

文章编号: 1000-2022(2003) 01-0130-06

一种绘制地面天气图及要素场等值线方法

戴泽军¹, 苗春生¹, 禹 伟², 陈建萍¹

(1. 南京气象学院 大气科学系, 江苏 南京 210044; 2. 湖南省气象台, 湖南 长沙 410007)

摘 要: 采用位图作为天气符号绘制天气图, 利用规则三角形网格法绘制等值线, 通过 Visual C++ 6.0 软件编程, 并在计算机上得到实现。结果表明, 此方法绘制的天气图显示清晰, 等值线分析准确、光滑, 完全可以满足气象台站日常业务要求。

关键词: 地面天气图; 位图; 天气符号; 规则三角形网; 等值线

中图分类号: TP317 **文献标识码:** A

随着计算机硬件和软件的飞速发展, 过去填图机填图后预报员进行分析的有纸化工作流程逐步被淘汰。目前大部分省地气象台站使用气象信息综合处理系统(MICAPS)进行天气图表的自动填绘及其他气象信息的加工处理, 这些强大、实用的功能已使 MICAPS 成为我国气象业务部门主要的天气分析与预报的业务工作平台。但实际工作表明, MICAPS 中的地面天气图自动绘制不尽如人意(例如有些天气符号分辨率不高、站点信息颜色搭配有待优化); 缩放与拖动等功能操作不方便。为此, 我们利用 Visual C++ 6.0 软件开发环境, 开发了从原始资料的采集到用位图作为天气符号的地面天气图显示和要素等值线分析的软件系统^[1-2], 适合省级和地市级气象台站使用。

1 地面天气图的绘制

地面天气图的绘制主要包括资料采集、坐标系的转换、地理底图的绘制和天气图诸要素的绘制。

1.1 资料采集

读取每个时次的地面资料原始报文, 通过解码^[3]、质量检查, 将所需要的站点资料以一个站点的信息为一记录保存在文件中。每一个记录按照站号、站名、纬度、经度、云底高、能见度、总云量、风向、风速、气温、露点、本站气压、海平面气压等要素顺序保存。

1.2 坐标投影及底图的绘制

1.2.1 坐标投影的转换

首先, 将地球坐标(L, B)转换成直角坐标^[4](x, y), 然后再把直角坐标(x, y)转换成屏幕坐标(x_{sc}, y_{sc})。即: 在屏幕上取一个宽 w 、高 h 矩形窗口, 矩形左下点(X_{LB}, Y_{LB})对应球面上坐标

收稿日期: 2001-01-15; 改回日期: 2002-05-08

作者简介: 戴泽军(1971-), 男, 湖南澧县人, 工程师, 硕士, 主要从事气象业务自动化研究, 现在湖南省气象台工作。

点 (N_{LB}, E_{LB}) 。将球面上左下点 (N_{LB}, E_{LB}) 与右上点 (N_{RT}, E_{RT}) 构成的区域投影到视窗上。球面点 (N_{LB}, E_{LB}) 和 (N_{RT}, E_{RT}) 分别对应直角坐标系中的点 (x_0, y_0) 与 (x_1, y_1) 。则球面任意一点 (L, B) 地球坐标被转换成投影直角坐标 (x, y) , 此点对应在屏幕上的坐标 (x_{sc}, y_{sc}) 为

$$\begin{aligned} x_{sc} &= X_{LB} + w(x - x_0)/(x_1 - x_0), \\ y_{sc} &= Y_{LB} - h(y - y_0)/(y_1 - y_0)。 \end{aligned}$$

1. 2. 2 底图的绘制

从 MICAPS 系统中取出地理特征文件, 文件包含全国各省省界、河流、海岸线等经纬点信息。将其读取后通过坐标转换, 用连线的方法在屏幕上绘出。同时配以每隔 5° 一条的经纬网格线, 就有了一个视屏上完整的平面地理地图。

1. 3 天气图诸要素的绘制

1. 3. 1 位图符号的填制

在位图符号的填制之前, 先制作好位图天气符号。过去由于计算机硬件与软件的限制, 制作天气符号的常用方法是将天气符号用字符制作工具制作成字符, 放入字符库中, 在软件运行时调用字符库。用字符制作的天气符号的主要缺陷是显示不够清晰。用位图制作成的天气符号, 颜色丰富, 显示清晰。Visual C++ 6.0 软件开发环境自带位图编辑器, 使用方便。将 127 个天气现象符号和 10 个云量符号逐个用位图编辑器点绘成 16×16 点阵的小位图, 位图符号按便于检索的顺序绘制, 绘制的颜色根据习惯的颜色调整(如降水符号用绿色、雾的符号用黄色), 127 个天气现象符号存放在一个位图文件里, 10 个云量符号存放在另一个位图文件里。这样, 位图文件就成为天气符号库。

