

地图线状要素注记自动配置的算法与实现

殷 勇,李成名,丁圣陶

(中国测绘科学研究院,北京 100830)

[摘要] 本文探讨了线状要素自动注记的几个关键问题,提出利用注记参考线计算线状要素的注记定位线,同时提出反 Douglas 算法进行注记参考线的动态分段,利用动态分段的结果集获取线状要素的平滑位置作为候选注记位,在此基础上获取注记中每个字的备选位置以及评分,最后采用禁忌算法解决注记之间的冲突,从而较好地解决了线状要素注记不可见或表达模糊的问题,实现了自动注记输出的完整性和美观性的统一。

[关键词] 线状要素;注记自动配置;参考线;反 Douglas 算法

[中图分类号] P208 **[文献标识码]** A **[文章编号]** 1009-1742(2013)05-0030-07

1 前言

线状要素的注记方式依地物类型和地图用途的不同有较大的差别,可分为点定位注记配置模式和线定位注记配置模式(曲线注记模式)两种。无论从地图注记的功能还是从美学角度出发,曲线注记模式都是最重要的线状要素注记方式。本文重点研究曲线注记模式,提出了一种快速而高效的线状要素曲线注记的自动配置方法,可以对自然要素(如线状水系)和人工要素(如道路和街道)进行统一处理。

在以往线状注记的研究文献中,较多采用线点注记代替线状注记^[1-3],不能处理曲线注记模式;注记压盖处理方式多采用栅格化的方式^[4-7],注记处理过程中数据进行矢栅互转时,既影响效率,同时也影响注记的精度;线状要素的注记方式一般较多探讨散列方式^[2-4],不能满足在实际应用中紧密型线状注记的需求;而避免由于点位密集引起的注记冲突和与地物的压盖—禁忌、回溯与神经网络等方法,已被证明取得了较好的结果^[3,8-10]。因此,本文的自动注记过程采用全矢量化的方式,有利于提高自动注记的速度以及精度;对于压盖处理方式,由于

采用全矢量化处理方式,这样方便利用 R 树索引来加快搜索速度;对于紧密型线状注记,本文采用整体移位的方式;对于散列型线状注记,本文采用禁忌法的单字移位达到整体最优;对于不足字长的线状要素,需要根据实际情况进行强制注记。

2 线状要素注记解决方案

线状要素注记自动配置的基本流程大致可以分为以下 6 个步骤。

- 1)全要素数据预处理。
 - 2)线要素的综合简化。
 - 3)选择注记方式。采用中心线注记还是平行线注记。
 - 4)求参考线。对平行线注记要先求平行线,并消除尖角、自相交、相近等。
 - 5)对较长的线要素进行分段处理,以便对其进行分段注记。
 - 6)对每一条曲线的注记进行注记定位与选优。
- 一般情况下,注记字的放置如图 1 所示,可以根据曲线段的两个端点构成的直线方位角判断是相对平行或垂直的线。相对平行的线,注记字从左往右放置,字头方向朝上,与曲线的切线垂直;相对垂

[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] 殷 勇(1982—),男,湖北黄冈市人,助理研究员,主要从事地图制图和综合自动化研究;E-mail:yinyong.985@gmail.com

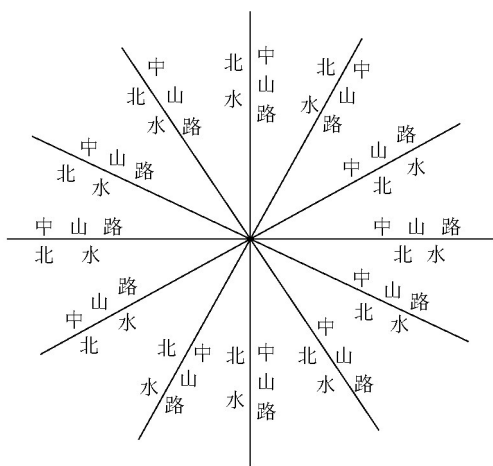


图1 线注记的走向
Fig.1 Directions of line label

直线的注记字从上往下放置,字头方向朝上,与曲线的切线平行。

3 线状要素注记关键算法

3.1 全要素数据预处理

此处的全要素数据预处理需要解决的是参与注记的所有线图层中的所有要素的合并与拓扑分段,便于后续处理得到高质量的定位线。

1)要素合并,根据数据特点合并具有相同属性值的弧段。

2)拓扑分段,当对两个相互交叉弧段进行注记时,只有一个弧段可以跨越交叉口进行显示,否则注记就会出现所属关系不明确,影响注记效果。这里首先根据经典拓扑理论对所有线要素建立拓扑关系,然后分两种情况进行拓扑分段处理:a.对于两个优先级高低有别的要素,在结点处,优先级高的合并(根据属性值是否相同),优先级低的打断;b.对于两个优先级高低相同的要素,在结点处,计算相邻弧段的长度,长度短的合并(根据属性值是否相同),长度长的打断。

