

# MapX 下 TrueType 字体符号的扩展

杨 伦<sup>1,2</sup>, 彭春光<sup>1</sup>, 黄 健<sup>1</sup>, 黄柯棣<sup>1</sup>

(1. 国防科技大学 机电工程与自动化学院, 湖南 长沙 410073;  
2. 空军第一航空学院, 河南 信阳 464000)

**摘 要:** 针对 MapX 点符号库的不足, 在分析 MapX 内部数据结构和 TrueType 字体符号的调用机制基础上, 结合 TrueType 字体文件的结构, 提出了一种扩充 MapX 中 TrueType 字体符号库的方法。该方法可以根据 GIS 应用系统的具体需求定制符合要求的 TrueType 字体符号库, 简单易行, 开发周期短, 丰富了基于 MapX 的 GIS 应用程序的 TrueType 字体符号库。

**关键词:** TrueType 字体 (TTF); MapX; 符号库

**中图分类号:** TP391.9

**文献标识码:** B

**文章编号:** 1673-9833(2007)05-0099-04

## Extension of TrueType Font Symbol under MapX

Yang Lun<sup>1,2</sup>, Peng Chunguang<sup>1</sup>, Huang Jian<sup>1</sup>, Huang Kedi<sup>1</sup>

(1. College of Mechanical Engineering and Automation, National University of Defense Technology, Changsha 410073, China;  
2. The First Aeronautic Institute of Air Force, Xinyang Henan 464000, China)

**Abstract:** According to the deficiency of MapX point symbol library and analysis on MapX internal data structure and call mechanism of TrueType font symbol, a new extending method of TrueType font symbol in MapX is put forward by combining the structure of TrueType font file. The method can customize satisfactory TrueType font symbol base according to the specific needs of GIS application system. With the merits of simple, convenience and short development period, it can enriches the TrueType font symbol library of GIS application system based on MapX.

**Key words:** TrueType font (TTF); MapX; Symbol librarys

## 0 引言

MapX 是 MapInfo 公司在为微软开发的可视化地图组件 DataMap 的基础上推出的, 基于 ActiveX (OCX) 技术的可编程控件, 以控件的方式提供了真正的对象链接与嵌入式的开发方案, 支持 Windows 环境下大多数标准的可视化开发环境, 如 Visual Basic、Visual C++、Delphi 等。开发人员可以选用熟悉的开发语言来创建 GIS (Geographical Information System) 应用程序, 通过对 MapX 的编程, 可以实现 MapInfo Professional 的大多数功能, 并且能够脱离 MapInfo Professional 软件平台运行。使用 MapX 开发 GIS 应用系统具有开发周期短、功能强大、可靠性好、易于移植、容易掌握、便

于维护的优点, 因而在实际的 GIS 项目开发中, MapX 的应用非常广泛。<sup>[1, 2]</sup>

MapX 采用空间实体和空间索引相结合的空间数据结构。空间实体是地理图形的抽象模型, 主要包括点、线、面 3 种类型<sup>[1]</sup>。在具体的实现中, MapX 采用图元 (Feature) 来表示这些空间地物, 不同的地物具有不同的样式库。MapInfo 提供的常用的样式和符号可以满足一般情况下的需求, 但是在特定的 GIS 应用系统中, 用户可能需要满足符合一定标准的特殊符号, 例如军事地理信息系统中的军队标号需要满足军队标号的标准, 而 MapInfo 的符号样式库并不能提供。因此, 为满足特殊领域的需求, 就需要开发者在 MapX 下定制自己的符号样式库。

收稿日期: 2007-07-06

作者简介: 杨 伦 (1982-), 男, 河南光山人, 国防科技大学硕士生, 主要研究方向为分布式仿真技术。

## 1 MapX 中的点符号

MapX 中使用的点符号分为位图符号、TrueType 字体符号和矢量符号 3 类。MapX 自带的位图符号有 55 个,存放于如下路径“Program Files\Common Files\MapInfo Shared\MapX Common\CUSTSYMB”。可以利用 Windows 中的画笔或 Visual C++ 的资源编辑器来修改或扩充这些位图点符号,但是位图点符号不能旋转,也不能够随地图的缩放而改变大小。

MapX 中的矢量点符号是按照字体文件的格式存储在 mapinfow.fnt 文件中,该文件在 MapX 中对应了 MapInfo 3.0 Compatible 字体,它的完整路径位于:“Program Files\Common Files\MapInfo Shared\MapX Common\mapinfow.fnt”<sup>[1]</sup>。修改和扩充矢量符号库就是针对 mapinfow.fnt 进行,需要用到 MapInfo Professional 自带的矢量符号生成应用程序 symbol.mbx,这是一个 MapBasic 程序,必须在 MapInfo Professional 平台下运行。利用该程序可以编辑、修改、删除和增加矢量符号,但是在实际制作时不易控制线条的尺寸和相互的比例关系,并且矢量符号在 MapX 中亦不能旋转。

