

复杂线状符号的设计及优化算法研究

吴小芳^{1,2,3} 杜清运^{1,2} 徐智勇^{1,2} 蔡忠亮^{1,2}

(1 武汉大学资源与环境科学学院, 武汉市珞喻路 129 号, 430079)

(2 武汉大学教育部地理信息系统重点实验室, 武汉市珞喻路 129 号, 430079)

(3 华南农业大学信息学院, 广州市五山路 483 号, 510642)

摘要: 针对地图线状符号中的需求, 提出了利用自定义线型设计线状符号的新方法, 以解决常规线型符号绘制的问题。此外, 考虑到线状符号在拐角处易产生严重变形以及出现断裂、自相交等情况, 提出了采用符号自适应性及双仿射变换方法的解决途径, 实现符号化绘制, 增强符号的可视化效果。

关键词: 线状符号; 符号化; 符号自适应性; 双仿射变换

中图法分类号: P283.1

地图符号是地图的语言, 是构成地图的基本元素。线状符号是地图符号中使用最多的一类, 也是地图符号中最复杂的一类。线状要素在符号化过程中要考虑多种情况, 程序设计算法复杂, 操作运算量较大, 耗费时间较长。因此, 如何有效地表达线状符号成为地图符号研究中的难点问题。本文针对地图线状符号中的常见问题, 提出了利用自定义线型设计线状符号的新方法, 以及采用符号自适应性和双仿射变换方法处理拐角处的变形问题。

1 线状符号优化算法

线状符号可以分为两部分: 线主体符号和在主体符号上叠加的基本图元。目前对于线状符号的绘制, 国内外采用的思路一般分为以下四类: 纯函数绘制、横向循环配置^[1]、纵向叠加^[2]、基于矢量栅格技术的绘制算法^[3]。在这些方法的基础上, 本文提出采用自定义线型与图元相结合的优化算法。

本文所采用的线状符号绘制方式是将自定义线型与图元相结合, 运用参数化的方式组织符号数据, 利用自定义的线型和图元沿着定位线循环配置合作的方式实现线状符号的绘制, 即是复

杂的线状符号拆分为由不同的线型和基本图元叠加组合而成。如采用此方法时, 铁路的基本图元由一条黑色实线和一条细白色虚线(为了显示图形, 白色采用灰色替代)组成, 如图 1 所示, 其中符号数据只包括黑色实线和白色虚线的线型、线宽、线色等参数化信息。在符号化时, 根据这些参数先设置 VC++ 中的自定义线型, 然后基于定位线的坐标先绘制黑色实线, 再绘制白色虚线, 即构成铁路, 其绘制速度快、效果好。下面分别从数据组织、算法效率和符号化效果方面进行分析和比较。

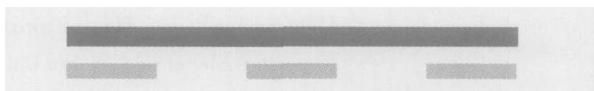


图 1 组成铁路的两条基线

Fig. 1 Two Basic Lines of Railway

1) 数据的组织结构比较

前四种方法中, 对于这类线型符号存储的数据是基本图元的属性及其几何坐标; 而本文方法存储的是线型符号的描述参数, 如线宽、线型、线色等, 不存储实际坐标, 采用参数化数据结构组织, 易于符号对象的描述, 且数据量小。

2) 绘制算法比较

下面针对各类绘制算法的思想及优缺点进行分析比较, 如表 1 所示。

表 1 绘制算法比较

Tab. 1 Comparison of Drawing Algorithm

方法	绘制算法思想	优缺点
纯函数方式	每种符号对应一种绘制函数	绘制速度快,效果较好,但符号可编辑性和维护性差,不易扩展
横向循环配置	基本图元沿定位线循环配置	绘制时,在定位线拐角处,需对图元作变形处理,符号化速度慢
纵向叠加	将线状符号拆分为多个基本的单元符号,并按纵向顺序分别绘制各单元符号	采用程序算法实现虚线线型的绘制,速度慢,且在尖锐的拐角处易出现变形,影响符号化效果
基于矢量栅格技术	利用栅格数据的加、或、异或、非等运算进行符号化	矢量与栅格相互转换的过程中会减慢符号化的速度,且栅格图形的符号表现效果欠佳
自定义线型与图元相结合	利用自定义线型和图元沿定位线循环配置合作的方式符号化	绘制速度快,效果好,且绘制曲线光滑,多适用于公路、境界、河流等以线型为主的符号

