

# MeteoInfo 用户手册

**Version 1.0.1**

# 简介

MeteoInfo 是一个免费的气象数据显示、分析软件，具有较强的综合数据图形显示、分析能力，操作简便、图形美观。软件支持多种常用气象数据格式，具备一些基本的 GIS 功能。MeteoInfo 还可以通过编写 IronPython 脚本程序自动运行，具备强大的数据分析功能。MeteoInfo 的核心是它的类库 MeteoInfoC.dll，使用该类库可以方便快捷地开发气象数据显示分析类软件。软件在 Microsoft 的 .Net 环境中用 C# 语言开发。本帮助文档是基于 MeteoInfo V1.0.1 编写。软件在不断更新中，欢迎任何意见和建议。

MeteoInfo 可以被免费使用和传播，作者不保证所有功能的输出结果一定正确，用户如发现问题请及时和作者联系，以便改进软件。

作者：王亚强

Email: [yaqiang.wang@gmail.com](mailto:yaqiang.wang@gmail.com), [wangyq@cams.cma.gov.cn](mailto:wangyq@cams.cma.gov.cn)

中国气象科学研究院

## 下载和安装MeteoInfo

MeteoInfo 的免费下载网址是 <http://www.meteothinker.com>。下载后将安装文件解压成 .exe 可执行文件。软件安装过程十分简单，只需要点击‘下一步’即可完成安装。软件安装后在安装目录下有‘Map’和‘Sample’子目录，‘Map’子目录中包含了一些常用的地图数据，‘Sample’子目录下有一些示例数据可用于软件学习。

Windows 操作系统中 MeteoInfo 软件必须有 .Net Framework 3.5 SP1 的支持，如果你的计算机中没有 .Net Framework 3.5 SP1，需要安装它以后 MeteoInfo 才能正常使用。.Net Framework 3.5 SP1 可以在 Microsoft 的官方网站免费下载。软件安装后在电脑桌面自动生成快捷按钮，双击可以运行 MeteoInfo。

在其它操作系统中，如 Linux，安装了 mono 以后也可以使用 MeteoInfo。但目前 Mono 对 .Net 的支持有限，软件的一些功能不能使用，比如 NetCDF 格式文件无法在 Mono 下打开。Linux 下安装的步骤：

1、安装 Mono（一些较新的 linux 操作系统可能已经自带了 Mono）。在此网站上下载 Mono 安装程序 <http://www.go-mono.com/mono-downloads/download.html>，使用二进制安装文件比较方便（如果有该系统的二进制安装文件），或者下载 Mono 源代码进行编译安装：

```
# ./config  
# make  
# make install
```

2、MeteoInfo 软件无需安装，只需将 .zip 文件解压后的 MeteoInfo 目录及其中的所有文件拷贝到计算机中即可。

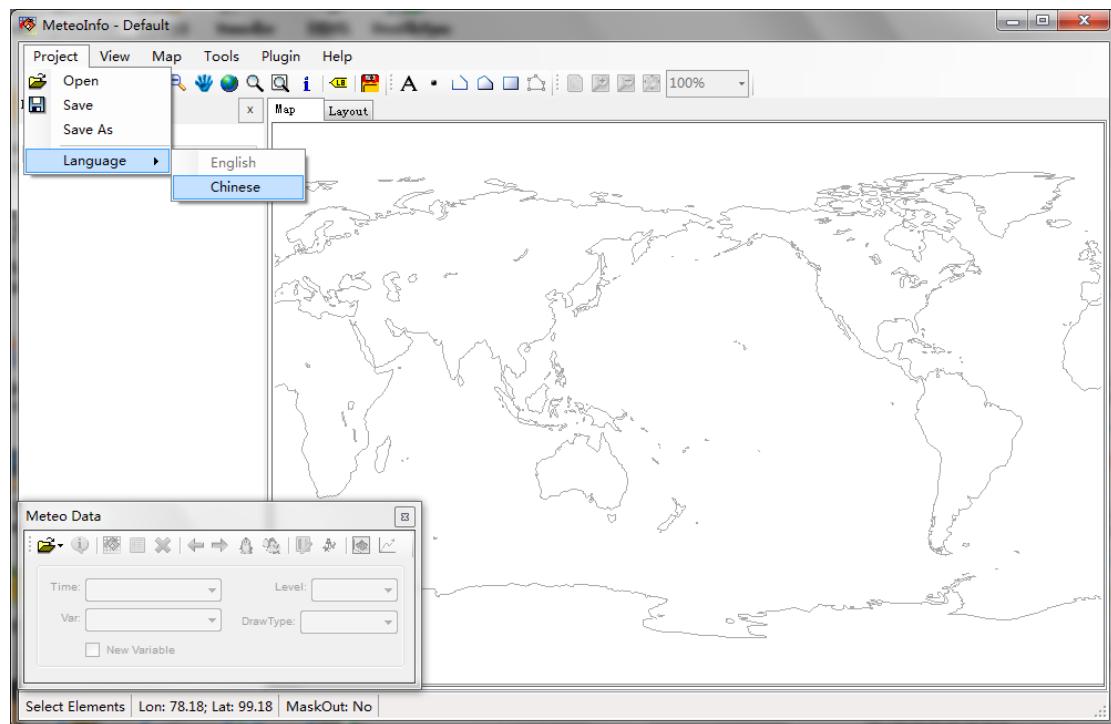
3、安装“WeatherSymbol.ttf”字体。

```
# cd /usr/share/fonts  
拷贝字体文件至上述目录。
```

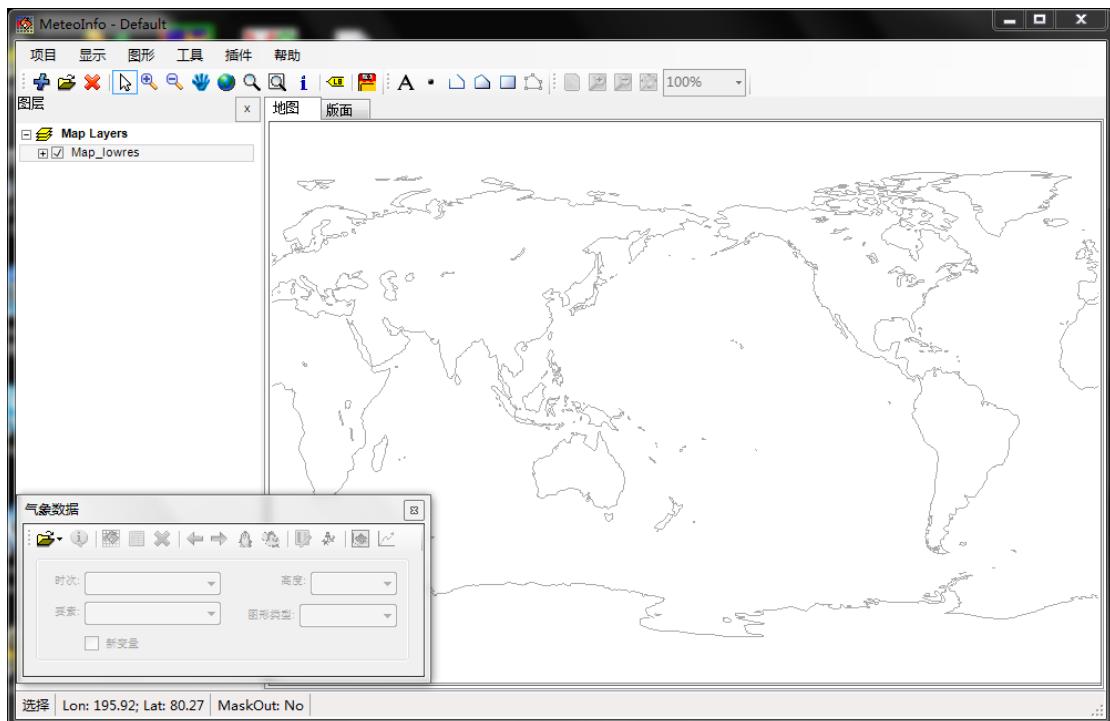
```
# mkfontscale  
# mkfontdir  
# fc-cache  
4、运行 MeteoInfo。  
# mono MeteoInfo.exe
```

## 多语言支持

MeteoInfo 软件具有多语言支持能力，目前支持中文和英文两种语言。软件第一次运行时的界面语言是英文。将软件界面更改为中文，只需点击菜单 ‘Project => Language => Chinese’ 即可。



软件中文界面如下。点击‘保存项目’子菜单，则将语言信息保存在软件启动文件中，下次启动软件时就会是中文界面。考虑到软件的使用范围，MeteoInfo 以英文为主进行开发，目前的中文支持并不完整。



## 支持的数据

### 地图数据

目前 MeteoInfo 支持以下格式的地图数据。

1. ESRI 的 shape 格式数据（点、线、面）。
2. GrADS 格式的地图数据。
3. MICAPS 格式的地图数据。
4. wmf 格式的地图数据（MeteoInfo 自定义的地图数据格式，能够用‘输出地图数据’功能输出该格式的地图数据）。

shape格式是ESRI（GIS业界的巨头，主要产品是ArcGIS）的一种GIS数据格式，ESRI公开了shape格式，所以已经成为大多数GIS 软件都支持的一种开放格式。这种GIS数据格式比较简单（没有拓扑结构），数据编辑较为方便。GIS数据大致分为点（point）、线（line）、面（polygon）三种。每个shape格式数据至少包括 3 个文件，这 3 个文件的文件名相同但后缀不同（.shp、.shx、.dbf）。.shp文件保存了地理信息，.shx文件是索引文件，.dbf文件保存了图元的属性信息。shape格式数据还可能有其它后缀的文件，比如投影过的数据会有.prj文件来保存投影信息。网络上有很多免费的shape数据资源，国内的数据可以在国家基础地理信息系统网站<http://nfgis.nsdi.gov.cn/> 上下载。国外的数据可以在此网站上下载：<http://www.diva-gis.org/Data>。请注意下载的涉及中国国界的数据一定要确认国界线符合我国规定后再使用。

## 气象数据

目前 MeteoInfo 支持下列气象数据格式：

- NetCDF 格点数据
- GRIB 1 和 2 格式数据
- GrADS 二进制格点和站点数据
- HDF EOS5 Grid 和 Swath 数据
- ARL 格式气象数据
- HYSPLIT 模式输出数据：Concentration（浓度）、Particle（颗粒）、Trajectory（气团轨迹）
- METAR 格式站点数据（目前针对 NOAA 的 cycle METAR 数据格式，<http://weather.noaa.gov/weather/metar.shtml>）
- NOAA ISH 数据
- MICAPS 第 1、3、4、7、11、13 类格式数据
- ESRI ASCII 格点数据
- Surfer ASCII 格点数据
- AWX 格式数据

气象数据格式通常比较复杂，MeteoInfo 会有考虑不到的地方，如发现打开错误的情况，请和软件作者联系（yaqiang.wang@gmail.com），并发送数据示例文件，以便改进软件。

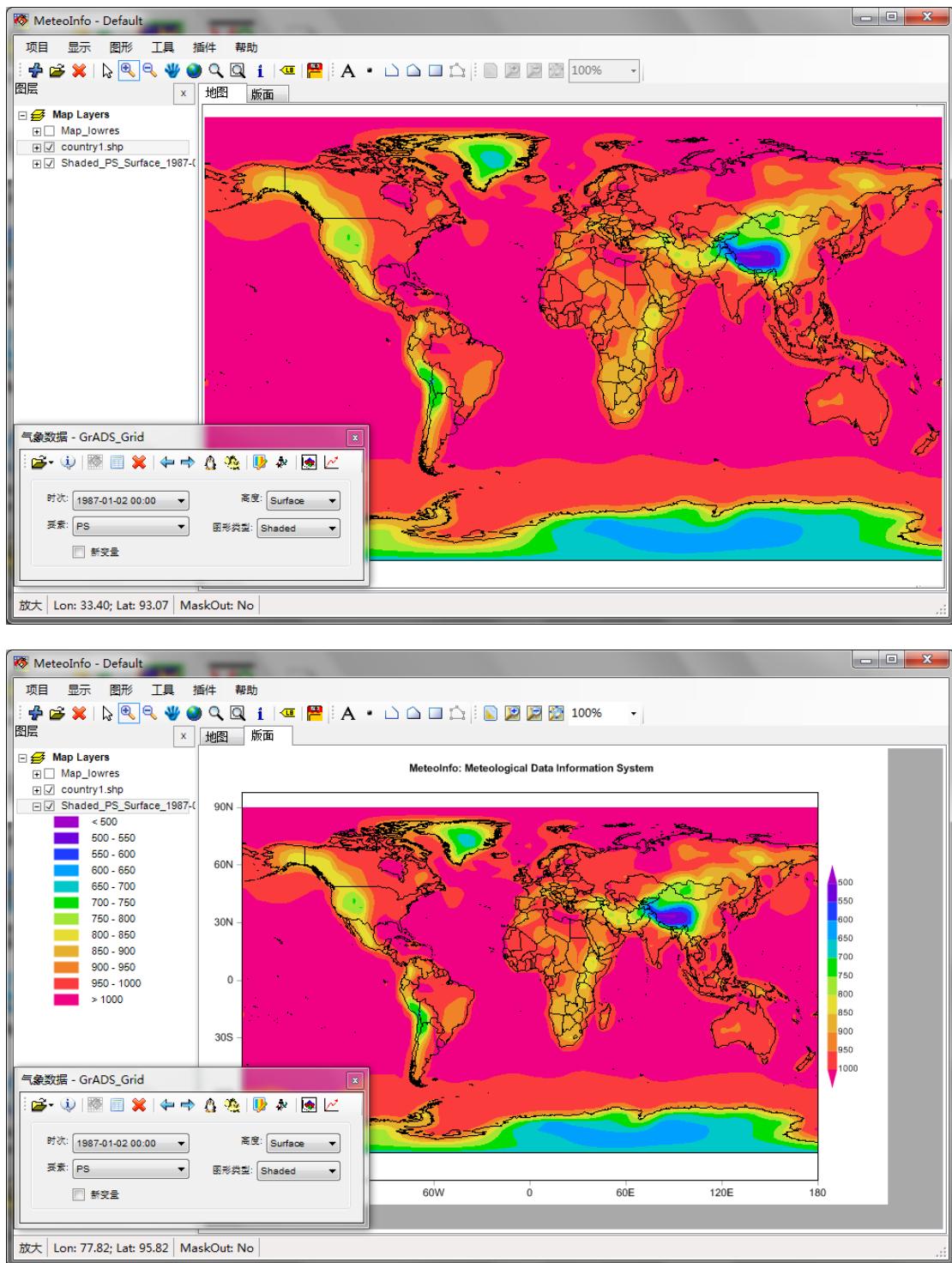
## 图像数据

MeteoInfo 支持常用的图像数据格式。

- Bmp
- Gif
- Jpg
- Tif
- Png

## 软件主界面

MeteoInfo 的主界面包括菜单、工具栏、状态栏、图层控制区、地图显示区和地图版本设计区。



## 图层和项目文件

在 MeteoInfo 中，数据最终以图层的形式进行显示和管理。各图层的属性、地图和版面的设置、投影参数等内容可以被保存在一个项目文件中，后缀为.mip。需要注意的是，从气象数据创建的图层只是在内存中存在，如果想将其保存在项目中，必须先将其保存为 shape 文件（选中该图层，点击鼠标右键，选择‘Save Layer’菜单将图层内容保存为 shape 文件）。

MeteoInfo 软件启动时，会自动加载一个缺省项目文件‘Default.mip’。

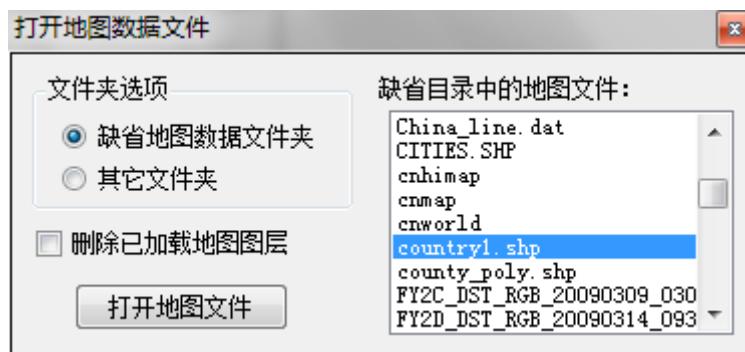
# 图层操作

在 MeteoInfo 中有三种类型的图层：

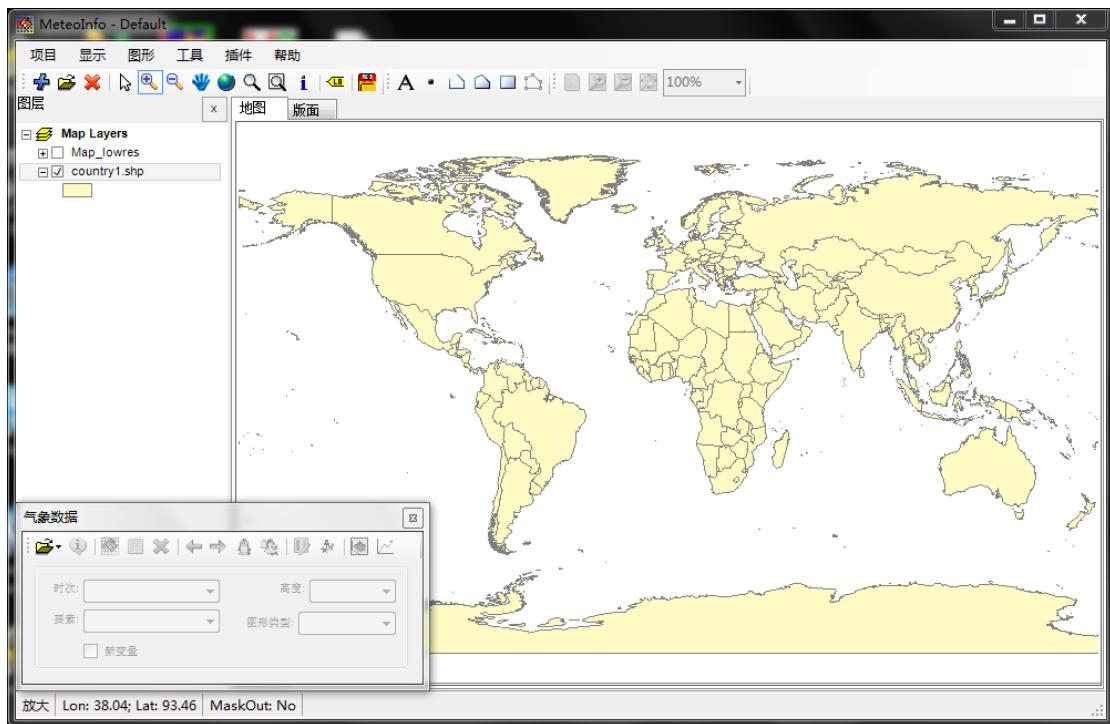
- 矢量图层（Vector layer），每个图层只有一种图元类型，图元类型包括点（Point）、线（Polyline）、面（Polygon），矢量图层有属性数据，可以包含多个字段，每一条属性记录和一个图元一一对应。
- 图像图层（Image layer），用来打开和显示图像数据（如 Jpg 等格式的图像文件），图像文件需要有地理定位文件（world file）来确定图像的显示位置。
- 栅格图层（Raster layer），用来显示格点数据，尤其是格点数目很多的格点数据适合用栅格图层来显示。

## 添加图层

在工具栏中点击‘添加图层’按钮  打开添加图层对话框。所有缺省地图数据目录（软件安装目录中的‘Map’子目录）下的文件会显示在右侧的列表中，选择其中的一个文件然后点击‘打开地图文件’按钮就能打开相应的地图数据文件。选中‘删除已加载地图图层’选项则在打开新的地图数据文件时删除以前加载的地图图层。如果要打开的地图数据文件不在缺省地图数据文件夹中，则选中‘文件夹选项’中的‘其它文件夹’，然后点击‘打开地图文件’按钮，在‘打开文件’对话框中选择要打开的地图数据文件。



例如，选择打开‘country1.shp’文件，那么在‘图层’窗口中就有两个地图图层。因为‘country1.shp’中的图元是多边形，默认打开成一个填成浅黄色的图层。

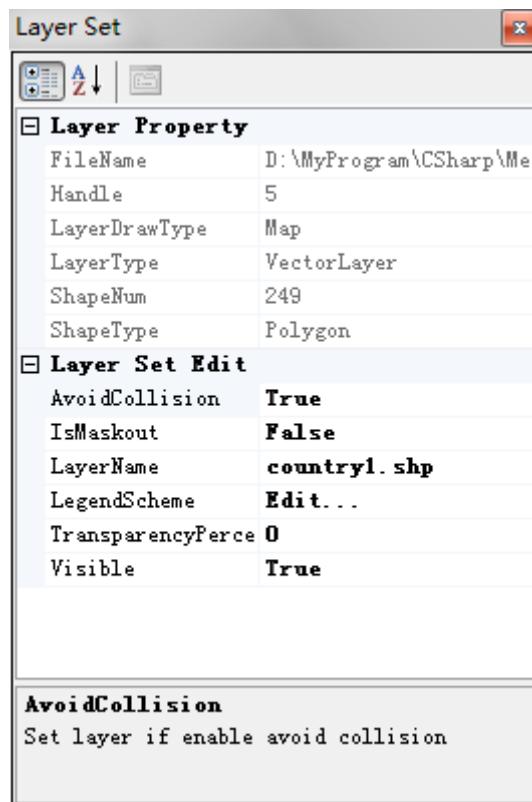


## 编辑图层属性

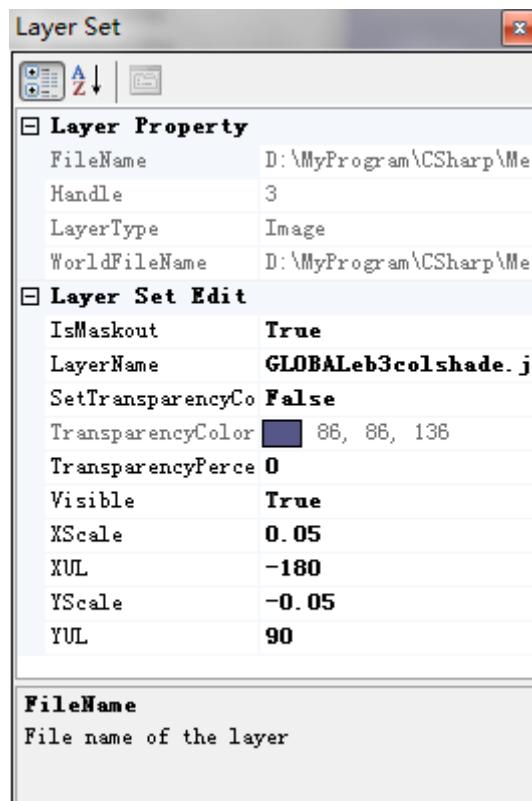
在‘图层’窗口中点击图层名称前的方框可以显示或隐藏图层。

图层中各图元的显示方式可以改变。双击‘图层’窗口中的某个图层名，则打开‘Layer Set’对话框。‘Layer Property’栏中的属性是只读的，‘Layer Set Edit’栏中的图层属性可以修改。

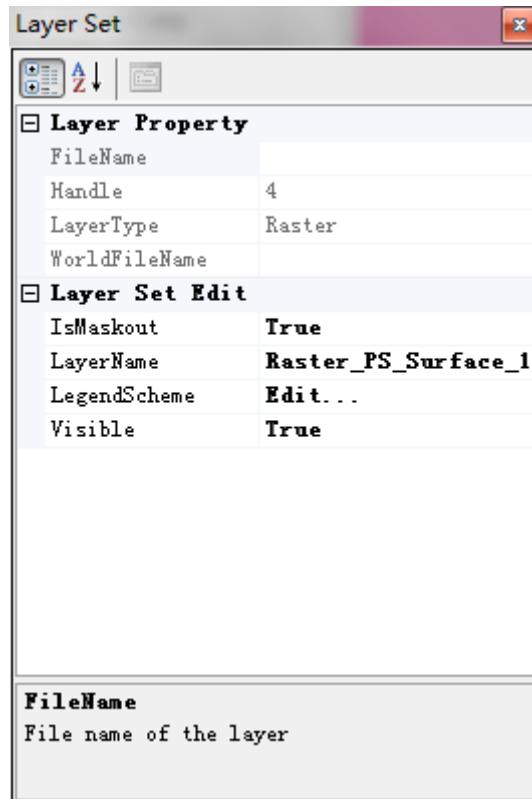
矢量图层属性：



图像图层属性：

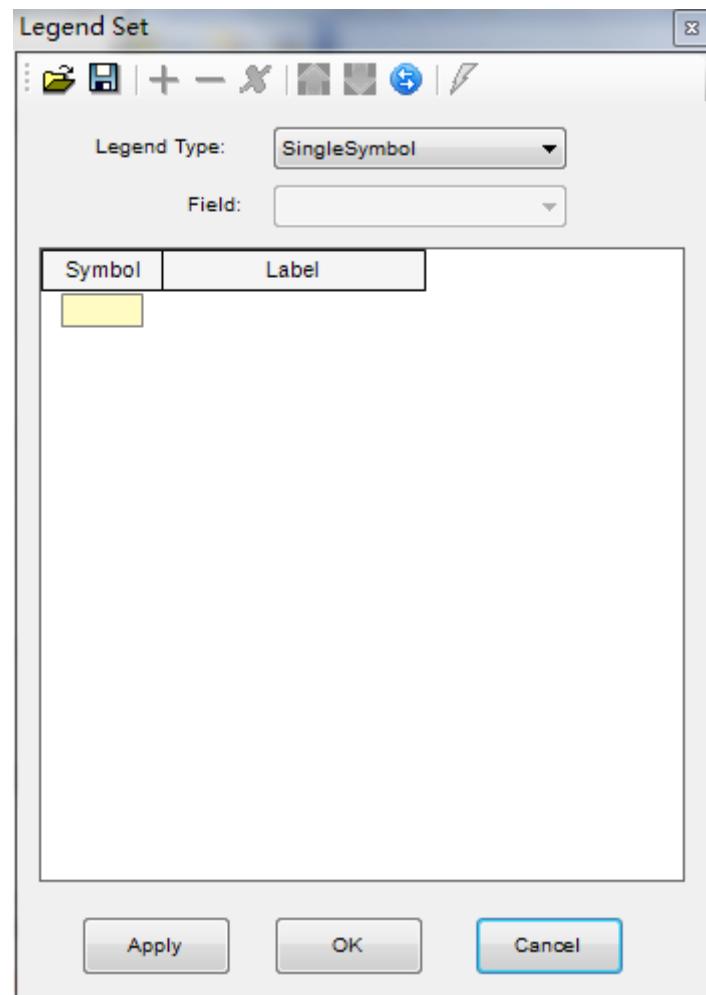


栅格图层属性：

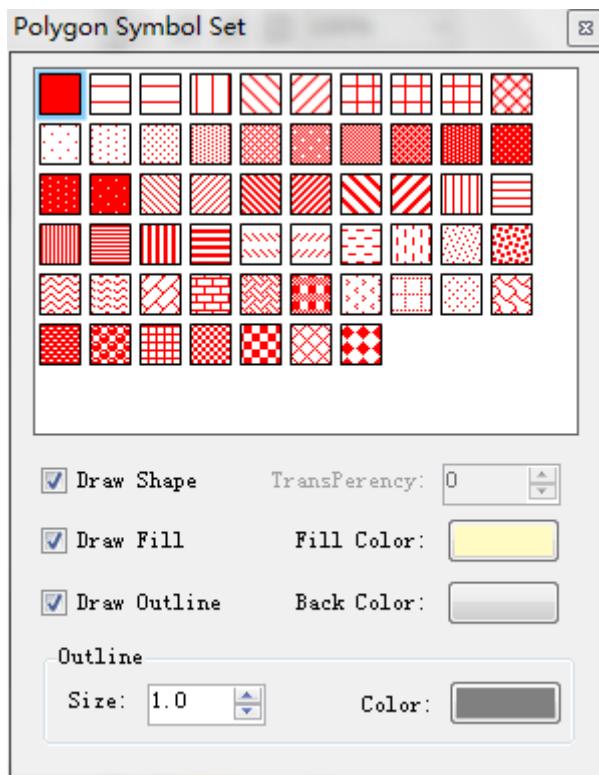


## 图例

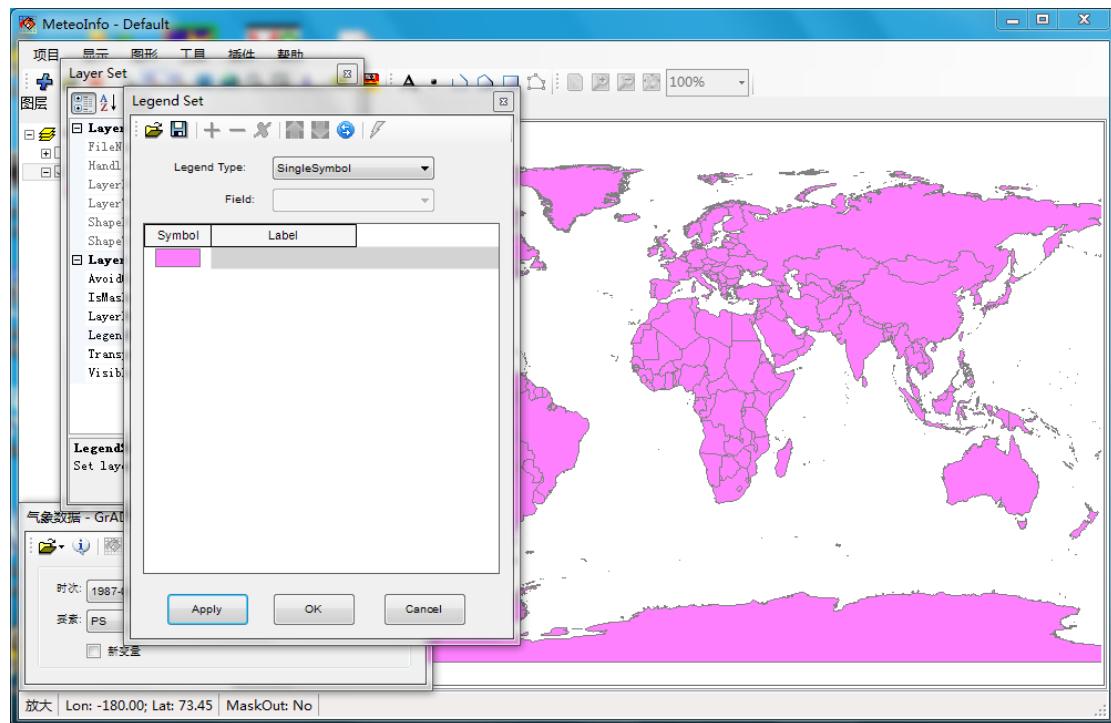
图层是依据其图例设置进行显示的。点击‘LegendScheme’属性的‘Editing...’栏，会在右侧出现一个按钮，点击这个按钮则打开‘Legend Set’对话框。



