

专题图符号扫描输出中的图案方法

艾自兴

摘 要

本文基于用图案来组成图形符号的基本思想,引入图案拓扑等价的观点,较系统地专题地图中点、线、面三类符号的绘制作了一些探讨。

【关键词】 专题图; 符号; 图案; 栅格

一、引 言

专题地图内容广泛,形式多样,其符号设计和使用千变万化。应当为每一种符号设计一种专门的绘图程序呢?还是应该设计较为通用的程序,使之能画尽可能多的符号?显然,对前者的回答应该是否定的。否则的话就会费大量的人力、物力。然而,要设计一种通用程序,使之能画出所有的符号也是不现实的。因此,应该寻求对后者的肯定的答复。

作者拟对图案的概念进行扩充,引入图案拓扑等价的观点,较系统地对专题地图中点、线、面三类符号的绘制作一些探讨。

图形信息能够以各种数字形式来表示。其中,对机助制图来说“矢量形式”与“栅格形式”最为重要。这两种形式各有其优缺点,互为补充。因此本文在数据处理过程中,并不限定使用某种固定形式,而试图将两者结合起来,灵活应用。

二、点状符号绘制的软件设计

1、基本思想

点状符号在专题地图中是用途较广的表示方法之一。如居民点、工农业企业、学校、气象站等。点状符号的数量很多,按其形状可分为三种:几何符号、字母(或汉字)符号、艺术符号。通过对这些符号的分析对比,我们发现这些符号本身可以看成是由某些图案所组成

本文1985年11月收到。

本文为研究生毕业论文的一部分,指导教师是张克权、徐庆荣副教授,黄伟讲师。

的。通过对这些图案的全等或相似变换，就可以得到所需绘制的符号了。

在手工绘图的情况下，人们对于一些比较简单的符号，就直接绘到地图上相应的位置。而对于一些比较复杂的符号（如艺术符号），则采用照相植字的方法。

机助制图可以模仿人工的过程。这时我们把简单符号和复杂符号都一样对待。使用图案符号库*的方法，将图案符号贮存起来。在需要绘制符号的时候，将所需要的图案符号从图案库中调出来，然后再置于地图上相应的位置。

2、基本算法

本文在试验过程中把整个图幅贮存在磁盘上，需要时从中取出某个有关部分来进行运算，这些有关的部分就称做窗口。

在专题地图中，当两个以上的点状符号距离很近时，常会出现符号相互重迭的现象，因此对其中的某些现象，需要进行消除“隐藏线”的处理。

点符号绘制过程中的主要框图如图1所示。

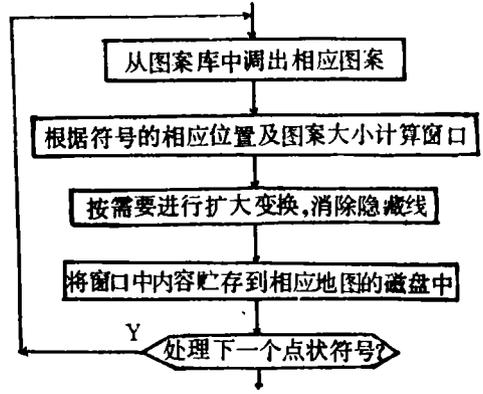


图1

三、线状符号绘制的软件设计

1、基本思想

线状符号在地图中使用是很广泛的。如境界、道路、河流等。其种类也是比较繁多的。一般说来，如果对图案这一概念的理解仅限于欧氏几何意义上的话，那么线状符号就不能是由图案来组成的。但是，如果我们对图案概念进行扩充，使它不仅可以进行欧氏几何中的变换，而且还可以进行拓扑变换的话，则线状符号是能够由图案来组成的。

这里，对图案的概念进行这样的扩充是可能的，也是必要的。事实上，在地图制图学中，常用图例的方法来说明地图上各种要素所代表的物体。如图2，其中a为图例，b为某线状物体（这种符号在地图上一般表示境界、道路等）。从地图制图学的角度来说，图形a与b都代表了同样的东西，比如说它们都代表了某种等级的道路。为什么图形a与b都代表同一级别的道路呢？或者说为什么根据图形a及图形b的图形结构，就能说它们代表的是同一类型的物体呢？我们发现，除非线状符号b是直的，否则图形a与图形b（任取其中两节）不是全等的，也不是相似的，也不会是其它在欧氏几何意义下的变换。