从表 1 可以看出,采用自定义线型与图元相结合的方式是在矢量栅格技术的绘制算法与纵向叠加方法的启发下,通过对线型符号进行有效拆分,设置不同的线型、线宽、线色参数,利用参数设置自定义线型,然后根据线型进行叠加,实现符号的绘制,绘制速度快,效果好。且在绘制 Bezier 曲线时,通过设置自定义线型参数,不需要将曲线离散为折线,即可实现符号的绘制。其绘制速度明显优于前面的方法,且绘制的曲线也十分光滑。

此外,自定义线型与图元相结合的方式根据线型叠加绘制线状符号,可避免平行线自相交的问题,从而解决符号相接处的图形问题,提高符号化质量。因此,对于公路、铁路、境界、河流等以线型为主的符号,均可采用此方法实现。

2 拐角点线状符号处理

自定义线型与图元相结合的方法中,自定义线型部分适合由不同线型组合而成的线状符号,图元部分则适合于非线型符号,如长城、陡坎等。对于由基本图元组成的线状符号,其符号化时仍是采用基本图元沿定位线循环配置的方法。在配置过程中,拐角点处基本图元的变形处理非常重要,拐角点处图元变形的好坏直接影响了整个符号的显示效果。

2.1 常见的问题

拐角点问题的产生是由于基本图元在沿定位线循环配置时,一个基本图元可能跨越在该定位线的多个直线段上而引发的问题。常见的问题有:拐角点基本图元的衔接问题(图 2(a));

拐角点基本图元的相交问题⁴⁾(图 2(b));拐角点基本图元异常(图 2(c));拐角点基本图元自相交问题(图 2(d))。

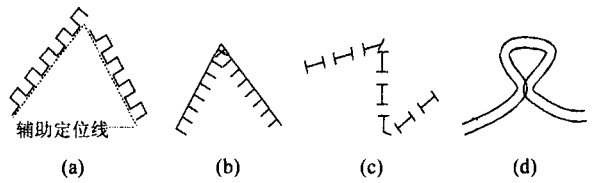


图 2 拐角点呈现的问题

Fig. 2 Problems in Corner

2.2 解决方法

根据组成线状符号的基本图元的不同特性,将其分为弹性图元和刚性图元两种。本文根据这两种图元不同特性,分别采用不同的方法解决拐角点处图元存在的问题。

2.2.1 弹性符号

解决弹性图元在线状拐角点处的配置问题时,有两种方法:通过符号自适应改变大小,避免基本图元跨越多个直线段,从而回避拐角点处的变形问题。采用双仿射变换方法,首先将基本图元沿曲线顶点拆分,然后对拆分后的两部分图元分别沿角平分线进行双仿射变换,最后在拐角点处配置变形后的图元。

2.2.1.1 符号的自适应性

符号的自适应性方法是根据拐角点的角度及角平分线等控制条件自适应改变符号的大小,从而避免拐角点处的符号产生变形。符号自适应性的算法思想如下。

1) 计算定位线中的每一直线段的长度,根据每一段的长度确定可配置的符号的数目以及每一段所余留的不足配置完整符号的长度。公式为:

$$n_i = l_i / (d + s), m_i = l_i \% (d + s)$$

其中, d 表示符号之间的间距; s 表示符号的宽度; l_i 表示每一直线段的长度; n_i 表示每一直线段可配置的符号数目; m_i 表示每一直线段所余留的空白长度。