MapX 中的 TrueType 字体符号将点符号广义扩展为字体文件,在 Windows 的 Fonts 目录下,安装完 MapInfo Professional 软件后会生成几个默认的 TrueType 字体文件,如 MapInfo Arrows、MapInfo Symbols、MapInfo Transportation 等。在 MapX 中,可以调用这些字体文件中的 TrueType 字体符号,并且 MapX 支持 TrueType 点符号的旋转。由于 Windows 操作系统的字体文件大多采用 TrueType 格式,很多 GIS 软件中也采用字体方式的点符号,因而,采用 TrueType 字体方式制作的点符号可以在各种 GIS 软件中共享。

## 2 TrueType 字体文件分析

TrueType 是 Apple 公司和 Microsoft 公司合作开发的页面描述语言(Page Description Language,简称 PDL) TrueImage 中的字形描述部分,采用二次 B-样条(Bezier-spline)曲线来描述字体的轮廓,既保证了轮廓曲线的光滑性,又提高了字形的还原速度<sup>[3, 4]</sup>。

TrueType 字体(TrueType Font,简称 TTF)文件由 3 部分组成:文件头(offset table)、描述表目录(table directory)和一系列的描述表(tables)<sup>[3]</sup>。文件头在 TTF 文件的头部,描述表目录紧跟在文件头后。TTF 文件的基础是描述表,其中有 10 个描述表是每个 TTF 文件必须包含的,它们是 cmap(字符代码到文字索引的映射表)、glyf(字模轮廓数据表)、head(文件头信息表)、hhea(水平度量头信息表)、hmtx(水平度量信息表)、loca(字符轮廓数据位置索引表)、maxp(最大值描述

表)、name(名字表)、post(Postscript 打印机控制)和 OS/2(OS/2 和 Windows 度量信息)<sup>[5]</sup>。

由于 TrueType 字体最初是由 Apple 公司定义的,而 Apple 公司的操作系统运行在 Motorola 的 CPU 上,因此 TrueType 字体中的所有数据都使用 Big-Endian 编码,最高位字节放在最前面<sup>[6]</sup>,与 Intel 格式的字节排列顺序正好相反,因此,在处理 TTF 文件的描述表数据时,需要先将 Motorola 格式转化为 Intel 格式。

下面介绍一些 TTF 文件中的描述表。

### 2.1 cmap 表

对于字库中的各个字模(glyph),字体文件本身对其有一个排列的顺序,一般不与系统的字符编码顺序一致,因此,用 cmap 表来建立字符编码到字模索引的映射关系。对于不对应字体文件中任何字模的字符编码,cmap 表将其映射为索引 0,称为丢失字符,通常用一个方框字模来表示。

cmap 表由表头、子表描述目录和一系列子表组成。cmap 表头占 4 个字节,包括 cmap 表的版本号和子表的个数。接下来是 cmap 子表描述目录,每个子表描述目录占 8 个字节,包含 3 个字段,分别是 PlatformID(平台代码)、EncodingID(编码体系代码)和 offset(子表位置偏移)。

为了在不同的平台下使用,TTF 文件采用了不同的译码体系,因而需要包含一系列的子表,每个子表对应于一个不同的译码体系。每个子表对应于一个子表描述目录。PlatformID 指定使用的平台,EncodingID 指定使用的字符集和编码方法。当 PlatformID 值为 3 时,代表 Microsoft 平台,为 1 时代表 Macintosh 平台。

一组 PlatformID 值和 EncodingID 值唯一确定一个 cmap 子表,offset 决定子表的位置。cmap 子表有 4 种可用的格式,分别是 format 0, format 2, format 4 和 format 6。Windows 中一般使用 format 0 和 format 4,其中 format 4 是微软公司标准的字符映射表,采用双字节编码方式,称为双字节分段映射表,结构如表 1 所示<sup>[7]</sup>。

表 1 Cmap 表的 format 4 格式子表

Table 1 Format 4 sublist of cmap table

数据类型	字段名称	描述
USHORT	format	子表格式
USHORT	length	子表长度
USHORT	version	版本号
USHORT	segCountX2	分段数目
USHORT	searchRange	查找范围
USHORT	entrySelector	入口选择
USHORT	rangeShift	范围调整
USHORT	endCount[ ]	段结束字符代码
USHORT	reservedPad	设置为 0
USHORT	startCount[ ]	段开始字符代码
USHORT	idDelta[ ]	段偏移
USHORT	idRangeOffset[ ]	子表偏移
USHORT	glyphIdArray[ ]	字模索引数组