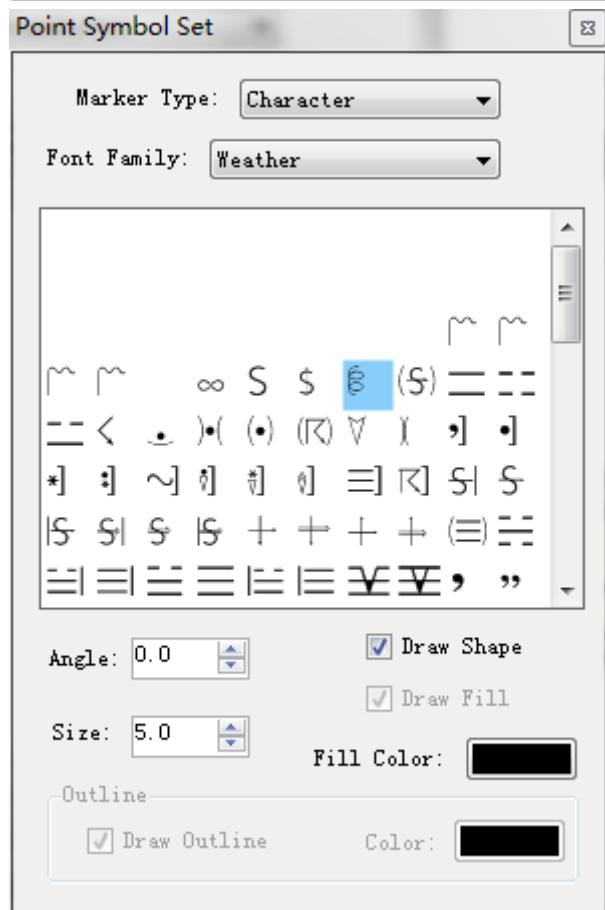
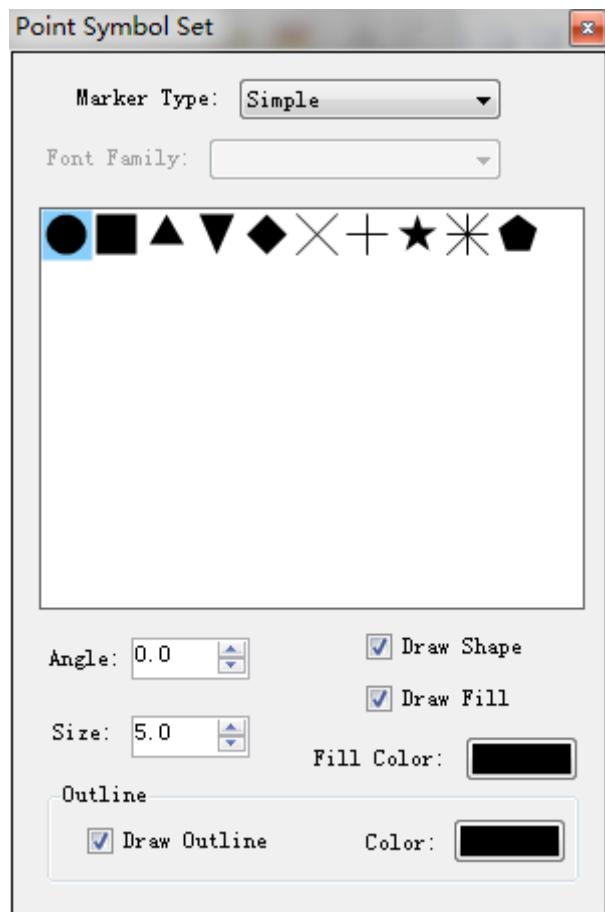
双击‘Symbol’列中的填色长方形，打开‘Symbol Set’对话框。

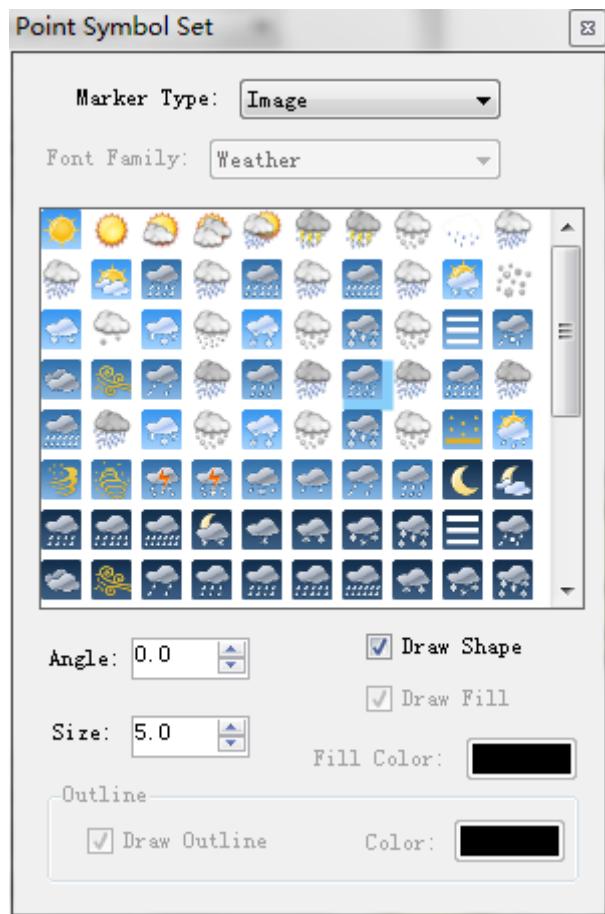


改变‘PolygonColor’的颜色然后点击‘Apply’按钮，就能看到图元颜色的改变。试着改变其它属性，看看有什么效果。

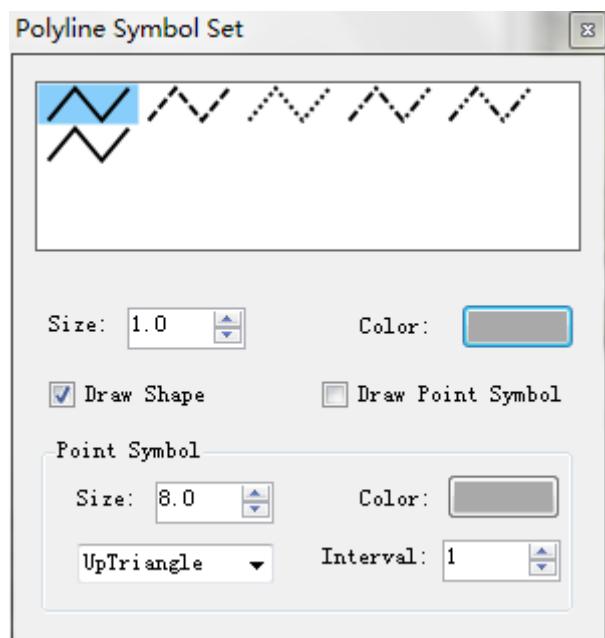


点图元符号设置的对话框如下。有三种符号显示方式：简单符号（Simple）,字体符号（Character）和图像（Image）。





线图元符号设置对话框如下：

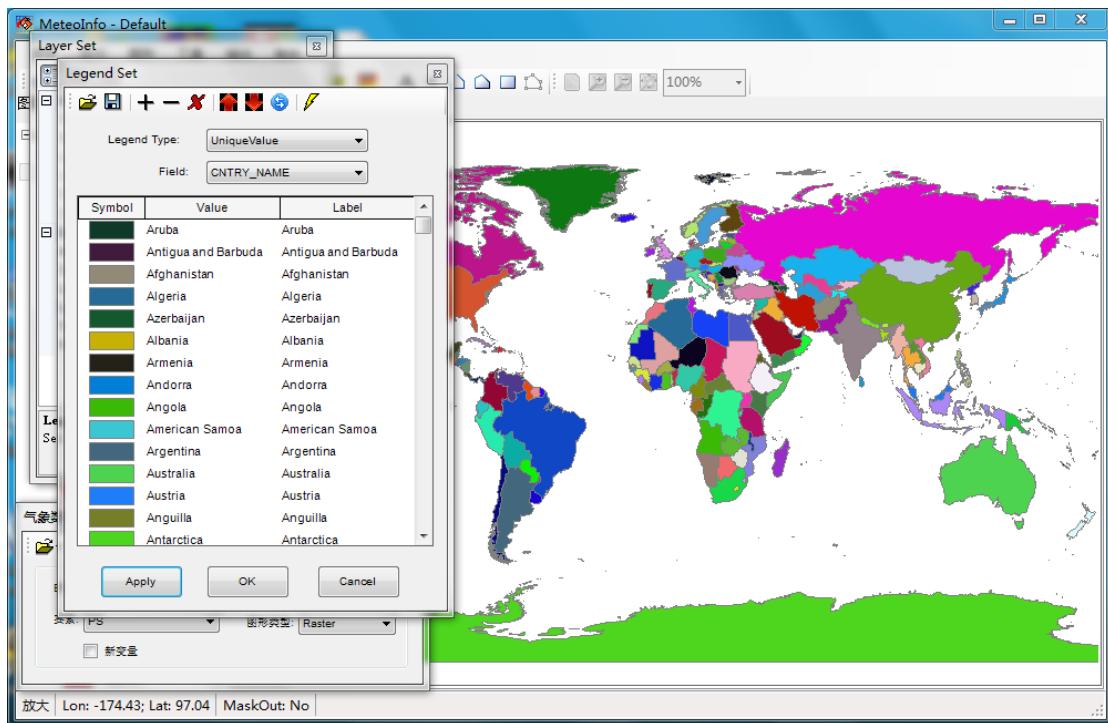


MeteoInfo 中有三种图例方案：

- ✧ ‘SingleSymbol’。图层中所有图元用同样方式显示。
- ✧ ‘UniqueValue’。图层中每个图元都用不同方式显示。

◆ ‘GraduatedColor’。图层中的图元被分成一些组，每组图元用同样的方式显示。

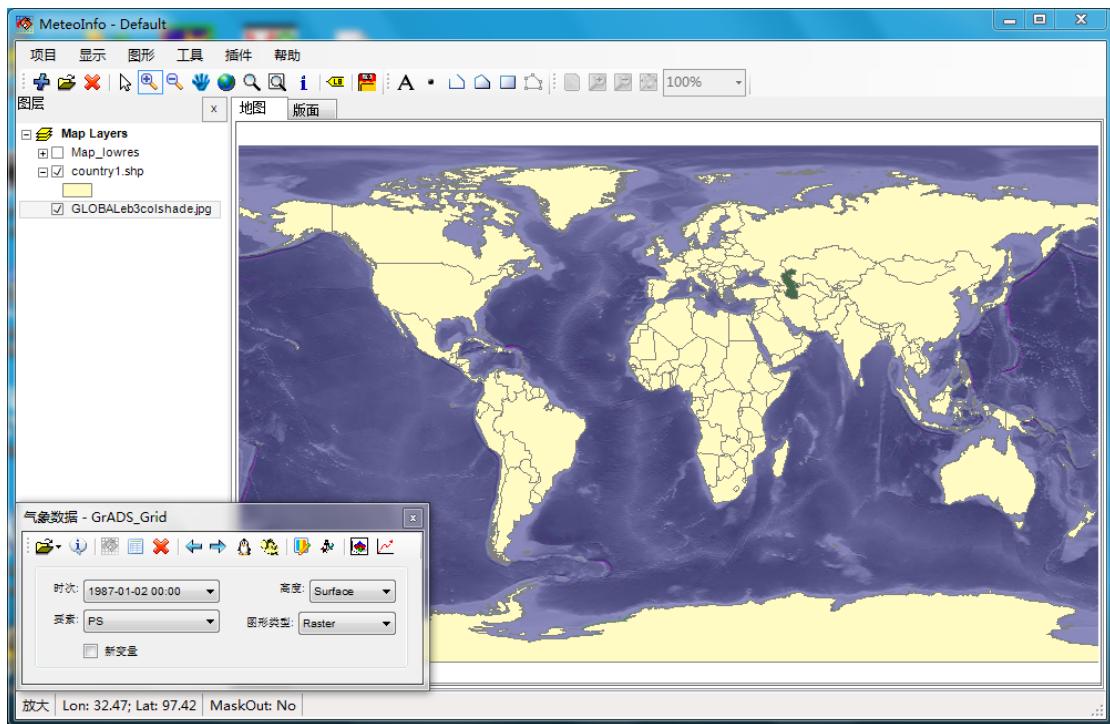
例如，改变‘Legend Type’为‘Unique Value’，然后点击‘Apply’按钮，得到下面的图形。



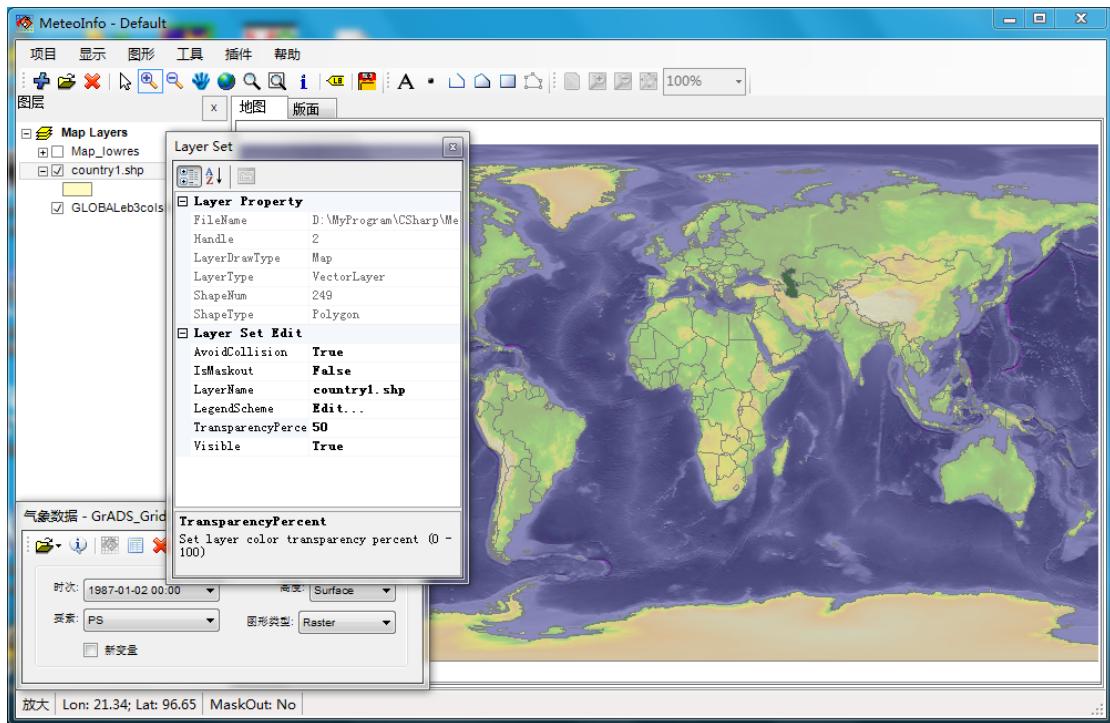
图例设置可以保存为图例文件，后缀为.lgs。以后在显示类似数据时可以加载相应的图例文件。对图例的改变只有在点击‘OK’后才能保存在图层中。图例编辑对话框中可以对图例进行添加、删除、移动等操作。

## 图层透明度设置

在项目中添加一个图像图层‘GLOBALeb3colshade.jpg’，可以看到图像的陆地部分被‘country1.shp’图层遮盖。



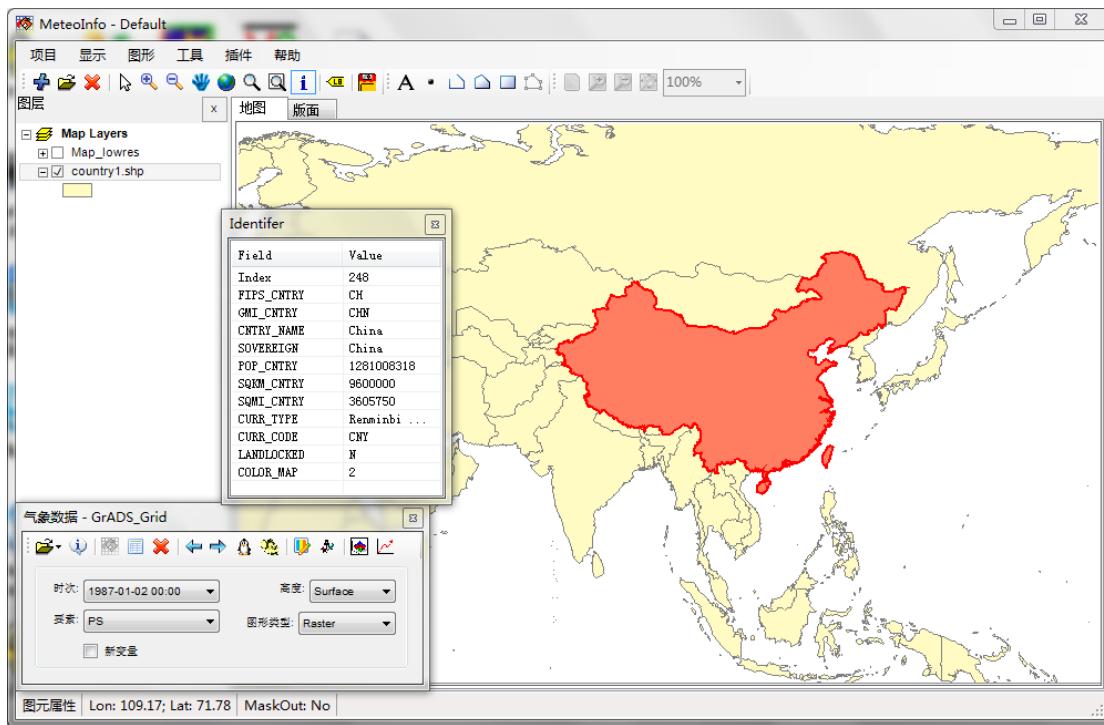
将‘country1.shp’图层的透明度（TransparencyPercent）设置为50（即50%的透明度）后的效果如下：



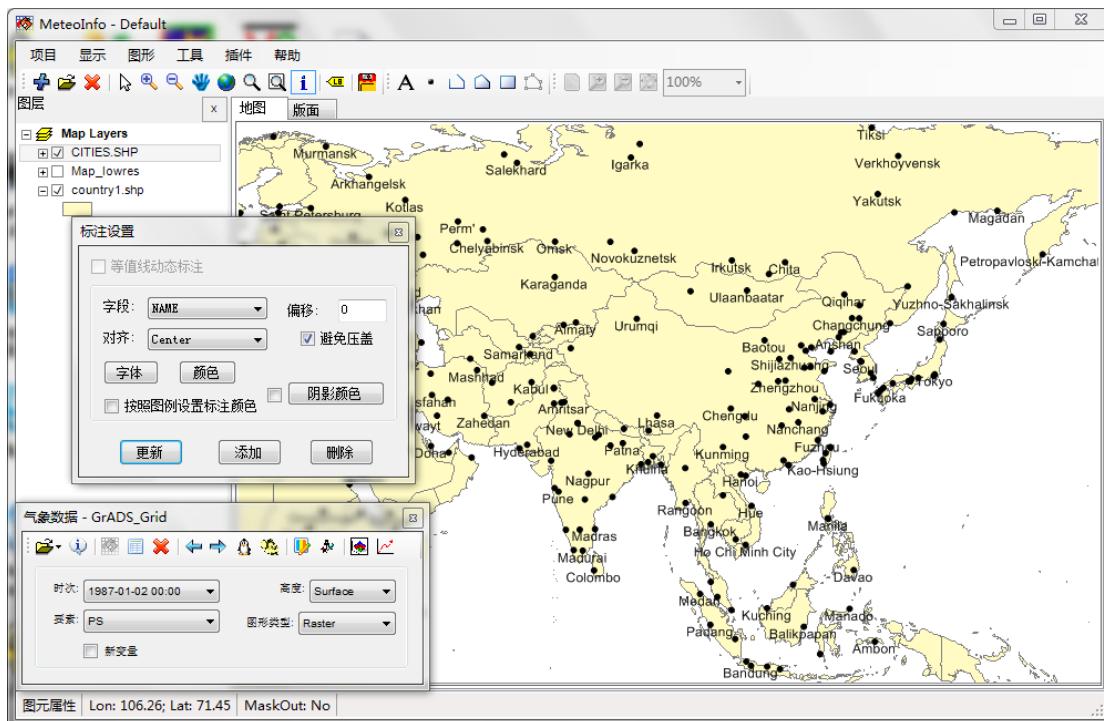
## 属性数据

矢量图层具有属性数据，单个图元的属性数据查询可以用‘Identifier’工具 ，被选

中的图元会高亮显示，该图元的属性在‘Identifier’对话框中被显示出来。



属性数据可以用来对图层进行标注。例如添加‘CITIES.SHP’图层，利用其‘NAME’字段的属性数据对各个城市进行标注。标注可以设置为‘避免压盖’，软件会自动根据地图显示范围仅显示不互相压盖的标注。图层标注的工具是 。



图层属性表可以通过‘显示’菜单的‘图层属性数据’子菜单显示：

Attribute Data - CITIES.SHP

	NAME	COUNTRY	POPULATION	CAPITAL
► 1	Murmansk	Russia	468000	N
2	Arkhangelsk	Russia	416000	N
3	Saint Peters...	Russia	5825000	N
4	Magadan	Russia	152000	N
5	Perm'	Russia	1160000	N
6	Yekaterinburg	Russia	1620000	N
7	Nizhniy Novg...	Russia	2025000	N
8	Glasgow	UK	1800000	N
9	Kazan'	Russia	1140000	N

## 使用图像图层

### 图像地理定位文件

图像是以栅格数据形式保存的，每一个栅格对应一个行号和列号。为了将图像显示为地理坐标，需要建立转换信息将图像坐标转换为地理坐标。转换信息保存在一个独立的 ASCII 文件中（地理定位文件），能够用任何文本编辑工具修改。转换信息也能在 MeteoInfo 中生成和编辑。

地理定位文件的文件名和相应的图像文件名一样，后缀取图像文件后缀的第一和第三个字母再加上字母‘w’组成。比如，图像文件为‘mytown.tif’，那么地理定位文件的文件名为‘mytown.tfw’。‘redlands.gif’的地理定位文件是‘redlands.gfw’。

地理定位文件的内容如下：

20.17541308822119	-A
0.0000000000000000	-D
0.0000000000000000	-B
-20.17541308822119	-E
424178.11472601280548	-C
4313415.90726399607956	-F

A = x 方向比例，一个象素在地理坐标中 x 方向的大小。

B, D = 旋转项。

C, F = 平移项，左上角象素的地理 x、y 坐标。

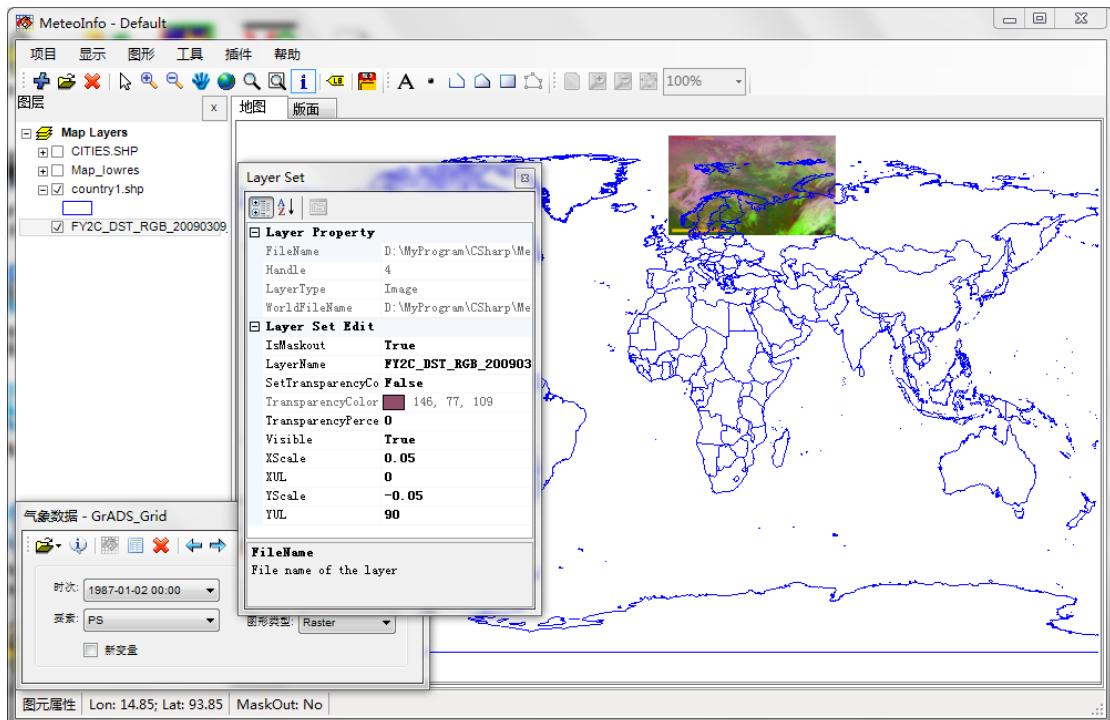
E = y 方向比例的相反数（负数），一个象素在地理坐标中 y 方向的大小。

注意：旋转项目前在 MeteoInfo 中未做处理。

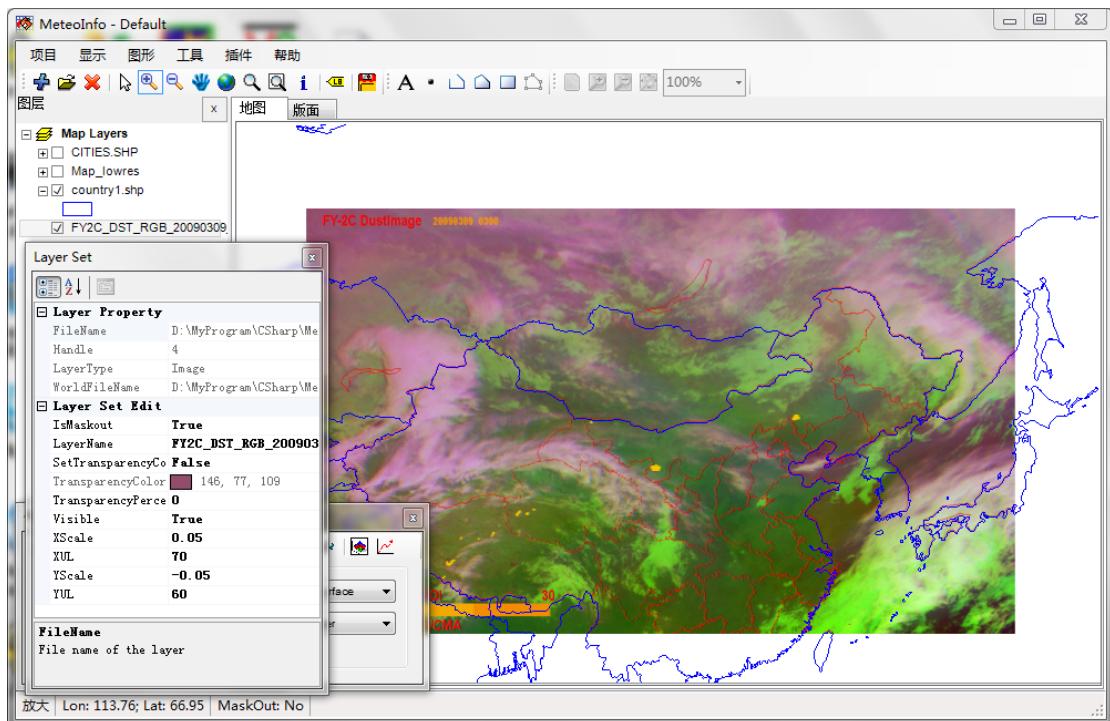
## 显示图像

如果图像文件有地理定位文件，MeteoInfo 会读取坐标转换信息对图像进行地理定位。如果没有地理定位文件，MeteoInfo 会自动生成一个，然后需要用户手动调整参数以便正确显示图像。

下面是一个打开没有地理定位文件图像的例子。双击‘图层’窗口中的图像图层，转换信息显示在‘Layer Set’对话框中。通过改变‘XScale’、‘XUL’、‘YScale’、‘YUL’的值来调整图像在地图中的位置，这个过程通常需要重复许多遍，需要一定的耐心。



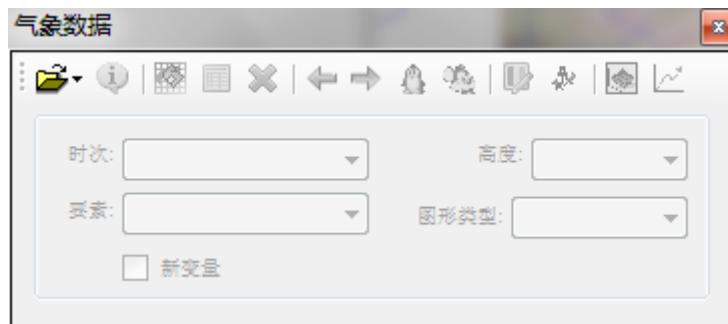
正确调整后图像的显示。



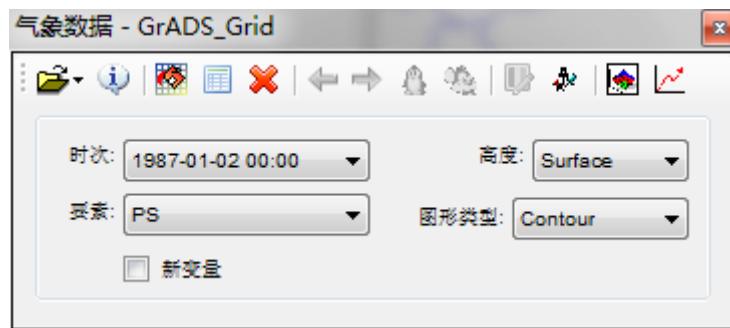
## 使用气象数据

### 打开气象数据

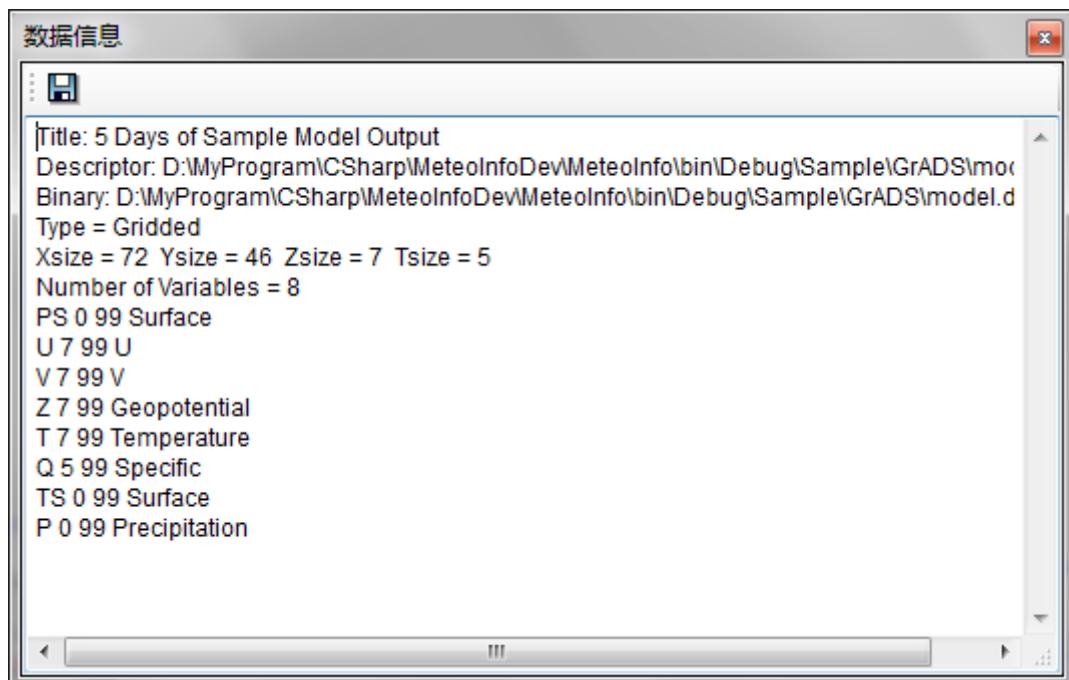
MeteoInfo 软件启动时会自动打开‘气象数据’对话框。该对话框关闭后可以通过点击‘打开数据文件’按钮重新显示。