2) 根据 m_i 的大小重新确定每一直线段可配置的符号数目。如果 $s \leq m_i < d + s$, $n_i + 1$, 即 n_i 的值在原有的基础上增加一个,同时相邻的下一直线段的长度减去 $d + s - m_i$ 后,再计算配置符号的个数,确定符号配置的位置;如果 $s/2 < m_i < s$, 则 n_i 保持不变,同时将 m_i 的值平均分给该线段上配置的最后两个基本图元,即最后两个图元所占的横向长度为 $s + d + m_i/2$; 如果 $m_i <$

$\sqrt{2}$, n_i 保持不变, 同时将 m_i 的值分给该线段上配置的最后—个基本图元, 即最后图元所占的横向长度为 $s + d + m_i$ 。其中, 若 $l_i < (d + s)$, 即 $n_i = 0$, 且 $l_i < d$, 可以参照下面的双仿射变换思想对符号进行变形处理, 确定符号的位置, 然后显示。依次循环类推, 求出定位线上所配置的图元数目及图元的定位点。

3) 在配置每个基本图元之前, 计算每个拐角点的角平分线。当配置拐角处附近的图元时, 计算图元的外接矩形与相邻的角平分线是否相交。若不相交, 按图元的原大小配置; 若相交, 根据交点的位置缩小外接矩形的高度或宽度, 使得原交点位于外接矩形的边线外, 即避免了图元与角平分线相交, 然后再绘制各图元。采用此方式可避免符号之间的相交情况, 保证整体视觉效果, 如图 3 所示。若其符号的缩放比例小于 30%, 则不显示符号, 以免符号太小而无法识别, 影响整体视觉效果。



图 3 陡坎符号

Fig. 3 Cutbank Symbol

此方法通过避开拐角点处的图元变形来满足整个线状符号的可视化效果, 而且在拐角点处通过自适应方法及角平分线的控制来调整图元的大小, 以保证符号整体的连续性效果。但这种方法仅适用于坐标稀疏的定位线, 因为对于坐标密集

的定位线, 由于每一个直线段比较短, 可能无法配置上一个基本图元, 因而每一—直线段上的图元均易发生大小变形, 导致整条定位线上的大多数图元均有变形, 从而影响整体效果。

2.2.1.2 双仿射变换

双仿射变换能保持边界的拓扑一致性, 该变换可把任意指定的四边形一对一地连续变换到另一—任意指定的四边形。实际上, 实现的过程是由两个仿射变换组成 ($\triangle 123$ 与 $\triangle 1'2'3'$, $\triangle 234$ 与 $\triangle 2'3'4'$), 如图 4 所示。该方法的特点是: 每一个三角形的仿射变换都只利用三角形三顶点的坐标条件, 故两邻接三角形公共边界上的点, 其变换的像是惟一的。边界 23 线上的任意点变换仅受其位置及 23、2'3' 的影响, 其像是同一的。该变换能解决不同图形区域之间的坐标变换, 十分稳定和迅速, 且保持边界的拓扑一致性。

利用双仿射变换解决图元拐角点处问题的基本思想是: 确定跨越拐角点的图元位置, 根据拐角点将图元的外接矩形拆分为左右两个矩形 ABCD 和 CDEF, 如图 5 所示, 同时将图元的几何数据也沿此点的垂直方向拆分为左右两个局部图元。计算拐角点的角平分线, 通过角平分线确定图元两边的矩形变形后的四边形坐标 $A'B'C'D'$ 和 $C'D'E'F'$, 如图 6 所示。根据图元两边的矩形坐标以及变形后两边的四边形坐标, 利用双仿射变换分别求得方程的参数。利用这些方程参数, 对两个局部图元的所有坐标数据进行双仿射变换, 转换到变形后的四边形坐标系内, 即实现了图元数据在拐角处的坐标变形处理。

