

基于特征元的符号库数据结构及算法探讨

蔡先华, 武 利

(东南大学 地理信息工程系, 江苏 南京 210096)

Study of Symbol Library Data Structure and Algorithm Based on Property Unit

CAI Xiarr hua, WU Li

(Department of Geographic Information Engineering, Southeast University, Nanjing 210096, China)

Abstract: Spatial data symbolization is an important method for GIS data visualization. It involves the most computer work in GIS processing. Therefore, to promote the efficiency of spatial data symbolization is vital for enhancing the capability of GIS software. This paper has proposed an algorithm design based on property unit, taking into account the global optimization and symbol object orientation. A property-based data structure and algorithm is also proposed. Application results show that it not only has more advantages over the old algorithm, but also expands the functions of symbolization.

Key words: vector data; symbolization; global optimization; data structure; algorithm

摘 要: 空间数据符号化是 GIS 中数据可视化的重要手段, 是目前 GIS 实现过程中计算机时间开销最大的工作之一。提高空间数据符号化效率, 是提高 GIS 软件性能的重要方面, 本文提出考虑全局优化、面向符号对象整体、基于特征元的算法设计思想。据此提出一种基于特征元的符号库的数据结构及实现算法。实验表明, 运用该算法及相关数据结构的空间数据符号化软件不仅效率优于传统算法, 而且扩展了符号库法实现空间数据符号化的功能。

关键词: 矢量数据; 符号化; 全局优化; 数据结构; 算法

1 引 言

符号化是 GIS 中空间数据可视化的重要手段, 多数 GIS 软件系统采用符号库法进行空间数据符号化。该方法计算工作量大, 时间开销大。用户在进行图形处理时, 有一个明显的等待过程。这种现象在数据量增大和符号形状复杂时更加明显。提高符号化速度, 尽可能减少用户等待时间是软件开发设计中必须追求的目标。有些 GIS 软件系统在图形编辑过程中采用不进行符号化来

提高速度, 但牺牲了空间数据可视化表现效果。在硬件条件不变的情况下, 通过设计更优化的符号库数据结构、提高符号化算法效率, 是解决这类问题的有效方法之一。

2 符号化基本原理

在处理及应用空间数据过程中必须符号化, 最初使用子程序库法, 目前主要用符号库法, 即根据符号库中存放的符号信息对空间数据进行符号化。符号库可以由软件开发商建立, 也可以由用

户通过编辑软件进行编辑。

采用符号库法对点状要素符号化非常方便, 只要对符号库中的符号特征点进行几何变换即可完成符号化; 对面状要素符号化也比较方便, 只需把符号库中面状图案有序地拼接起来, 对边缘部分进行裁剪即可实现; 但对线状要素符号化比较复杂。实现时必须根据线状地理要素的空间几何位置进行符号变形处理。符号化程序设计算法复杂, 运算量大, 耗费时间长。

本文主要针对线状符号进行研究。对于点状、面状符号, 实现算法及信息块数据结构相对容易, 本文不作说明。

2.1 符号库法符号化的基本思想

用来描述线状地理要素的符号一般是由某一图案沿线状要素的定位线串接而成的, 这种在线状符号中重复出现表示的图案是线状符号的基本单元, 故又称线状符号的重复元。如图1所示是陡坎、铁路符号及其重复元。

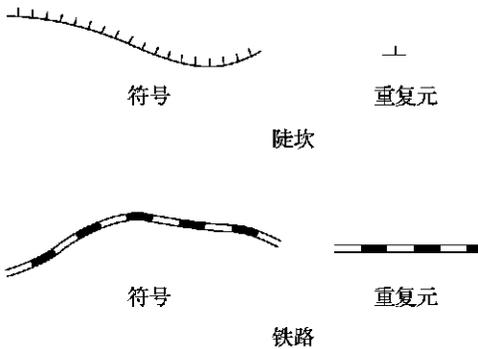


图1 线状符号图案重复元

Fig. 1 Repeat pattern of line symbol

线状要素符号库中存放符号重复元的矢量或栅格数据, 为符号化提供必要的符号图形信息, 通过一个符号化程序实现空间数据符号化。一般线状符号信息块数据结构框架如图2所示^[2]。

颜色 码	特征 点数	图案 长	抬落 笔 1	x_1	y_1	...	抬落 笔 n	x_n	y_n
---------	----------	---------	-----------	-------	-------	-----	-------------	-------	-------