3.2 要素综合简化

对于一个地理信息系统(GIS),由于受硬件条件的限制,数据量的大小以及数据的质量将对整个系统的运行速度和效率产生直接的影响,如果是注记动态配置功能模块,要不断动态响应GIS用户的操作,对速度有更高的要求,精简的数据就显得更为重要。在注记自动配置中,要素简化不仅可以删除冗余数据,减少数据存储量,同时降低线状要素形状复杂度,这对于后期线目标的裁剪、平行线

的求取以及提高注记动态配置的响应速度都很有意义。

本文采用的综合简化算法是Douglas算法。其基本步骤如下。

1)设定数据压缩的阈值,这个值一般情况下不会太大,是一个基本上不会影响视觉效果的值。

2)连接首尾两点,找出与两点连线有最大偏移距离的点。

3)如果这个点的偏移距离小于这个阈值,则去掉两点间所有的点,否则,把曲线以这个点为分段点分成两条曲线,执行第2步。

这是一个比较简单的容易实现的算法,可以用一个递归算法来实现。

3.3 求参考线与平行线

对矢量数据平行线的求取,以往一般采用角平分线法,这种算法具有高精度、运行速度快等优点。图2给出了角平分线法的原理,其中, α 为角度, d 表示平行线与中心线的距离。

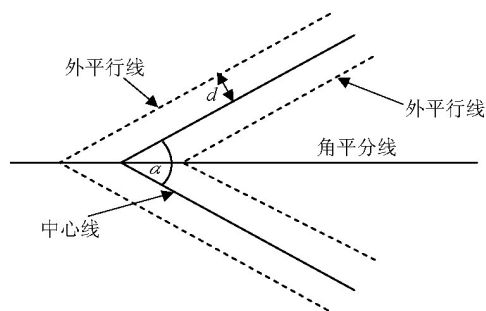


图2 角平分线法
Fig.2 Internal bisector

这种方法生成的平行线有一个缺点,那就是小角度上的点,不管平行线是在角的外侧还是内侧,所求得的点都会离中心线非常远。如果是在外侧,平行线上的点就会非常突出,伸向很远的地方;如果是在内侧,平行线上的点就会缩到很远的地方,如果点两侧的边比较短的话,平行线就会因为这个点而改变其原有的形状。这个问题一直是一个待解决的问题。

为了解决这个问题,笔者分两步进行。

1)解决平行线内侧点回缩而改变形状的问题。这个问题的解决方法是生成平行线的参考线,通过生成的参考线来求取平行线。

具体的方法是扫描曲线上的每一个点,求这个点的夹角,如果角度 $\alpha < 90^\circ$,且平行线是在这个角

的内侧,则计算两条边的长。如果两条边的长度都小于 $d/\tan(\alpha/2)$,则删除该点和角两端的点。如果只有一边的长度小于 $d/\tan(\alpha/2)$,则用另一条边上距离该点与这条边的长度相同的点代替该点。

2)解决平行线外侧点突出的问题。这个问题的解决方法是改进角平分线法,在不求取平行线时添加辅助点。

具体的方法是在用角平分线求平行线的过程中,对于角度 $\alpha < 90^\circ$,且平行线在这个角的外侧,则用角平分线上距离该点为 d 的点代替平行线上的点,在左右两侧各加一个辅助点。

上述求平行线的过程可以用图3来表示,图3中求了中心线的左平行线,中心线的方向是从左往右,图中先通过连接点1和2,3和4,生成了参考线,然后再通过改进的角平分线法生成了平行线。

在求完平行线以后,平行线可能会出现自相交和自相近的情况。对于自相交的情况,在以往的研究中已经给出了解决方法,但对于自相近的情况,却没有人注意。在这里先介绍平行线自相交的处理方法,给出平行线自相近的描述、对注记的影响以及处理方案。