图2

为了了解图形a与b之间究竟有什么关系，我们可以设想图形b是这样得到的。设有一根与b等长的橡皮条，按照a的形状在其上画成如图3a所示。然后将橡皮条的中心线重合

* 限于篇幅，对图案符号库的设计和使用的详述。

到线状符号的中心线上去，就可得到图 3 中图 b 的情况了。在这个过程中，不能把橡皮条撕裂开，也不能把不同部分粘起来。由此可知，图 3 中的两个图形是拓扑等价的。然而这是一个特殊的拓扑变换，它要求沿着中心线的宽度及各节图形在中心线上的长度不变。

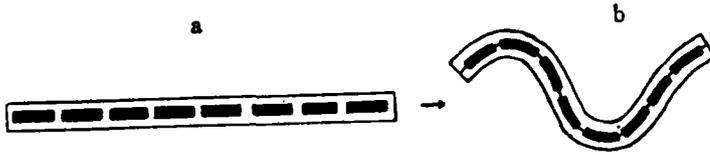


图 3

进而我们又可知图 2 中图形 a 与图形 b 中的任两节图形是拓扑等价的。因而我们也可以将图形 b 看成是由一节节的图形 a 经拓扑变换后再串接而成的。

如果将上面的图形换成实线、双线路等线状符号，其结论还是相同的。

图 4 中 a 是一节放大的图案符号，b 则是经过相应的拓扑变换后的图形。现在要求出拓扑变换 F 。即要求出一种能在计算机上实现的将图形 a 变换成图形 b 的一种算法。

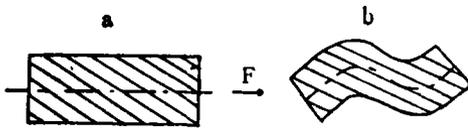


图 4

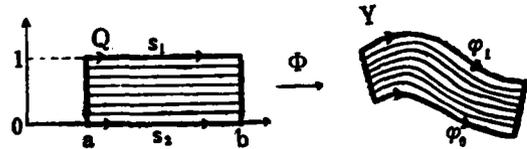


图 5

我们发现，拓扑学中曲线同伦的几何意义（图 5）与图 4 有不少相似之处。

定义：令 φ_0 与 φ_1 为空间 Y 中两条曲线，它们的定义域是同一个区间 $[a, b]$ 。则 φ_0 到 φ_1 的同伦是一个矩形 Q 到 Y 的映射，使 Q 的底边映成曲线 φ_0 ，而顶边映成 φ_1 。明确地说，令 Q 为两变量 (t, τ) 平面中的矩形： $a \leq t \leq b$ 及 $0 \leq \tau \leq 1$ 。则 φ_0 到 φ_1 的一个同伦是一个映射 $\Phi: Q \rightarrow Y$ 使得

$$\Phi(t, 0) = \varphi_0 t; \text{ 与 } \Phi(t, 1) = \varphi_1 t$$

对于所有的 $t \in [a, b]$ 。其几何意义如下：

画出由水平线段 S_τ 组成的矩形 Q ，每一值 $\tau \in [0, 1]$ 。有一条 S_τ 。 Φ 在线段之一 S_τ 上的限制决定一由 $\varphi_\tau t = \Phi(t, \tau)$ 所定义的曲线 $\varphi_\tau: [a, b] \rightarrow Y$ 。于是得到一族曲线，对于 0 与 1 之间每一个值 τ ，就有一条曲线（图 5）。如果把 τ 看作时间变量，那么就可把这族曲线看成是单独一条动曲线的各种位置。 Q 中每一个铅垂线段映成动曲线的一点所描出的道路。