图2 线状符号信息块数据结构

Fig. 2 Data structure of line symbol information block

2.2 线状地理要素符号化实现方法

数据符号化是将符号信息块特征点坐标(x, y)变换至空间坐标(X, Y)并按给定顺序连线的过程。其重复元必须在空间数据定位线上分段串接, 在线段转折时作变形处理。重复元信息块的

x 轴与定位线的方向一致, y 轴与定位线垂直, 如图3所示。

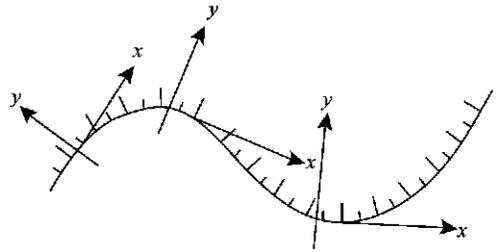


图3 重复元信息块坐标系在符号化的变化

Fig. 3 Change of repeat pattern coordinator system

符号库法目前还不能解决所有的线状地理要素的符号化, 其符号化的前提是线状符号可以分解为重复元。因此, 面向重复元的符号库法只能符号化可分解为重复元的线状符号。然而, 有些线状符号是不能分解为重复元的, 如地形图中的电力线、通讯线、渐变单线河流等的符号在特征点上变化。这类符号的符号化一般采用其它方法进行(如针对某一符号编写程序)。

3 符号库法符号化程序算法分析

3.1 面向符号整体与面向重复元的符号化算法分析

面向重复元的算法设计基本思想: 在程序设计中考虑如何绘制一个重复元, 使“重复元”串联形成线状符号。

面向符号整体的算法设计基本思想: 在程序设计中考虑符号的整体性, “重复元”为符号化提供图形信息, 实现时以点符元、线符元为基本图形符号化。

由于两种算法的基本思想不一样, 在符号化算法设计时, 运行效率有较大的区别, 通过如下基于不同设计思想的平行线符号化算法可以说明这一点。

面向重复元的平行线绘制算法框架——算法

1:

```

Draw Parallel- 1(data xy, 重复元 pattern)
{ 据重复元数据提取相应信息;
  for (i = 0; i < n; i + +) // n 为平行线节点数
  { 计算当前相邻两节点距离(当前处理直线);
    if (未处理重复元长度 < 相邻两节点距

```

```

离) then
{ 处理直线开始部分图案(考虑弯曲因素);
  while(处理定位线长度 > 一个重复元长度)
  { 绘制一个重复元;
    处理定位线减一个重复元长度; }
}
考虑弯曲因素, 处理重复元落在直线的结束部分;
计算重复元未处理长度; }

```

面向符号整体设计思想的平行线绘制算法框架——算法 2:

```

Draw Parallel- 2(data xy, 线宽 t)
{ 准备绘制第一条直线段的首点及第一条直线相关参数;
  for (i = 2; i < n; i++) // n 为平行线特征点数
  { if (相邻三点不在一条直线上)
    { 计算机 i 节点对应平行线上坐标;
      绘一平行线直线段; }
  }
  处理最后直线段;
}

```

在面向重复元的算法中, 有一个循环嵌套。分别在外层循环开始与结束处对重复元进行变形处理, 一个特征点处需处理两次。内层循环处理落在当前直线段上重复元的绘制, 当前直线越长, 循环次数越多。两重循环的总次数与线状符号所要绘制的重复元个数(重复元个数与曲线长度成正比)相关。表示线状几何位置的点数只与外层循环相关, 是算法中总循环次数的决定因素之一(另一个重要因素是地理要素的空间长度)。面向符号整体设计思想的绘制算法中只用了一个循环, 循环的次数是线状要素的特征点数减 2, 而与线状符号的长度不直接相关, 弯曲部分处理在循环内, 一个弯曲只需处理一次, 算法也比较简单。

算法 1 的时间复杂度为

$$T_1(n) = O_1(n * m)$$

式中, n 为地理要素定位线特征点点数, m 为组成曲线的直线段平均重复元个数。

算法 2 的时间复杂度为

$$T_2(n) = O_2(n)$$

由此可见, 由于绘制平行线的两种算法出发点不同, 算法的时间复杂度也不一样。算法 1 的时间复杂度是平方阶, 算法 2 的时间复杂度是线性阶。这主要是因为面向重复元的算法中, 所耗费时间与线状要素的长度密切相关, 而与其形状的复杂程度不直接相关。面向整体符号的算法所耗费时间只与线状要素的形状的复杂程度密切相关, 且处理弯曲的方法简单。