填制时取出天气现象或云量信息, 计算出对应的天气符号在位图库中的位置坐标 (x_{Bi}, y_{Bi}) , 将位图符号填制在屏幕坐标 (x_{sc}, y_{sc}) 处。

具体步骤为:

首先定义位图对象和设备句柄

```
CBitmap bitMap1, bitMap2; CDC dcMem1, dcMem2;
```

再加载位图

```
bitMap1. LoadBitmap(IDB_TIANQI); bitMap2. LoadBitmap(IDB_YUNLIANG);  
dcMem1. CreateCompatibleDC(pDC); dcMem1. SelectObject(bitMap1);  
dcMem2. CreateCompatibleDC(pDC); dcMem2. SelectObject(bitMap2);
```

最后显示天气符号

```
pDC-> StretchBlt(sc_x, sc_y, 16, 16, &dcMem1, Bit_x, Bit_y, 16, 16, SRCAND);  
pDC-> StretchBlt(sc_x, sc_y, 16, 16, &dcMem2, Bit_x, Bit_y, 16, 16, SRCAND)。
```

1. 3. 2 风的绘制

风的绘制公式为

$$\begin{aligned} x_1 &= x_{sc} + r_0 \sin w; & y_1 &= y_{sc} - r_0 \cos w; & (1) \\ x_2 &= x_{sc} + (r_0 - \text{countf}_4 \times r_2) \sin w; & y_2 &= y_{sc} - (r_0 - \text{countf}_4 \times r_2) \cos w; & (2) \\ x_3 &= x_2 + r_1 \sin(w + \pi/3)/2; & y_3 &= y_2 - r_1 \cos(w + \pi/3)/2; & (3) \\ x_4 &= x_2 + r_1 \sin(w + \pi/3); & y_4 &= y_2 - r_1 \cos(w + \pi/3)。 & (4) \end{aligned}$$

其中 (x_{sc}, y_{sc}) 为风向杆起始点坐标, w 为风向弧度, countf_4 为风速杆控制数, $\pi= 3. 141\ 592\ 6$, r_0 为风向杆长度, r_1 为 4 m/s 风速杆长度, r_2 为风速杆间隔(本文取 $r_0= 24, r_1= 14, r_2= 4$)。

(x_{sc}, y_{sc}) 和 (x_1, y_1) 两点的连线为风向杆, (x_2, y_2) 和 (x_3, y_3) 两点的连线为 2 m/s 的风速杆。 (x_2, y_2) 和 (x_4, y_4) 两点的连线为 4 m/s 的风速杆。当风速大于零时, 首先绘制风向杆。风速大小绘制如下: 当风速大于 20 m/s 时, 先绘制三角形表示 20 m/s 的风速后将风速减去 20 m/s ; 风速大于等于 4 m/s 时, 先绘制 4 m/s 的风速杆后将风速减去 4 m/s ; 风速为 2 m/s 时, 绘制 2 m/s 的风速杆。每绘一条风速杆时, 控制数 countf_4 加 1。任意两点 (x_1, y_1) 、 (x_2, y_2) 连线的图形函数为

$$\text{pDC} \rightarrow \text{MoveTo}(x_1, y_1); \quad \text{pDC} \rightarrow \text{LineTo}(x_2, y_2)。$$

1.3.3 数字字符串的填制

用图形设备对象填制其他的天气要素。先将天气要素信息转换成字符串 str , 用函数 $\text{pDC} \rightarrow \text{TextOut}(\text{sc-x}, \text{sc-y}, \text{str})$ 填字符串 str , 用函数 $\text{pDC} \rightarrow \text{SetTextColor}(\text{clr})$ 设定字符串颜色, 函数中 $(\text{sc-x}, \text{sc-y})$ 为屏幕坐标, clr 为字符串颜色。