cmap 子表采用 format 4 格式时, 字符编码分成了多个连续的段, 每一段由 startCount、endCount、idDelta 和 idRangeOffset 4 个参数决定, 用以在段中映射字符编码, 其中 startCount 和 endCount 决定了每一段的起始和结束字符代码, 最后一段的起始和结束字符代码都必须是 0XFFF, 作为段结束标志。给定一个字符编码, cmap 子表利用 format 4 的映射算法就可以得到对应的字模索引, 映射算法见如下的 C++ 代码段:

```
// 进行从字符编码到字模索引的映射
for(USHORT charcode=m_FirstCharCode+1;
    charcode<=m_LastCharCode; charcode++)
{
    USHORT index;
    int j;
    for(i=0; i<uSegCountX2/2; i++)
    { if(charcode<=uEndCount[i])
        { j=i;
          break;
        }
    }
    if(charcode>=uStartCount[j])
    { if(uRangeOffset[j]==0)
        { index=charcode+uIdDelta[j]; }
      else { memcpy(&index, buffer+uRange-
        Offset[j]+(charcode-uStartCount[j])*2+2*j, 2);
          index=SWAPWORD(index);
          if(index!=0)
              { index=index+uIdDelta[i]; }
        }
      }else { index=0; }
    m_CharCode2GlyphId[charcode]=index;
  }
}
```

### 2.2 maxp 表

maxp 表确定字体文件对内存的需求, 该表结构中 包含 15 个字段, 以该表的版本号字段开始, 紧接着是字模个数字段, 其它字段用以指定一系列参数的最大值, 例如, maxComponentElemmaxpents 字段指定复合字模包含的最大简单字模个数。

### 2.3 post 表

post 表包含了在 Postscript 打印机上使用 TrueType 字体所需要的额外信息, 其中的数据包括 FontInfo 目录项和字体中所有字模的 Postscript 名字。因此, 可以为字体文件中的每个字模指定一个有意义的 Postscript 名称, 在选择字模时就可以达到“见名知意”的目的, 便于在 MapX 中调用时对字体符号图元的选取。

post 表的结构以格式字段开始, 指定该表使用的格式。post 表有 4 种格式, 分别是 Format 1.0、Format 2.0、

Format 2.5、Format 3.0。其中 Format 2.0 是 Windows 字体使用的格式。

采用 Format 2.0 格式时, 字模名称数组将字体中的字模映射到名称索引。如果名称索引在 0 到 257 之间, 该名称索引被看作是 Macintosh 字体文件标准顺序下的字模索引; 如果名称索引在 258 和 32 767 之间, 利用该值和 258 之差在 post 表格文件尾部的字模名称表中进行索引。索引数值 32 768 到 65 535 作为保留值, 不被使用。对于不需要 PostScript 名称的字模, 可以使用索引值 0, 映射到名称 .notdef。

## 3 MapX 下 TTF 符号库的调用

点符号结构简单, 一般由简单的图形单元(直线、矩形、圆弧、多边形等)组成, 在军事符号库中可用来表示指挥所、武器装备等。首先需要根据具体的应用确定所需 TTF 符号的种类, 对符号进行归类, 对同一类的点符号归到同一个字体文件中。由于在具体的应用中, 点符号需要按照一定的规范进行绘制, 因而可以选用矢量图形软件, 如 CorelDraw, 来设计和绘制点符号, 并导出为 TTF 文件。最后将各个 TTF 点符号统一进行编码命名和建库。

对于已建立的字体库, 需要安装在系统的字体目录下, 才能在 MapX 中进行调用。为了便于符号的选择, 可以设计一个符号点选界面, 如图 1 所示。调用 API 函数 GetGlyOutline 读取字模的字形信息<sup>[8]</sup>, 再转化为位图图象显示。用户使用时就可以根据点符号的位图进行选择, 程序将用户选择的字体库名称及符号的字符编码传递给 MapX, 就可以将用户选择的字体符号在地图图层上以点图元的形式显示出来。

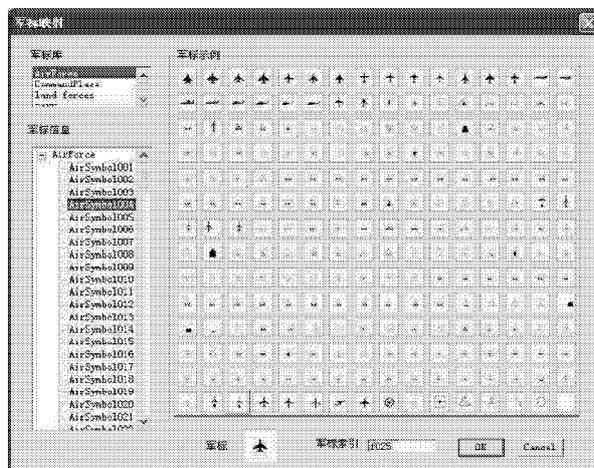


图 1 TTF 符号选择

Fig. 1 Selection of TrueType font symbol

在 MapX 中, 地图图层上的点符号是作为点图元来对待的, 需要首先创建一个图元对象, 再指定图元的类型。MapX 中图元对象包含 Type 属性, 该属性描

述图元的类型,取 FeatureTypeConstants 常量的值<sup>[1]</sup>。对于 TTF 点符号,图元的 Type 属性值需要设置为 miFeatureTypeSymbol,表示该 TTF 点符号在地图图层上代表一个点图元。接着需要对图元的样式进行设置,用以指定 TTF 点符号的样式,如表 2 所示<sup>[8]</sup>。

