

# 基于多重变换组合的动态地图符号设计

祝国瑞<sup>1,2</sup>

徐智勇<sup>1,2</sup>

吴小芳<sup>1,2</sup>

(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)

**摘 要:**采用面向对象的设计方法, 将符号的状态描述(参量)与过程操作(对不同参量实施不同变换)封装, 采用关键帧-层-图元-几何/非几何属性结构进行符号组织, 从而将地图符号的时间维信息纳入到符号设计之中, 建立了不同的变化速率模型, 实现属性的动态变化; 通过图元与图元之间的二维 Morph 函数变换矩阵, 结合仿射变换, 对符号的几何信息进行快速提取, 从而实现了符号的动态显示。

**关键词:**动态符号; 符号设计; Morph; 可视化

**中图法分类号:**P283.1

纸质地图只能展现地理现象的状态性信息, 而电子地图在此基础上可以跟踪描述过程性信息, 即动态特征<sup>[1]</sup>。文献[2]通过对传统的视觉变量进行扩充, 引入动态特征描述, 提出了动态地图符号的动态参量描述, 并提出了动态地图的概念。

将时间维纳入到动态地图中, 促进了地图学的发展, 使得电子地图在视觉表达形式上有了新的发展, 同时也对动态地图可视化的问题提出了新的挑战。作为地图的语言, 地图赖以存在的基础和信息得以传输的媒介——地图符号, 在动态地图发展的促使下, 产生了动态符号。

## 1 动态地图符号设计

### 1.1 动态符号的视觉参量<sup>[2-5]</sup>

Koussoulakou 与 Kraak 通过引入动态视觉变量(时间变量)的概念来描述 GIS 的动态特征, 指出时间变量包括显示次序、持续时间和变化速率三个主要的视觉变量, 并指出动态视觉变量可以应用于地图中点状、线状和面状符号当中, 表现符号随时间变化的动态特征。

从制图实用的角度看, 动态视觉变量主要包括发生时长、变化速率、变化次序和节奏等变量, 这些动态变量需要借助符号的上述静态变量的变化来描述<sup>[5]</sup>。

### 1.2 动态符号设计

在进行动态符号设计时, 可以利用时间帧来控制符号动画的进程, 然后根据帧与帧之间符号(多个图元组合)的变化, 反映出符号的发生时长、变化速率、变化次序和节奏等动态视觉变量。由于符号图元内在的属性基本与静态视觉变量对应, 故能反映符号的静态视觉变量, 同时由于内在的时间关联、递进从而表达动态视觉变量。

笔者借鉴 Flash 中动画设计的思想, 结合动态符号的视觉参量, 提出了多重变换组合的动态地图符号设计, 其基本思想是: 采用关键帧-层-图元-几何/非几何属性的表达符号, 在符号动态设计时, 利用时间帧来控制符号动画的进程, 建立动态符号的关键帧, 每一关键帧由若干个层组成, 每个层上有一个图元, 每个图元具有几何属性(形状、位置)与非几何属性(颜色、明度、尺寸、旋转等)。相邻关键帧之间, 各个符号层之间的非几何属性采用变换函数获取, 而几何坐标则通过二维 Morph 变换得到, 这样就建立了符号的持续时间内的所有动画信息, 即建立了动态符号的模型, 如图 1 所示。

显然, 关键帧为某一属性发生变化的地方, 起始帧、结束帧也是关键帧。起始帧和结束帧之间的时差表示发生时长, 关键帧之间的时间关系确定了符号的显示次序, 关键帧之间图元的变换由

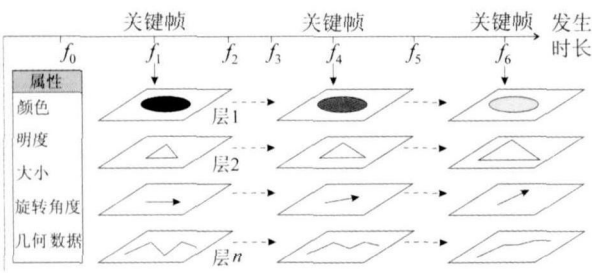


图 1 动态符号模型

Fig. 1 Dynamic Symbol Model

变换函数确定。变换函数可以采用等速变化、变速变化(加速变化或减速变化)、周期变化来表达变化的快慢,即变化速率,如采用加速变换表示人口增长,可反映出人口的快速增长,给读者以较强的视觉冲击力。因此,关键帧中图元之间的不同组合反映出符号这一时刻的静态变量,所有的关键帧之间的图元联合变化则反映了动态视觉变量:发生时长、显示次序与变化速率。