点击‘打开数据’按钮，在下拉菜单中选择要打开的数据类型，如选择‘GrADS 数据’打开‘Sample’目录中的‘model.ctl’示例数据。数据的时次、高度、要素等信息会显示在对话框中。



点击‘显示数据信息’按钮能够看到数据更详细的信息。



## 显示格点数据

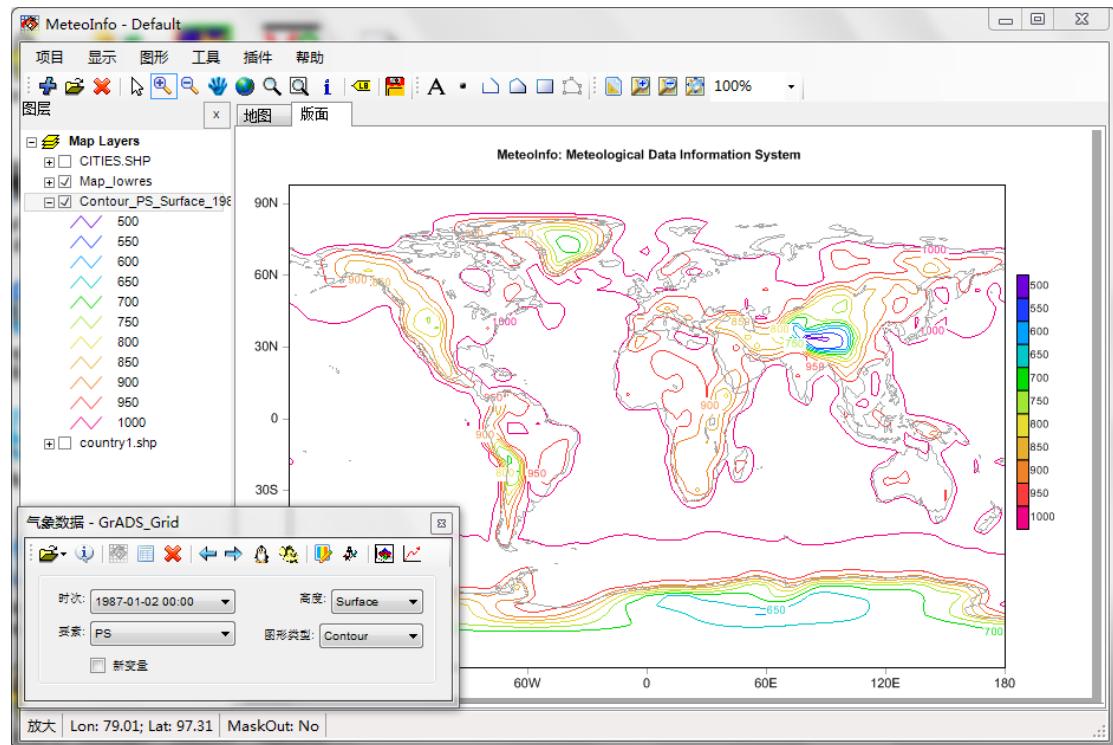
格点数据能够被显示为如下图形类型

- ✧ Contour (等值线)
- ✧ Shaded (等值线填色)
- ✧ Grid Fill (格点填色)
- ✧ Grid Point (点图)
- ✧ Vector (矢量图—显示风场)
- ✧ Barb (风向杆图—显示风场)
- ✧ Streamline (流线图—显示风场)
- ✧ Raster (栅格图)

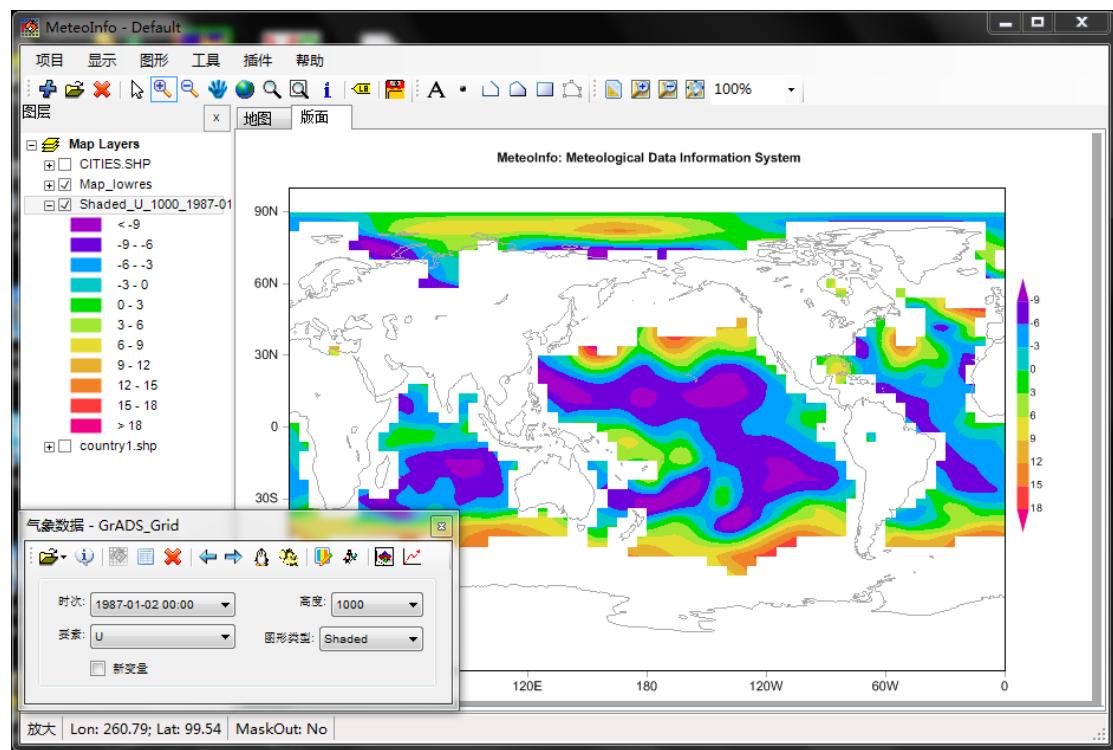
下面的示例图用的是‘model.ctl’和‘model.dat’示例数据画出来的。

## 等值线及填色图形

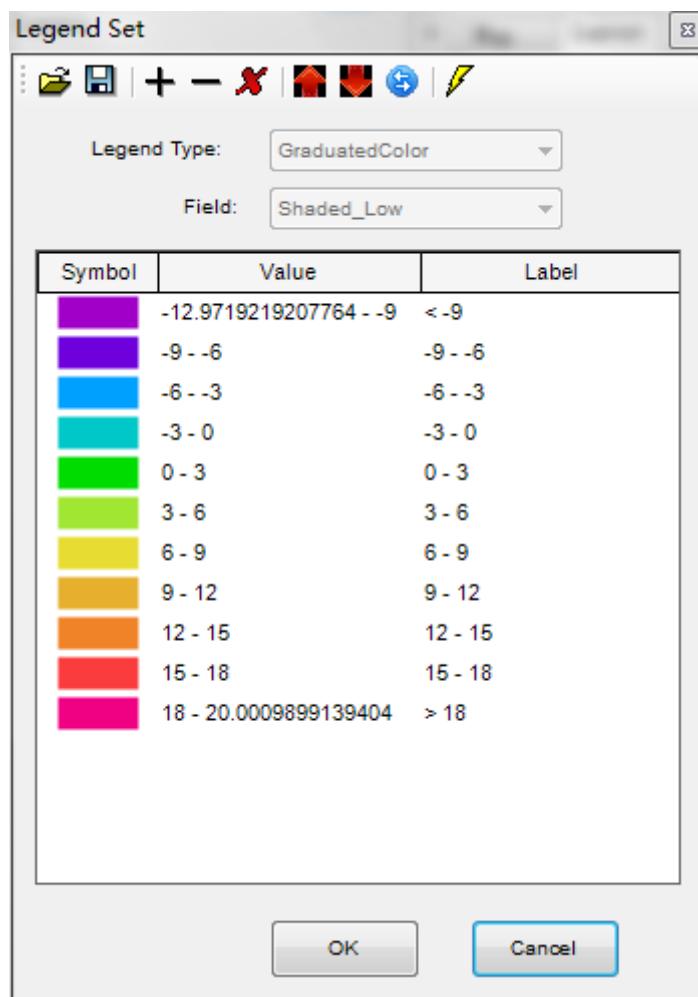
设置‘时次’、‘高度’、‘要素’，选择‘图形类型’为‘Contour’，点击‘显示数据图形’按钮。软件将按照设置追踪格点数据等值线并生成等值线图层，新生成的图层显示在‘图层’窗口中，图层名自动按照图形类型、要素、高度和时间生成。



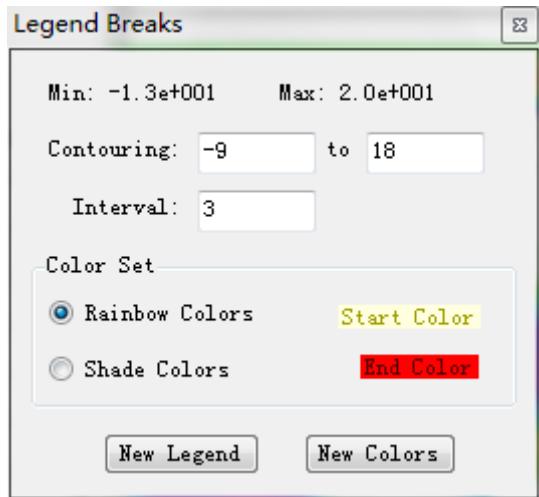
等值线填色图形显示如下：



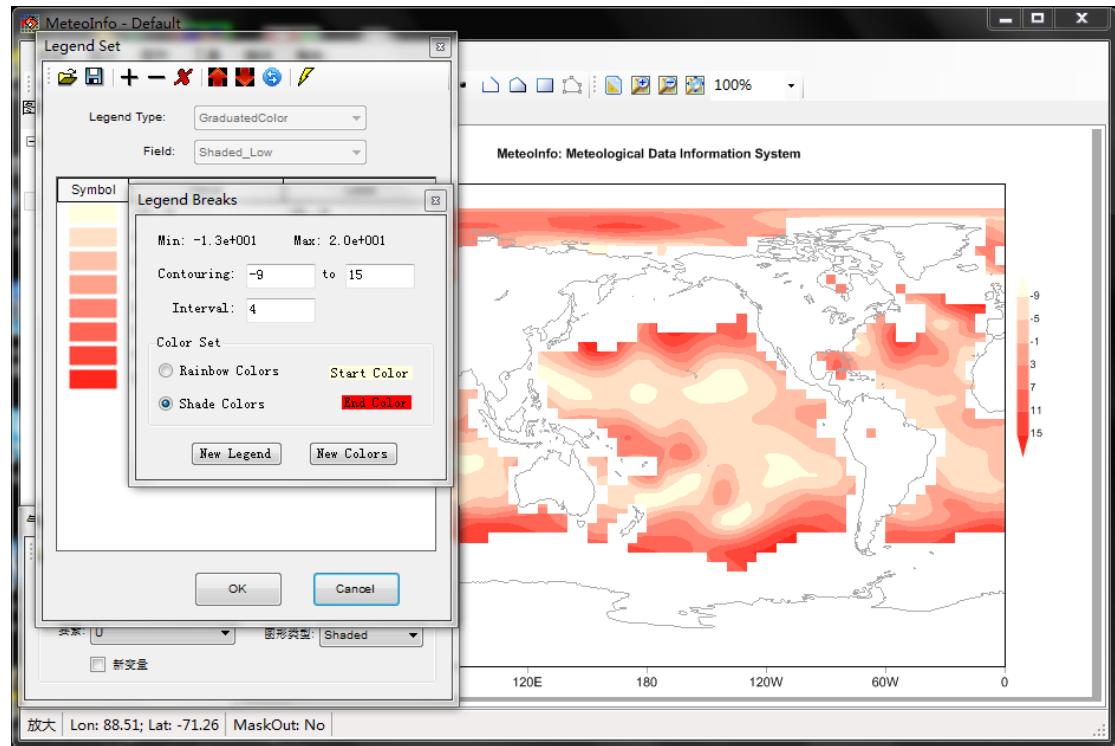
软件会根据数据自动设置等值线间隔，并将颜色缺省设为‘Rainbow’（彩虹）色彩系列。也可以人为设置等值线值及显示颜色。点击‘图例设置’按钮  打开‘Legend Set’对话框。



这个‘Legend Set’对话框和前面在地图图层显示部分中讲的类似。不过它们有一个重要区别，在这里改变图例设置后软件会根据新的图例设置重新追踪等值线，生成具有不同图元的新图层。而在‘图层’窗口中改变图层的图例设置只是改变图元的显示方式，图元对象本身不会改变。图例设置可以手动改变，也可以通过‘New Legend or Color’对话框依据设置自动生成新的图例设置。

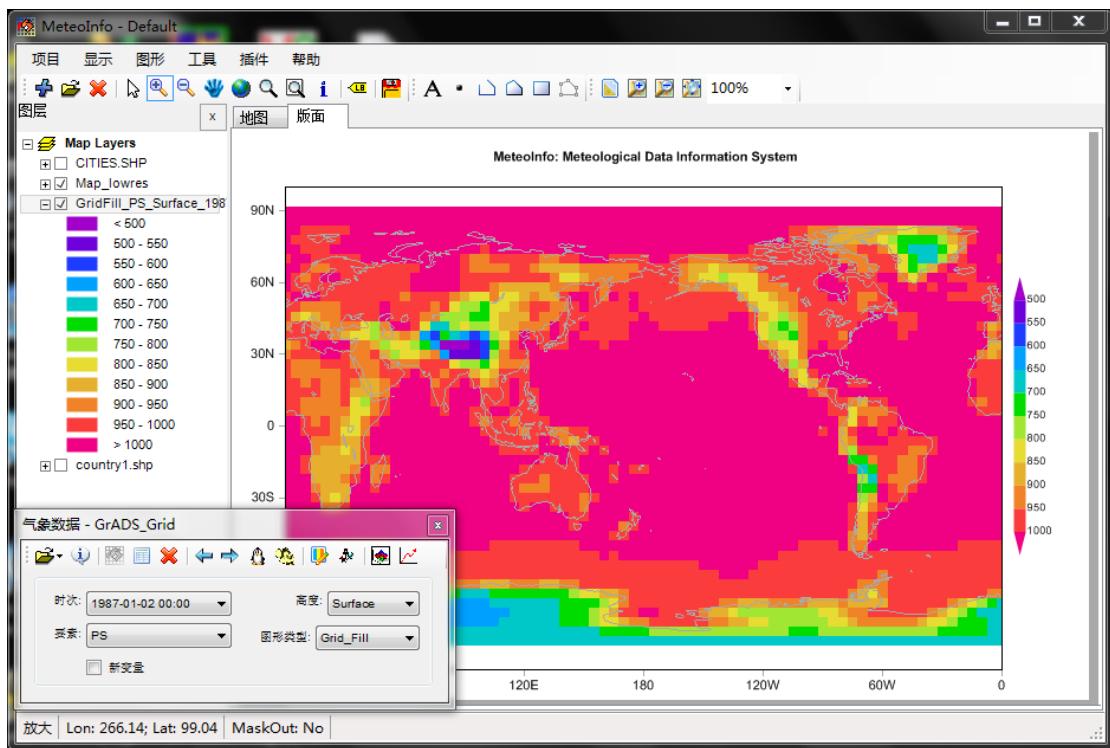


将‘Interval’改为4，‘Color Set’中选中‘Shade Color’（通过起始颜色和结束颜色自动生成系列颜色）。然后点击‘New Legend’按钮，再点击‘Apply’按钮可以看到下图。只有点击‘OK’按钮后改变的图例设置才能被保存。