结合本文的实际情况，由于本文所绘的图形都是在二维平面上进行的，故我们还可以进一步得到一种特别的同伦，即线性同伦。对于 Q 中的各对 (t, τ) ，定义线性同伦

$$\Phi(t, \tau) = (1 - \tau)\varphi_0 t + \tau\varphi_1 t$$

即线性同伦 $\Phi(t, \tau)$ 为把从 $\varphi_0 t$ 到 $\varphi_1 t$ 的线段以比 $\tau: 1 - \tau$ 分割的分点（图 6a）。 Φ 在 Q 的各竖直线段上的限制是一相似映射，因为比的保持是相似映射的特征。图 6b 给出线性同伦的说明，图中已画出动曲线在时刻 $\tau = 0, \frac{1}{4}, \frac{3}{4}$ 与 1 时的位置；以及动曲线上对应于 $t = 0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}$ 与 1 的各点所描出的直线道路。

由此，我们可以得到这样的启发。设曲线 l （其长度为图案符号的长度）相应于某线状符号的中心线的一段。直线段 n 的中点与 l 的起点相交，且 n 与 l 在其交点处的切线垂直，并设 n 的长度为原图案符号的宽度（图 7a）。这样，曲线 l 就相当图 6 中 $\tau = \frac{1}{2}$ 时动曲线的位置，而 n 则相当于动曲线上对应于 $t = 0$ 的各点所描出的直线道路。现假设线段 n 从 l 的起点沿着 l 向 l 的终点运动。在运动的过程中，保持 n 的中点与 l 相交，且垂直于 l 在该交点处的切线。如果把 n 运动时所掠过的区域用晕线表示出来，那么当 n 从 l 的起点运动到 l 的终点时，就可以得到一个区域图形（图 7b）。这正是我们所需要的图形。

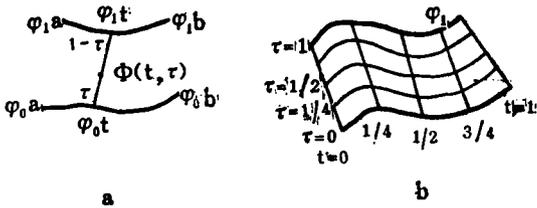


图 6

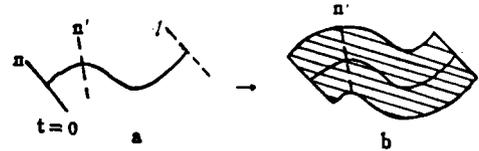


图 7

2、基本算法

(1) 单节符号的情况

① 单线符号

在图 8a 中， i, i' 分别为线段 l 的起、止点， l 为线状符号的中心线的一段，其长度与图案的长度相等。我们采用逐点加粗的办法来代替前面所说的线段 n 的扫描运动。从 l 的第一个象点 (i 点) 起，使加粗的宽度与图案宽度相等，逐点加粗，直到 l 的末点 (i' 点)。图

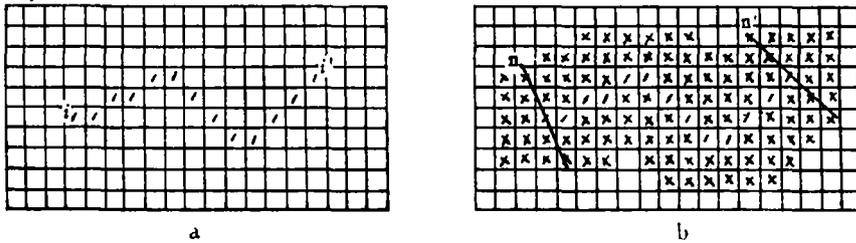


图 8

8b 给出了用八向邻域逐点加粗后的结果。但该图形还不是所要求的结果，在起、止两端点处，还有一些多余的象点，即线段 n 的左面以及线段 n' 的右面的那些点。使用一定的法则，就可以去掉那些多余的点。最后可得结果如图 9 所示。这样我们就完成了图案与其相应的拓扑等价图形转换的问题。