同时,通过在关键帧中引入层的概念,并采用层的 ID 大小表示层的绘制顺序(ID 小的先绘制),这样能表达层的压盖信息,相邻两关键帧之间图元的属性变化,是通过相邻关键帧之间的层建立一一对应关系(层 ID 相同的建立动画关联)。

这样的设计思想具有非常大的灵活性,几乎可以表现任何想表现的内容,它类似于电影的播放模式,很适合表演细腻的动物。另外,可将符号组织成数据库的形式,便于符号的检索、存储、更新、定义和管理。

由动态符号设计的思路可知,建立动态符号的过程就是建立每一关键帧,以及关键帧之间的变换映射的过程。因此,关键帧的确定和设计是核心,在构建动态符号时,首先必须进行关键帧的规划,然后逐关键帧进行设计。在设计一个动态符号时,必须注意的是,不宜采用过多的属性变化表现动态形式,以免影响动态符号的整体表达。

1.3 动态符号提取

由于采用数据库的方式存储符号,因此,在提取符号时,可以充分借鉴数据库检索的优势。符号化时,主要涉及到的是查询操作,即根据符号的 ID 和当前的时间  $curTime$ , 生成所需的符号。步骤为: 提取出当前时间  $curTime$  相邻的两个关键帧  $KeyFrame_1$ 、 $KeyFrame_2$ , 及其图元集合信息  $Elements_1$ 、 $Elements_2$ ; 根据层编号之间的对应关系,建立图元与图元之间变换组  $G(Element_i - Element_j)$ , 其中  $Element_i \in Elements_1$ ,  $Element_j \in Elements_2$ , 而且  $layerID(Element_i) =$

$layerID(Element_j)$ ; 根据插值  $curTime$  和图元组分别得到非几何属性(线性插值)和几何坐标信息(二维 Morph); 循环每一图元,得到其变换后的几何/非几何信息,按照  $layerID$  的大小组织即得到  $curTime$  处的符号。

1.4 图元属性数据提取

动态提取符号时,图元的属性信息可根据相邻两帧的权重系数  $K$  (可反映变化速率)确定,而相邻两关键帧之间的属性变化速率可采用等速变化、变速变化等模式。设  $K = 0$  时,前一关键帧对生成的符号的影响为 1,后一关键帧对符号的影响为 0;  $K = 1$  时,前一关键帧对符号的影响为 0,后一关键帧对符号的影响为 1; 据此可建立不同的变化速率模型。

设通过 SQL 提取出的相邻两关键帧对应的符号为  $Symbol_i(keyTime_i, width_i, height_i, rotation_i, Geometry_i)$  和  $Symbol_{i+1}(keyTime_{i+1}, width_{i+1}, height_{i+1}, rotation_{i+1}, Geometry_{i+1})$ , 当前显示的时间为  $curTime$ , 显然  $keyTime_i < curTime < keyTime_{i+1}$ 。首先将时间归一化处理,得到归一化后的时间  $T$ :

$$T = (curTime - keyTime_i) / (keyTime_{i+1} - keyTime_i) \tag{1}$$

然后建立  $K = f(T)$  之间的关系,得到符号速率变化模型。图 2 为不同的符号变化速率模型。

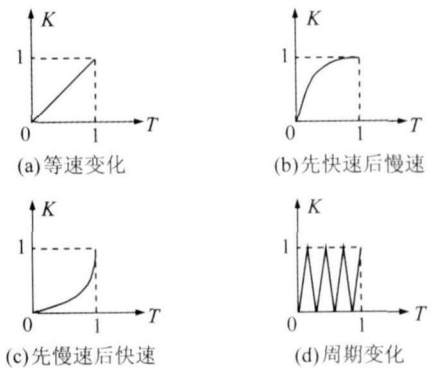


图 2 符号变化速率模型

Fig. 2 Model of Symbol Change Velocity

确定权重系数  $K$  后,图元的属性信息可根据式(2)计算:

$$Value = (1 - K) \times Value_i + K \times Value_{i+1} \tag{2}$$

式中,  $Value \in \{color, size, rotation, \dots\}$ 。

对图元的每一属性,按式(2)计算,即可得到两帧之间图元的所有属性值。

1.5 图元几何数据提取

一般地,基本图元可分为点、直线、折线、曲线、椭圆、三角形、矩形、多边形、字体和位图等<sup>[6]</sup>。组

织图元时,可将图元的坐标单位归一化,称为单位图元,不同大小、方向的图元均根据单位图元按照仿射变换(先按长/宽比例拉伸,然后根据旋转角度进行旋转)生成<sup>[7]</sup>。笔者根据动态符号设计的需求,提出了一种较快的、占用内存资源较少的、可扩展的二维图元变换算法。首先考虑单位图元之间的变换(图元类型相同的情况下,不需考虑),然后对生成的图形进行仿射变换得到所需的图元。