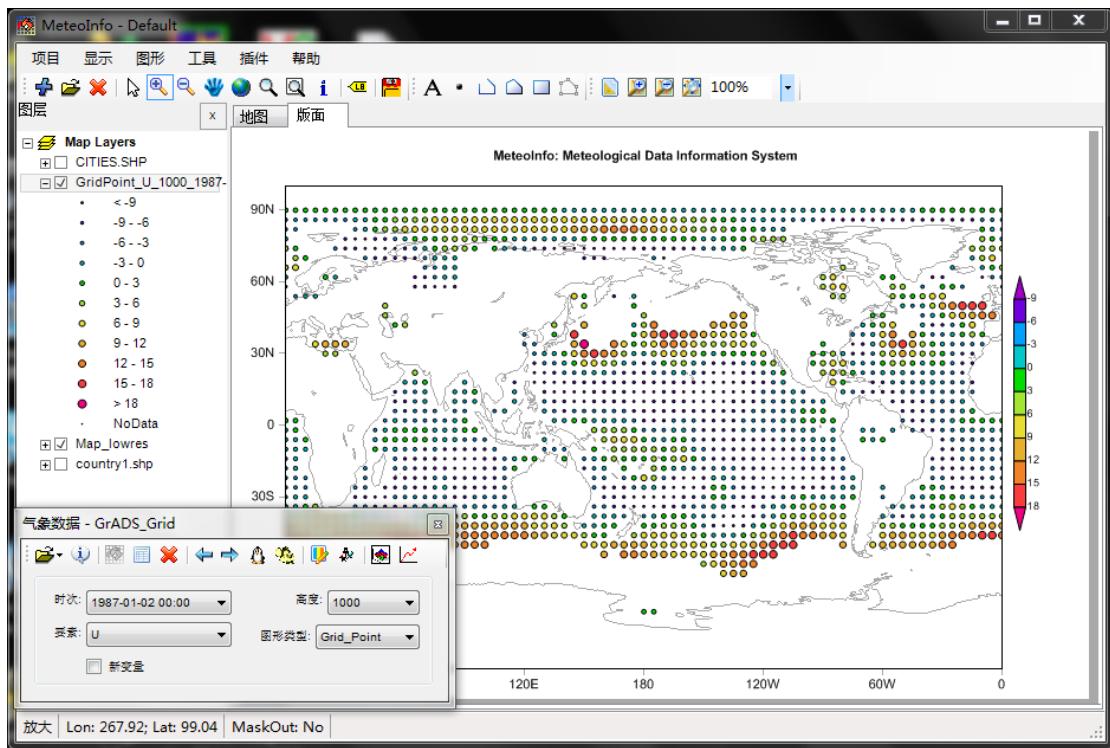


## 格点填色和点图

选中格点填色（Grid Fill）图形，格点数据将依据其值显示成同样大小、不同颜色的长方形。

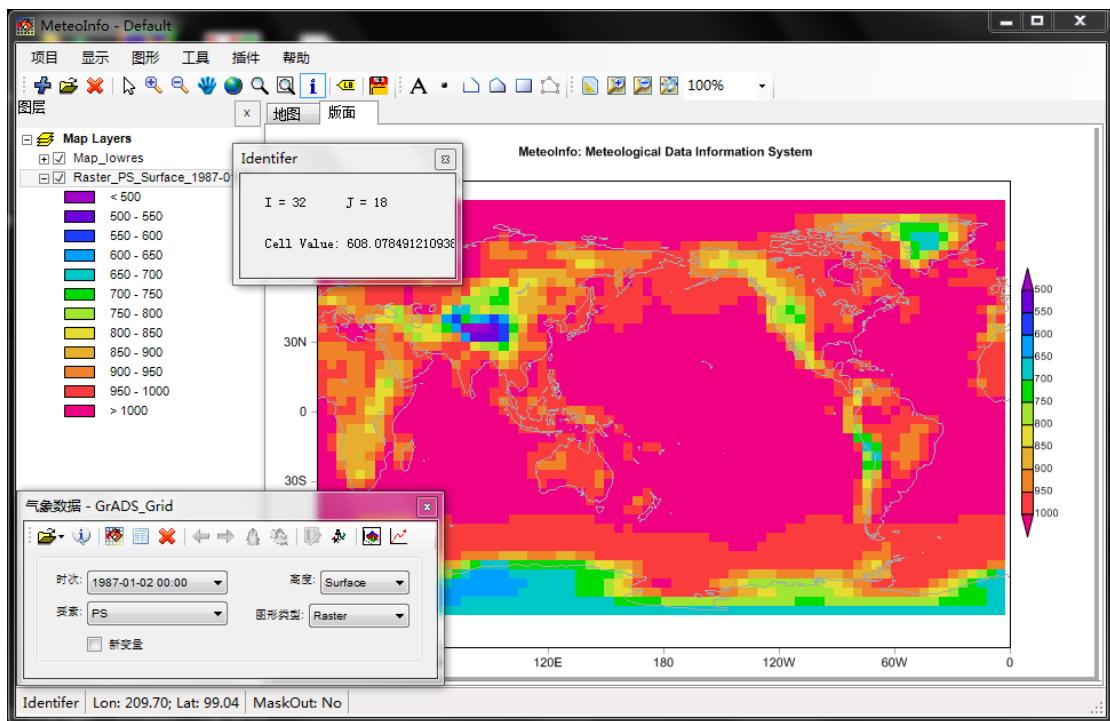


选中格点点图（Grid Point）图形，格点将依据其值显示成不同大小、颜色的点。



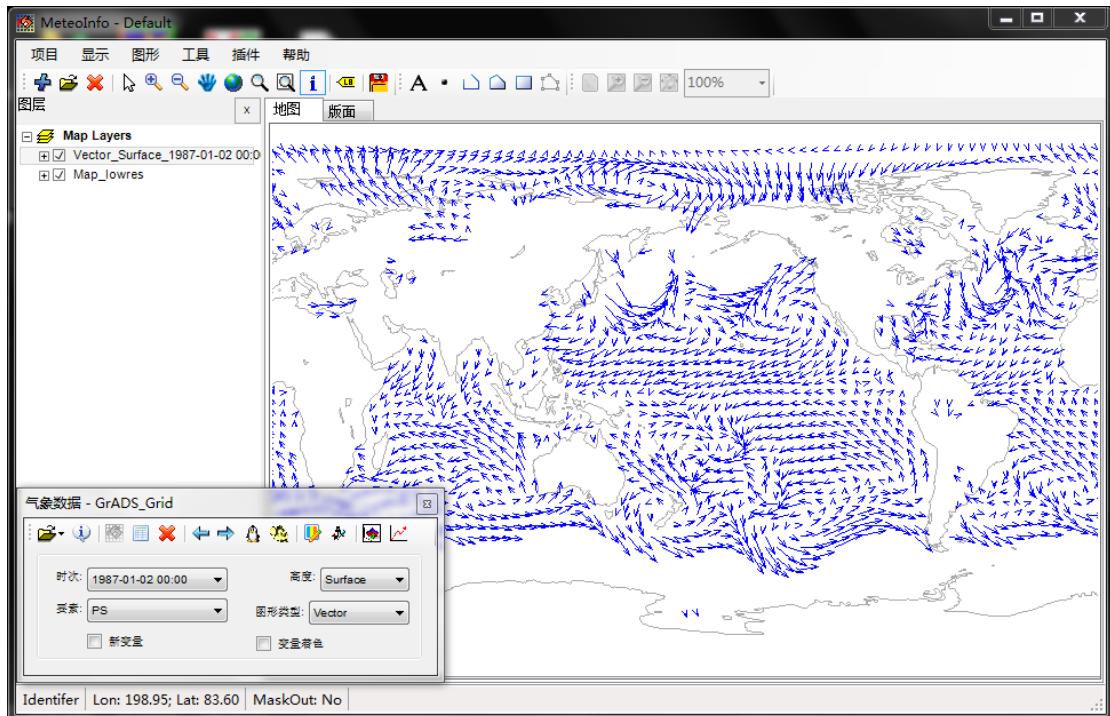
## 栅格图形

栅格图形适合格点数目很多的格点数据，每个格点的数据可以用‘Identifier’工具获取。

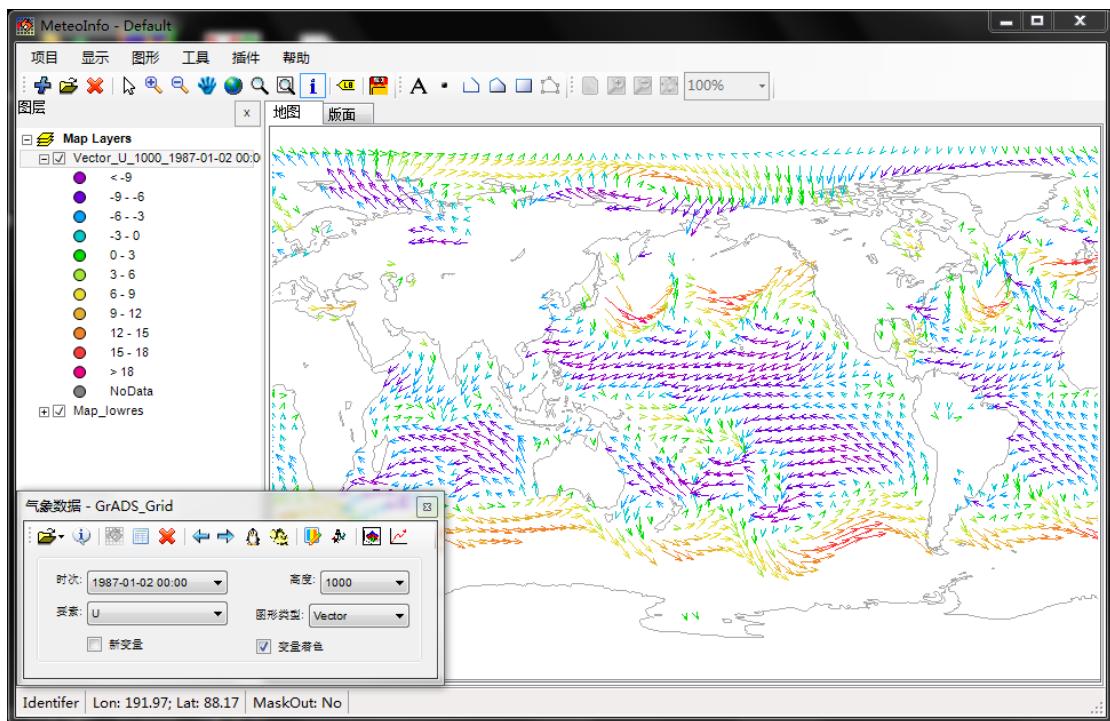


## 矢量图形

通过 U、V 分量合成的风场矢量图。

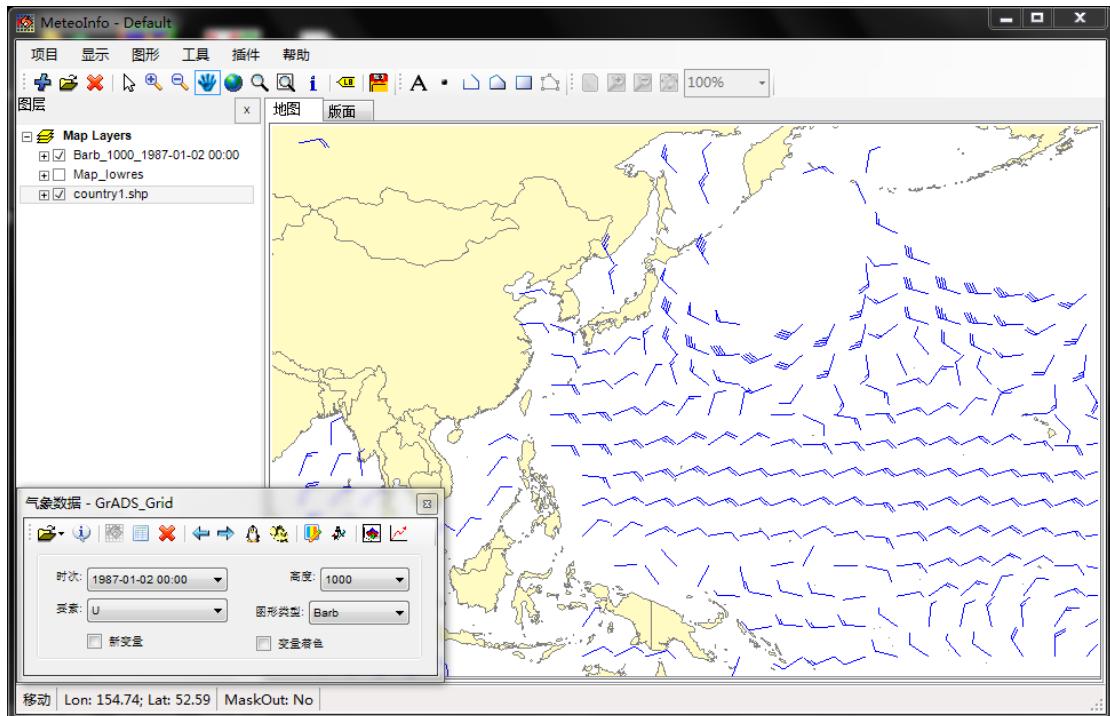


选中‘变量着色’选项，风场矢量能够根据第三个变量显示成不同颜色。

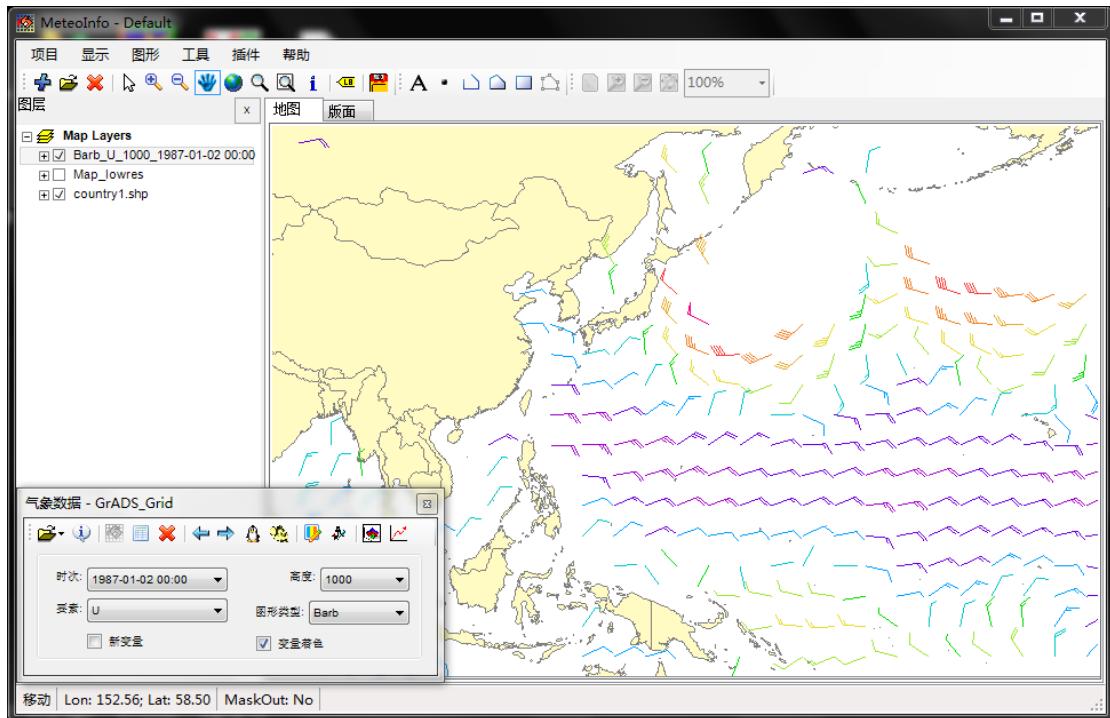


## 风向杆图形

通过 U、V 分量合成的风向杆图。

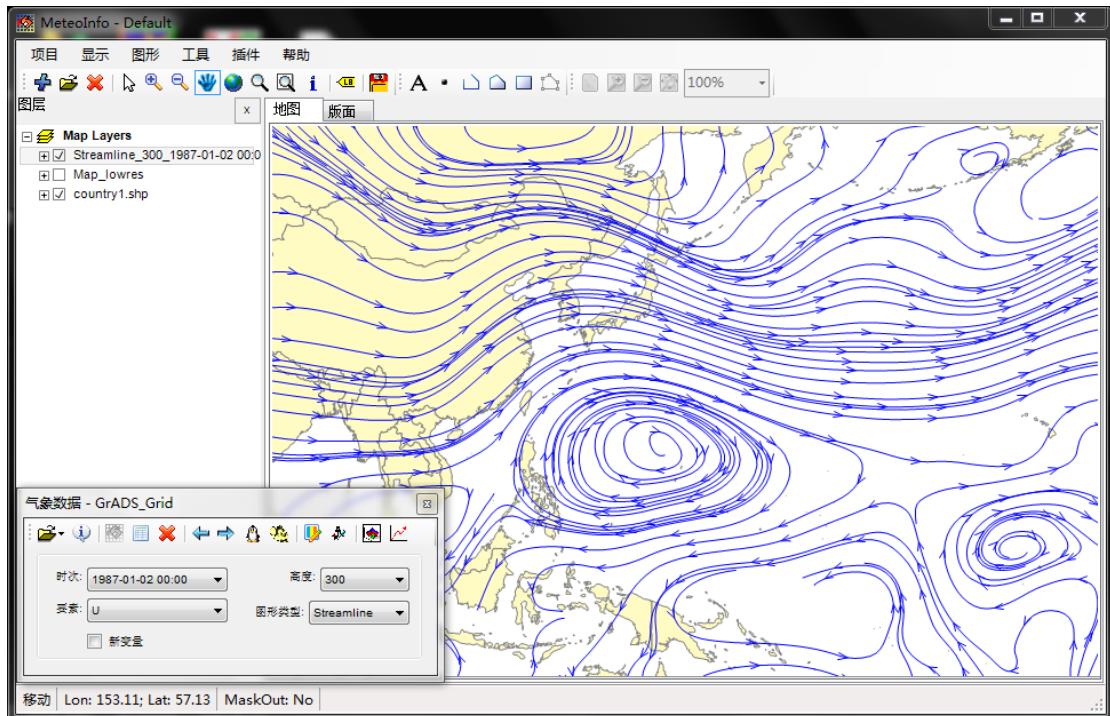


根据第三个变量显示着色后的风向杆图。

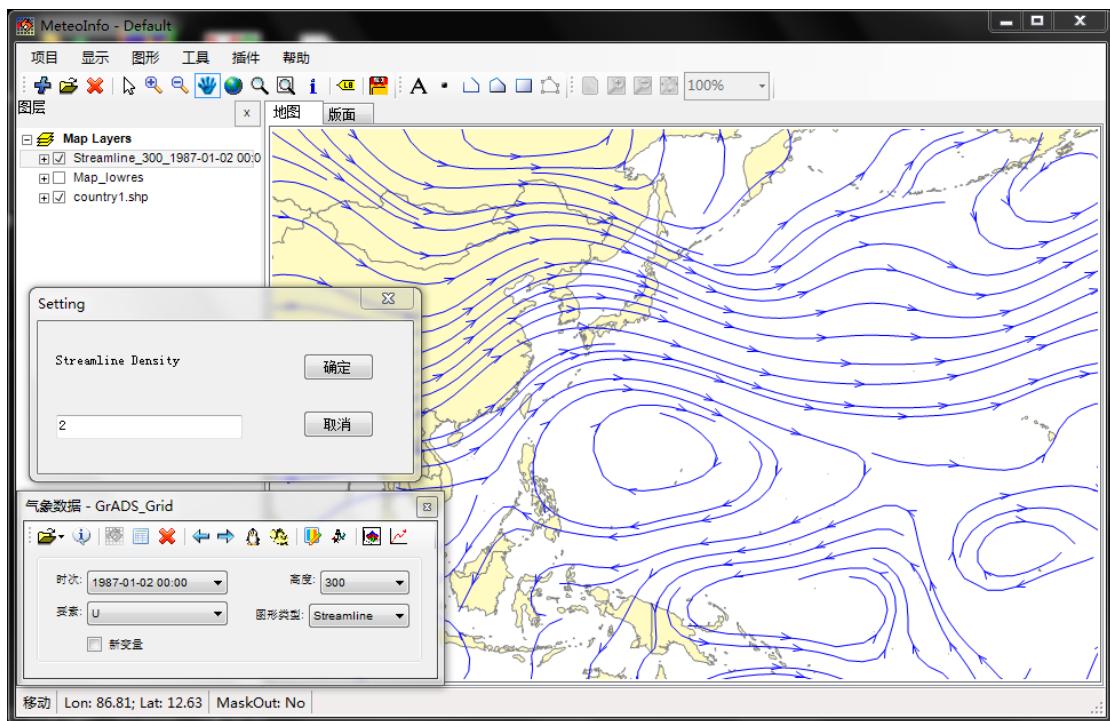


## 流线图

通过 U、V 分量生产流线图。



流线密度可以在 1 至 10 之间设置，缺省设置为 4。将其设置为 2，会使流线变得稀疏。



## 显示站点数据

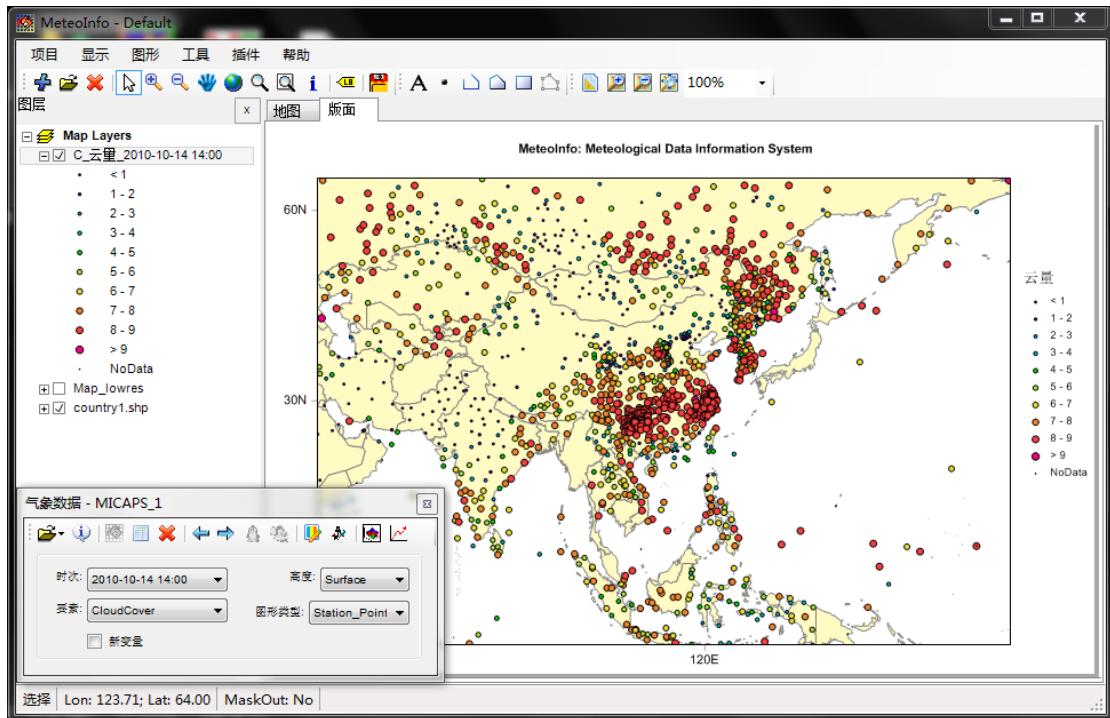
站点数据能够被显示成如下图形类型：

- ◆ Station Point （点图）
- ◆ Station Info （点图，所有字段添加入属性数据）
- ◆ Contour （等值线，插值计算）
- ◆ Shaded （等值线填色，插值计算）
- ◆ Barb （风向杆）
- ◆ Weather Symbol （天气现象）
- ◆ Station Model （站点填图）

下面的示例图形是由 MICAPS 第一类数据 ‘10101414.000’ 绘制出来的。

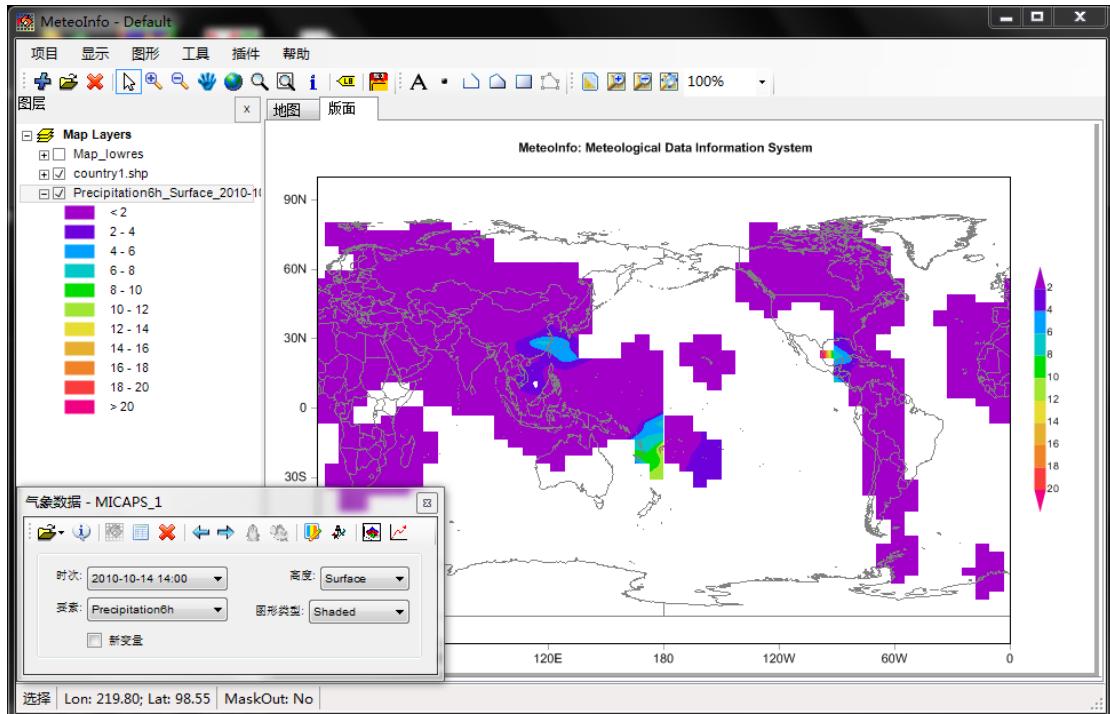
## 站点点图

设置 ‘图形类型’ 为 ‘Station\_Point’。

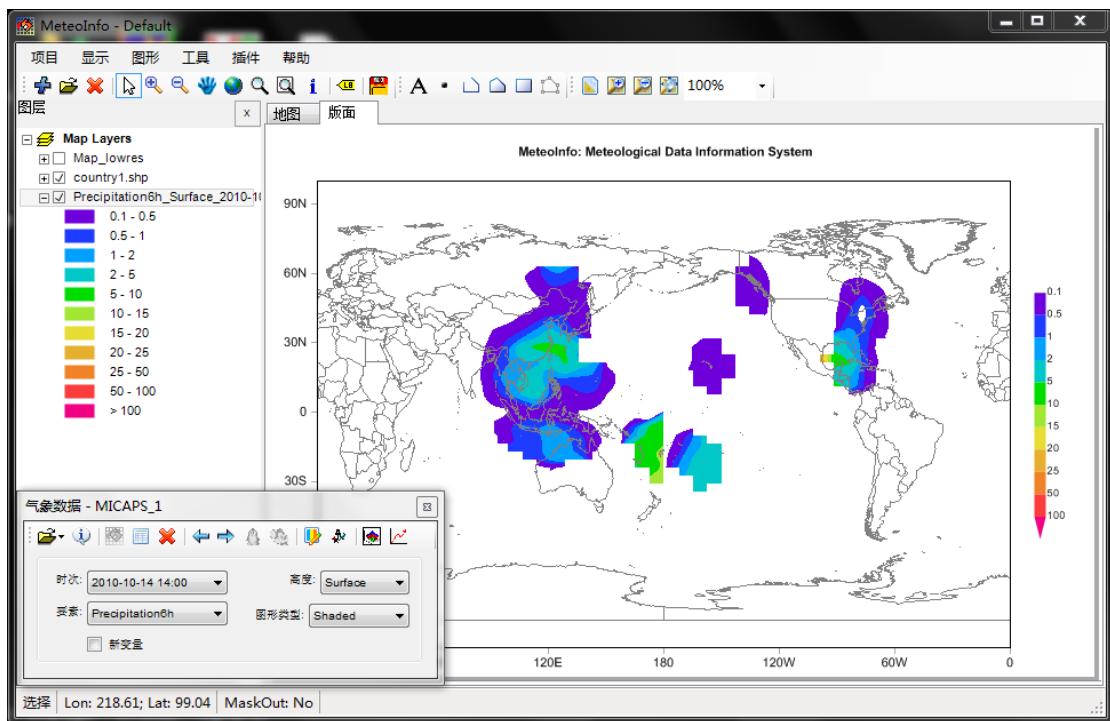


## 等值线和填色图形

要绘制站点数据的等值线，必须先将站点数据插值为格点数据。软件会自动设置插值的参数和图例，本例中的自动设置并不合适，必须重新设置。



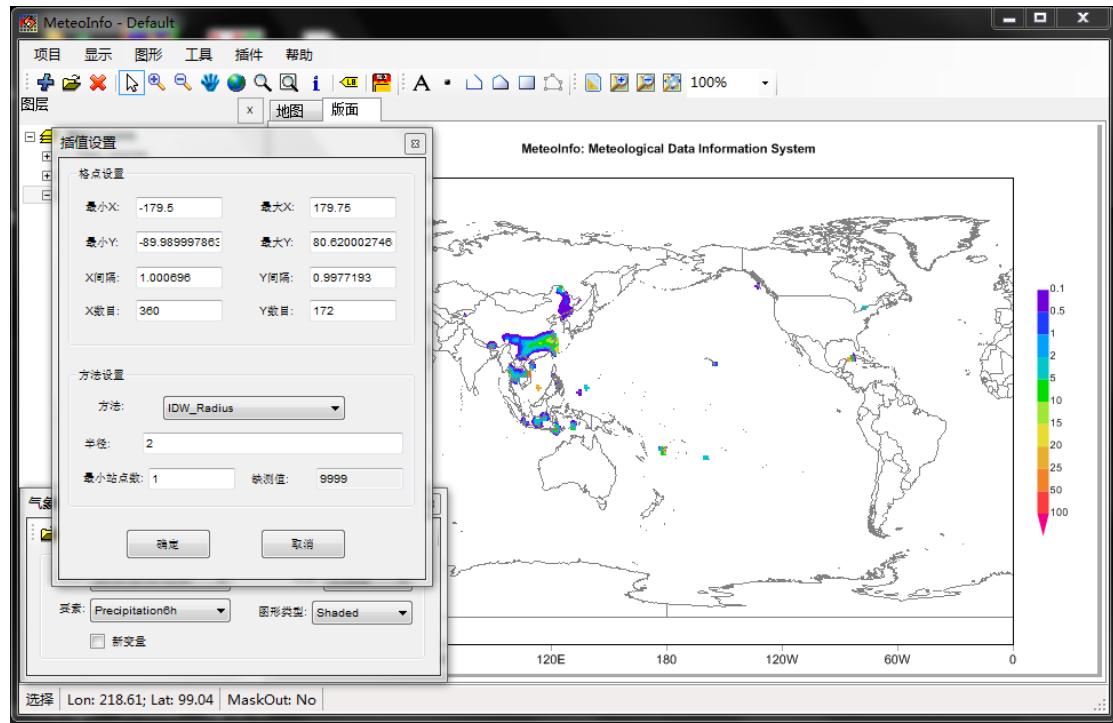
图例重新设置后效果如下图，插值成的格点间隔过大，效果并不好。我们再试着改变插值参数设置。



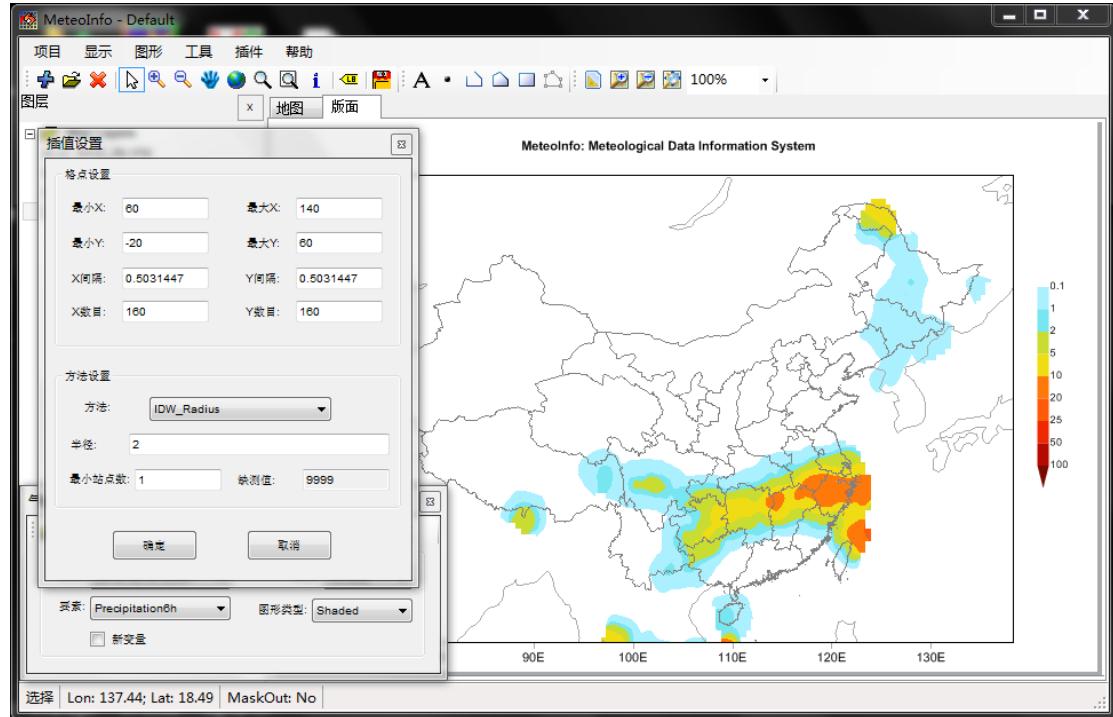
点击‘设置’按钮，打开‘Interpolate’对话框。目前软件只提供 IDW (Inverse Distance Weighted, 反距离权) 方法，该方法有两个子方法：IDW\_Radius 和 IDW\_Neighbors。IDW\_Radius 方法中只有在搜索半径内的点才参与计算格点值，搜索半径内无站点数据的格点被赋值为无效值。IDW\_Neighbors 方法中格点值由最近的几个站点值计算获得，一些分布比较连续，变化较小的气象要素（如气温）有时候适合用这种方法。



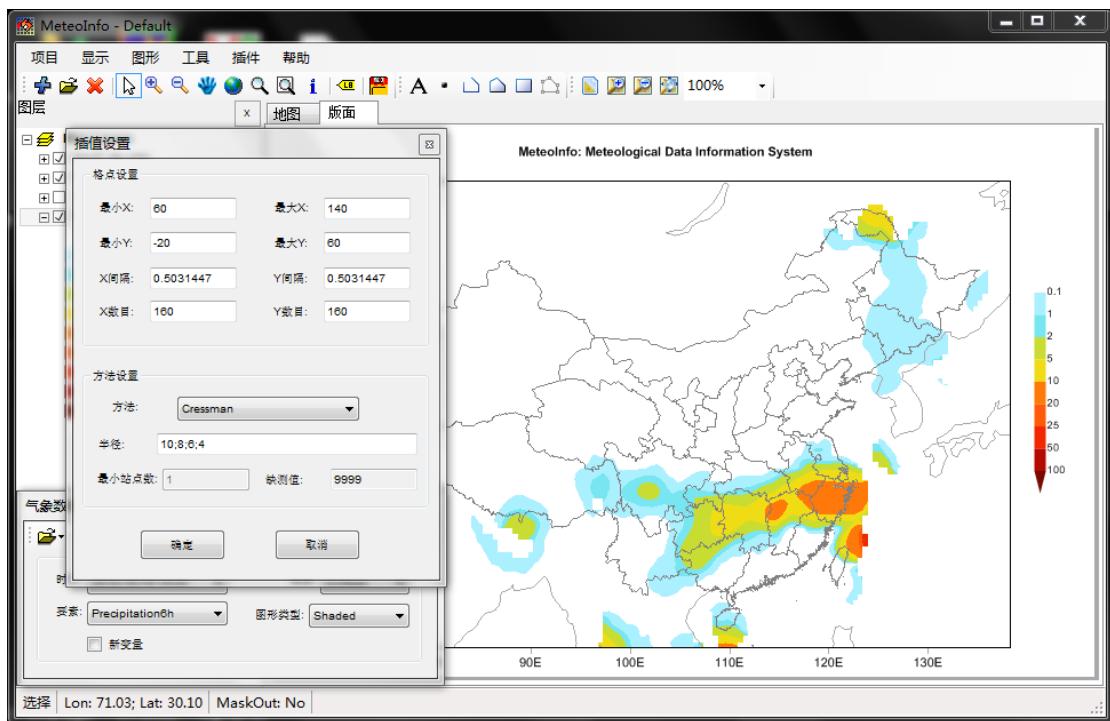
改变插值设置为下图所示，点击‘OK’按钮，软件将根据新的插值设置重新计算格点值并追踪等值线。



如果我们只对某个区域感兴趣，如中国，我们可以缩小格点数据范围以节省计算时间。

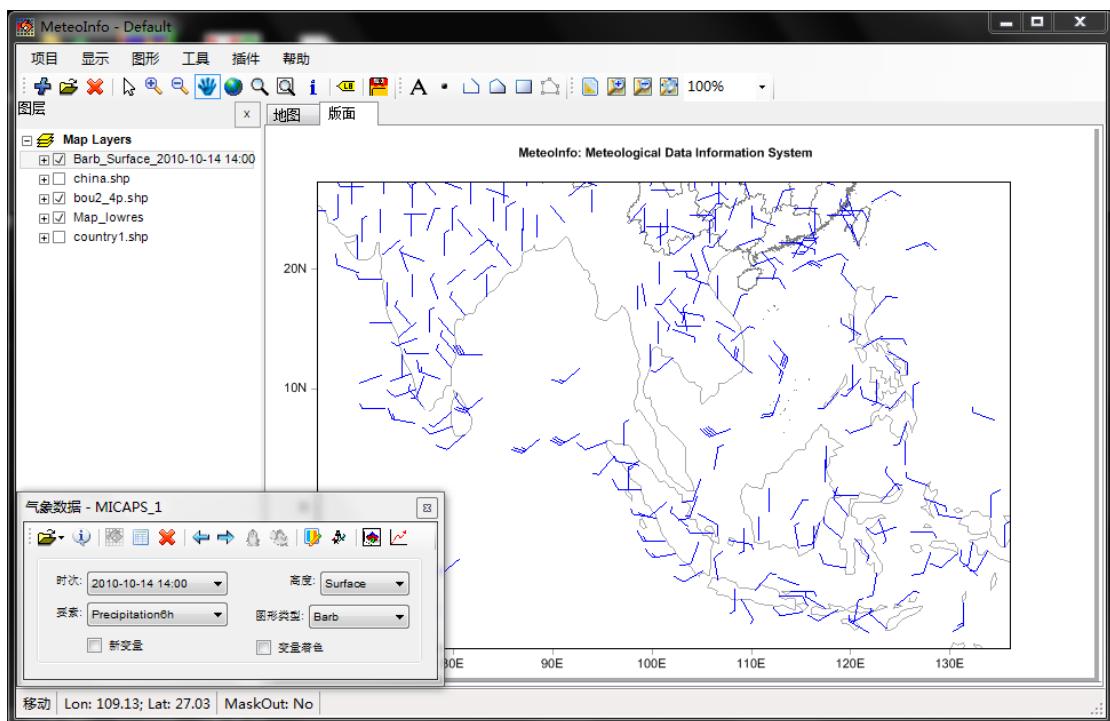


Cressman 客观分析法（Cressman, 1959）是一种迭代分析方法。下图是一个例子。



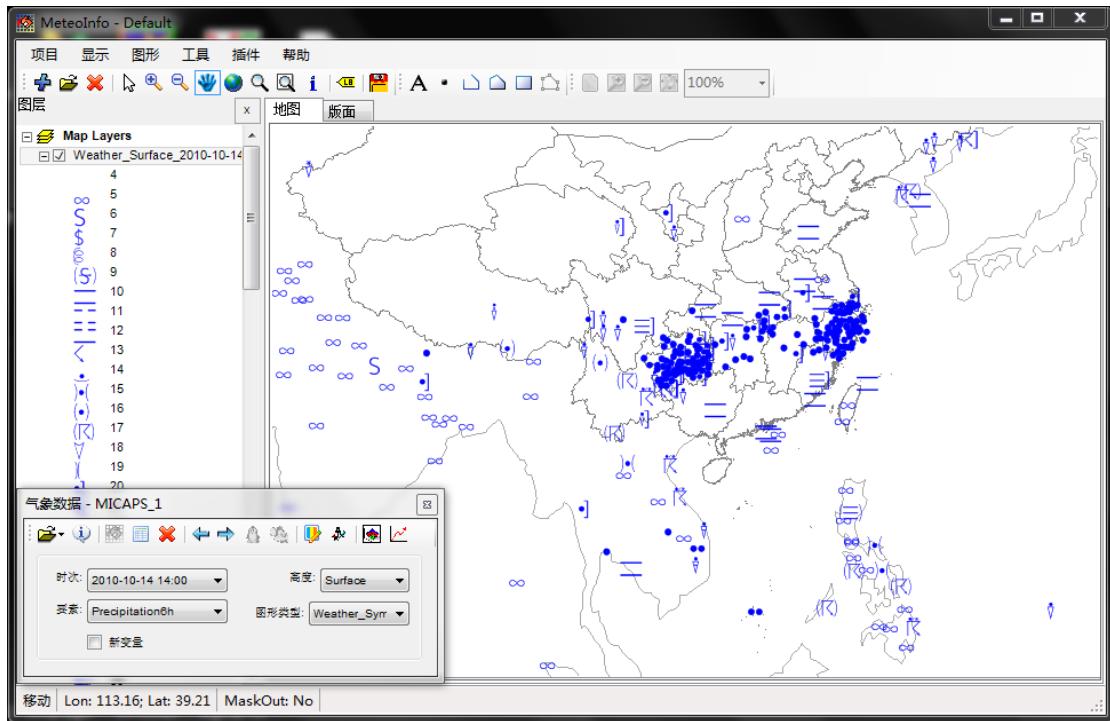
## 风向杆图形

选择图形类型为‘Barb’，图形显示如下。



## 天气现象图形

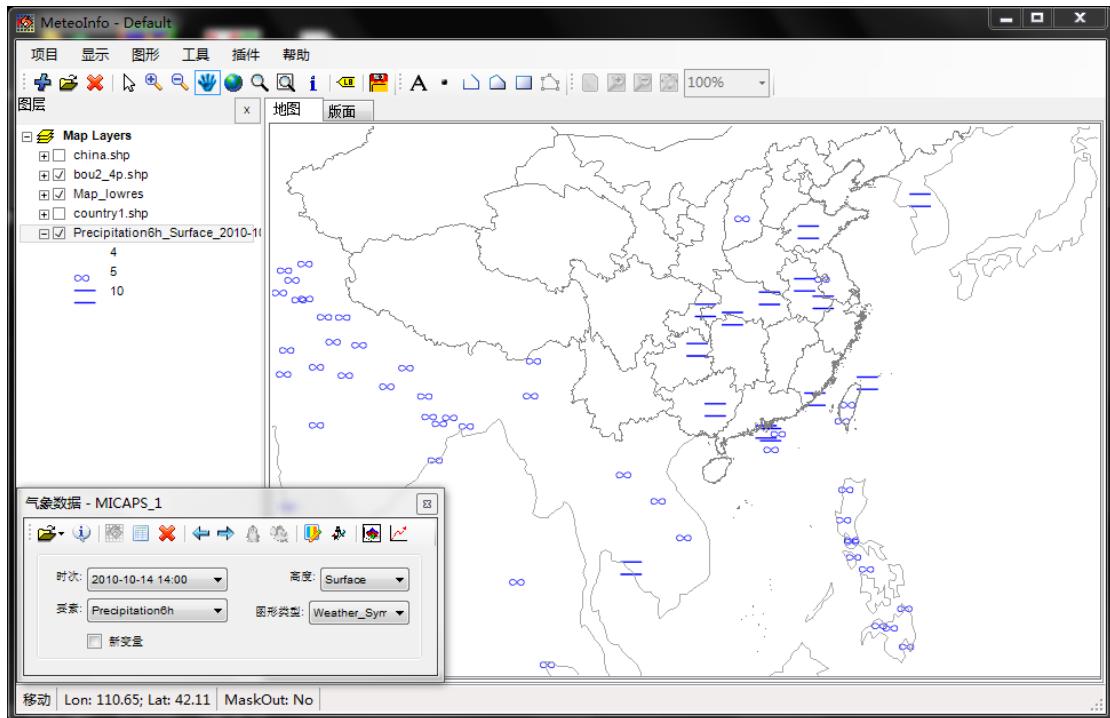
设置图形类型为‘Weather Symbol’，缺省状态下所有天气现象都显示在图中。