设平行的定位线有 N 个坐标点,则包含的线段数为 $N-1$,消除定位线自相交情况的具体步骤如下。

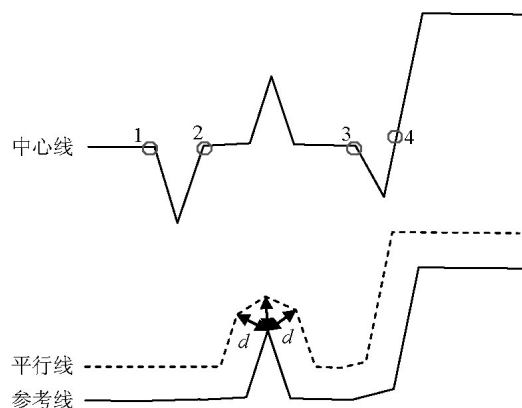


图3 改进的角平分线法

Fig.3 The reformative method of internal bisector

- 1)输入定位线的坐标数和坐标串。
- 2)从第1段到第 $N-1$ 段线段,依次检查与每段线段相交的第一段线段。
- 3)两条相交的线段形成一个环路,环路从交点开始又终于交点,将环路之间顶点删除,将新的交点作为新顶点加入定位线坐标串中,并修改定位线的顶点数目。
- 4)输出最后的顶点和线段。

平行线自相交的示意图以及处理方案如图4所示。

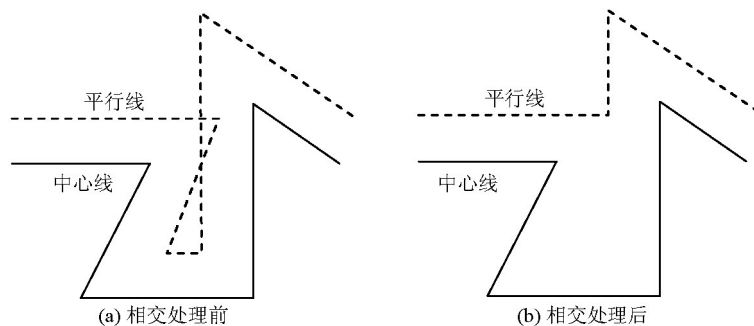


图4 平行线自相交示意图

Fig.4 The parallel line intersect

平行线自相近一直是一个被忽略的问题,但在注记过程中又很可能存在。如图5所示,其处理方法可以分以下几步。

- 1)输入坐标点串。
- 2)从第一段开始,依次比较该段与后面线段的关系是否相近,并记录下最后一个与其相近的线段。
- 3)如果是端点相近,则连接端点,如果是端点

与线段相近则连接这个端点和这个点在其相近线段上的垂足。

- 4)删除连接两点之间的点。
- 5)输出剩余的点作为平行线。

3.4 动态分段

有些线状要素在地图上的长度比较长,需要进行分段注记,也就是说在注记之前,要对线状要素进行分段处理,然后再每一段或者选择其中的几段

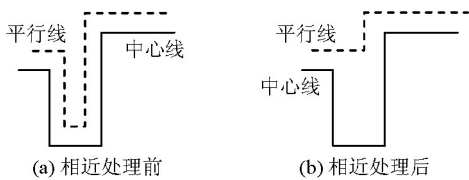


图5 平行线自相近示意图
Fig.5 The parallel line close

进行注记。在以往的曲线分段方法中,有两种分段方法,一种是根据方位角改变分段的方位角法,另一种是根据曲线的单调性进行分段的单调段法。方位角法在一定程度上可以将曲线分成较为光滑的曲线段,但是对一些复杂的曲线,特别是在一些线段比较短,方位角变化比较大的曲线中,容易把曲线划分得太细,以至于无法进行注记;单调段法能够保证曲线注记的通视性,但是不能提取出相对平滑的曲线。

方位角法和单调段法两种方法都不能满足在光滑部分注记的目的,为了解决这个问题,受 Douglas 算法的启发,提出了反 Douglas 算法。算法的基本思想是以曲线上所有的线段为基础,逐步合并相对光滑的曲线段。

在介绍反 Douglas 算法的具体实现步骤以前,先来进行如下两点说明。

1) 相邻曲线高度 h , 这是指当相邻的两条曲线合并成一条曲线时,曲线上所有点到曲线两端点所成直线的最大距离。

2) 相邻曲线相对高度 h_{comp} , 设两曲线的长度分别为 l_1 和 l_2 , 两曲线的合成曲线两端点的距离为 d , 则 $h_{\text{comp}} = hd^2 / (l_1^2 + l_2^2)$ 。

