGroups 中的每一组进行分类,最后一个等高线相同的组划分为一类。对每一类包含关系等高线群进行处理。对于每一类,把这类中每一组的序号颠倒过来,建立一个子树,它的根节点包含的数据为各组的第一条等高线,取各组等高线群的第二条等高线,建立子节点,把第一个等高线节点作为父节点。如果各组等高线群的第二条等高线有重复的,只建立一个子节点。依次取各组等高线群的第二条、第三条……等

等高线进行前述操作,直到所有组的等高线都搜索结束,同样方法建立下一类对应的子树。

4)建立闭合等高线树。把每个子树的根节点作为子节点,加入到总的闭合等高线树中。最后建立的闭合等高线树 Close Tree。

如图3所示,按照等高线形成的闭合面的面积升序排列,得到序列[9, 5, 8, 6, 7, 0, 4, 3, 10, 2, 1];按照包含关系分组得到等高线群 ContainContourGroups[]: [[L₉, L₁₀], [L₅, L₄, L₃, L₂, L₁], [L₇, L₁₀], [L₈, L₁₀], [L₆, L₂, L₁], [L₀]]; 将其划分为3类,分别建立子树;最后建立了闭合等高线树 Close Tree。

3.3 不闭合等高线建树

不闭合等高线的建树过程如图4所示。

1)确定起始等高线。对 intersectNodes[]中所有点 N_i (i 是交点的序号,如果 N_i 为最后一个交点,则 N_{i+1} 是第一个交点)进行判断,如果满足 N_i 和 N_{i+1} 在同一条等高线 L_k 上,则把 L_k 添加到起始等高线组 StartContours[]中,并且对 N_i 和 N_{i+1} 进行标记。对起

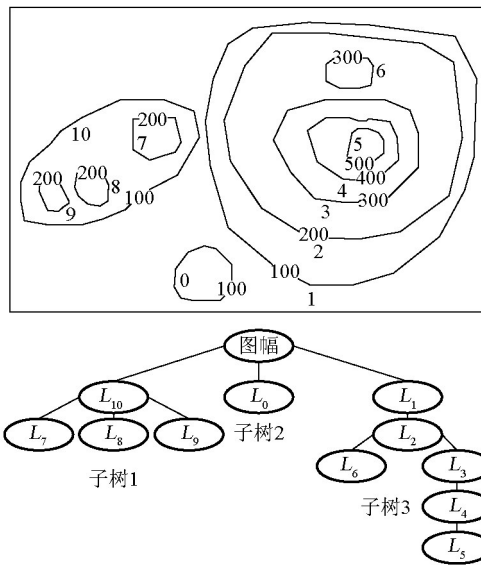


图3 闭合等高线及其对应的等高线树结构 Close Tree
Fig.3 Close contours and the corresponding Close Tree

始等高线组中的每一条等高线创建一个独立节点, 进入2)。

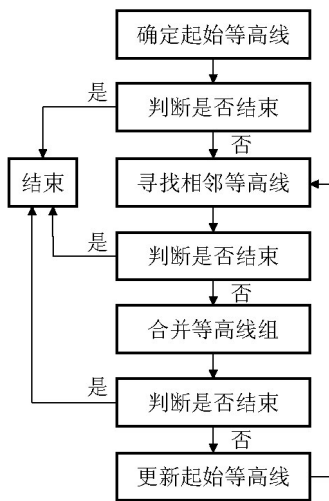


图4 不闭合等高线的建立过程
Fig.4 Process of building the Open Tree

如图5所示, 起始等高线为 L_{13} 、 L_{15} 、 L_{19} 、 L_{17} 、 L_7 、 L_9 和 L_1 。

2) 搜索结束的条件。判断是否所有的节点都被搜索过了, 如果不是则表示没有结束, 继续下一步; 如果是, 则把起始等高线对应的节点作为子节点, 添加到以整个图幅作为根节点的树结构中。