点击‘设置’按钮，在‘Weather Symbol Set’对话框中选择‘Smoke, Haze, Mist’，点击‘确定’按钮。

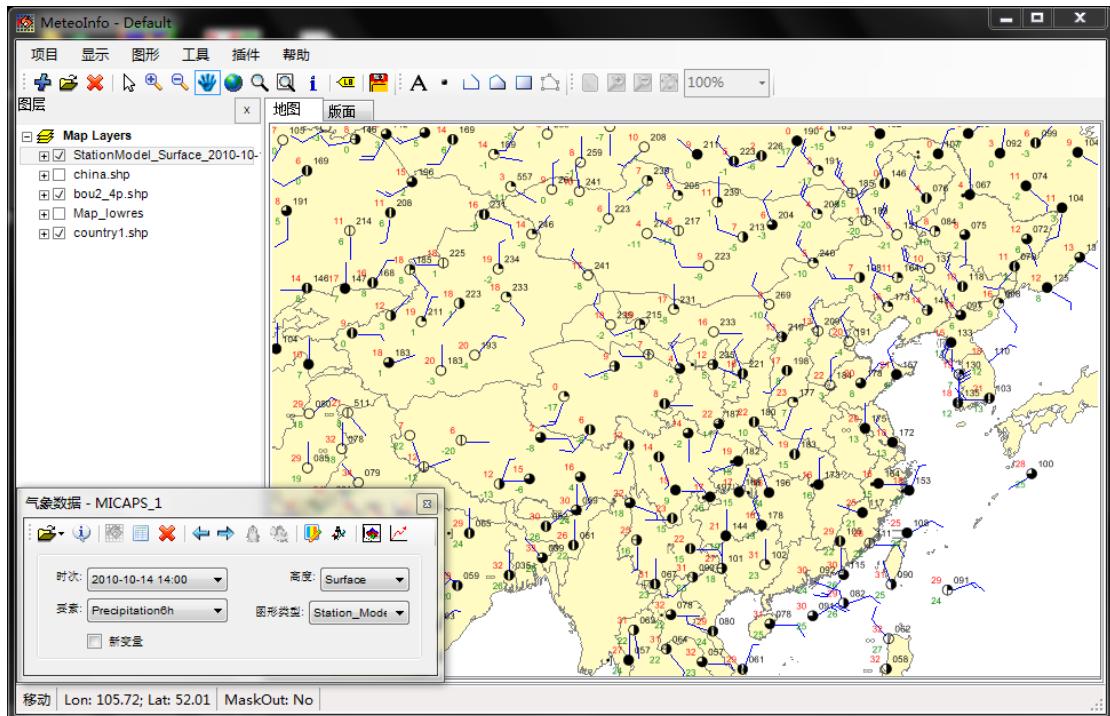


图形显示如下：



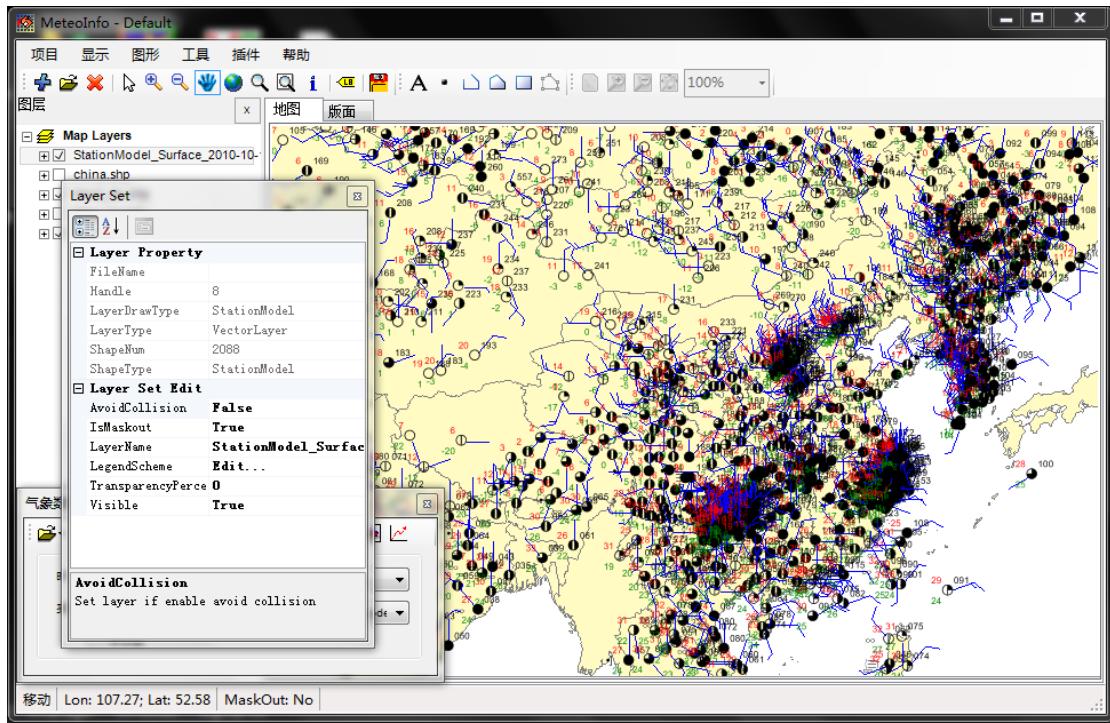
## 站点填图

设置图形类型为‘Station\_Model’，画出站点填图图形。



在生成站点填图图层时，‘AvoidCollision’（避免重叠）属性会自动设为‘True’，只有那些不互相压盖的站点填图符号才会显示。放大或缩小地图范围时，软件会根据地图范围自动调整站点的显示。

如果设置 ‘AvoidCollision’ 属性为 ‘False’，所有地图显示区域范围的站点填图符号都会显示出来，这时可能发生严重的重叠遮盖现象。

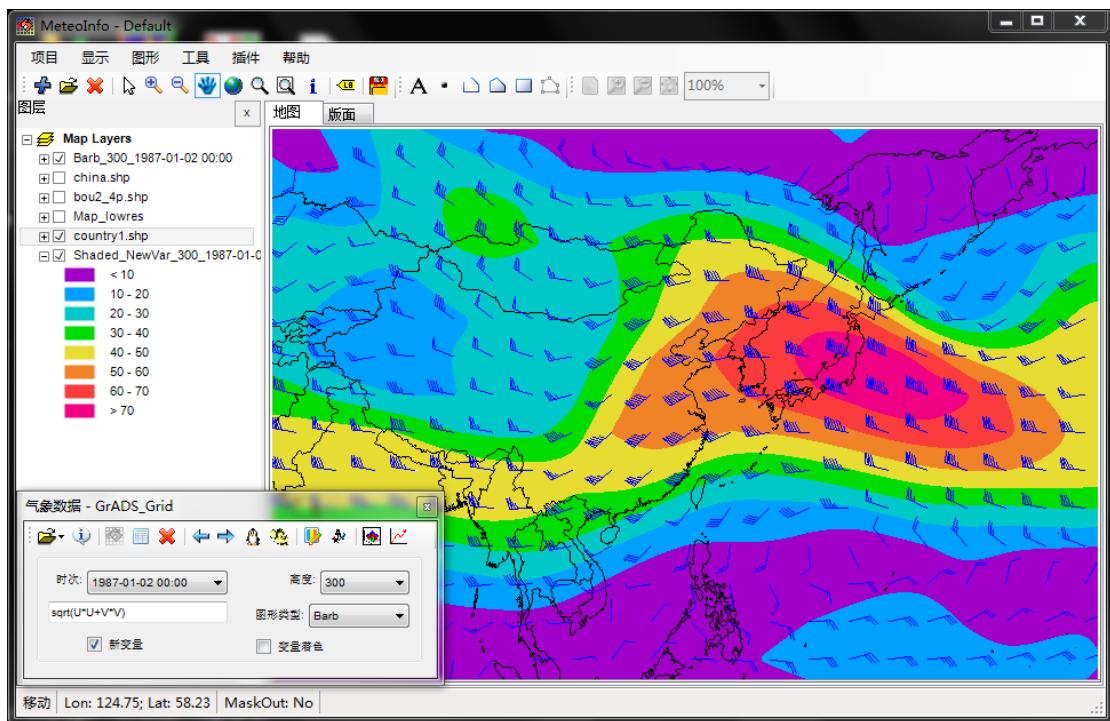


## 显示新变量

软件包含了公式解释功能，可以用已有的变量编写一个公式生产新的变量。例如用 U、V 变量编写公式 ‘ $\sqrt{U^2+V^2}$ ’ 生产风速变量。

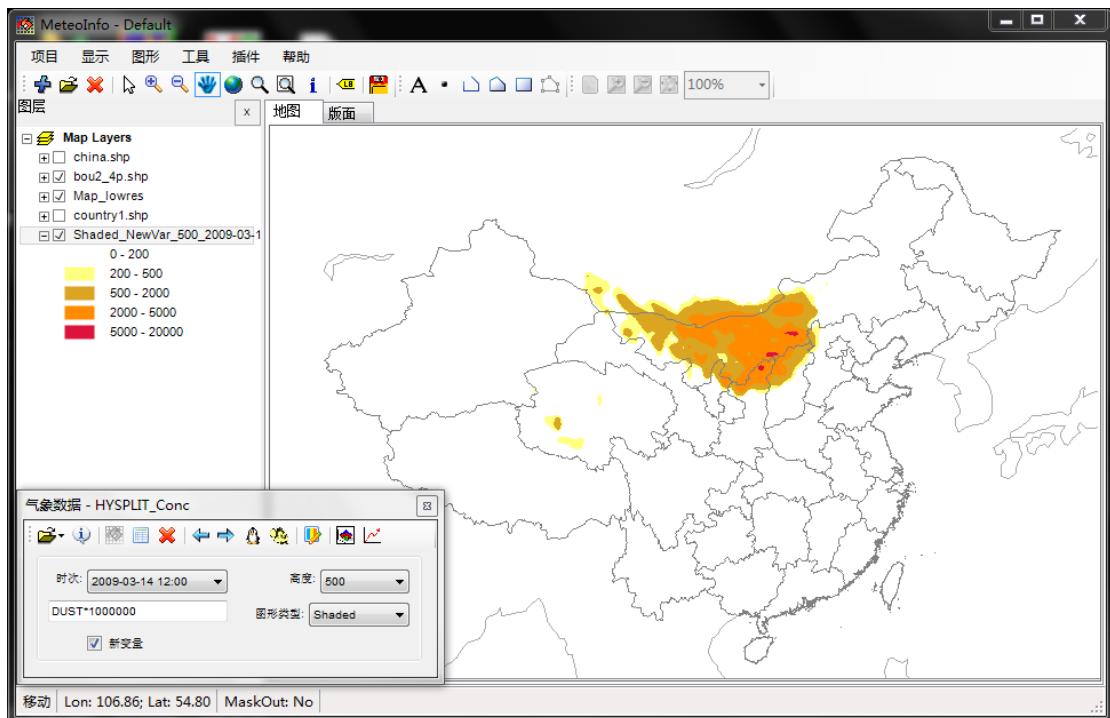
公式中可以用到的运算符包括：‘+’，‘-’，‘\*’，‘/’，‘%’，‘^’。

公式中可以用到的函数包括：‘abs’，‘acos’，‘asin’，‘atan’，‘cos’，‘exp’，‘log’，‘log10’，‘sin’，‘sqrt’，‘tan’。

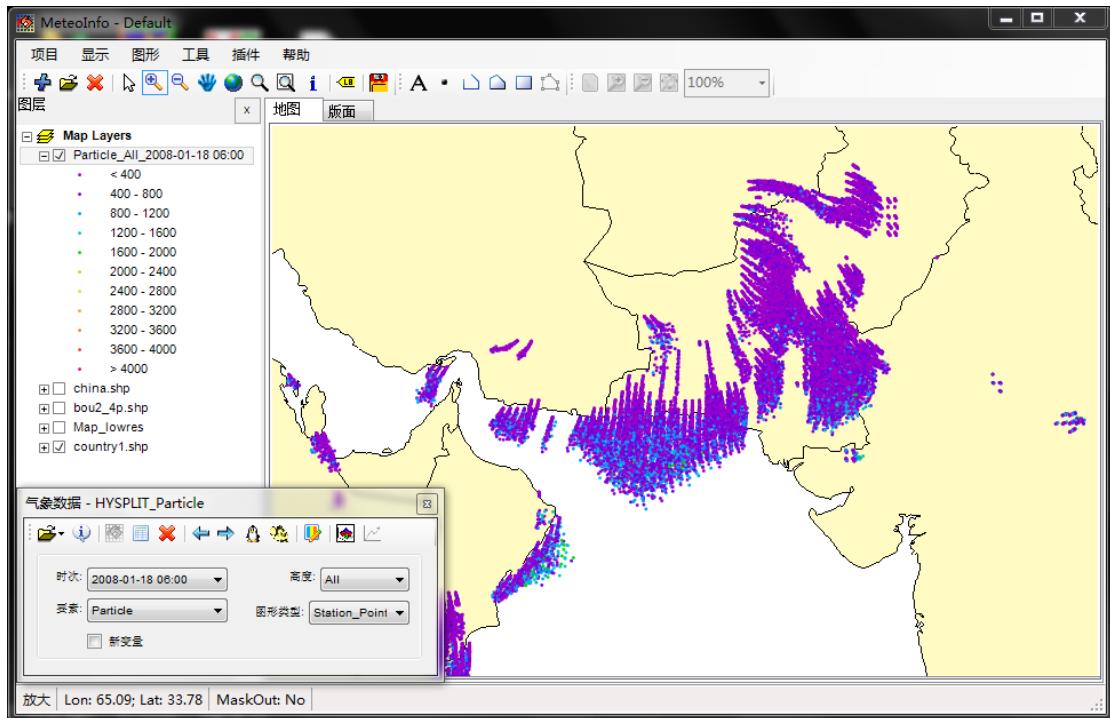


## 显示HYSPLIT模式输出数据

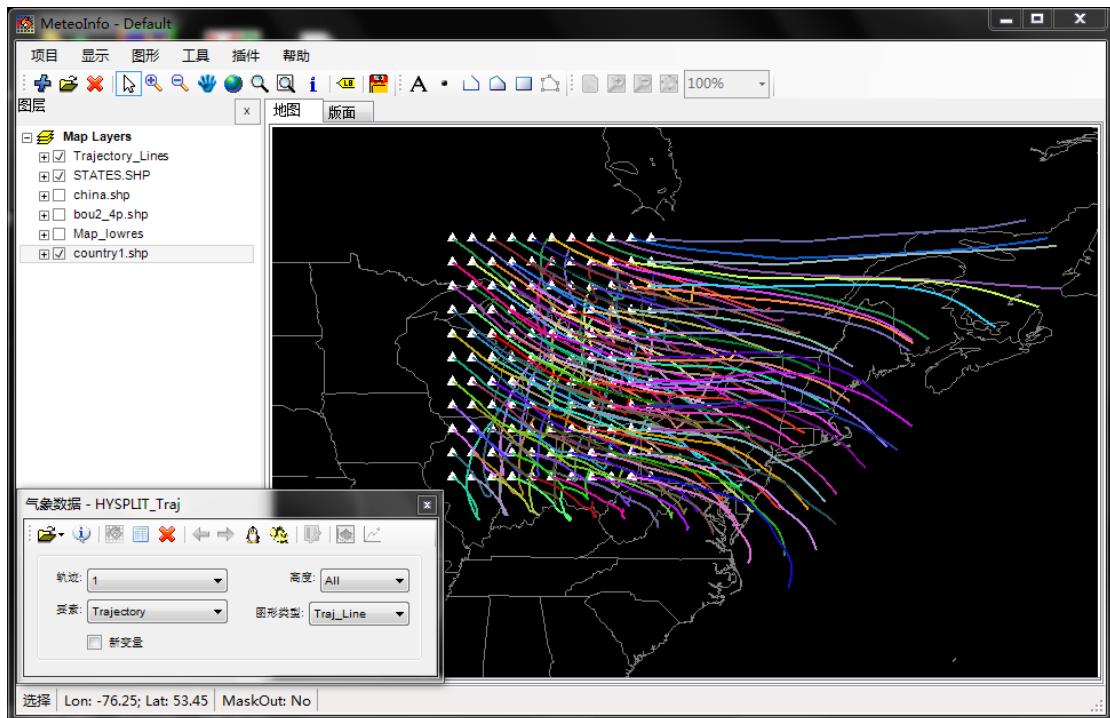
浓度数据图形。



大气颗粒分布图形



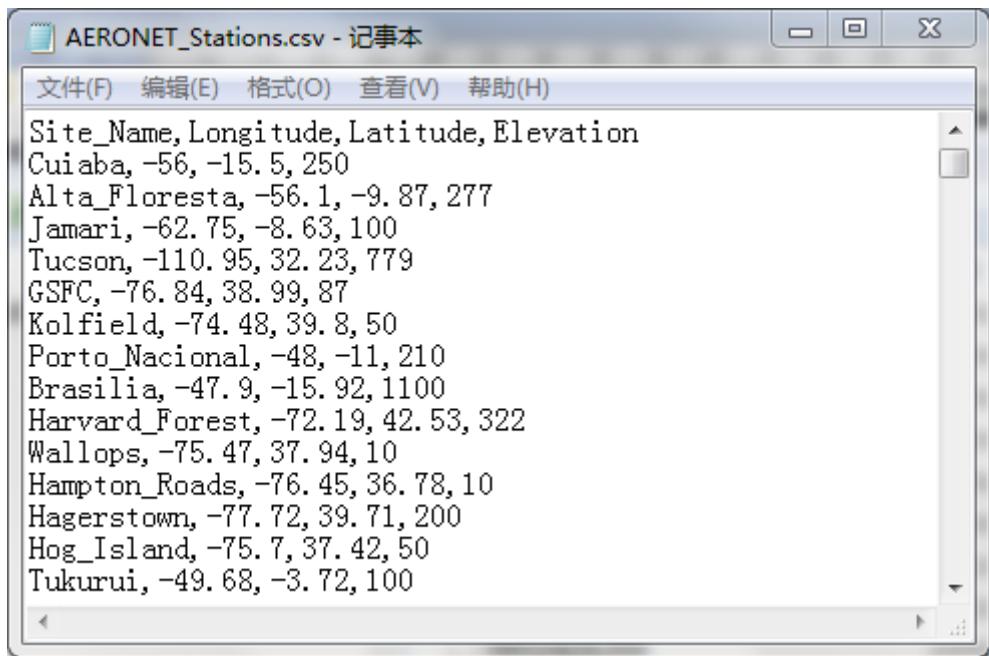
气团轨迹图形。



## 站点（经纬度）数据

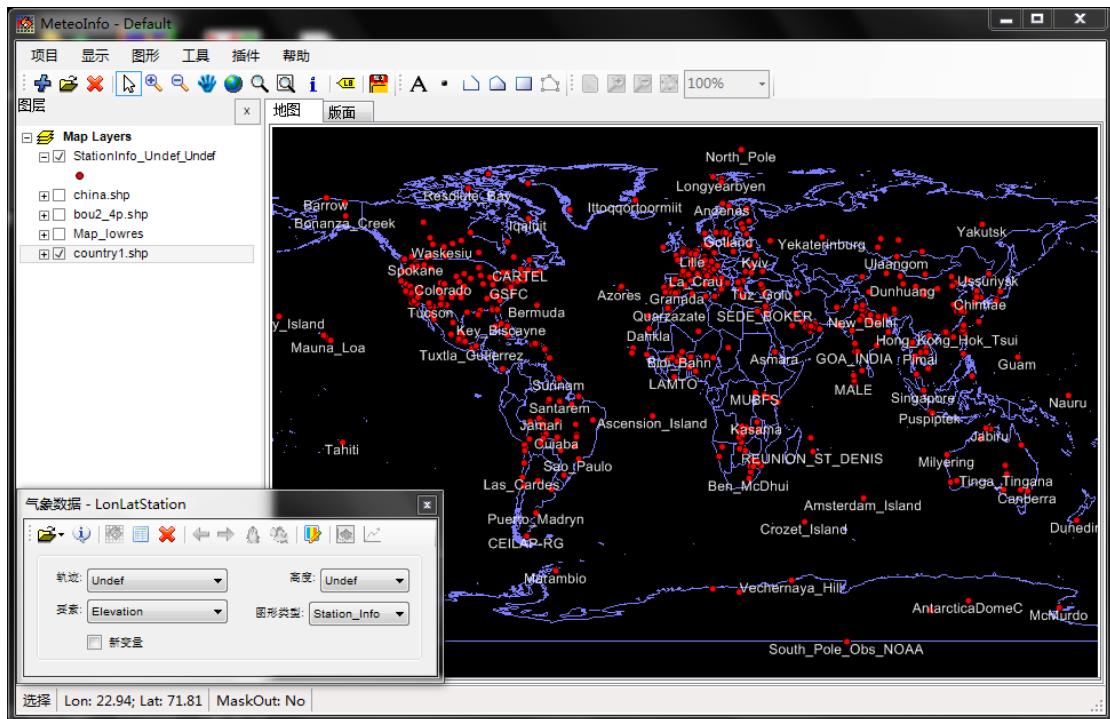
MeteoInfo 支持有经纬度信息的站点文本文件。文件头只有一行，是每一列的名称。文件的第一列是站点标识（站号或者站名），第二列和第三列是站点的经度和纬度，第四列以后是各个要素。列和列之间要用逗号分隔（不能是中文的逗号）。如果你的站点数据不是这

种格式，只需要在 Excel 里按照上述说明处理一下，另存为 CSV 文件即可。数据示例如下：



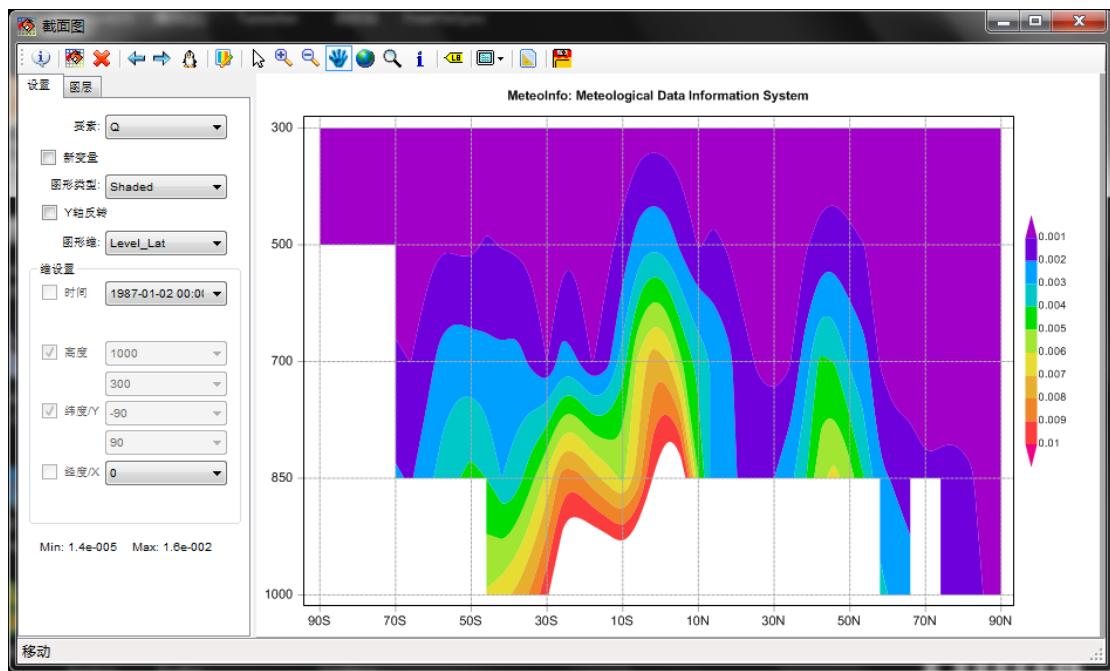
```
Site_Name,Longitude,Latitude,Elevation
Cuiaba,-56,-15.5,250
Alta_Floresta,-56.1,-9.87,277
Jamari,-62.75,-8.63,100
Tucson,-110.95,32.23,779
GSFC,-76.84,38.99,87
Kolfield,-74.48,39.8,50
Porto_Nacional,-48,-11,210
Brasilia,-47.9,-15.92,1100
Harvard_Forest,-72.19,42.53,322
Wallops,-75.47,37.94,10
Hampton_Roads,-76.45,36.78,10
Hagerstown,-77.72,39.71,200
Hog_Island,-75.7,37.42,50
Tukurui,-49.68,-3.72,100
```