反 Douglas 算法的计算步骤如下。

1) 设定两曲线可以合并的曲线相对高度的阈值 H_{comp} , 合并长度阈值 $L_{\text{合}}$, 这两个阈值的设定与注记字符串的长度和字大有关。

2) 输入注记曲线的点串。

3) 计算每个基础线段的长度 l_i 。

4) 以每个基础线段为初始的曲线段, 每个曲线段的长度为 L_k 。

5) 从每一个曲线段开始扫描, 计算相邻曲线相对高度 $h_{k,k+1}$, 并且记录下最小的相对高度 h_{min} 。

6) 判断是否终止。如果 $h_{\text{min}} > H_{\text{comp}}$, 转下一步。当 $h_{\text{min}} < H_{\text{comp}}$ 时, 合并相对高度等于 h_{min} 的相邻曲线, 并求出新曲线的长 $L_{k,k+1} = L_k + L_{k+1}$ 。如果

$L_{k,k+1} < L_{\text{合}}$, 用新的曲线替换原来的两条曲线, 转第 5 步。否则从剩下的相对高度中取出最小的值来替换 h_{min} , 重复第 6 步。

7) 输出曲线段, 对每一个曲线段进行注记。对于长度大于 $L_{\text{合}}$ 的基础线线段, 或将线段平均分割, 再进行注记。

反 Douglas 算法有两个优点: a. 能够提取出相对光滑的曲线, 相对原来的方法来说, 对弯曲度的适应度较高; b. 可以限定合并长度, 除非有长度较长的直线段, 否则能够进行注记的线段没有必要进行分段注记。

3.5 注记定位与选优

求取定位参考线(平行线)以后, 沿定位参考线(平行线)的自动最优注记定位问题是一个组合优化问题。本文采用禁忌搜索算法求解全局最优, 为此, 需要建立每个定位参考线(平行线)注记的备选位置组合以及合理有效的目标评价函数。线状要素注记定位过程可以概括为以下 3 步。

3.5.1 获取注记备选位置

1) 注记配置将注记看成一个由其尺寸决定的矩形框。选择按照注记规则生成的一条定位参考线(平行线), 放置注记的方式是先中间后两头。整个注记串放置的中点称为注记字串的定位点。如果注记包含奇数个汉字, 先确定整个注记串的中点, 配置中间的一个汉字, 再配置紧靠中间汉字的两个汉字, 以此类推。

2) 注记配置的控制参数组 (ParallelDis, FontSpace, LabelPos) 唯一确定一个备选位置。其中 ParallelDis (PD) 为注记定位参考线(平行线)与线状要素的偏离距离, FontSpace (FS) 为注记字符的间隔, LabelPos (LP) 为注记串的定位点, 即注记字符连接线的中点。

给定上述 3 个参数的变化范围和变化步长, 可以获得所有的备选位置。先定义几个参数: 定位参考线(平行线)长度 L , 注记字符串字数 CN , 单个字宽 CW , 注记首(末)字到定位参考线(平行线)起(终)点的距离 FL , 定位参考线(平行线)中点位置 MP 。便于后续计算, 对 3 个参数的变化步长这里我们统一取值为 5。

偏离距离 PD 的范围 $[\text{Min}D, \text{Max}D]$, 对于紧密注记, 取 $\text{Min}D = \text{Max}D = 0$; 对于散列注记, 取 $\text{Min}D = \sqrt{2} \times CW / 2$, $\text{Max}D = CW$ 。

字符间隔 FS 的范围 $[\text{Min}S, \text{Max}S]$, 对于紧密注

记,取 $\text{Min}S = \text{Max}S = 0.1\text{CW}$; 对于散列注记,取 $\text{Min}S = 3, \text{Max}S = 5$ 。

注记定位点 LP 的范围 $[\text{MP}-\text{Delta}, \text{MP}+\text{Delta}]$, 对于任意给定的字符间隔 FS, 注记定位点 LP 的活动邻域 Delta 表达式为:

若字数为奇数,则 $\text{Delta} = L/2 - (\text{FL} + \text{CN} + 1/2\text{CW} + \text{CN}/2\text{FS})$;

若字数为偶数,则 $\text{Delta} = L/2 - (\text{FL} + \text{CN}/2\text{CW} + \text{CN}/2 - 1\text{FS} + \text{FS}/2)$ 。

3.5.2 对备选注记位置评分

一般认为,最好的注记位置是离线状要素最近、位置越接近中心、字符间隔最小的备选位置。对备选位置的评价考虑注记位置优先级、注记最小外接矩形框是否相交、注记最小外接矩形框是否压盖背景注记(一般指面注记或其他限制条件)等3个因素,从而建立目标函数

$$F = \sum_{i=0}^n C_i$$

式中, $C_i = a \cdot \text{Preference}(i) + b \cdot \text{Conflict}(i) + c \cdot \text{Overlap}(i)$, $\text{Conflict}(i) = \sum p_j$, n 表示待配置注记的总数, C_i 表示每个点注记的代价值。 a, b, c 分别为注记位置优先级、冲突量和压盖量的权重值,本文中这3个权重值都为1。 $\text{Preference}(i)$ 为注记位置优先级, p_j 为每个线状要素的等级大小, $\text{Preference}(i)$ 与 p_j 设计为0.2的倍数取值, $\text{Overlap}(i)$ 为压盖背景注记的个数。