3.2 考虑全局优化、面向符号对象整体、基于特征元的符号化算法基本思路

在符号库法空间数据符号化中, 现有实现算法多是基于重复元进行设计的(算法 1 的设计思想)。通过对面向整体线状符号对象的子程序库法算法效率分析可知, 这种方法比面向重复元处理的算法速度快, 效率高。本文提出基于特征元的符号库数据结构, 提出考虑全局优化、面向符号对象整体、基于特征元的符号化程序设计思路。

4 基于特征元的符号库信息块数据结构

4.1 相关约定

为了便于说明, 对相关名词作如下约定: 本文所涉及坐标系均采用笛卡尔坐标系。

定位线特征点: 本文中简称特征点, 组成空间要素定位线的所有几何点, 一组有序的定位线特征点组成一条定位线。

符号特征点: 组成符号信息的所有几何点, 描述符号的形状。

点符元: 具有一定形状、大小、方向的简单几何图形。点符元在特定线状符号中的定位点位置根据其所在特征元中的定位点位置确定, 形状不因线状要素的定位线弯曲而变化, 方向与其定位点在直线符号定位线上投影点有关。当投影点在两特征点之间时, 方向垂直于两特征点连线; 当投影点在特征点上或在距特征点很小的数值内时, 方向应在该特征点相邻折线的角平分线上。大小随特征元的缩放而缩放。如: 圆(空心、实心)、十字丝、矩形(空心、实心)等。

线符元: x 坐标单调增加的折线组, 在信息块中线符元多是由两点组成(即一条直线), 也可能有两个以上的点组成。在绘制线符元时, 用一定宽度和颜色的实线绘制, 定位线弯曲处必须进行适当弯曲变形处理。

特征元: 描述符号形态特征的基本图案, 在点

状符号中,是点状符号的形态图案;在一般线状符号中是线状符号的重复元;在类似电力线、通讯线的符号中,是特征点处的一个图案特征信息。

4.2 建立符号库特征元信息块的基本原则

根据对点符元、线符元的约定,建立特征元符号库时应遵循如下原则:

1. 特征元由点符元、线符元组成。
2. 线符元符号特征点 x 坐标 ≥ 0 ,当符号特征点在定位线左侧时, y 坐标为正;当符号特征点在定位线右侧时, y 坐标为负。
3. 点符元描述方法:在特征元中的定位点 (x, y) ,形状信息(由符号特征点和相关属性组成);
4. 线符元描述方法:线段在特征元中位置(用符号特征点 $x_1, y_1, x_2, y_2, \dots, x_n, y_n$ 描述)、宽度、颜色及其它属性特征(其中 $x_1 \leq x_2 \leq \dots \leq x_n$)。

根据符号库特征元信息块建立的基本原则,可推出以下结论:①符号特征点 y 坐标为 0 时,点落在线状要素定位线上;②符号特征点 y 坐标不为 0 时,点落在线状要素定位线左侧或右侧;③如果线符元只有两个在定位线上的特征点且其长度为特征元长,则这个线符元所要描述的图形即为线状要素定位线;④如果两个特征点不在定位线上,但 y 坐标相同且线符元长度为特征元

长,则这个线符元所描述的是平行于定位线的平行线。

4.3 线状符号库特征元数据结构

根据基于特征元的符号化设计思想,特征元的框架结构如下

```

{ SymbolID; // 符号特征信息标识
  SymbolLength; // 符号特征元长
  SymbolAttribute; // 符号相关属性信息
  NumberOfElement; // 点符元、线符元数
  FirstElementID; // 第一个图元标识
  ElementAttribute; // 图元属性信息
  ElementInformation; // 图元形状信息
  ...;
  LastElementID; // 最后一个图元标识
  ElementAttribute; // 图元属性信息
  ElementInformation; // 图元形状信息
}

```

5 基本算法框架

算法框架如图 4,在这个算法中充分考虑了符号的整体性,仅用实线、平行线表示的符号时间开销降低,而在所绘制的符号中,很多符号的特征元是由平行线、实线组成的,本算法提出了符号化效率。

