

# 专题制图符号库的设计与实现

李 伟 周勇前

(武汉测绘科技大学测绘遥感信息工程国家重点实验室,武汉市珞喻路 39 号, 430070)

**摘 要** 介绍用面向对象方法实现专题图符号库设计,对符号库的组织方式、符号的分类及如何实现用户自行设计符号库进行详细的分析和论述,最后对本文所提出的符号设计思想进行总结。

**关键词** 面向对象;符号;符号库;软件

**分类号** TP 392; P 285

## 1 符号设计

专题符号可分为 3 类:点状符号、线状符号、面状符号。注记可当作点状符号处理

程序实现完全采用面向对象的概念和方法,在 WINDOWS NT 平台下用 VISUAL C<sup>+</sup> 开发。

### 1.1 点状符号设计

点状符号是指定位于某一点的个体符号,又称定位符号,如普通地图上的控制点、独立地物、非比例居民地符号、专题图上的定点符号和定位统计图表等。本系统中的点状符号不仅用作点状地物的符号化,还是线状、面状符号的前期准备。

#### 1) 图元分类及操作

符号是由若干基本图元对象类实例聚集而成的复杂对象。根据符号设计的具体要求,可以将符号分解为以下几种基本图元对象类:点 (CSymPoint)、线 (CSymLine)、折线 (CSymPolyline)、椭圆 (CSymEllipse)、圆弧 (CSymArc)、矩形 (CSymRect)、圆角矩形 (CSymRoundrect)、多边形 (CSymPolygon)、B 样条曲线 (CSymBezier)、子符号 (CSubSym)。按照面向对象的概念,可以将不同类中具有相同数据成员或操作方法的类抽象出来,生成一个更高层次的超类。所以,可从以上基本图元类中抽象出超类 CSymElement,如图 1 基本图元对象类的数据成员已包含了位置、线形、填充图案等必要数据,从而使这些图元能满足符号设计的要求。

#### 2) 操作

对符号的任意组成部分进行操作,包括修改位置(图元平移、旋转),修改图元属性(线形、填充图案),删除或复制图元,并能利用子符号(调

整子符号的缩放比例和位置)组合成新符号(如图 2)。为方便符号编辑,用格网和标尺精确定位。

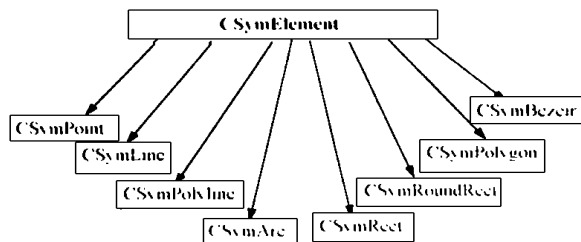
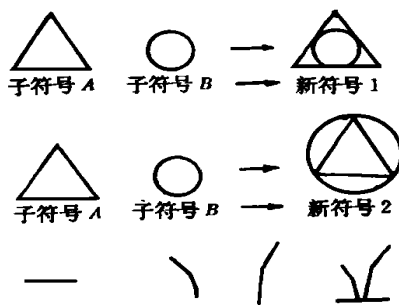


图 1 图元对象类层次图

Fig. 1 Hierarchy Chart of Symbol Element Classes



图元 A (线) + 图元 B (折线) + 图元 C (折线) → 新符号

图 2 由图元或子图组合生成新符号

Fig. 2 Creating New Symbol by Symbol Element or Sub-Symbol

#### 3) 符号图上表达须考虑的问题及解决方法

· 符号大小: 由于在程序设计中实现了符号缩放功能,所以我们将符号大小固定,避免重复存贮大小形状完全一样,只是调用时有所区别的符号,如某些符号在不同比例尺地图中有一些相同的符号,在符号调用中根据实际需要按一定比例缩放。