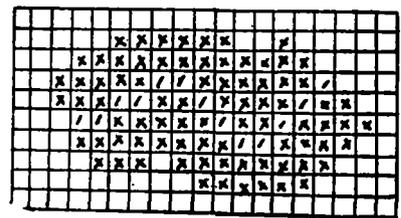


图 9

② 双线符号

图 10a 表示某节双线符号的图案。先把该图案看成是实心的图案。用前面讨论过的方法，进行相应的拓扑变换，得到结果如图 10b。再作一个比图形 a 窄，而长度相同的实心图案。用同样的方法经拓扑变换后得到图 10c。然后再用图 10c “减” 图 10b，最后即可得到

图10a 经相应的拓扑变换后的图形了 (图10d)。

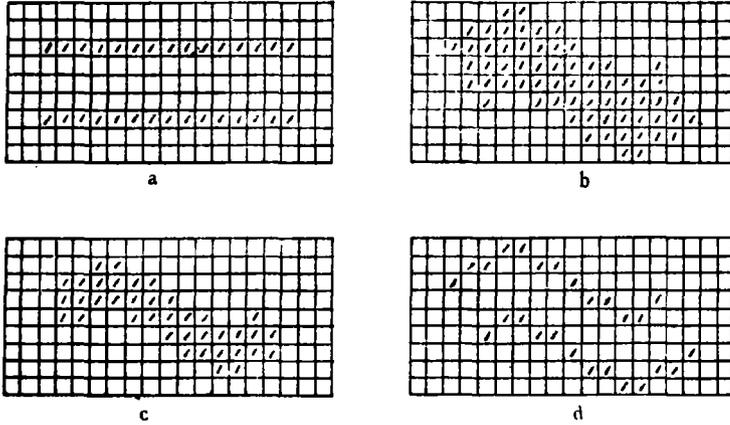


图10

(2) 多节符号的情况

上面讨论了如何将线状符号的图案经过拓扑变换后, 变到线状符号的某节相应位置的情况。下面再讨论这些经过拓扑变换后的图案是如何串接而成我们所需要的线状符号的。

① 连续线状符号的情况

先看单线符号的情况。当所需求的线状符号的中心线是直线, 或有弯曲, 但该弯曲不好在两节图形符号的邻接处, 则一般不会有什问题。如果线状符号的中心线正好在两节相邻图形符号的连接处拐弯, 这时就会出现图 11a 中的情况。即在两节相邻图形符号的连接处, 出现了一个空白区。这是不符合制图要求的, 必须进行处理。

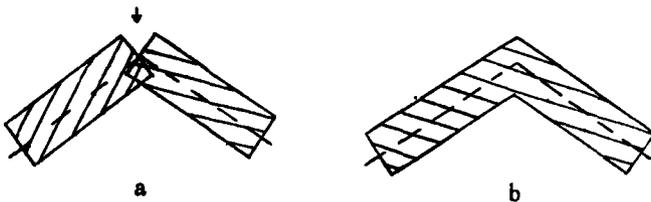


图11

通过分析, 我们发现, 这个空白区的象元点, 在对各节图案符号的拓扑变换过程中, 在逐点加粗时, 曾经

是被算出过的 (图 8)。只是后来又被作为多余的点子而被去掉了。因此在这里, 我们将保留这些“多余”的点, 而只是对整个线状符号的第一节图形的起始端和最后一节图形的末端才去掉这种多余点, 并在取窗口时使之略大于各节图案, 由此可得图形 (图11b)。

通过分析, 我们发现, 这个空白区的象元点, 在对各节图案符号的拓扑变换过程中, 在逐点加粗时, 曾经

对双线符号的情况, 当所需绘出的线状符号的中心线是直线, 或有拐弯, 但该拐弯不在两节相邻图形的连接处附近时, 则属于正常情况。但是当线状符号的中心线在两节相邻图形符号的连接处附近拐弯时, 不管线状符号的走向是从 i_1 到 i_n , 还是从 i'_1 到 i_n , 当 P 点与 i_n 点之差的绝对值小于某个定数 (比如说图案符号宽的一半) 时, 就会出现图 12a 中的现象。这是不允许的, 应去掉那些多余的点。