表 2 TrueType 字体符号的样式  
Table 2 Style of TrueType font symbol

属性名称	属性类型	属性含义
SymbolType	SymbolTypeConstants	TTF 类型
SymbolCharacter	Integer	字符编码
SymbolFont	OLE_FONT	字体属性
SymbolFontBackColor	OLE_COLOR	背景颜色
SymbolFontColor	OLE_COLOR	前景颜色
SymbolFontHalo	Boolean	光晕效果
SymbolFontOpaque	Boolean	透明效果
SymbolFontRotation	Integer	旋转角度
SymbolFontShadow	Boolean	阴影效果

在 MapX 中设置 TrueType 字体符号样式(Style)的 C++ 代码如下:

```
// 设置 TTF 符号 Style
CMapXFeature ftr;
ftr.CreateDispatch(ftr.GetClsid());
ftr.Attach(m_ctrlMapX.GetDispatch(FALSE));
m_objLayer.AddFeature((LPDISPATCH)ftr);
ftr.GetStyle().SetSymbolType(miSymbolTypeTrueType-Font);
ftr.GetStyle().GetSymbolFont().SetName("air service");
ftr.GetStyle().SetSymbolCharacter(46);
ftr.GetStyle().SetSymbolFontRotation(45);
ftr.GetStyle().SetSymbolFontColor(RGB(255,0,0));
```

利用以上代码在某基于 MapX 的态势显示系统中调用定制的 TTF 符号库形成态势图,如图 2 所示。

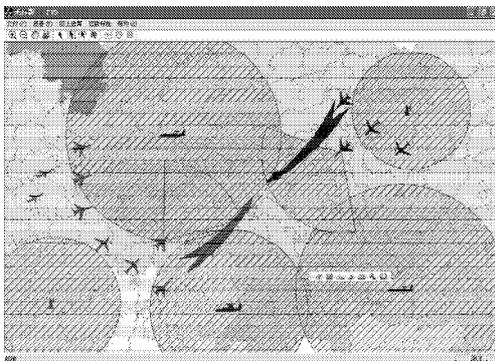


图 2 MapX 中定制 TTF 符号库中符号的调用

Fig. 2 Symbol call of customized TTF library in MapX

## 4 结语

本文针对 MapX 自带的点符号库无法满足基于 MapX 的 GIS 具体应用需求的问题,通过研究 MapX 内部数据结构及其对 TrueType 字体符号的调用机制和分析 TrueType 字体文件的结构,提出了一种扩充 MapX 下 TrueType 字体符号的方法,该方法简单易行,开发周期短,可以根据具体需求开发相应的符号库,丰富了基于 MapX 的 GIS 应用程序的字体符号库。

## 参考文献:

- [1] 齐锐,屈韶琳,阳琳赞.用 MapX 开发地理信息系统[M].北京:清华大学出版社,2003.
- [2] 陈正江,汤国安,任晓东.地理信息系统设计与开发[M].北京:科学出版社,2005.
- [3] 王瑜,黄源,张福炎.Windows 中 TrueType 字形数据的存取技术[J].小型微型计算机系统,1997,18(11):75-81.
- [4] 吕强,史磊,杨季文.TrueType 字体文件格式初探[J].计算机研究与发展,1995,32(11):23-31.
- [5] 朱开玉,宋国辉,燕存良,等.CAPP 系统中动态创建 TrueType 字体的研究[J].计算机工程,2006,32(8):20-22.
- [6] (美)袁枫(Feng Yuan).Windows 图形编程[M].英宇工作室译.北京:机械工业出版社,2002.
- [7] Microsoft Corporation. TrueType 1.0 Font Files Technical Specification (Revision 1.66) [EB/OL]. [1995-08-30]. <http://www.microsoft.com>.
- [8] 杨建红,刘蓉,余泽太.TrueType 字体在图形图像处理软件中的应用[J].武汉大学学报:工学版,2004,37(6):110-112.
- [9] MapInfo Corporation. MapInfo MapX Developer Guide V4.5.1 [EB/OL]. [2001-02-20]. <http://www.mapinfo.com>.