打开 AeroNet 全球站点分布数据文件，选择图形类型为 ‘Station\_Info’，生成站点图层。

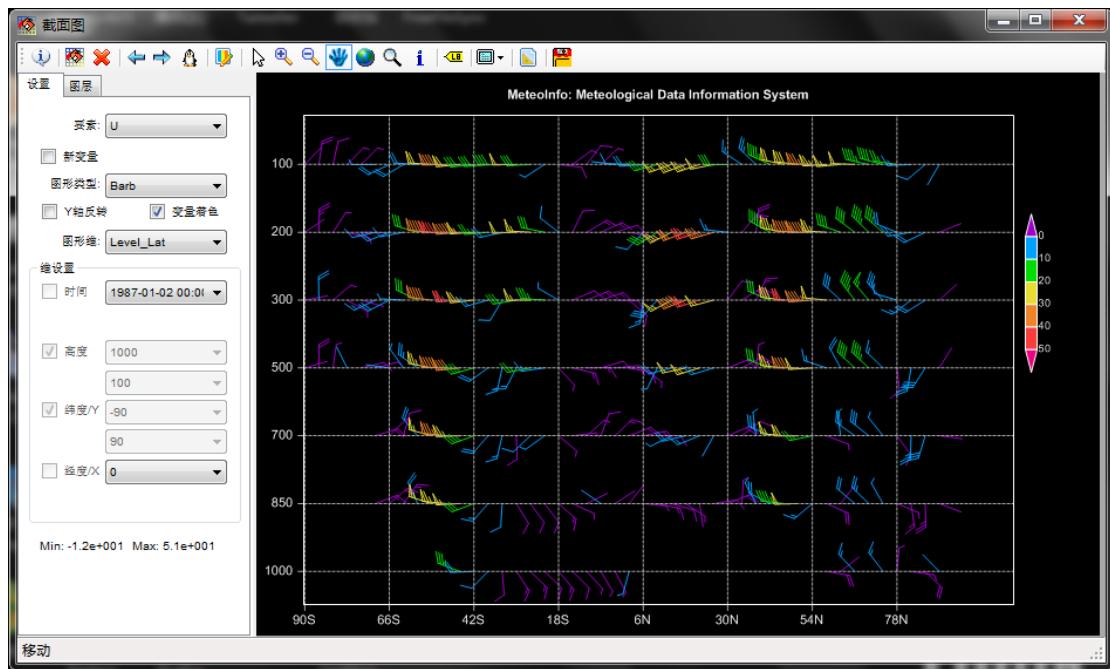


## 截面图

点击‘截面图’按钮打开下面的窗口。选择‘要素’、‘图形类型’和‘图形维’，设置相应的维，点击‘显示数据图形’按钮画出截面图。截面图有两个可变维。

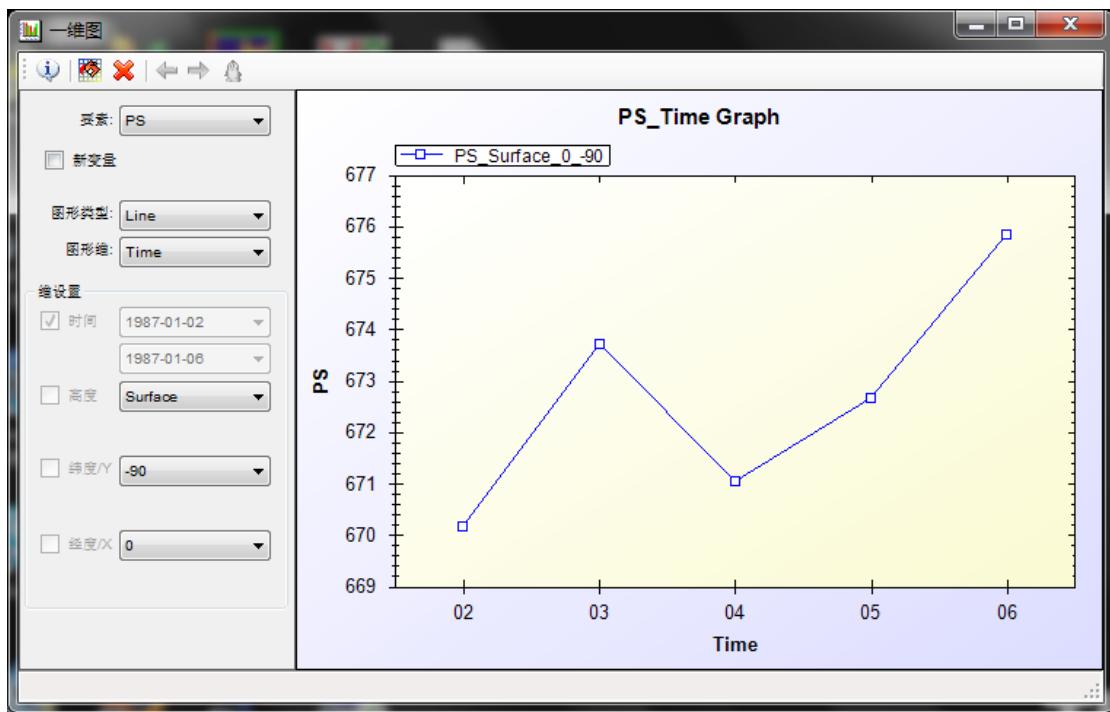


风廓线图。



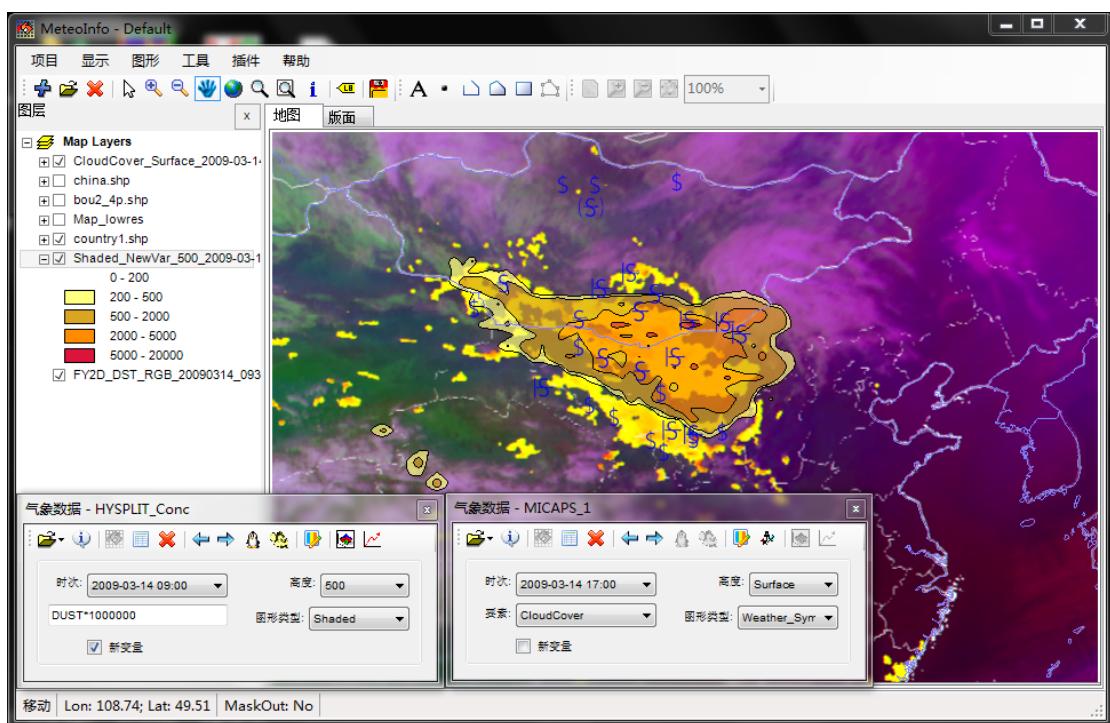
## 一维图

点击‘一维图’按钮打开下面的窗口。一维图的设置与截面图类似，但一维图只有一个可变维。



## 模式结果验证的例子

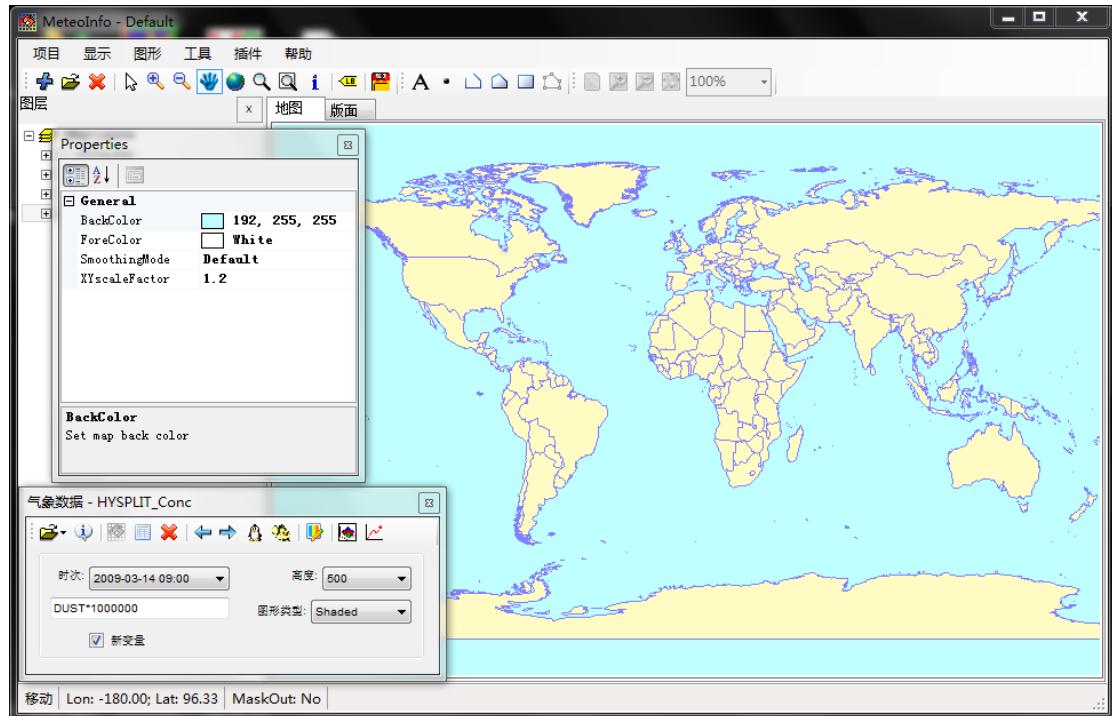
MeteoInfo 中可以显示模式结果、站点观测和卫星观测图像，因此可以很容易通过比较模式和观测进行某些模式结果的验证。下图是一个沙尘暴模式预报和站点观测、卫星观测对比的例子。MeteoInfo 中多边形图层可以设置颜色的透明度，从而更容易和卫星观测进行对比。



# 地图设置

## 地图属性

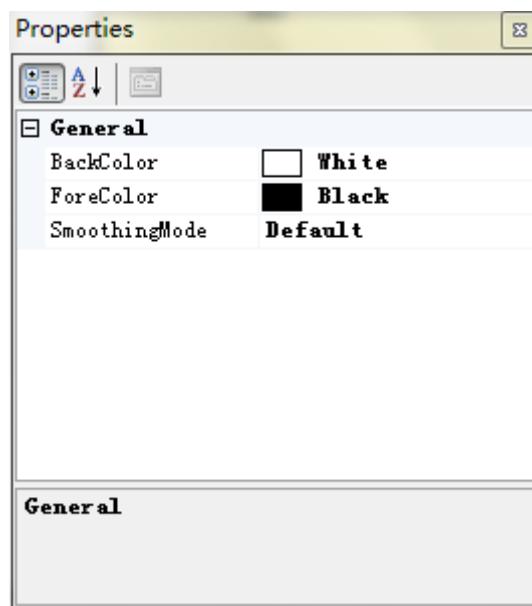
点击‘图形’菜单下的‘地图属性’子菜单打开‘Map Property Set’对话框。可以设置图形背景色、前景色、SmoothingMode、X/Y比例。



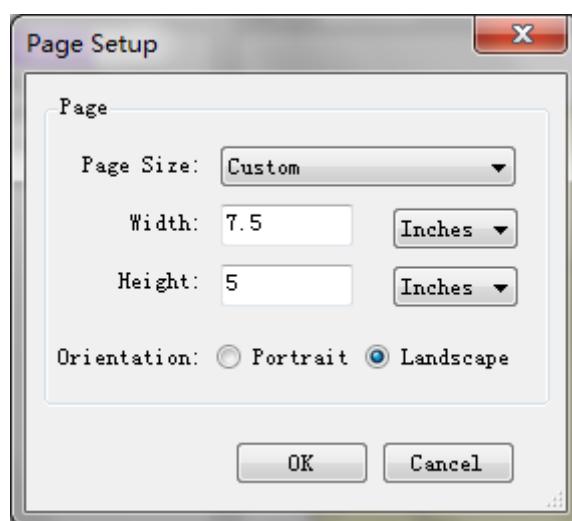
MeteoInfo 软件中使用 GDI+绘图技术，缺省状态下图形显示较为快速但有锯齿，为了获得更高质量的图形，可以设置‘SmoothingMode’为‘HighQuality’或‘AntiAlias’消除锯齿。

## 版面属性

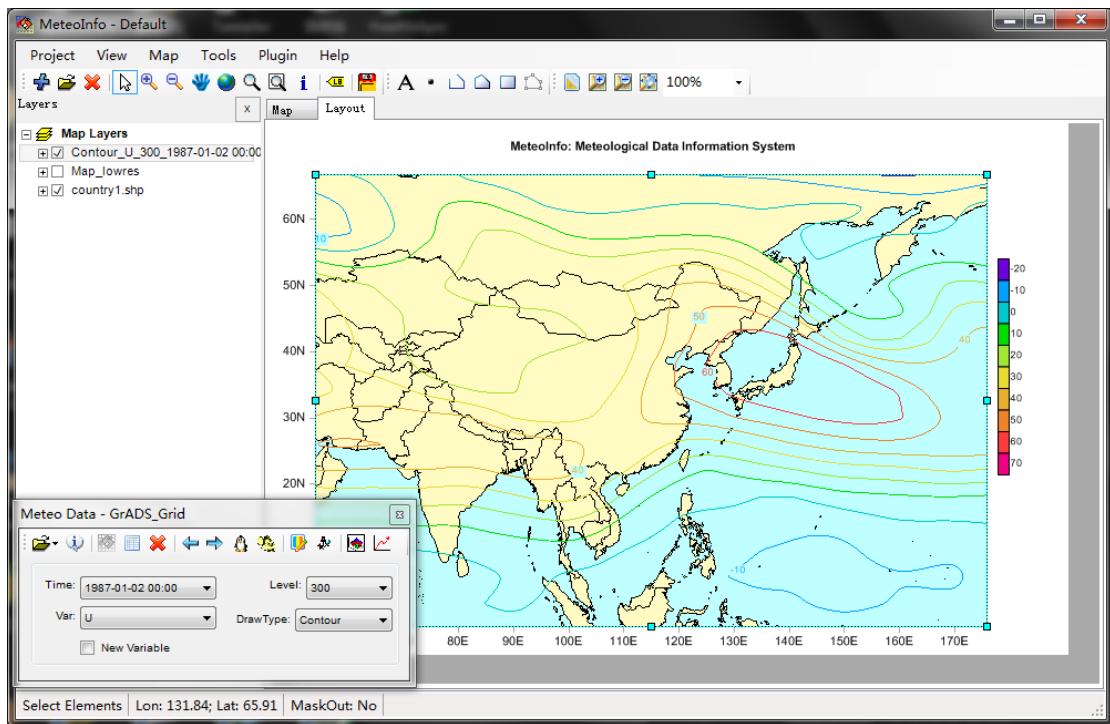
版面（Layout）主要是用来对地图图形进行排版，以便能输出更专业、美观的图形。包含的主要组成部件（Element）有：地图、图例、图名、插图、用户图形等。点击‘图形’菜单下的‘版面属性’子菜单打开属性对话框。可以设置版面的背景色、前景色、SmoothingMode。



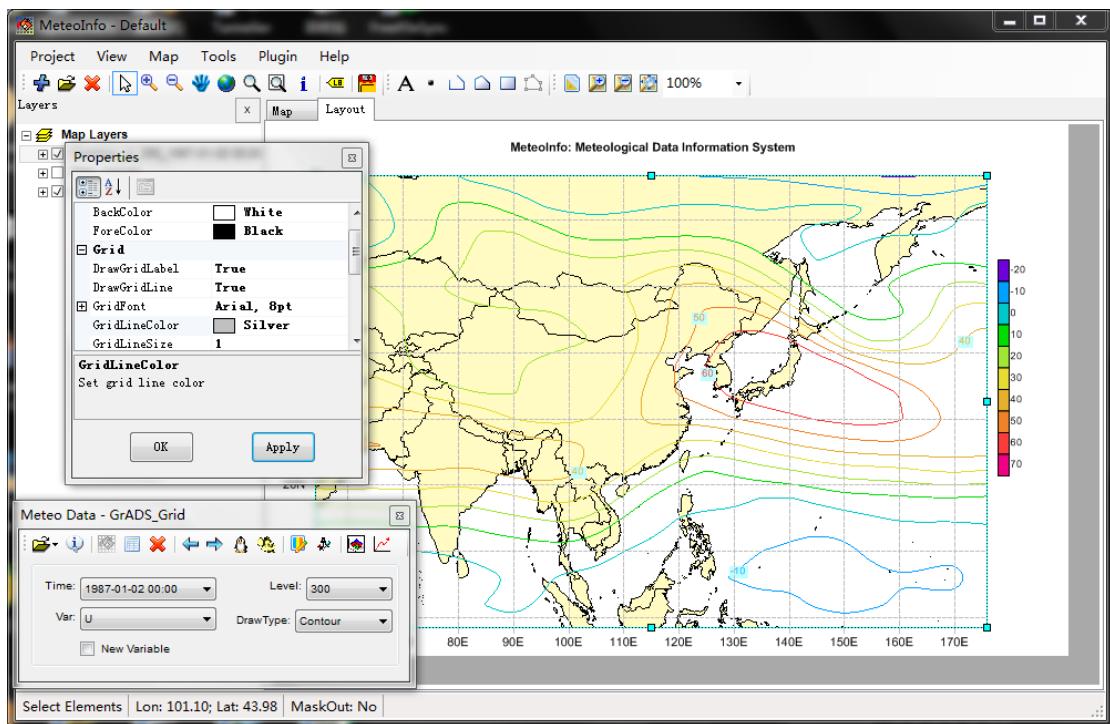
当版面页处于显示状态时版面工具栏处于可用状态。版面的页面大小可以用放大页面、缩小页面等按钮来实现。版面的实际尺寸可以通过点击‘页面设置’按钮来进行设置。



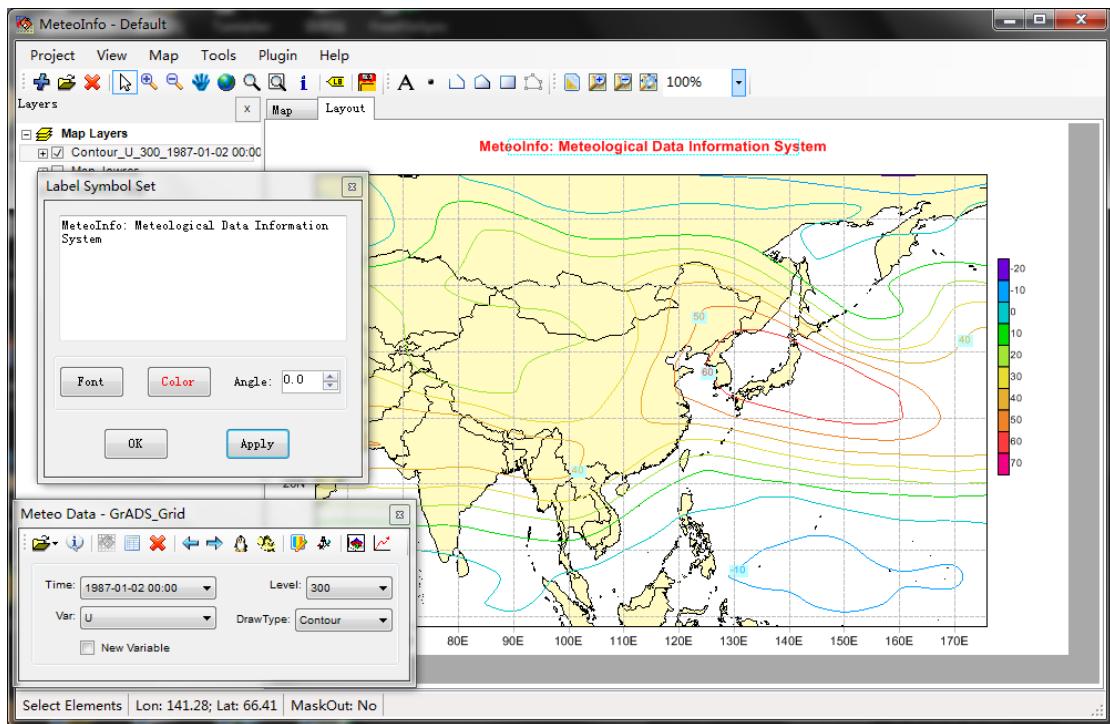
当工具栏中的‘选择’工具被选中时，可以用鼠标选中版面中的某个部件，并对其进行移动、缩放等操作。



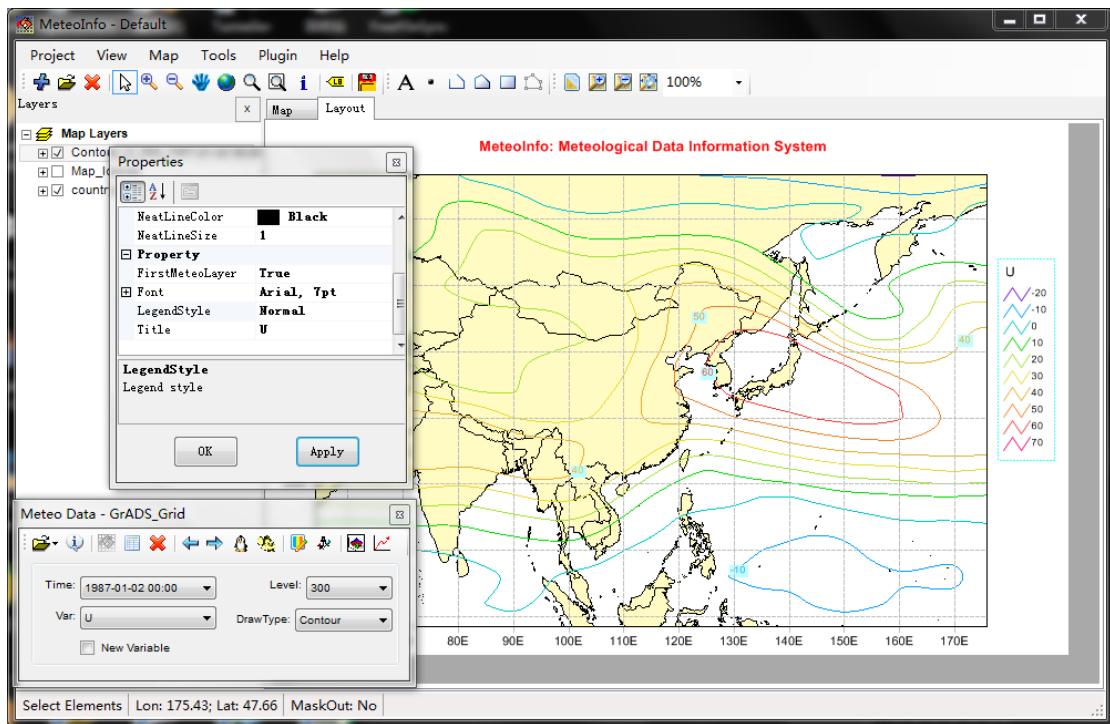
用鼠标双击地图部件打开其属性编辑对话框。这其中包括对经纬线和标注的设置。



用鼠标双击图名部件可以打开其属性编辑对话框。

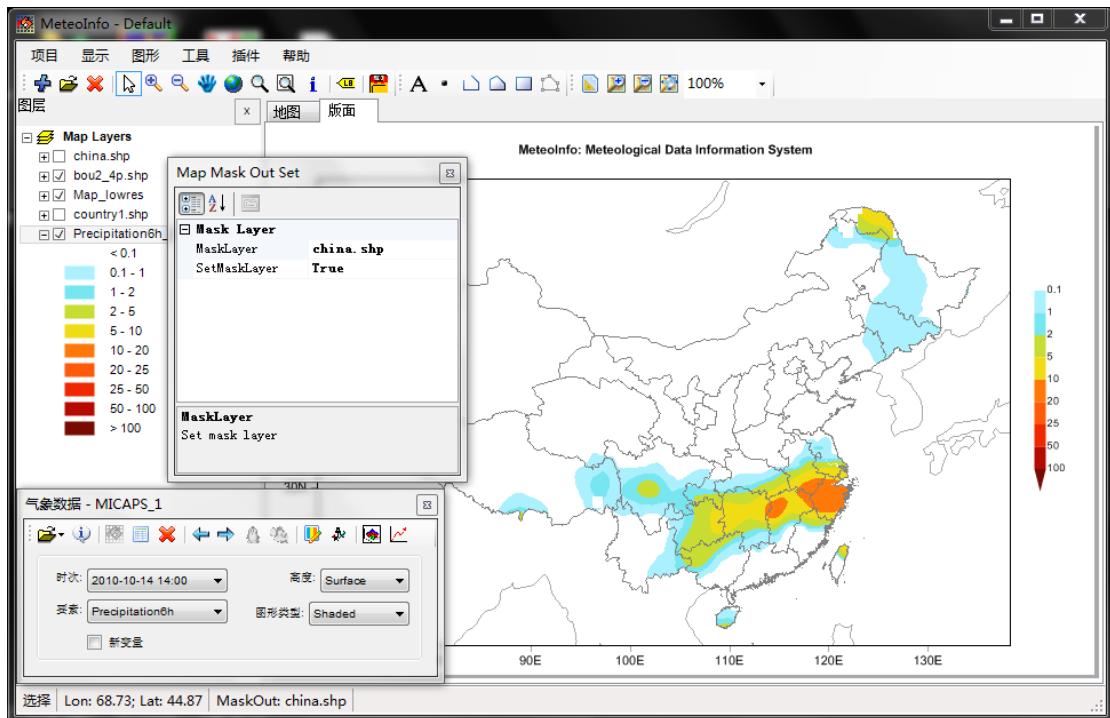


用鼠标双击图例部件打开其属性编辑对话框。当‘FirstMeteoLayer’属性设为‘True’时，图例部件会自动显示最上面的一个气象数据图层的图例。反之，显示最上面的一个在图层控制区被展开图例的图层。图例是‘SingleSymbol’的图层不会被自动显示。图例部件有三种显示类型：竖直色标栏（Bar\_Vertical）、水平色标栏（Bar\_Horizontal）和普通色标（Normal）。普通色标是指点、线、面都以本来的图元类型来显示图例。下图是将显示类型改为‘Normal’后的效果。



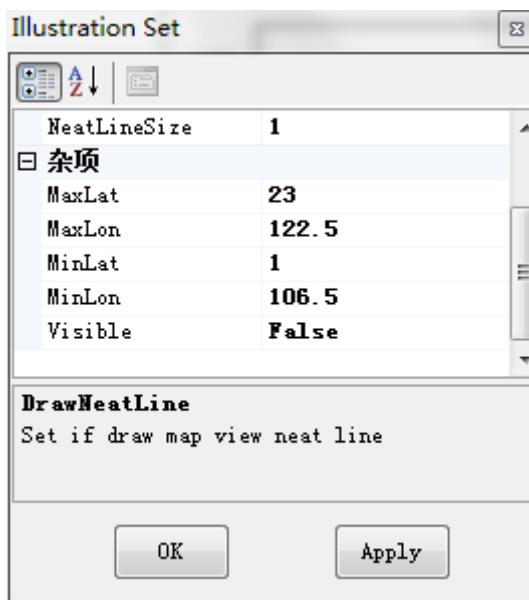
## Mask (屏蔽) 图层

MeteoInfo 可以设置屏蔽 (mask) 图层以达到屏蔽某个制定区域外图形的目的。首先要保证在打开的地图数据图层中有符合做屏蔽图层要求的图层，即必须是多边形 (Polygon) 图层。将 ‘SetMaskLayer’ 改为 ‘True’，然后选择一个屏蔽图层 (MaskLayer)。下图中选择了 ‘china.shp’ 图层为屏蔽图层，则中国区域外的气象数据被屏蔽掉。

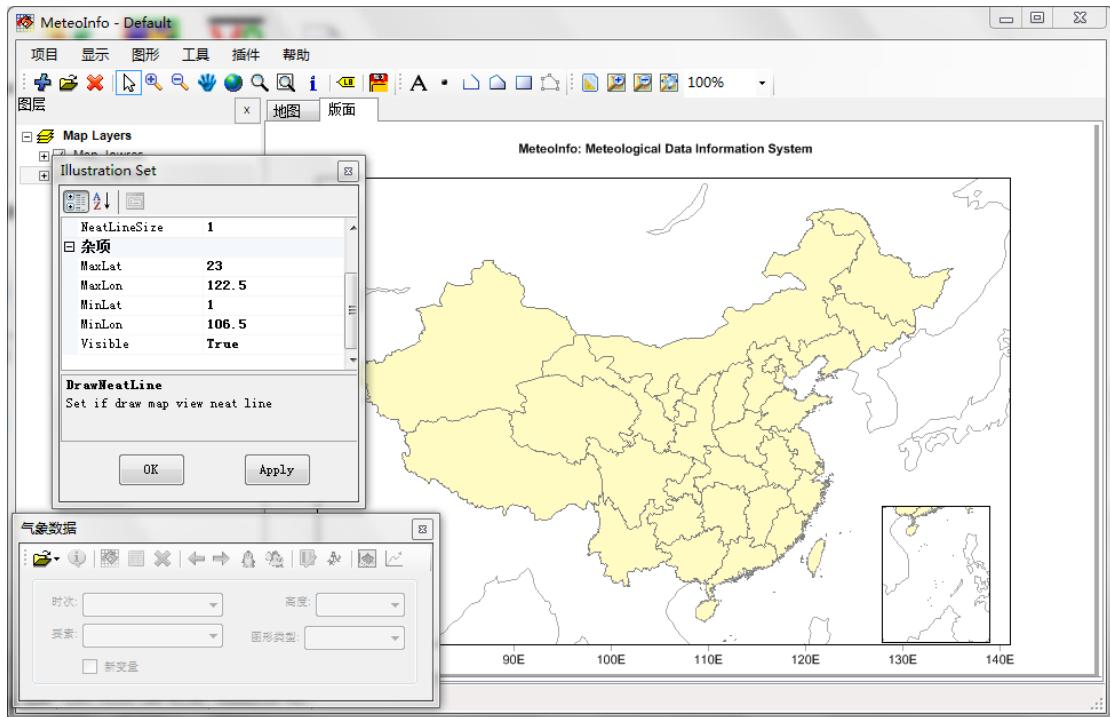


## 插图

在‘图形’菜单下选择‘插图’子菜单打开‘Illustration Set’对话框。



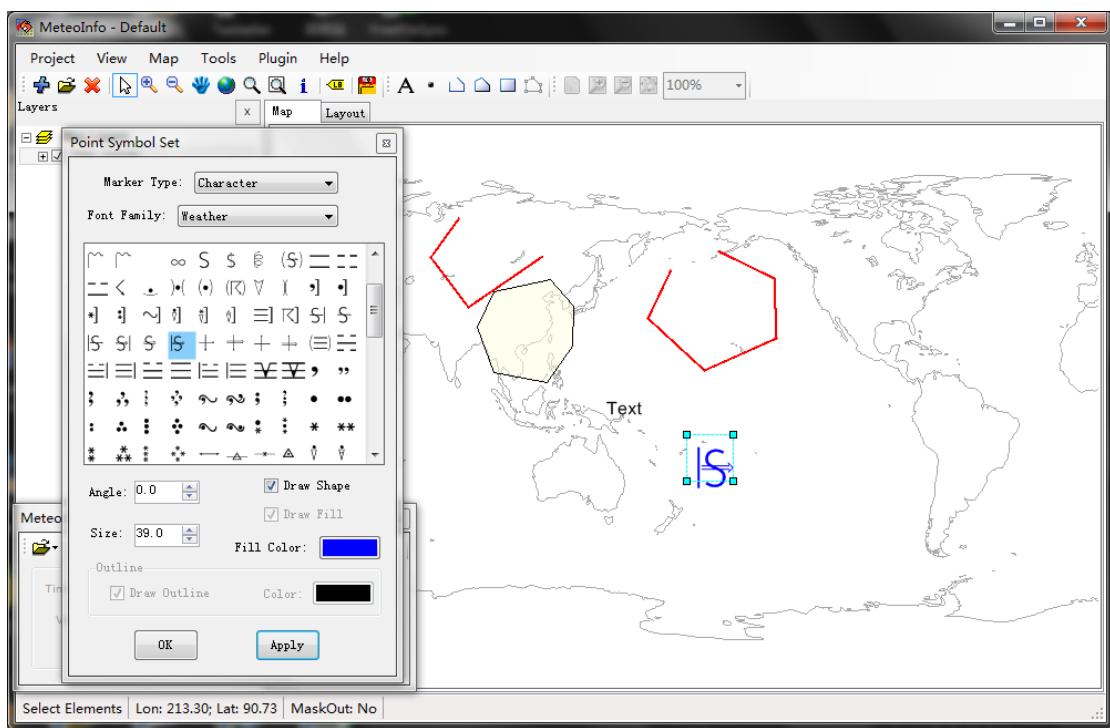
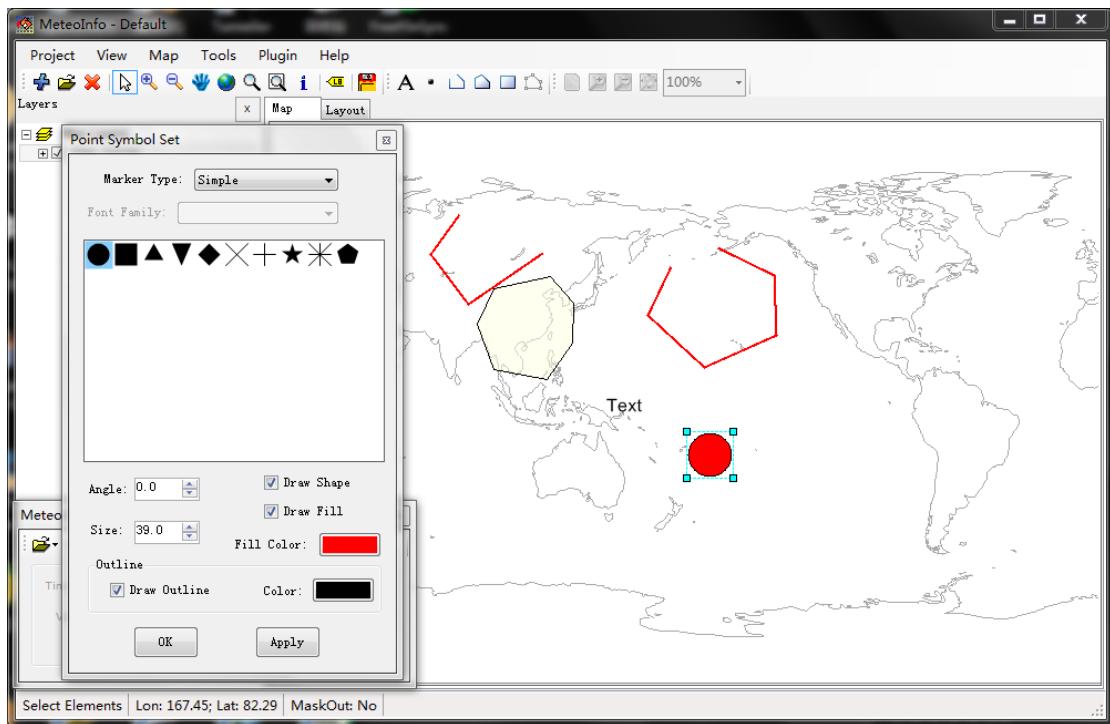
设置‘Visible’属性为‘True’，缺省参数设置将在地图显示区域右下脚显示中国南海地区为插图。插图位置和范围可以改变。