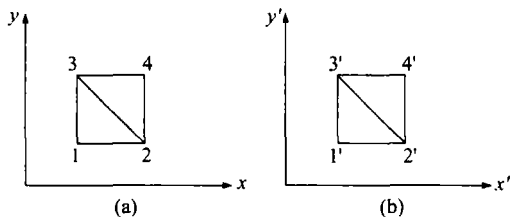


图 4 双仿射变换示意图

Fig. 4 Diagram of Double Affine Transformation

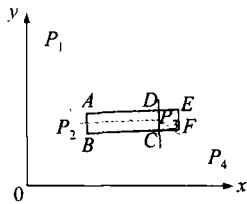


图 5 拆分后的图元

Fig. 5 Parted Graphic Element

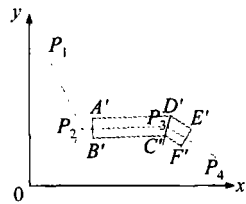


图 6 变形后的图元

Fig. 6 Transformed Graphic Element

采用双仿射变换的算法思想绘制的长城符号如图 7 所示。可以看出, 在拐角点处, 根据双仿射变换方法, 图元产生变形, 其外形、尺寸发生变化, 避免了符号相交的情况, 且图元衔接自然, 无断裂情况出现。此方法对坐标稀疏或密集的定位线均适用。

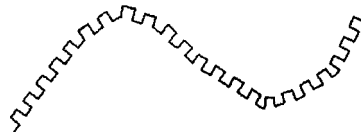


图 7 长城符号

Fig. 7 Great Wall Symbol

但当拐角特别尖锐时, 图元的外形易发生较大的变化, 对于这种情况, 需要采用不绘制图元或缓解图元变形的方法进行处理, 也可通过在拐角处增加节点, 将尖锐的拐角平缓化, 缓解图元的变形。

2.2.2 刚性符号

刚性符号由于需要保持外形不变, 在跨越拐角点时不允许进行变形处理, 因而这类符号的处理比较简单, 只需按常规计算图元的定位点, 对图元进行平移变换和旋转变换, 然后在定位点上依次配置图元即可^[6]。当然, 也可采用上面所提的符号的自适应方式对其方法进行改进, 在每一线段配置可容纳的图元数目后, 不考虑所余留的长度, 不对图元进行缩放, 直接进行下一段的图元配置, 即可形成整个线状符号的配置。

参 考 文 献

[1] 赵江洪, 殷赣华. 通用地图符号库的算法设计及其

实现[J]. 测绘通报, 2002(4): 41-42

- [2] 谈晓军, 边馥苓, 何忠焕. 地图符号可视化系统的面向对象设计与实现[J]. 测绘通报, 2003(1): 11-14
- [3] 蔡先华, 张远. 基于矢量栅格技术的矢量数据符号化方法研究[J]. 东南大学学报(自然科学版), 2003, 33(6): 763-766
- [4] 郭庆胜, 郑春燕. 地图线状符号图案单元的优化配置方法[J]. 武汉大学学报·信息科学版, 2002, 27(5): 499-504
- [5] 吴小芳. 基于 GDI+ 的高精度地图符号库的设计与实现[D]. 武汉: 武汉大学, 2003
- [6] 何忠焕. GIS 符号库中复杂线状符号设计技术的研究[J]. 武汉大学学报·信息科学版, 2004, 29(2): 132-134

第一作者简介: 吴小芳, 博士。现从事地图制图学、组件电子地图、地图符号和地理信息系统理论与方法研究。

E-mail: somewxf@163.com

Design and Algorithm Optimization of Complex Linear Symbol

WU Xiaofang^{1,2,3} DU Qingyun^{1,2} XU Zhiyong^{1,2} CAI Zhongliang^{1,2}

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

(2 Key Laboratory of Geographic Information Science, Ministry of Education, Wuhan University, 129 Luoyu Road, Wuhan 430079, China)

(3 College of Information, South China Agricultural University, 483 Wushan Road, Guangzhou 510642, China)

Abstract: A new method of user-defined linetype is put forward to design linear symbol. This method can solve the drawing problems and improve the drawing speed. In addition, with respect to the serious distortion of symbol on the vertices of line, such as symbol breakage, symbol intersection, the methods of symbol self-adaptability and double affine transformation are used to manipulate the distortion, which promotes the visual effect of symbol and meets the need of linear symbol with different styles.

Key words: linear symbol; visualization; symbol self-adaptability; double affine transformation

About the first author: WU Xiaofang, Ph. D. Her major research interests include cartography, COM electronic map, map symbol and GIS theory.

E-mail: somewxf@163.com