2 要素场等值线的绘制

2.1 规则三角形网格法

2.1.1 规则三角网的形成

先将要绘制的区域映像为一个 $m \times n$ 个格点的矩形区域。组成 $(m-1) \times (n-1)$ 个大小相等的小矩形。再将每个小矩形按对角线划分成两个三角形。其中任意一个三角形的三个顶点与三个对边建立了三角形的内部关系, 每个边或在矩形区域边缘上, 或是相邻三角形的边。这样通过三角形的边组成了矩形区域内规则三角形网络的三角形关系网^[5]。

具体分为以下 3 步(设网格规格为: X 横向 m 个点, Y 纵向 n 个点):

第 1 步: 形成三角形网并分析三角形的顶点、边、编号之间关系。

按图 1 所示(此处 $m=6, n=5$), 将 $m \times n$ 个格点的矩形区域形成三角形网, 将顶点(格点)和三角形编号。设某一网格点坐标 (I, J) , $I=0, 1, \dots, m-1$; $J=0, 1, \dots, n-1$ 。其编号为

$$sn = (I + m \times J) + 1, \quad (5)$$

则 $I = (sn - 1) \% m$, $J = (sn - 1) / m$ ($\%$ 号为求余, $/$ 号为整除), 如图 2 所示。

小矩形的左下角为奇数号三角形, 三角形编号为

$$s = (I + (m - 1) \times J) \times 2 + 1。 \quad (6)$$

三个顶点的编号、次序、该点对边次序、对边相邻三角形编号分别为

左下角点 $(I, J): sn', 1, 4, s + 1$;

右下角点 $(I + 1, J): sn' + 1, 2, 5, s - 1$;

左上角点 $(I, J + 1): sn' + m, 3, 6, ss。$

小矩形的右上角为偶数号三角形, 三角形编号 s 为

$$s = (I + (m - 1) \times J) \times 2 + 2。 \quad (7)$$

三个顶点的编号、次序、该点对边次序、对边相邻三角形编号分别为

右下角点 $(I + 1, J): sn', 1, 4, ss$;

右上角点 $(I + 1, J + 1): sn' + m, 2, 5, s - 1$;

左上角点 $(I, J + 1): sn' + m - 1, 3, 6, s + 1。$

其中 ss 为该点对边和相邻三角形的编号, $ss = s + (-1)^{(s \% 2)} \times (2 \times (m - 1) - 1)$, 三角形的边在区域上时, 将其编号为 $ss = -2$ 。由(5)和(6)式或者(5)和(7)式可以推出: $sn' = s/2 +$

$s/(2 \times (m-1)) + 1。$

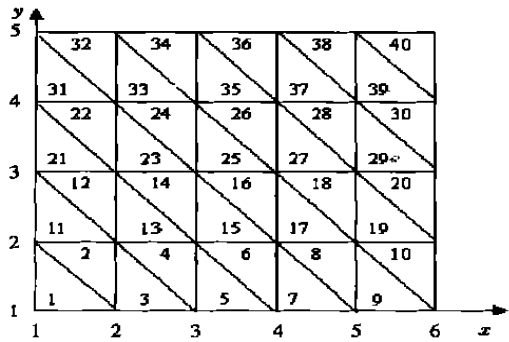


图 1 三角形编号示意
Fig. 1 The sketch map of triangles' serial numbers

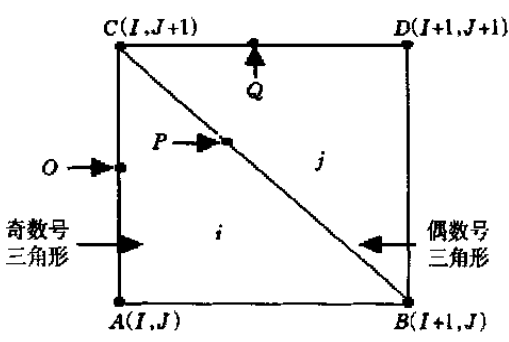


图 2 三角形顶点与插值点示意
Fig. 2 The sketch map of triangles' vertexes and interpolation points

第 2 步: 根据三角形的编号求出信息量。

由 $m \times n$ 个网格点组成编号从 1 到 $2 \times (m-1) \times (n-1)$ 个三角形。由第 1 步可知, 根据三角形的编号, 可以得出此三角形的六个信息量即三角形的三个顶点和三个对边的相邻三角形编号。另增加一个信息量赋初值为零, 在程序中用来判断此三角形是否被搜索过 (搜索过后赋值为 1, 下次检索到此三角形时跳过)。存放三角形信息量的次序为: 1、2、3 号位放三个顶点的编号, 4、5、6 号位分别放前面 1、2、3 号位顶点对边的相邻三角形编号, 7 号位为访问信息位置并赋初值为零。为了程序优化, 将顶点对应边在区域上的顶点和对应三角形编号信息移到 1、4 号位。三角形三个顶点的编号次序为 1、2、3, 并约定点 1、2 组成 3 号边, 点 1、3 组成 2 号边, 点 2、3 组成 1 号边。