· 符号定位中心: 一个符号的定位中心选择

的不同会影响符号的缩放、旋转效果。

· 符号方向: 符号在图上的表示有一定的方向,有些符号一律正北方向,有些则垂直于定位线排列

### 1.2 线状符号设计

线状符号是指表示顺线状延伸分布的物体或制图现象的符号。线状符号有一条定位线,可进一步分解为平行线、虚线、连续配置点状符号等。如果将实线也看作一种特殊符号,则线状符号可以分解为一系列点状符号与空(实际是间距)的组合序列:

$$\text{PointSym 1} + \text{Space 1} + \text{PointSym 2} + \text{Space 2} + \dots + \text{PointSym N} + \text{Space N} = \text{LineType}$$

每个点状符号按不同角度和方向沿线配置。事实上,线状符号由不同的点状符号组合而成。所以设计线状符号,应首先设计单点符号,然后组合成线状符号,如图 3



图 3 生成线状符号

Fig. 3 Creating Line Symbol

### 1.3 面状符号设计

面状符号是指地图上用来表示呈面状分布的物体或地理现象的符号,其特点是用不同大小、间距、排列方式的符号填充来表示质量特征或数量差异。面状符号的设计过程与点状符号完全一样,只是用途和调用机制不同。面状符号实际是点状符号在面状区域按某种方式进行排列组合。也就是说,点状符号实际上可以为面状符号做区域填充。调用时要考虑填充阵列方式(水平阵列或斜线阵列)、符号间距(X方向间距和Y方向间距)和缩放比例等因素。设置不同的参数,会产生不同的填充效果。如图 4

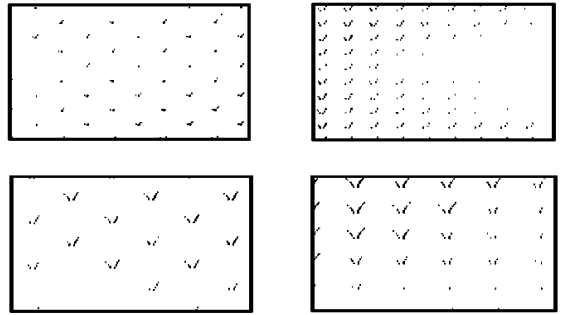


图 4 同一符号填充参数设置不同,产生不同的填充效果

Fig. 4 The Different Filling Effect by Different Parameter Setting

库,选择库中符号,删除库中符号,保存符号库等。最终结果是生成一个符号库。

· 符号编辑,包括增加图元,删除图元,修改图元,线形填充图案等,最终生成符号。

### 2.2 界面

如何让用户自行创建符号库和设计符号,是符号库设计系统的关键。因此,要设计容易被用户接受和使用的界面,要求界面功能简洁而明确,界面应隐藏系统内部细节,在能够实现全部功能的前提下,以便于用户操作为主导思想。

界面主要由标尺、格网、工具条、状态条和菜单组成。标尺及格网便于图元定位,而菜单和工具条保持一致,方便用户操作,并且状态条能够及时提示用户操作结果,如图 5

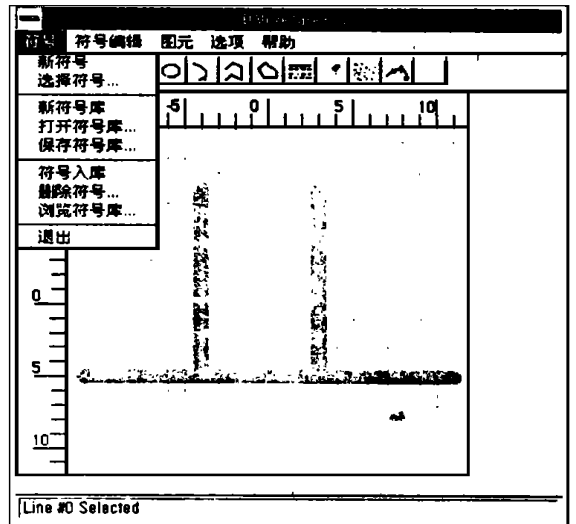


图 5 系统界面

Fig. 5 System Interface

## 2 符号库创建

不同用户的需求各不相同,我们只能在为用户提供一个尽量完备的缺省符号库同时,提供交互式的符号库设计软件,让用户能按自己的需要创建任意符号,编辑符号,并能对符号库进行编辑、修改、增加、删除等操作,建立适合用户需求的符号库。