对1)中的结果判断是否结束, 如果没有结束则进入3)。

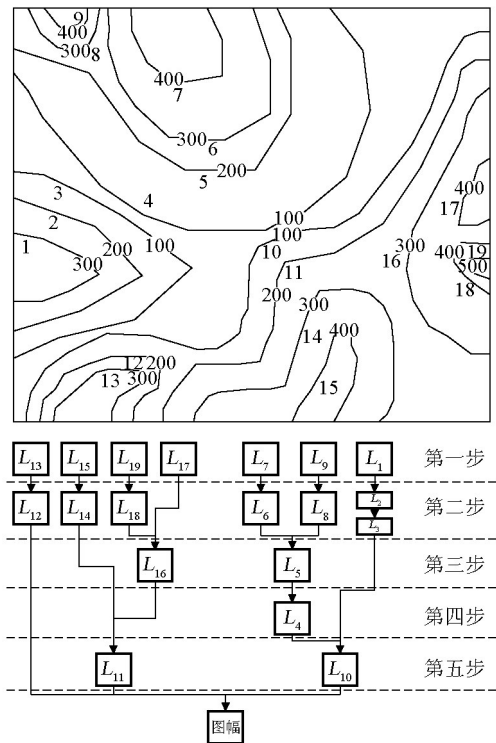


图5 不闭合等高线及其对应的等高线树结构 Open Tree
Fig.5 Open contours and the corresponding Open Tree

3) 寻找唯一相邻等高线。对于起始等高线中的等高线 C_i , 它的起点和终点分别为 N_s 和 N_e (s 和 e 表示交点的序号, 交点 N_0 的前一个交点是 N_{2n} , 交点 N_{2n} 的下一个交点是 N_0 , 共有 n 条等高线)。判断 N_s 的前一个交点 N_{s-1} 和 N_e 的后一个交点 N_{e+1} 是否在同一条等高线上。如果不位于同一条等高线上, 则不进行其他操作; 如果位于同一条等高线 C_j 上, 则 C_i 和 C_j 相邻, 并对等高线 C_j 创建节点, 作为 C_i 节点的父节点, 并用 C_j 代替 C_i 存放到期始等高线组 Start-Contours[]中, 对 C_i 的起点和终点 N_{s-1} 、 N_{e+1} 进行标记, 再对 C_j 进行与 C_i 一样的操作。对起始等高线中的每一条等高线进行同样的操作。

所有的起始等高线处理结束后, 判断搜索是否结束, 不结束则进入4)。

4) 合并起始等高线组。对于起始等高线组中的等高线, 按照起点的位置升序排序。找出其中连续相邻的起始等高线 $[C_i, C_j, \dots, C_k]$, 这里的连续相邻是指前一个等高线的终点与后一个等高线的起点相邻。判断它们中第一条等高线 C_i 起点的前一个节点和它们中最后一条等高线 C_k 终点的后一个节点是否未标记且在同一条等高线上。若是, 且它们在同一条等高线 C_l 上, 则对 C_l 建立节点, 把这些

起始等高线对应的节点作为 C_i 对应节点的子节点, 并从起始等高线组中删除这些起始等高线。对 C_i 的起点和终点进行标记。

判断搜索是否结束, 不结束则进入3)。

整个建树的过程和结果及不闭合等高线树 Open Tree 如图5所示。

3.4 建立整个等高线树

前面两步建立的等高线树是相互独立的, 这一步就是要把闭合和不闭合等高线树结合起来, 表示出图幅内所有等高线的空间关系, 具体步骤如下。

1) 对 Close Tree 根节点的所有子节点 $cNODE_i$, 取它们代表的闭合等高线 C_i 。同时, 对 Open Tree 根节点的所有子节点 $oNODE_j$, 取它们代表的不闭合等高线 C_j 。

如图6所示, 下侧左边的树为 Close Tree, 其根节点的子节点代表的等高线为 L_{14} 、 L_{15} 、 L_{16} 和 L_{17} , 下侧右边的树为 Open Tree, 其根节点的子节点代表的等高线为 L_{10} 、 L_{11} 和 L_{12} 。