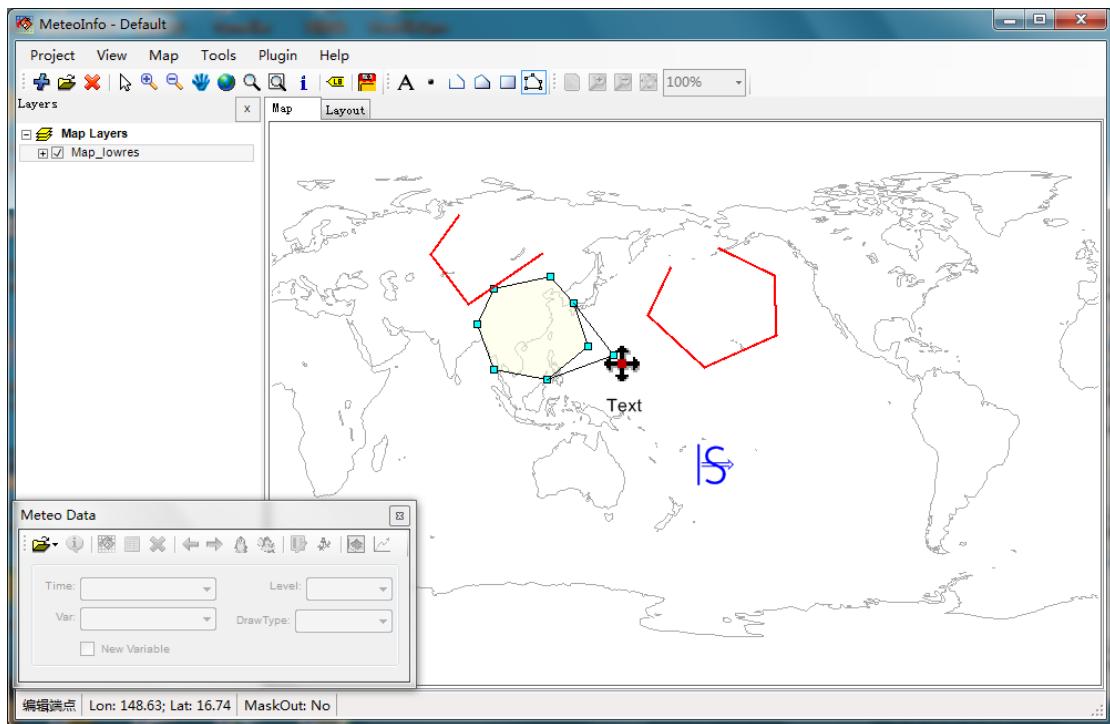
## 交互式绘图工具

通过交互式绘图工具可以用鼠标绘制标注（文本）、点、线、多边形、长方形等图形，图形可以绘制在地图和版面区域中。在地区区域绘制的图形和地图有同样的坐标系，随着地图移动、缩放、投影变换而同步变化。在版面区域绘制的图形位置是固定的，不随地图的变化而变化。

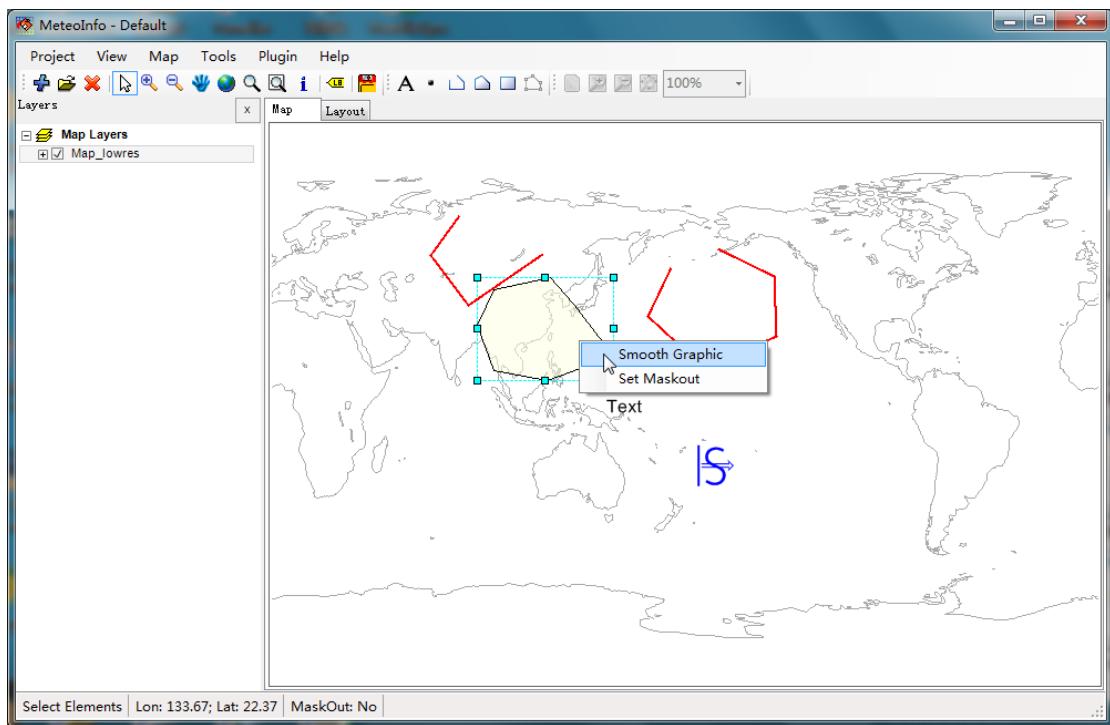
用鼠标点击选择某个绘图工具即可进行相应的图形绘制。标注和点可以通过单击鼠标左键将其添加在图形区域，对于线和多边形可以通过单击鼠标左键逐个添加端点，最后一个点通过双击鼠标左键结束该图形的绘制。图形被绘制后，选中‘选择’工具 ，可以点击鼠标左键选中某个被绘制的图形，对其进行移动和缩放，或者在键盘上点击删除键（Delete）删除该图形，在图形范围内双击鼠标左键可以打开该图形的属性对话框对图形属性进行编辑。

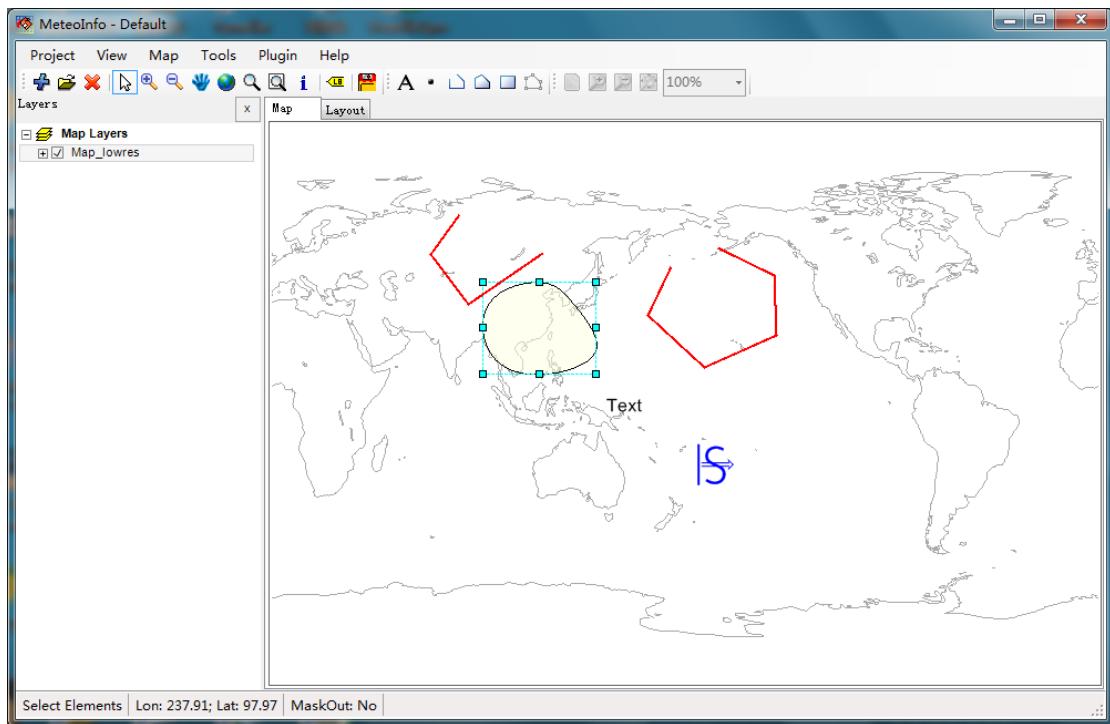


线或者多边形图形被选中后，‘编辑端点’工具  处于可用状态，选择该工具可以对图形的每个端点进行移动。



线和多边形图形被选中后，点击鼠标右键出现快捷菜单，选择‘Smooth Graphic’可以对图形进行平滑。





## 地图投影

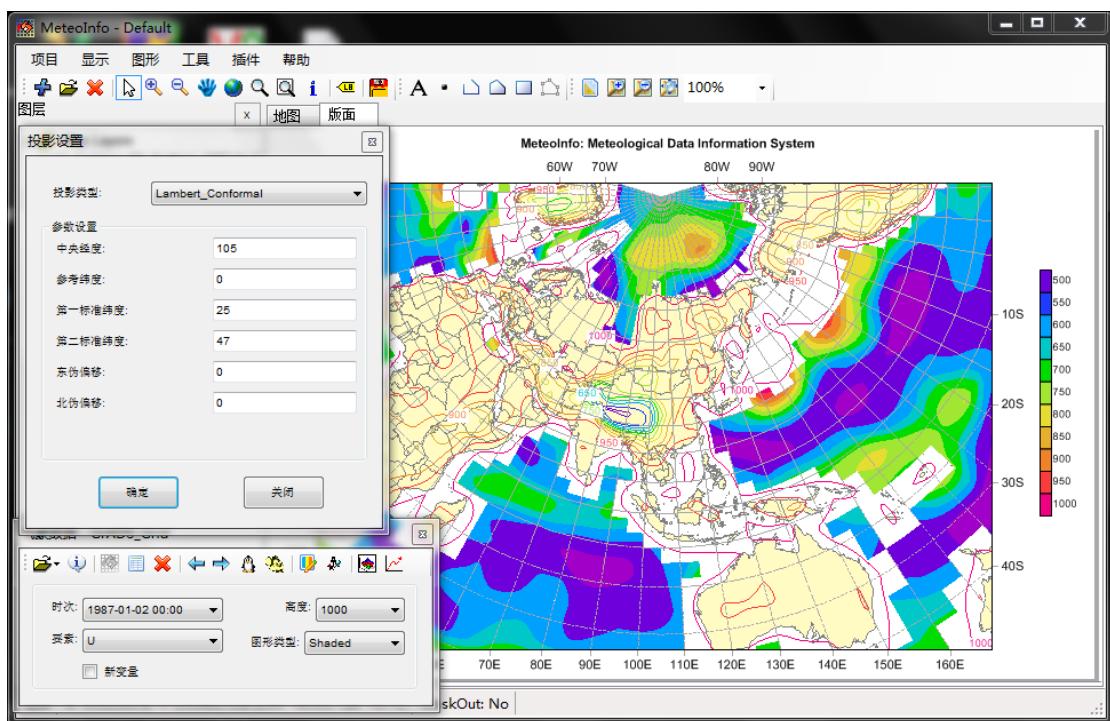
支持的地图投影类型：

- ◆ Lambert\_Conformal (兰伯特正形圆锥投影)
- ◆ Albers\_Conic\_Equal\_Area (亚尔勃斯等积圆锥投影)
- ◆ North\_Polar\_Stereographic (北极方位立体投影)
- ◆ South\_Polar\_Stereographic (南极方位立体投影)
- ◆ Mercator (墨卡托投影)
- ◆ Robinson (罗宾森投影)
- ◆ Mollweide (摩尔威德投影)
- ◆ Orthographic (正射投影)
- ◆ Geostationary (地球静止轨道投影)
- ◆ Oblique\_Stereographic (斜立体投影)
- ◆ Transverse\_Mercator (横轴墨卡托投影)

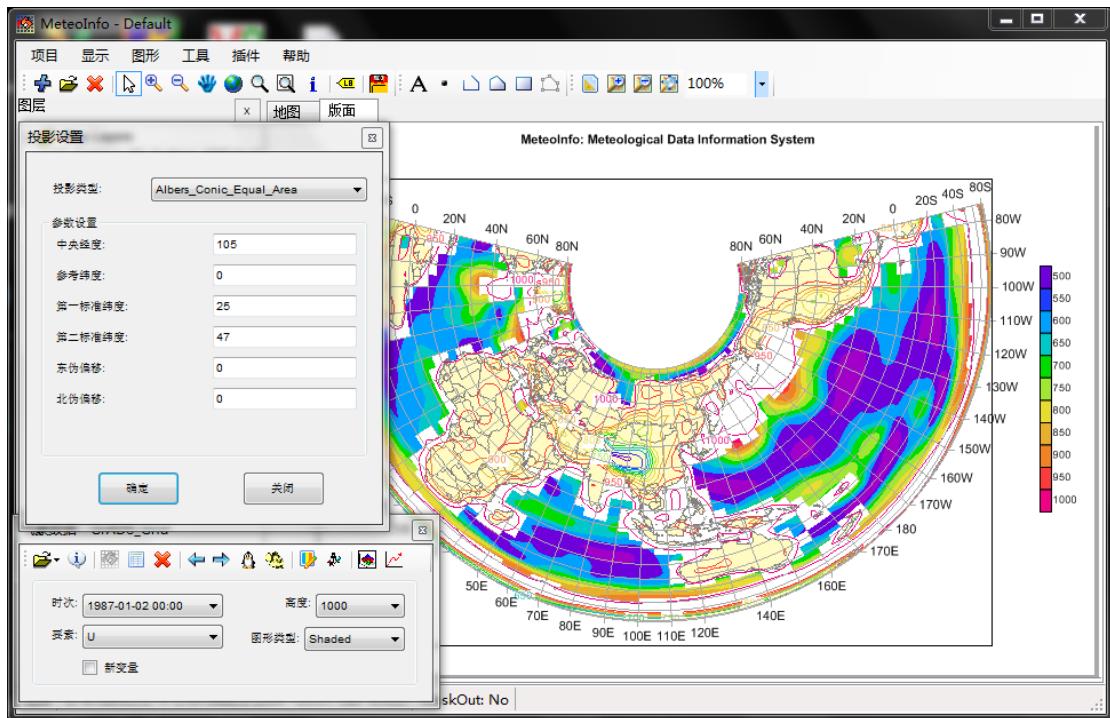
点击‘地图’菜单下的‘投影’子菜单打开‘Projection Set’对话框。通过选择‘Projection’和设置相应的参数来设置地图投影。



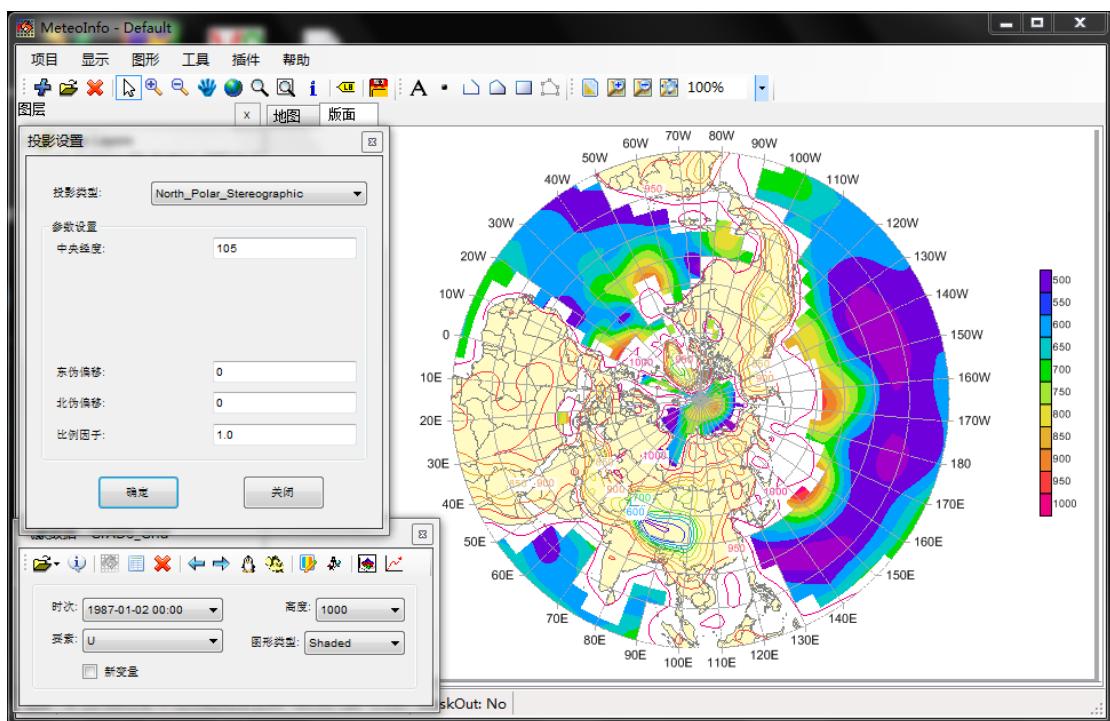
Lambert Conformal 投影:



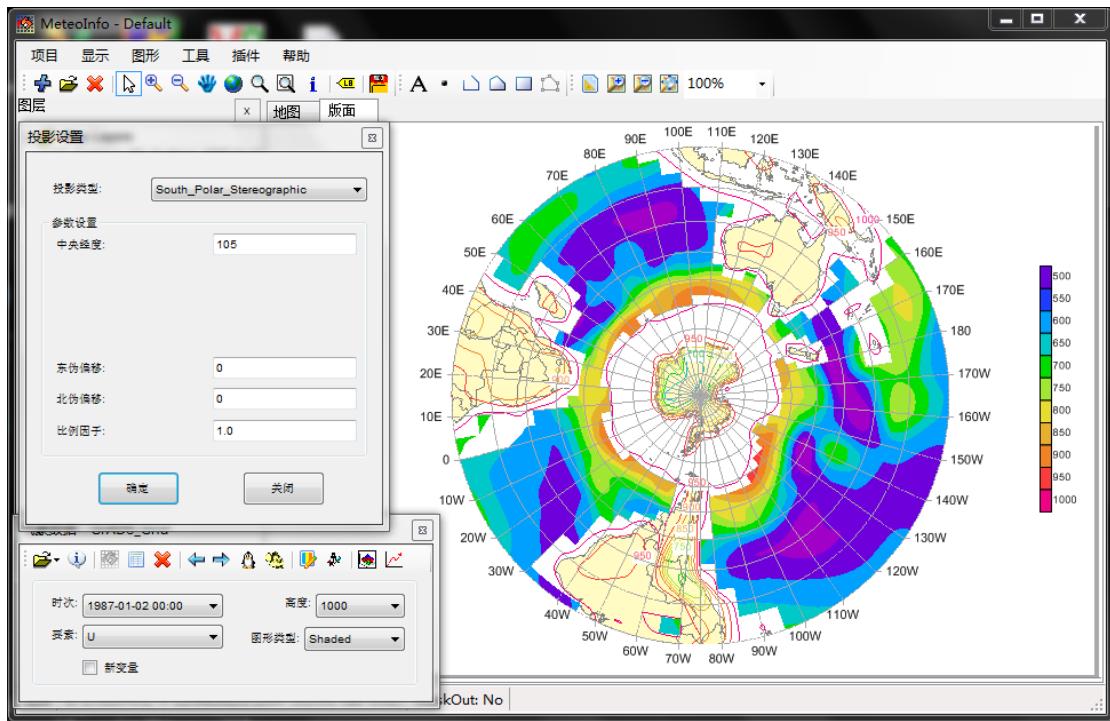
Albers Conic Equal Area 投影:



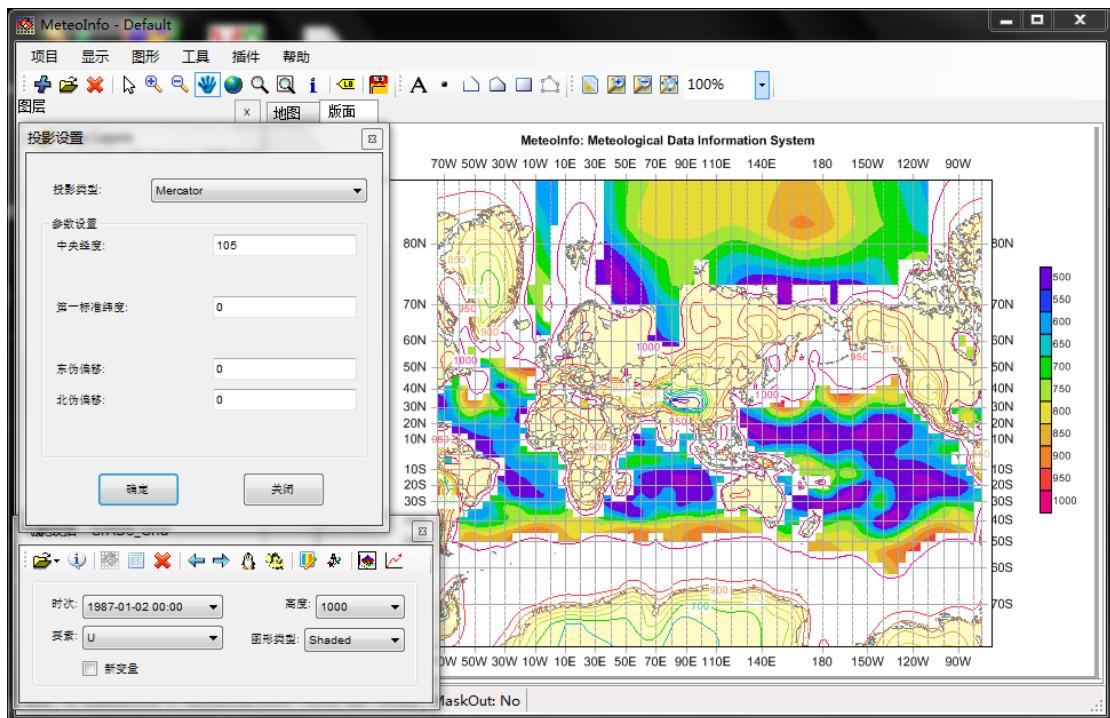
North\_Polar\_Stereographic 投影:



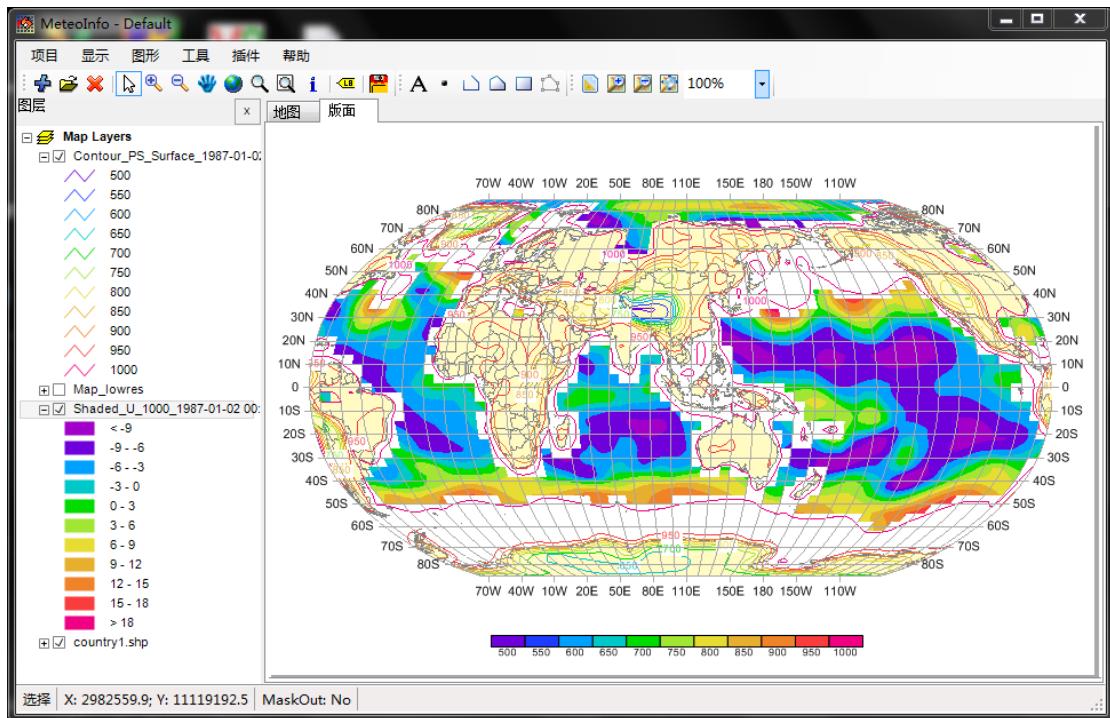
South\_Polar\_Stereographic 投影:



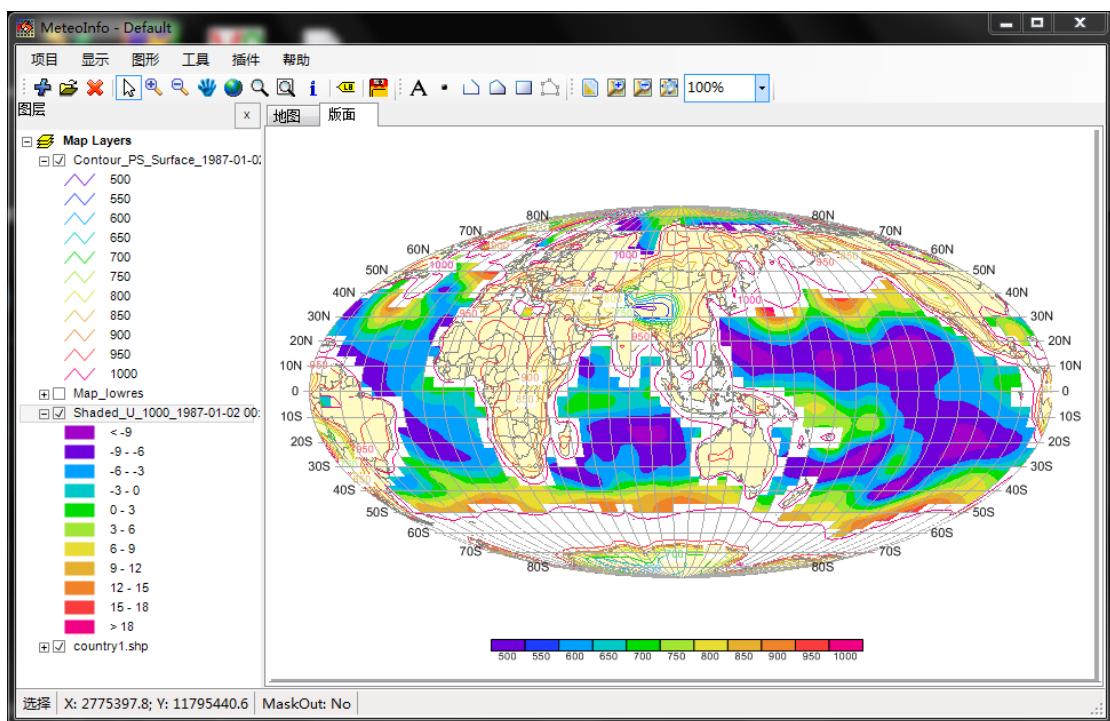
Mercator 投影:



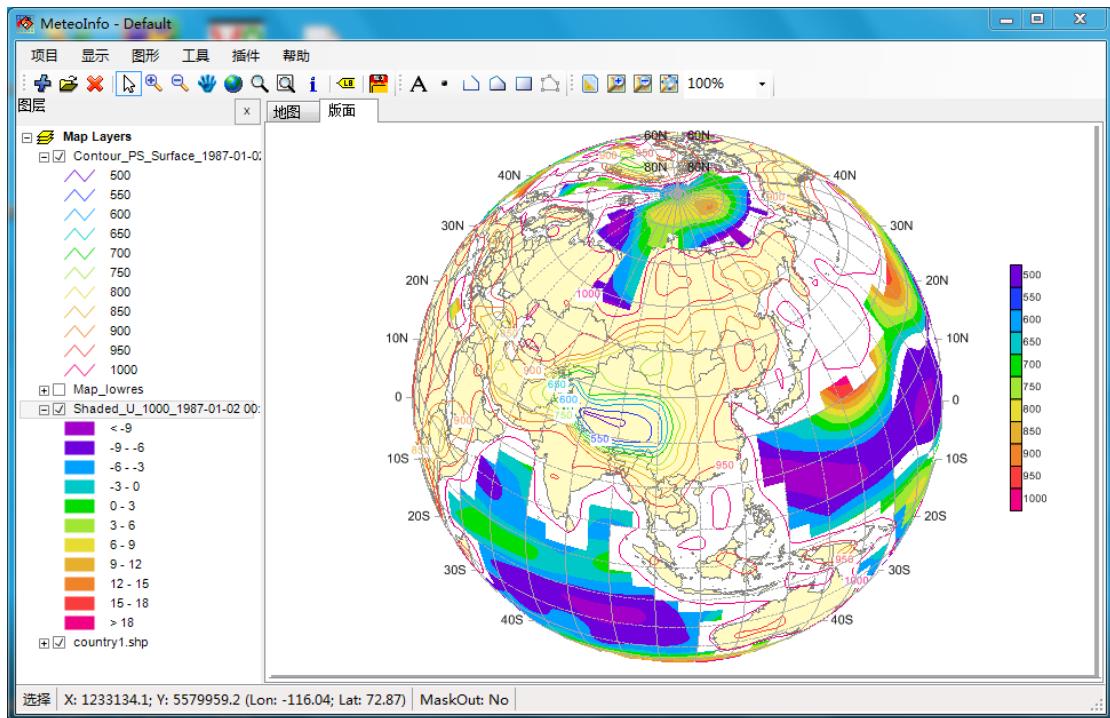
Robinson 投影:



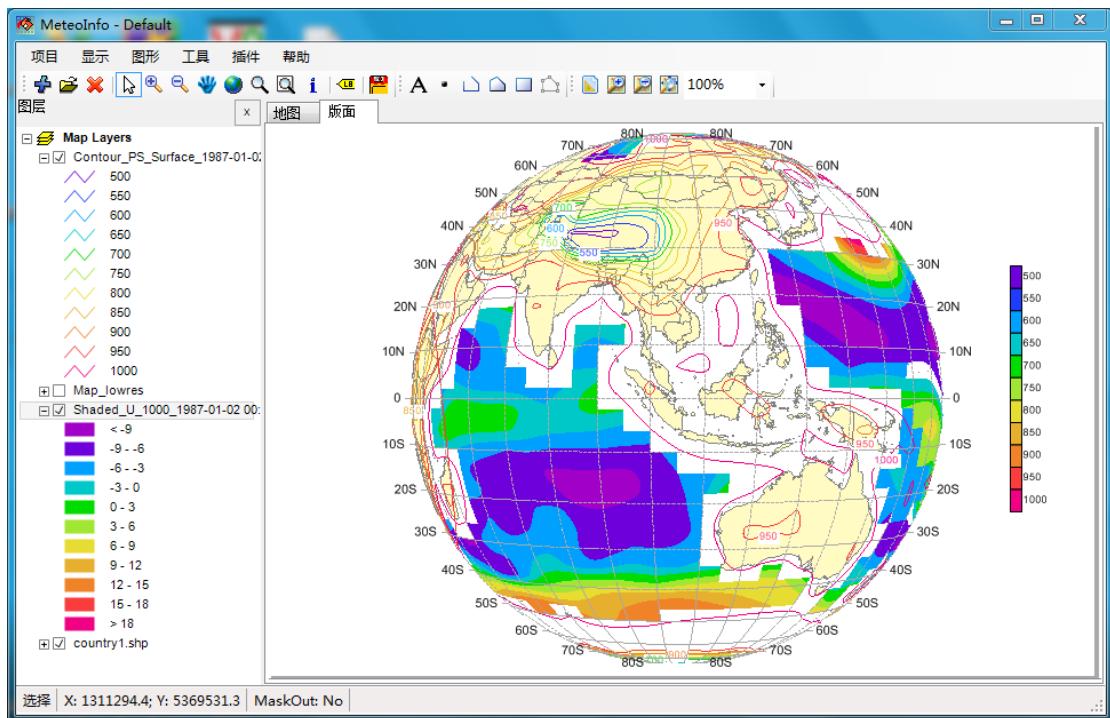
Mollweide 投影:



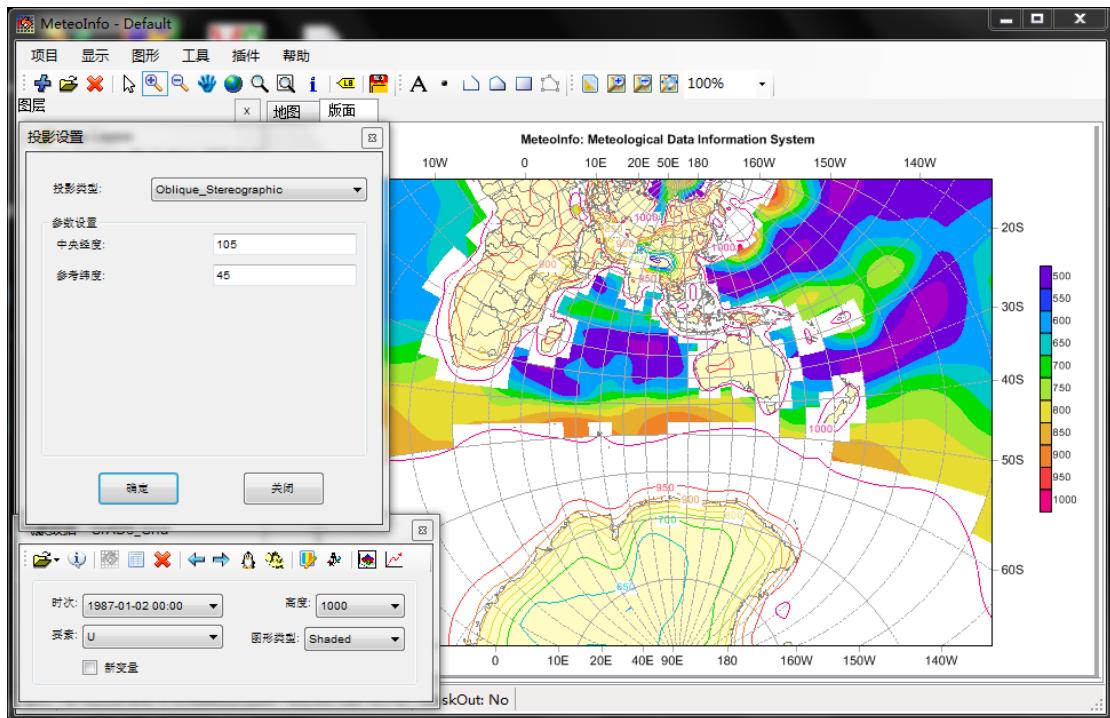
Orthographic 投影:



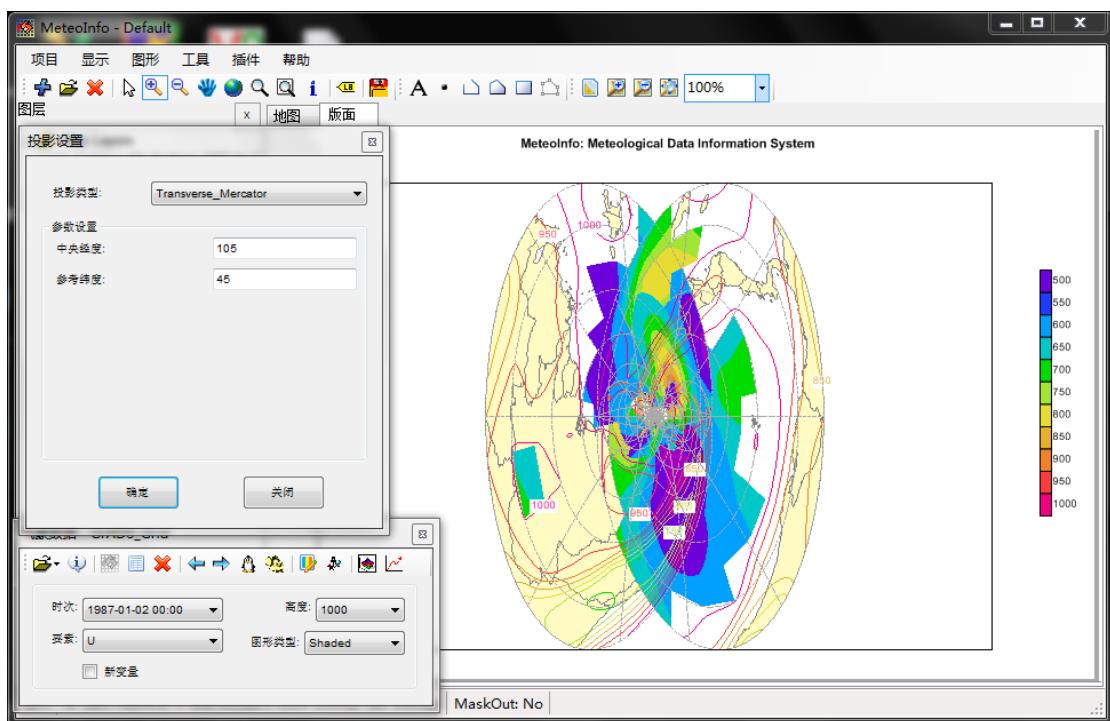
Geostationary 投影:



Oblique\_Stereographic 投影:



Transverse\_Mercator 投影:

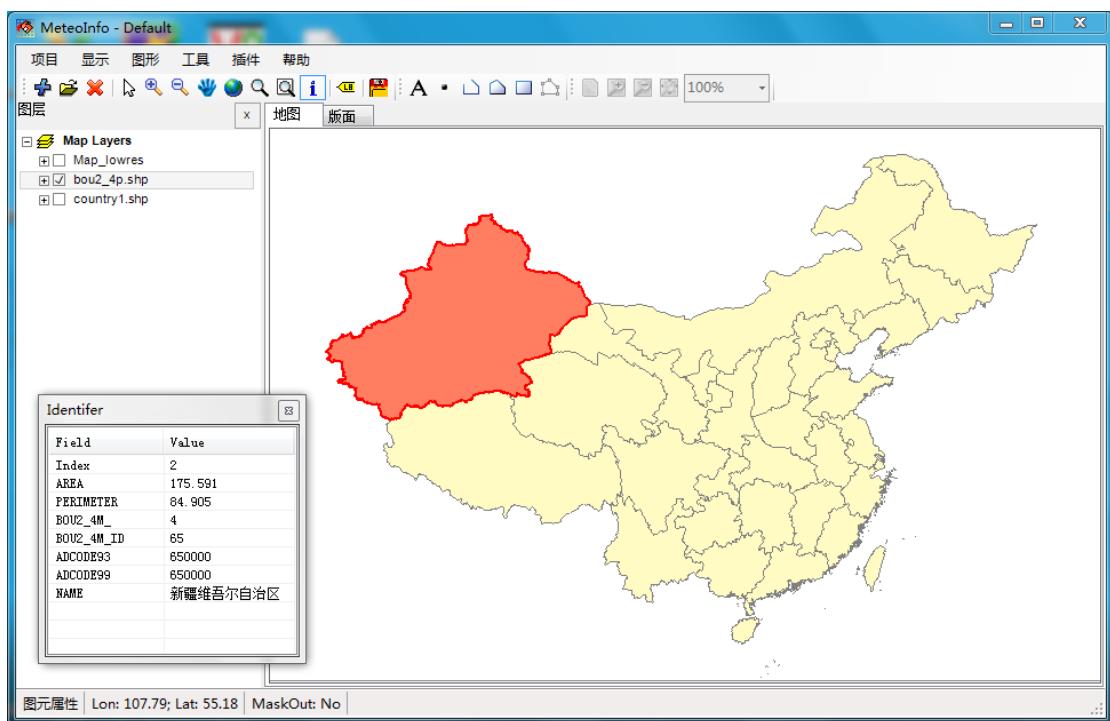


# 工具

## 输出地图数据

利用这个功能可以将某地图图层中选中的图元按照设置的格式输出，从而获得自己需要的地图数据。下面的示例获得 wmf 格式的新疆地图数据。

打开 ‘bou2\_4p.shp’ 地图数据，选中 ‘图元属性’ 工具，用鼠标左键点击地图上的新疆区域，显示其图元属性，记住其 ‘Index’ 为 2。



点击‘工具’菜单下的‘输出地图数据’子菜单，打开‘输出地图数据’对话框。

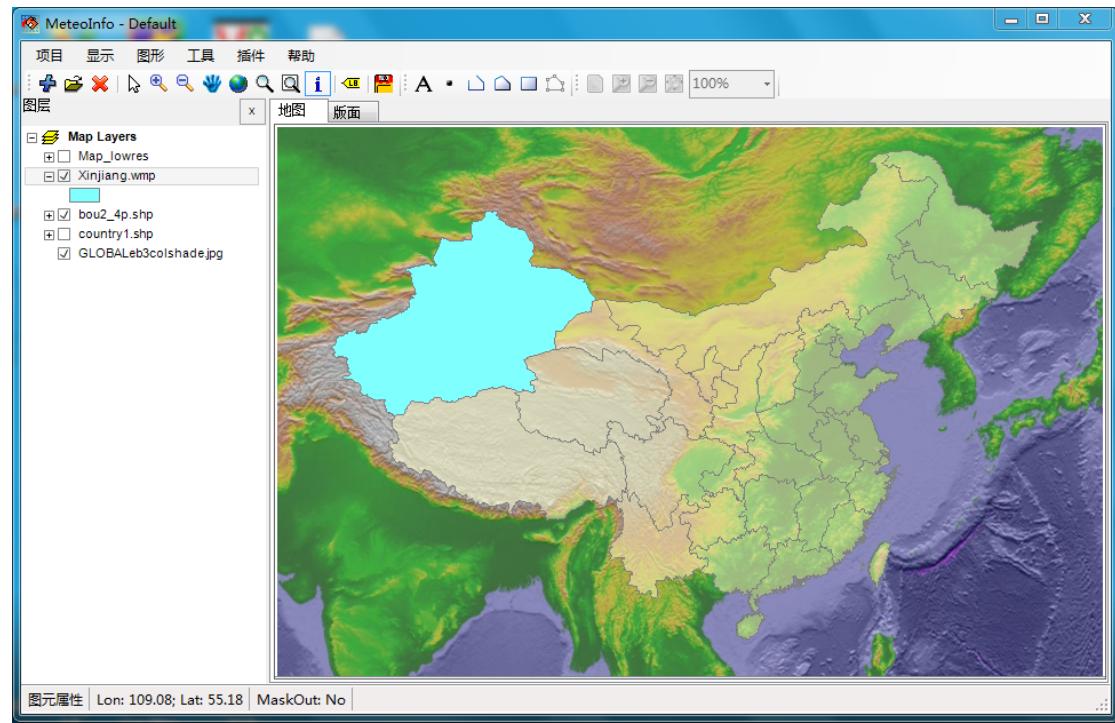


选择地图图层为‘bou2\_4p.shp’，选择序号为 2 的图元（新疆）。设置数据输出格式为‘ASCII wmp File’，然后点击‘输出’按钮即可将台湾省的地图数据（边界线经纬度信息）输出为 wmp 格式文件。wmp 格式文件是 ASCII 文件，可以用写字板等文本编辑器打开。

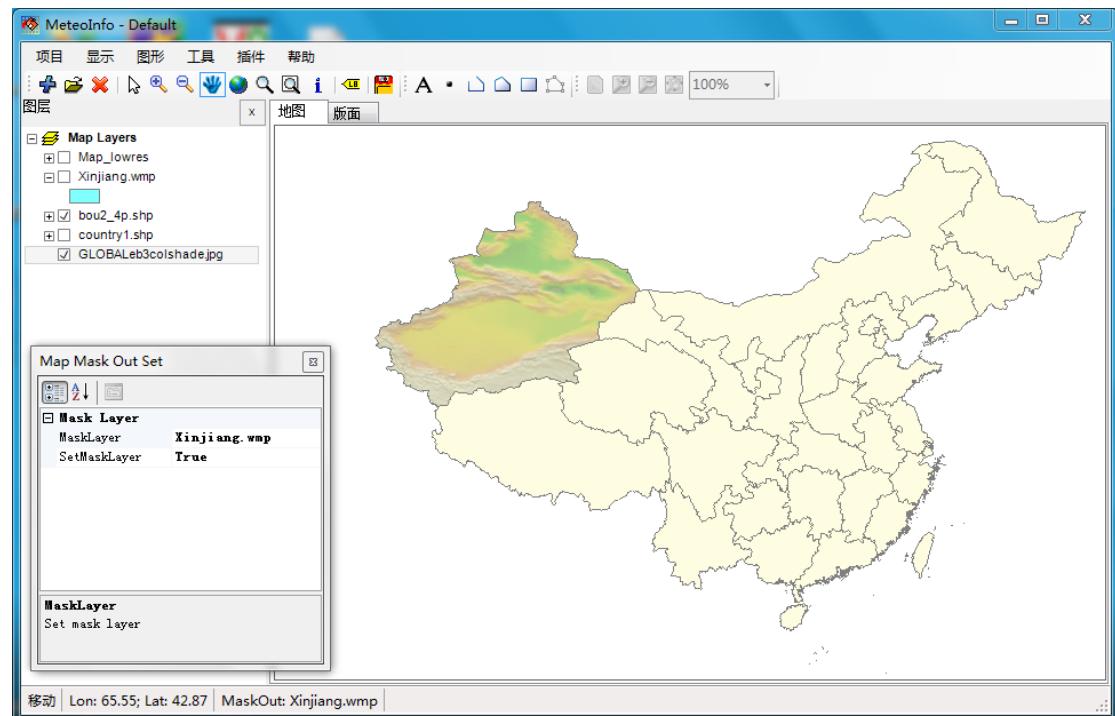
```
Polygon
1
9051
96.3832855224609, 42.7269554138184
96.3599090576172, 42.7096900939941
96.0966415405273, 42.599666595459
96.0597457885742, 42.3987350463867
96.0255279541016, 42.1109313964844
96.0546340942383, 42.0099906921387
96.0221405029297, 41.936824798584
95.9984741210938, 41.9179420471191
95.9914016723633, 41.8989448547363
95.8591766357422, 41.8483543395996
95.6668701171875, 41.8285522460938
95.5684509277344, 41.8516387939453
```

在 wmp 格式文件中，第一行是图元类型（点 Point、线 Polyline、面 Polygon），第二行是图元个数，后面是数据体，每个图元数据体第一行是该图元经纬度数据对的个数（点 Point

图元没有这个信息), 然后是该图元组成点的经纬度信息。经纬度之间用逗号分割, 经度在前。wmp 格式数据可以作为地图数据文件在 MeteoInfo 中使用。



因为 ‘Xinjiang.wmp’ 是一个面图层, 它可以被当作屏蔽图层来使用。

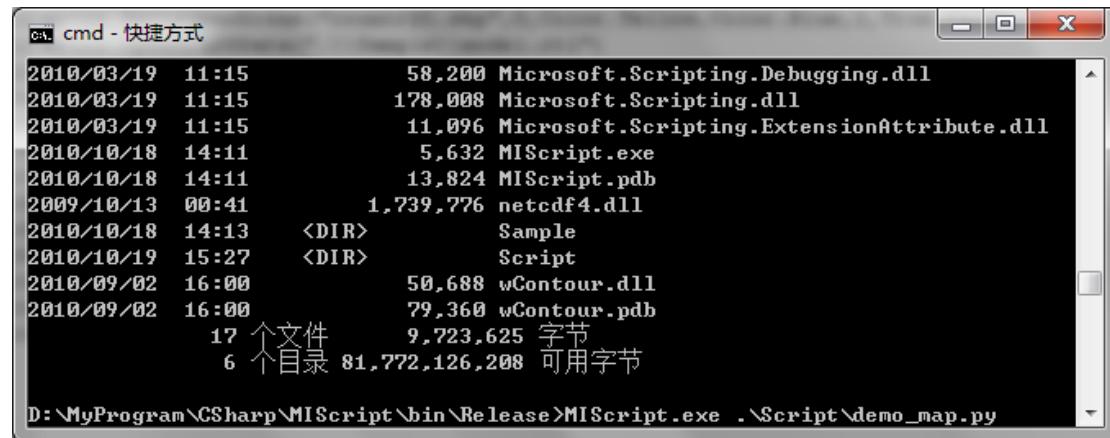


# MeteoInfo脚本程序

可以通过编写脚本程序来自动运行MeteoInfo并输出数据、图形。脚本语言是IronPython (<http://ironpython.codeplex.com/>)。

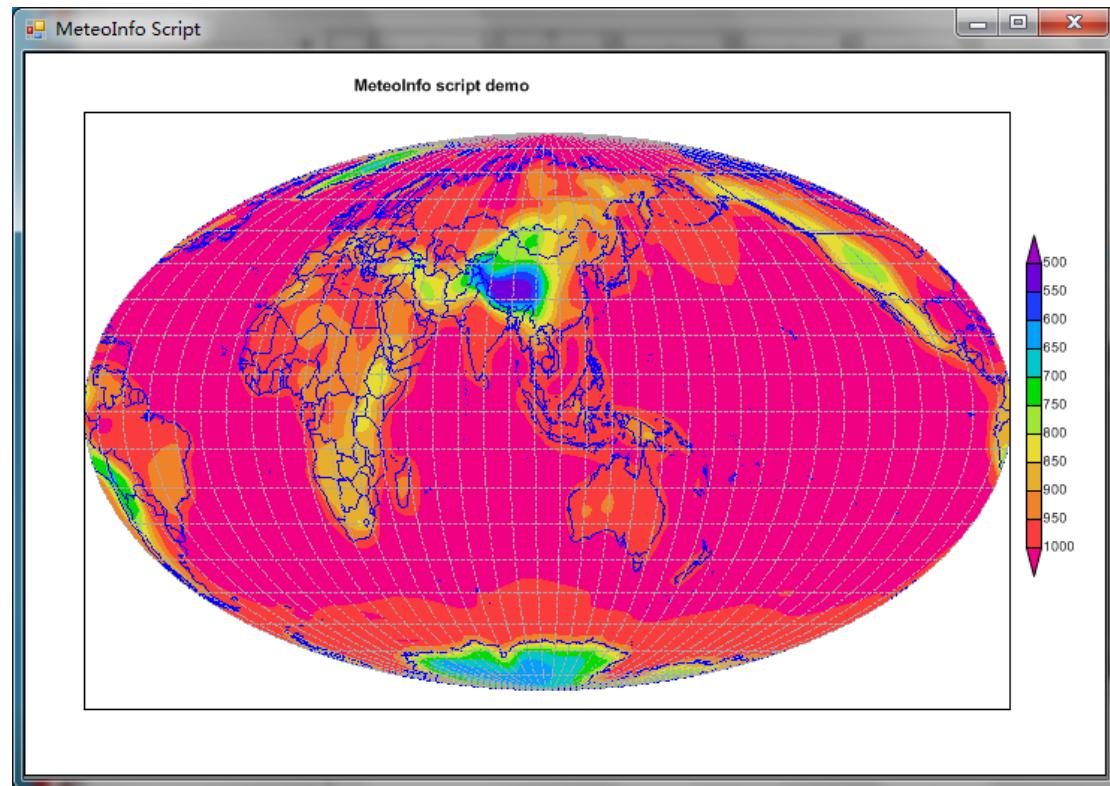
在软件安装路径的‘Script’目录下有一些示例脚本程序。

MeteoInfo 软件包里已经包含了 IronPython 运行所需的各种库文件，以及 MIScript.exe，可以在命令行界面中运行。安装 MeteoInfo 软件后，可以将 MeteoInfo 所在的路径设为系统 path，这样运行 MIScript.exe 就更为方便了。当然也可以在电脑中安装 IronPython，用 ipy.exe 来运行 MeteoInfo 的脚本程序。



```
2010/03/19 11:15      58,200 Microsoft.Scripting.Debugging.dll
2010/03/19 11:15      178,008 Microsoft.Scripting.dll
2010/03/19 11:15      11,096 Microsoft.Scripting.ExtensionAttribute.dll
2010/10/18 14:11      5,632 MIScript.exe
2010/10/18 14:11      13,824 MIScript.pdb
2009/10/13 00:41      1,739,776 netcdf4.dll
2010/10/18 14:13 <DIR>      Sample
2010/10/19 15:27 <DIR>      Script
2010/09/02 16:00      50,688 wContour.dll
2010/09/02 16:00      79,360 wContour.pdb
17 个文件          9,723,625 字节
6 个目录          81,772,126,208 可用字节

D:\MyProgram\CSsharp\MIScript\bin\Release>MIScript.exe .\Script\demo_map.py
```



上图中结果所用的脚本程序如下：

```

import clr
clr.AddReferenceByPartialName("System.Windows.Forms")
clr.AddReferenceByPartialName("System.Drawing")
from System.Windows.Forms import *
from System.Drawing import *
clr.AddReference("MeteoInfoC.dll")
from MeteoInfoC import *

myApp = MIApp()
myApp.OpenLayer(".\\Map\\country1.shp")
myApp.SetLegendBreak("country1.shp",0,Color.Yellow,Color.Blue,1,True,False,True)
myApp.OpenGrADSData("\\Sample\\model.ctl")
myApp.SetDrawType("shaded")
t = 2
myApp.TimeIndex = t
myApp.Display("PS")
myApp.MoveLayerToTop("country1.shp")
myAppSetTitle("MeteoInfo script demo")
myApp.ProjectLayers("+proj=moll+lon_0=105")
myApp.MapLayout.Refresh()
#myApp.SaveFigure("\\try.png")
Application.Run(myApp)

```

## 加载CLR和.NET类库

IronPython 的一个重要特点是能够直接调用.Net 平台强大的类库。.Net 平台的核心是 CLR (Common Language Runtime, 公共语言运行时), IronPython 调用.Net 类库首先要 import clr, 然后加载各种命名空间, 再从命名空间中引入类。

编写 MeteoInfo 脚本程序需要了解 Python 语言的基本语法以及.Net 类库的基础知识, ‘IronPython in Action’ 是一本好的参考书。

```

import clr
clr.AddReferenceByPartialName("System.Windows.Forms")
clr.AddReferenceByPartialName("System.Drawing")
from System.Windows.Forms import *
from System.Drawing import *

```

## 加载MeteoInfo类库

MeteoInfo 类库是.Net 类库, 也可以通过 clr 来加载。MeteoInfo 类库中专门设计了一个 MIApp 类来简化脚本程序的编写, 该类包含了一些常用的功能。

```
clr.AddReference("MeteoInfoC.dll")
```

```
from MeteoInfoC import *
```

创建一个 ‘MIApp’ 类的对象

```
myApp = MIApp()
```

然后就可以使用 MIApp 中的功能函数了。

## 打开数据

打开数据文件的函数根据数据类型有所不同，具体如下：

- 添加图层数据文件: `OpenLayer(string aFile)`
- 打开 GrADS 格式数据文件: `OpenGrADSDData(string aFile)`
- 打开 NetCDF 格式数据文件: `OpenNCData(string aFile)`
- 打开 GRIB 格式数据文件: `OpenGRIBData(string aFile)`
- 打开 MICAPS 格式数据文件: `OpenMICAPSData(string aFile)`
- 打开 ARL 格式数据文件: `OpenARLData(string aFile)`
- 打开带经纬度的站点数据文件 (Lon/Lat station): `OpenLonLatData(string aFile)`
- 打开 HYSPLIT 浓度数据文件: `OpenHYSPLITConc(string aFile)`

例如：

```
myApp.OpenGrADSDData(".\\Sample\\model.ctl")
```

## 设置维

时间和高度维设置：

- 时间: `myApp.TimeIndex = i`
- 高度: `myApp.LevelIndex = i`

缺省设置均为 0。格点和站点数据依据维的设置进行读取。

## 设置图形类型

图形类型可以用 `SetDrawType(string drawTypeString)` 函数来设置，‘drawTypeString’ 可以设置为：

- “contour”
- “shaded”
- “grid\_fill”
- “grid\_point”
- “vector”
- “barb”
- “streamline”
- “station\_point”
- “weather\_symbol”

- “station\_model”
- “station\_info”
- “traj\_line”
- “traj\_point”
- “traj\_startpoint”
- “image”
- “raster”

例如:

```
myApp. SetDrawType ("shaded")
```

## 读取气象数据

MeteoInfo 中有两个很重要的类: GridData 和 StationData, 用来分析和显示气象数据。

- 读取格点数据: `GetGridData(string varName)`, 返回一个 GridData 对象
  - 读取站点数据: `GetStationData(string varName)`, 返回一个 StationData 对象
- 函数中的参数是气象数据中的变量名。

例如:

```
UGridData = myApp.GetGridData("U")
rainSTData = myApp.GetStationData("Precipitation6h")
```

## 生成气象数据图层

一些重载的 Dislay 函数用来从格点或站点数据生成气象数据图层。

- `Display(string varName)`
- `Display(GridData aGridData)`
- `Display(string U, string V)`
- `Display(string U, string V, string varName)`
- `Display(GridData UGridData, GridData VGridData)`
- `Display(GridData UGridData, GridData VGridData, GridData XGridData)`
- `DisplayWind(string windDir, string windSpeed)`
- `DisplayTraj()`

例如:

```
myApp.Display("PS")
```

```
UGridData = myApp.GetGridData("U")
```

```
VGridData = myApp.GetGridData("V")
```

```
myApp.Display(UGridData, VGridData)
```

## 气象数据计算、分析

格点和站点数据可以进行各种计算和分析。可用的运行符包括：‘+’、‘-’、‘\*’、‘/’。  
在 ‘MeteoInfoC.Data’ 命名空间中的 ‘DataMath’ 类包含了一些常用的数学函数：

- [Abs](#)
- [Acos](#)
- [Asin](#)
- [Atan](#)
- [Cos](#)
- [Sin](#)
- [Tan](#)
- [Exp](#)
- [Pow](#)
- [Sqrt](#)
- [Log](#)
- [Log10](#)
- [Magnitude](#)
- [Cdiff](#)
- [Hcurl](#)
- [Hdivg](#)

例如：

```
...
#从 'MeteoInfoC.Data' 中引入所有类
from MeteoInfoC.Data import *

...
#读取 U/V 格点数据
UGridData = myApp.GetGridData("U")
VGridData = myApp.GetGridData("V")
#计算风速（方法一）
Speed = DataMath.Magnitude(U, V)
#或者（方法二）
Speed = DataMath.Sqrt(U * U + V * V)
#生成风速面图层
myApp.SetDrawType("shaded")
myApp.Display(Speed)
```

## 其它函数

- [ProjectLayers\(string projStr\)](#)

实现地图投影功能，参数是Proj4 的投影字符串。具体请参考Proj4 的帮助文档，也可以参照此网站：[http://remotesensing.org/geotiff/proj\\_list/](http://remotesensing.org/geotiff/proj_list/) .

例如：

- `myApp.ProjectLayers("+proj=moll+lon_0=105")`
- `Zoom(double minX, double maxX, double minY, double maxY)`  
缩放地图功能。  
例如：  
`myApp.Zoom(75, 135, 15, 55)`
- `ZoomEx(double minX, double maxX, double minY, double maxY)`  
按给定范围精确缩放地图，地图区域的高度和宽度比例可能发生变化以便使范围精确。  
例如：  
`myApp.ZoomEx(0, 360, -90, 90)`
- `SetMaskout(string layerName)`  
设置屏蔽图层，参数是图层名。  
例如：  
`myApp.SetMaskout("china.shp")`
- `SetIllustration(bool visible)`  
设置是否显示插图。  
例如：  
`myApp.SetIllustration(True)`
- `SaveFigure(string aFile)`  
将图形保存至文件。  
例如：  
`myApp.SaveFigure(".\\try.png")`
- `SetInterpolation(double minX, double maxX, double minY, double maxY, int xNum, int yNum, string aInterMethod, float radius, int minNum)`  
设置插值（站点数据插值为格点数据）方法和参数。  
例如：  
`myApp.SetInterpolation(60,140,-20,60,160,160,"IDW_Radius",1,1)`
- `SetLegendScheme(string legendFile)`  
根据图例文件设置图例。  
例如：  
`myApp.SetLegendScheme(".\\Legend\\rain.lgs")`
- `SetLegendBreak(string layerName, int brkIdx, Color aColor, Color outlineColor, float outlineSize, bool drawOutline, bool drawFill, bool drawShape)`  
设置某个图例要素。  
例如：  
`myApp.SetLegendBreak("country1.shp",0,Color.Yellow,Color.Gray,1,True,False,True)`
- `Title(string aTitle)`  
设置图形名称。  
例如：  
`myAppSetTitle("MeteoInfo script demo - Grid data calculation")`
- `MoveLayerToTop(string layerName)`  
移动某个图层至最上面。  
例如：  
`myApp.MoveLayerToTop("country1.shp")`