单位图元之间的变换是生成新坐标序列的过程,这些坐标序列根据单位图元的类型从而确定自身的类型(点、线或多边形)。由于从一个单位图元变换到另一个单位图元,其算法不是惟一的,据此,笔者提出了单位图元变换函数矩阵。该矩阵的一个元素  $Matrix(i, j)$  表示图元  $i$  和图元  $j$  之间的变换函数指针,如图3所示。显然,该矩阵为非对称矩阵,矩阵的对角线上某些元素为 NULL(不需要进行坐标计算),如圆和圆、矩形和矩形之间的变换;而三角形和三角形、多边形和多边形之间的变换指针就不为 NULL,需要设置变换函数指针。图4为圆和正方形之间的变换示意图,变换时,只需根据两个图元之间的关键点(图4中的小黑点)建立变换关系,采用线性插值即可生成任意时刻的几何图形。

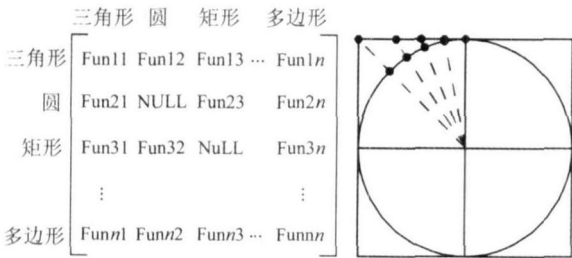


图3 图元变换函数矩阵

Fig. 3 Element Transform Function Matrix

图4 圆和正方形的变换

Fig. 4 Transform Between Circle and Square

建立了这样的矩阵之后,用户可以通过两种方式对动态符号库进行扩展: 增加新的图元,只需要将新增图元与其他已知图元建立函数关系,将该函数注册到图元变换函数矩阵即可; 增加优化的图元变换算法,以获取较好的变换精度和效果。当然,也可将矩阵内的函数指针定位到通用变换算法中,通过侦测图元类型,将图元按一定的精度离散为折线或多边形,然后调用通用变换算法生成图形。

在两个相邻关键帧之间生成的图形,其所含有的坐标数据是不变的(也有变化的可能性),因

此,可采用缓冲的方法,即相邻关键帧之间的坐标生成时,开辟的内存暂不释放,而是直到关键帧发生变换或者需要开辟更多的内存为止,这样能保证变换的速度。

在单位图元之间 Morph 变换生成中间图元后,可根据相邻两帧权重系数  $K$  值,按照式(2)计算图元的宽(width)、高(height)、旋转角度(rotation)。最后对图元进行不等比例拉伸、旋转等仿射变化,形成中间图元,然后逐层叠加构成最终动态符号。

在目前的地图设计中,常常采用三维地图符号,如三维的柱体、长方体、圆锥、球等来表示专题信息,这些图元之间的 Morph 变换,可将现有的3维 Morph 变换算法,注册到图元变换函数矩阵中,实现二维、三维符号的 Morph 的变换一体化。

## 2 动态地图符号应用

图形符号动态化是增强量化地图吸引力的方式之一,它能动态地表达事物的变化特征。通常动态表达可分为两类: 与事物经历的时间相关联,表示事物实际历史动态变化过程; 利用动画产生具有连续移动的可视化效果,引起人们视觉上的注意。即动态符号可以表示时态变化,也可以表达实体的重要性程度、质量差异、数量分级等非时态特征。总地来说,动态符号在地图显示中可以表示三个方面的信息内容: 空间定位、属性特征以及时态变化<sup>[2]</sup>。

1) 突出空间定位。有关视神经感受实验表明,闪烁、跳跃、变化着的现象比静止现象对视觉感受有更强的吸引力,因此对重要地物或需引起用户特别注意的地物可通过动态闪烁符号表示,使用户能在很短时间感受到其存在,从而获取其空间位置。

2) 表达属性时态变化。动态符号对地物的质量差异、数量等级、甚至是多维属性数据均可以可视化的方式表达<sup>[8,9]</sup>,如表达人口数量随时间的变化、车流量随时间的变化等。而且动态符号在空间数据不确定性、定位精度、属性精度的可视化方面有其独到之处<sup>[10,11]</sup>,此外,Lubos Mitas 采用动态制图模型模拟了景观的变化<sup>[12]</sup>,William Acevedo 等将其应用到城市的发展变化研究<sup>[13]</sup>。