假定该线状符号的走向是从 i_1 到 i_n 的方向, 我们可以将图 12a 中 i_1 到 i_n 这节图形符号加长, 将该节符号的末端加长到 i'_n 点, 使得 i'_n 到 P 点的距离大于该符号宽度的一半, 问题就解决了 (图 12b)。

如果这时该线状符号的走向是以 i'_1 到 i'_n 点的方向, 我们仍可用调整图案长度的方法来

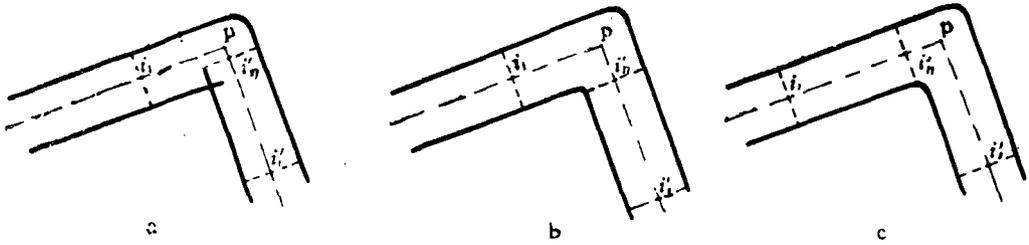


图12

解决问题。调整 i_1' 与 i_2' 间的长度，使其变为从 i_1' 到 i_2' ，结果如图 12c 所示。

② 非连续线状符号的情况

当所要画的线状符号的中心线转角不是很大，或图案之间的间隔较大时，则各节符号的串连并不须作另外的特殊处理。但是当所要绘的线状符号的中心线较为曲折且图案间隔又较小时，会出现两节图形符号相互压盖的现象（图13a）。这时可调节 i_1 到 i_2 的长度，使其成为如图 13b 所示即可。

非连续双线符号也有类似的现象，其处理方法基本相同，这里就不再赘述。

上面讨论的绘制线状符号的方法，都是基本的方法。在此基础上，很容易得到其它一些线状符号的绘制方法。如在各节图案之间加上点状符号，即可得到点线符号。逐渐改变各节图案的宽度，就可得到渐粗线的画法等。

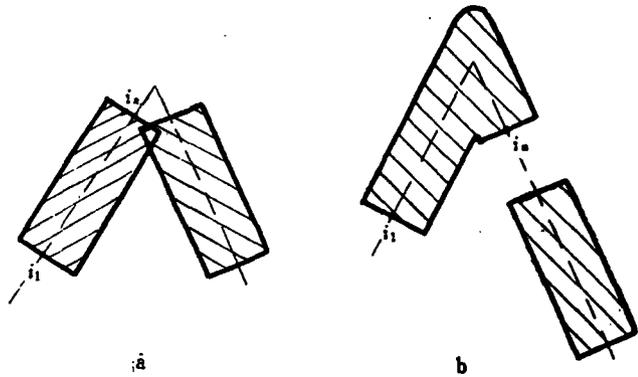


图13

另外，用本法处理双线路交叉时情况也很容易。其基本思想是先把所有的双线路都按同宽的实线来运算，并把运算结果存入磁盘中。然后沿各双线路中心线位置上，作比双线路窄的实线，将其与磁盘上的双线路同宽的实线相“减”即可。

四、面状符号绘制的软件设计

1、基本思想

面状符号的种类很多，它在专题地图中的使用是很普遍的，如各种使用质底法、范围法来表示其内容的地图，以及各种统计地图等，都可以看到面状符号的使用。

绝大部分面状符号都可以看成是由图案组合而成的。因此，使用图案来组成各种面状符号是很自然的。

将图案组合成所需的面状符号的基本思想如下：

设给定面状符号的区域轮廓如图 14a 所示。使用某种方法将该区域内“填实”（图 14b）。根据选定的图案，将其排列成规则图形其面积应能复盖整个区域（图 14c）。用图 14c 与图 14b 作“逻辑乘”，得到图 14d。再将图 14d 中残缺符号清除掉，最后就得到所需的面状符号（图 14e）。