### 2.1 功能需求

· 符号库操作,包括新建符号库,打开符号

### 2.3 创建符号库主要步骤

- ① 新建符号库;
- ② 新建符号;

③ 符号编辑,修改符号组成图元位置(图元平移、图元旋转),修改图元属性(线形、填充图案),删除或复制图元,并能利用子符号(调整子符号的缩放比例和位置)组合成新符号;

④ 将该符号加入符号库;

⑤ 将编辑好的符号保存在磁盘上。

### 2.4 编辑符号库

在建立好一个符号库之后,如果要对符号库进行修改,则需要进行如下操作:

① 打开符号库;

② 选择符号;

③ 符号编辑,修改符号组成图元位置(图元平移、图元旋转),修改图元属性(线形、填充图案),删除或复制图元,并能利用子符号(调整子符号的缩放比例和位置)组合成新符号;

④ 删除库中某些不需要的符号;

⑤ 若要增加符号,则要新建一个符号,然后将该符号加入符号库;

⑥ 保存符号库,将编辑好的符号保存在磁盘上。

## 3 结 论

本文提出的面向对象方法实现的符号库设计系统具有如下特点:

1) 点状符号:点状符号是整个符号库的基础。点状符号不但用来表示点状地物,还是生成线状符号和面状符号的前期准备。线状符号和面状符号都可以由点状符号派生而来。

2) 线状符号:用组合的方法生成线状符号,从而将复杂的工作化简,易于理解和操作。

3) 面状符号:因为矢量符号和栅格符号各有优缺点,所以本符号库设计系统中各取其长,库文件存贮的是矢量数据,在填充时采用的是栅格填充,其中间转换过程是通过程序实现。下面是矢量符号与栅格符号比较:

	填充时考虑边界及裁剪	存贮数据量	符号缩放	变形否
矢量符号	考虑	很小	易	不变形
栅格符号	不考虑	大	难	变形

同一符号根据参数不同可生成不同的填充图案。本系统不但能够达到与 WINDOWS提供的系统图案相当的填充效果,而且符号库远比 WINDOWS所能提供的系统图案更为丰富。

4) 实现了符号设计可视化,用户设计符号过程直观,界面简洁易用,所见即所得。

### 参 考 文 献

- 1 Cheng P, Gong J. Research of Map Symbols Organization Method in Object-Oriented GIS. *GeoInformatics Proceedings of "GeoInformatics 96 Wuhan-International Symposium"*. Wuhan, 1996.
- 2 徐庆荣,杜道生,黄伟,费立凡.计算机制图原理.武汉:武汉测绘科技大学出版社,1993.
- 3 尹贡白,王家耀,田德森,黄采芝.地图概论.北京:测绘出版社,1991.
- 4 张克权,黄仁涛等.专题地图编制.北京:测绘出版社,1991.
- 5 祝国瑞,苗先荣,陈丽珍.地图设计.广州:广东省地图出版社,1993.

## The Design and Implementation of a Symbol Base in Thematic Mapping

Li Wei Zhou Yongqian

(National Laboratory for Information Engineering in Surveying, Mapping and Remote Sensing, W TUSM, 39 Luoyu Road, Wuhan, China, 430070)

**Abstract** The design and implementation of a symbol base, by object-oriented approach in thematic mapping, is introduced. Then organization of the symbol base, symbol classification and how to develop a symbol base by the user are discussed. Finally, a brief summary about the symbol designing thoughts is presented.

**Key words** object-oriented; symbol; symbol base; software