3) 表达空间位置、属性时态变化。用动态符号来表现地理现象动态变化过程是最直接的手段。依据时态综合原则在符号的动态参量与实际变化的特征之间建立映射关系,采用符号动态参量表现地理实体的时态特征和变化规律。

图 5 为台风随时间的变化图。其中, 红线为台风中心位置的变化, 周边的多边形范围表示台风的影响区域。图中, 随着台风的移动, 其空间位置、风速大小、影响范围等均随着时间而发生变化。在这一过程中, 笔者采用多重变换组合的动态符号方式来模拟台风的运动过程, 运用动态轨迹表示台风中心的移动, 运用多边形的动态变化反映台风影响范围以及风速等属性变化。

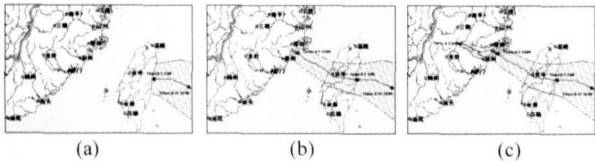


图 5 台风随时间的变化

Fig.5 Change of Typhoon with the Time

参 考 文 献

[ 1 ] MacEachren A M. Time as a Cartographic Variable [ J ]. Visualization in GIS, New York, 1994: 115-129

[ 2 ] 艾廷华. 动态符号与动态地图[ J ]. 武汉测绘科技大学学报, 1998, 23(1): 47-51

[ 3 ] Bertin. Graphics and Graphic Information Processing[ M ]. Berlin: Walter de Gruyter, 1981

[ 4 ] Mennojan Kraak, Arjen Klomp. A Classification of Cartographic Animations: Towards a Tool for the Design of Dynamic Maps in a GIS Environment[ C ]. The Seminar on Teaching Animated Cartography, Spain, 1995

[ 5 ] 祝国瑞. 地图学[ M ]. 武汉: 武汉大学出版社, 2003

[ 6 ] 程朋根, 龚健雅. GIS 中地图符号设计系统的设计与实现[ J ]. 中国图像图形学报, 2000(12): 38-43

[ 7 ] 吴小芳, 徐智勇, 蔡志亮, 等. 基于 GD1+ 的高精度地图符号库的设计与实现[ J ]. 武汉大学学报·信息科学版, 2004, 29(10): 928-932

[ 8 ] Kuo Chunlin, Hanashima M, Tomobe K. Design of Visualization for Historical Statistic Datasets[ C ]. PNC 2004 Annual Conference, Taipei, 2004

[ 9 ] Cook D, Symanzik J, Majure J J, et al. Dynamic Graphics in a GIS: More Examples Using Linked Software[ J ]. Computer & Geoscience, 1997, 23(4): 371-385

[ 10 ] Fisher P F. Visualizing Uncertainty in Soil Maps by Animation[ J ]. Cartographica, 1993, 30(2)

[ 11 ] Ehlschlaeger C R, Shortridge A M, Goodchild M F. Visualizing Spatial Data Uncertainty Using Animation[ J ]. Computers & Geosciences, 1997, 23(4): 387-395

[ 12 ] Mitas L, Brown W M, Mitasova H. Role of Dynamic Cartography in Simulations of Landscape Process Based on Multivariate Fields[ J ]. Computers & Geosciences, 1997, 23(4): 437-446

[ 13 ] Acevedo W, Masuoka P. Time Series Animation Techniques for Visualizing Urban Growth [ J ]. Computers & Geosciences, 1997, 23: 423-435

第一作者简介: 祝国瑞, 教授, 博士生导师, 主要研究兴趣: 地图设计, 地理信息工程和三维可视化。  
E-mail: zhugr123@sohu.com

Design of Dynamic Map Symbol Based on Multi-transform Assembly

ZH U Guorui<sup>1,2</sup>    *X U Zhiyong*<sup>1,2</sup>    *W U Xiaofang*<sup>1,2</sup>

(1 School of Resources and Environment Science, Wuhan University, 129 Luoyu Road, Wuhan 430079, China)  
(2 Key Laboratory of GIS, Ministry of Education, 129 Luoyu Road, Wuhan 430079, China)

**Abstract:** Multi-transform assembly is brought forward to design dynamic map symbol. The data structure of key frame, layer, graphic element, geometry and attribute are used to organize the dynamic symbol. The models of change rate are built to implement the dynamic change of symbol attributes, and the expandable 2D Morph transform matrix of graphic elements is provided, and the affine transform is united to implement the dynamic extraction and visualization of symbol.

**Key words:** dynamic symbol; symbol design; Morph; visualization