2、基本算法

在实际应用中，有时面状符号并不独立，而某一面状符号与另一面状符号连接在一起。这时，我们可以使用前面曾多次用过的开窗口的办法，一个窗口装一个区域。这样，就可以分别对这些区域进行处理。其主要过程如图 15 所示。

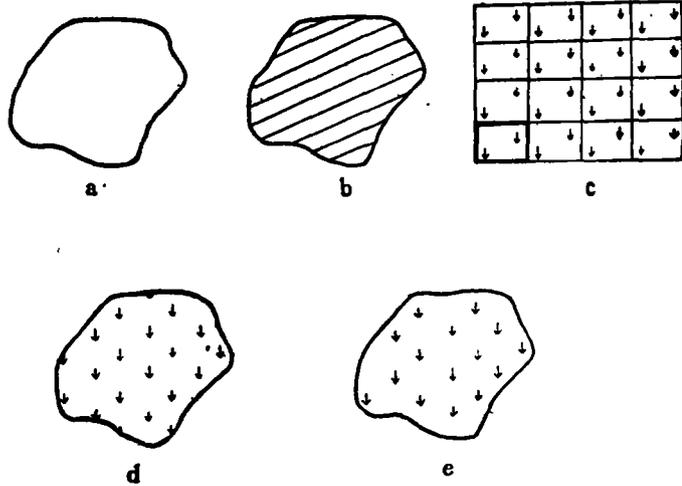


图14

将充满图案的矩形区域与填实后的区域范围作“逻辑乘”后，会出现两种情况。一种是类似晕线类型的图案所组成的面状图形符号，它们本身的图形不象一些用个体的图案符号组成的面状符号图形那样，总能构成完整的个体图形。另一种由个体图案组成的面状符号却会出现图形不完整的现象，如图 14d。故这时应清除掉那些残缺图形。

设某面状符号的区域轮廓由象元序列 $1, 2, \dots, K$ 个象元点组成。从 i 等于第一点起，直到最后一点 K ，逐步按 i 的八向邻域进行搜索。如果在 i 的八向邻域内，除轮廓线外，其它象元出现了不为零的情况，则说明在 i 点处有残缺的个体图形。由此就可进一步将那些残缺图形逐个清除掉了。