3.5.3 注记点位的搜索与选优

1) 读入所有待注记的定位参考线。

2) 产生初始解:产生初始解 X , 设置迭代次数 $k=0$, 最佳解向量 $X_{\text{best}} = X$, 若目标函数值为0, 直接返回。

3) 产生一组候选解:按代价值从大到小得到一组可行的候选解。

4) 搜索邻域:在上述每个候选解中通过改变3个控制参数 ($\text{ParallelDis}, \text{FontSpace}, \text{LabelPos}$), 计算比较每个代价值,通过最小代价值得到一个最好的候选解 X^* , 如果 X^* 不在禁忌表中,或者即使在禁忌表中,但已经满足释放准则,则用 X^* 更新 X ; 如果该解在禁忌表中,但是不满足释放准则,则寻找次优解,并重复此过程。

5) 更新禁忌表:将寻优得到一个最好的候选解 X^* 压栈到禁忌表中,如果禁忌表已满,则首先释放最先记录的候选解。

6) 更新 X_{best} : 如果 X^* 的目标函数值小于 X_{best}

的目标函数值,则用 X^* 更新 X_{best} , 否则 X_{best} 保持不变。

7) 判断终止条件:若连续 n 次迭代之后目标函数值没有改进,或者达到最大允许迭代次数,则停止优化,输出结果;否则迭代次数加1,并转到步骤3),继续优化。

初始解 X 以偏离距离 PD 的最小值 $\text{Min}D$ 、字符间隔 FS 的最小值 $\text{Min}S$ 、LP 的活动邻域 $\text{Delta}=0$ 求取所有注记备选位置。在产生初始解 X 后根据所有注记最小外接矩形框建立 R 树,在步骤4)中 X^* 更新 X 过程中动态更新 R 树,通过 R 树提高注记相交与压盖的查询速度。

4 结语

图6给出了直接用角平分线法求取的左右平行线的效果图,图6中暗灰色部分是中心线,黑色是平行线,星号是对平行线的分段点。由图6可以看出角平分线法的不足,以图中A所示的点为例,它对应的平行线上的点发生了点的远离,其外侧点突出到B所示的位置,内侧点回缩到C所示的位置,而图中D和E所示的位置则发生了平行线自相交的情况,如果平行线的距离再小一点,那么E所示的位置就会出现平行线自相近的情况。

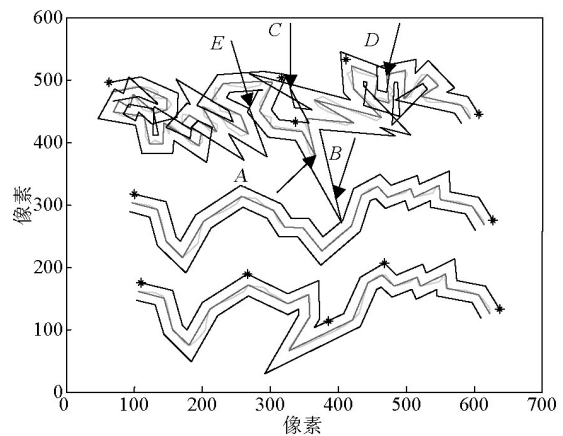


图6 角平分线法求平行线

Fig.6 Parallel line obtained according to the internal bisector

平行线即注记线,是地图注记过程中注记字所在的线,所注记字的中心就在平行线上,而图6中所示的平行线显然无法进行注记。

图7给出了用改进算法求取上述平行线,然后用反 Douglas 算法对平行线进行分段的效果图。与

The algorithm and realization of linear feature automatic label placement

Yin Yong, Li Chengming, Ding Shengtao

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

[Abstract] This paper discussed several key problems in the automatic annotation of linear elements, proposed to calculate the positioning line of a linear element using reference annotation line and to achieve dynamic segmentation of reference annotation line using reverse Douglas algorithm, using the segmentation result to choose smooth parts of linear elements as candidate annotation positions. On this basis, get the candidate position and score for every letter, then resolve the conflicts between annotations using taboo algorithm. The problem of vague and invisible representation of linear element annotations was solved and the integrity and aesthetics of automatic annotation output were achieved.

[Key words] linear elements; automatic annotation configuration; reference line; reverse Douglas algorithm