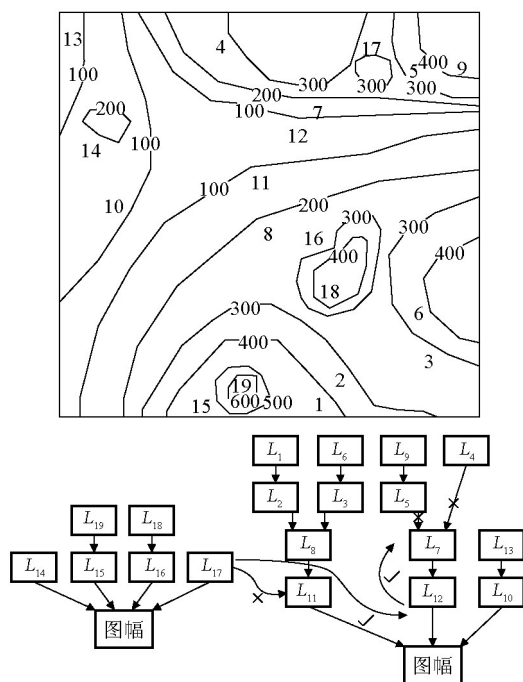


图6 闭合等高线树、不闭合等高线树及完整等高线树的建立

Fig.6 Close Tree, Open Tree and process of building the contour neighbor Tree

2) 对于 C_j , 按照等高线起点到终点的方向, 沿着图幅边进行闭合, 形成闭合等高线 C_j' 。判断 C_j' 是否包含 C_i 。如果不包含, 取 Open Tree 根节点的下一个子节点进行同样的判断; 如果包含, 将此 $oNODE_j$ 当作根节点重复2)。如果 Open Tree 根节点的所有子节点都不包含 C_i , 则把 C_i 添加到 Open Tree 中, 图幅根节点作为 $cNODE_i$ 的父节点。

3) 直到遇到父节点包含 C_i , 但所有子节点都不包含 C_i 的情况, 把此父节点作为 $cNODE_i$ 的父节点。取闭合等高线树根节点的下一个子节点, 进入2)。

如图6所示, 对于等高线 L_{14} , 它被等高线 L_{10} 和图幅左上角点形成的闭合环包含, 但是不被等高线 L_{13} 与图幅左上角点形成的闭合环包含, 因此把节点 L_{14} 作为节点 L_{10} 的子节点。对于等高线 L_{17} , 判断出 L_{11} 与图幅边角点形成的闭合环没有包含关系; 但判断出和下一个子节点 L_{12} 有包含关系; 和 L_{12} 的子节点 L_7 也有包含关系; 但和 L_7 的子节点 L_5 和 L_4 都没有包含关系, 因此把 L_{17} 作为 L_7 的子节点。

4) 当 Close Tree 根节点所有的子节点都找到父节点, 等高线树建立完毕。

4 结语

树结构是表达等高线空间关系的一种良好工具, 如何正确快速地对大数据量的等高线数据建立树结构对于提取地形线和地貌综合等工作具有重要意义^[3,4]。本文提出一种基于等高线相邻关系生成等高线树的新方法, 此方法简单易懂, 不需要预处理, 便于使用计算机程序实现。经过实际程序验证, 其对闭合不闭合等高线均能处理。

参考文献

- [1] 毋河海. 等高线树的自动建立及其应用[J]. 测绘科技动态, 1996(1): 2-7.
- [2] 张琳琳, 武芳, 王辉连. 等高线空间关系的确定及应用[J]. 测绘通报, 2005(8): 19-22.
- [3] 王涛. 地貌信息提取中的结构化问题研究[D]. 武汉: 武汉大学, 2005.
- [4] 张琳琳. 地貌自动综合的结构化方法研究[D]. 郑州: 信息工程大学, 2005.

An algorithm of neighbor relationship oriented contour tree building

Guo Peipei, Li Chengming, Yin Yong, Ding Shengtao

(Chinese Academy of Surveying and Mapping, Beijing 100830, China)

[Abstract] A contour tree has been found many important applications in extraction of terrain structure lines, terrain analysis, etc. A regular contour tree expressing the spatial relationship of contours is difficult to build, even with data of high quality. A new method is proposed in this paper: classify contours into close group and open group; build contour tree for each group respectively; finally amalgamate these two contour trees into one. With the elevation values, spatial relationship of contours is of easy accessibility. The result shows that the above mentioned method is correct and effective.

[Key words] contour tree; neighbor relationship; elevation value; close contour; unclosed contour