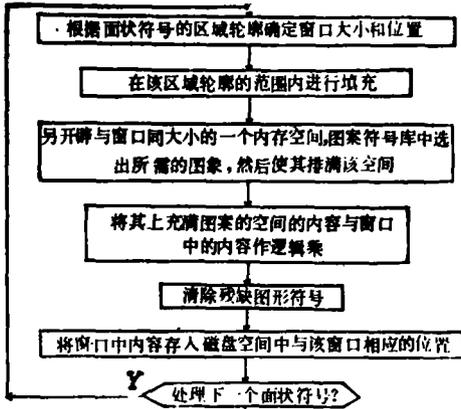


图15

五、结 束 语

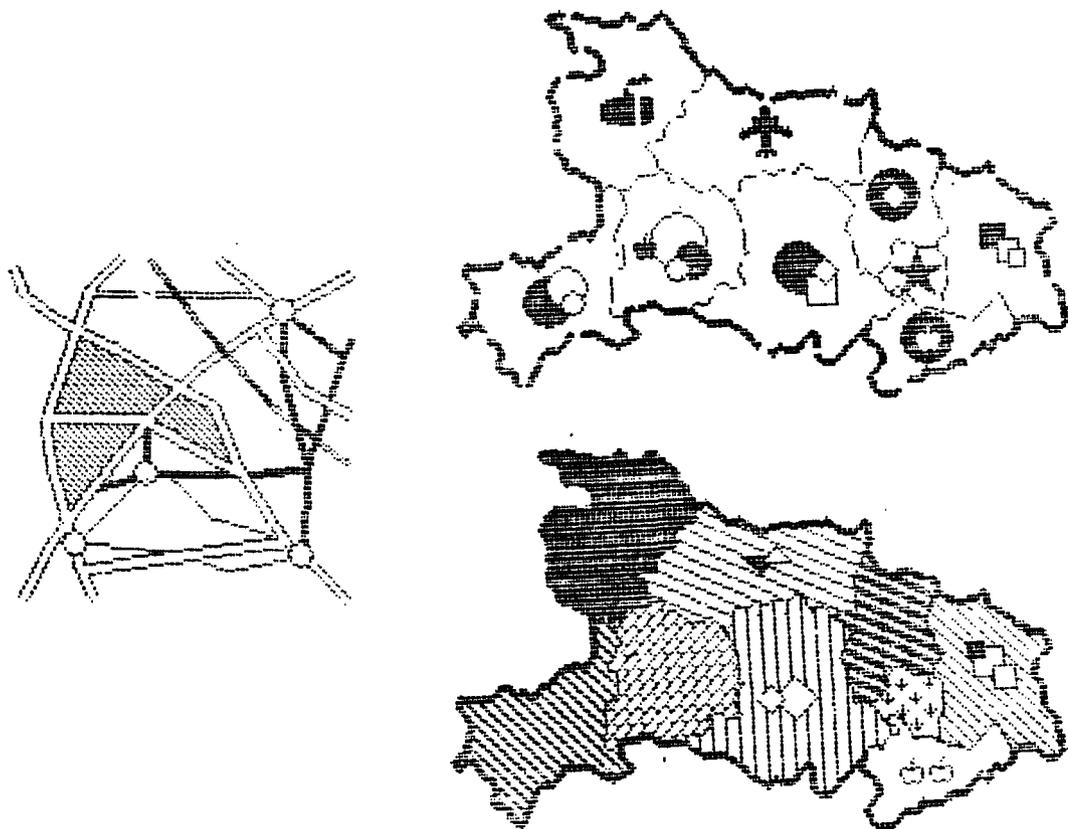
本文基于图案来组成图形符号的基本思想，对专题图的点、线、面三大类符号绘制作了较系统的探讨，初步达到了用少量的“通用”程序来画尽可能多的专题图符号的目的。

本文论述的方法，还对专题地图各要素之间相互关系的处理带来了很大的灵活性（如

点、线、面各符号之间的配合使用等)。

由于设备的限制,本文是用行式打印机来代替栅格绘图机输出图形的。因此图形的质量、精度等不太理想,如果是使用栅格绘图机来输出图形的话,其结果将会好得多。

下面是由 Cromenco 微型机上打印输出的部分结果:



参 考 文 献

- 〔1〕 张克权、黄仁涛等, 专题地图编制, 测绘出版社, 1982.
- 〔2〕 国际地图制图协会, 机助地图制图学讲习班文献, 武汉测绘学院印, 1981.
- 〔3〕 陈锡驹、斯延路德 著, 蒋守方、江泽涵译, 拓扑学的首要概念, 上海科学技术出版社, 1984.

A Patterning Method for Mapping by Using Thematic Symbols in Raster Form

Ai Zixing

Abstract

There are many symbols in thematic cartography. These symbols can be classified into three kinds: point symbols, line symbols and area symbols. For computer-aided cartography it would be wasteful if we give each symbol a special drawing program. Therefore, We try to make a commonly used program that draws as many thematic symbols as possible.

This paper is a trial in this aspect. The paper is based on the idea that we can use patterns to form the thematic symbols. Some researches were made systematically for drawing the point, line and area thematic symbols.

The basic ideas of drawing the point, line and area thematic symbols are as follows

1. Point symbols

The point thematic symbols include geometric symbols, literal symbols and art symbols. The symbols themselves can be regarded as patterns. The patterns are stored in the pattern base. when a symbol is to be drawn, the symbolic pattern is obtained from the pattern base. Then after equal or similar transformation the pattern is put on the correct place of the map.

2. Line symbols

we can say the line symbols are made of patterns if we make some topologic transformation on the patterns. The line thematic symbols can be formed by a series of patterns on which the topologic transformation is made.

3. Area symbols

we suppose that the area symbols can be made by filling regularly in the defined area with the symbolic patterns.

Furthermore, the paper discusses also the method of creating and using the pattern base.

[Key Words] thematic map, symbol, pattern, raster