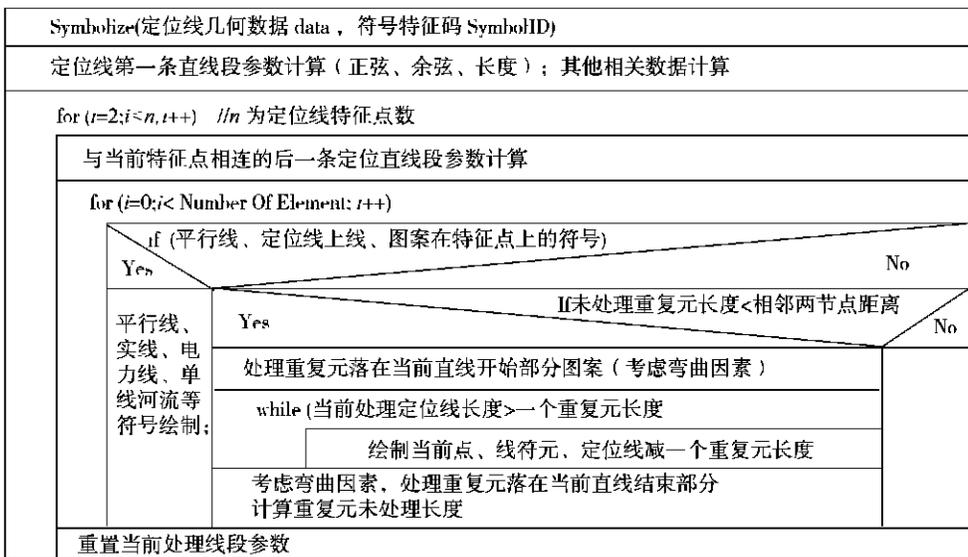


图 4 面向特征元、考虑整体优化的信息库法符号化算法框架图

Fig. 4 Algorithm diagram of symbolization

不仅如此, 由于考虑了全局优化, 面向符号对象实现符号化, 解决了符号库法中如电力线、单线河流等不能分解为重复元的线状符号的绘制。

图 5 是运用本算法实现的一种任意线状符号的图形, 可以看出这种算法可以绘制非常复杂的图形符号。

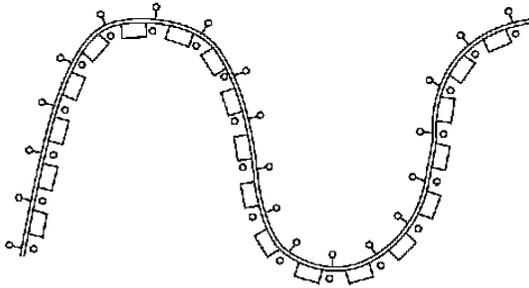


图 5 用本算法实现的一种复杂符号图形

Fig.5 A kind of complex graphic using this algorithm

6 应用与结论

基于特征元、考虑全局优化、面向符号整体的符号库法及相应算法已在作者参与研制的城镇地籍管理信息系统中应用。用 P II 处理器、333 MHz 主频、128 M 内存的微型计算机, 对数据量为 1.2 M、复杂符号(平行线、陡坎、铁路等)分别为 10%、20%、30% 的地图分别用两种算法程序进行符号化。试验结果如表 1:

表 1 两种设计思想符号化软件速度对比

Tab. 1 The comparison of symbolization software speed for two design thoughts

复杂符号/ 比例	基于重复元/ s	基于特征元/ s
10%	0.5~ 1	0.5
20%	1.5~ 2.5	1~ 1.5
30%	3~ 4	2~ 2.5

试验证明, 本文所提出的方法比基于重复元的算法及符号库符号化速度有较大幅度的提高。用户在进行图形编辑处理时, 明显感觉到等待时间的缩短。

矢量数据符号化是 GIS 实现过程中计算机时间开销较大的一项工作。基于特征元、考虑全局优化、面向符号整体的空间数据符号库数据结构为提高符号化速度提供了一种高效算法和符号库数据结构, 基于这种数据结构的空间数据符号化算法可以绘制设计出各种符号, 在为用户扩展符号提供了实现方法的同时, 也提高了符号化实现的速度。

参考文献:

[1] CAI Xiarr hua. The Algorithm on Drawing River Symbol[J]. Bulletin of Surveying and Mapping, 1997, (11): 8-10. (in Chinese)

[2] XU Qing rong, DU Dao sheng, HUANG Wei, FEI Li fan. Principle of Computer Mapping [M]. Wuhan: Publishing House of Wuhan Surveying and Mapping Technical University, 1993. (in Chinese)

[3] YU Liarr sheng, WANG Tao. Map Appearance [M]. Beijing: Publishing House of Surveying and Mapping, 1995. (in Chinese)

[4] CHENG Per gen, GONG Jiarr ya, SUI Hai gang. Design and Implement of Map Symbol Design System in GIS[J]. Journal of Image and Graphics, 2000, (12): 1 006 1 011. (in Chinese)

[5] TAN Xiarr jun, BIAN Fu ling, HE Zhong huan. Object oriented Design and Implementation of Map Symbol Visualization System [J]. Bulletin of Surveying and Mapping, 2003, (1): 11-13. (in Chinese)