第 3 步: 保存三角形网的信息。

一条边在矩形区域边缘上的三角形, 称之为边三角形, 其他的称为内部三角形。为了优化程序, 节省时间, 先搜索边三角形, 分析出经过边三角形的所有非闭合等值线和闭合等值线, 再搜索内部三角形, 分析出剩下的闭合等值线。为此, 将 $2 \times (m+n-1)$ 个边三角形的编号放在前面, 内部三角形编号放在后面。依照上述步骤可形成三角形网的所有信息: 将 $2 \times (m-1) \times (n-1)$ 个三角形的信息按三角形编号从小到大的顺序依次生成, 再将三角形的编号按照边三角形在前、内部三角形再后的次序排列, 最后保存在文件中, 形成三角形网文件保存。

2. 1. 2 三角网分析的原理

三角网分析等值线的基本原理是: 等值点既是当前三角形的出口点又是下个三角形的入口点, 当前三角形出口点所在边的对应顶点包含有入口三角形的三角形编号信息, 由三角形关系网数据进行等值点的内插和跟踪。假设 X 、 Y 和 H 代表三角形顶点的坐标值与顶点场值, Z 为当前所绘等值线的值。对于任意两个相邻的顶点 1 和顶点 2, 仅当 $(H_1 - Z) \times (H_2 - Z) < 0$ 时, 等值线穿过顶点 1 和顶点 2。

穿过点 0 的坐标值为

$$\begin{aligned} X_0 &= X_1 + (X_2 - X_1) \times (Z - H_1) / (H_2 - H_1); \\ Y_0 &= Y_1 + (Y_2 - Y_1) \times (Z - H_1) / (H_2 - H_1)。 \end{aligned}$$

如图 2 所示,运用以上公式可以算出等值线从 i 号三角形 C 、 A 两点的连线中的 O 点穿过,下一点穿过 B 、 C 两点的连线中的 P 点,为了保证等值线不过三角形的顶点,插值时将顶点处的值根据实际操作加一小量(如 0.01)。 B 、 C 两点对应 i 号三角形的 A 点,可以取出 A 点对应边相邻的三角形编号 j 。 i 号三角形的 B 、 C 两点的编号可以得到,进入 j 号三角形后,与 j 号三角形的三个顶点编号比较,下一个等值点穿过 BD 或 CD ,经插值分析,等值线从 C 、 D 两点的连线中的 Q 点穿过,如此类推……。

2.2 等值线绘制

如果初始场为矩形网格场,直接进行等值线绘制。如果初始场为离散点,先将离散点的值利用某种插值方法(本文采用最靠近插值点的九点作线性内插)插值到矩形网格场上再绘制。在绘制某个值的闭合等值线时,将等值线经过的三角形编号保存下来,可以得出包围闭合线的最小矩形域,保存两个对角端点坐标。寻找下一个值的闭合等值线时,若上次保存的矩形域没有包含本次保存的矩形域,搜索矩形区域内的极值后标识中心(高低、冷暖等);反之,则保存本次闭合线的矩形域,将上次保存的矩形域删除,依次下去。

本方法选用三次 B 样条曲线进行光滑处理^[6-7]。三次 B 样条曲线光滑处理方法程序容易实现,而且平滑效果较好,用于许多工程绘图中。

3 地面天气图效果的比较

如图 3 和图 4 所示,使用本文方法地面天气图的显示效果比 MICAPS 好,天气现象显示更为清晰。从两张图可以看出,在同样大小区域内,MICAPS 显示的站点数要少一些。由于 MICAPS 也使用 Visual C++ 6.0 软件开发环境,因而两者在计算机处理速度方面无明显区别。MICAPS 中缩放和拖动功能,需要相当一段时间才能刷新屏幕,而且操作不方便。我们利用工具对话框上的湖南、华中、长江中上游、中国、东亚、欧亚六个区域的选择按钮,可快速显示所选区域的地面天气图。

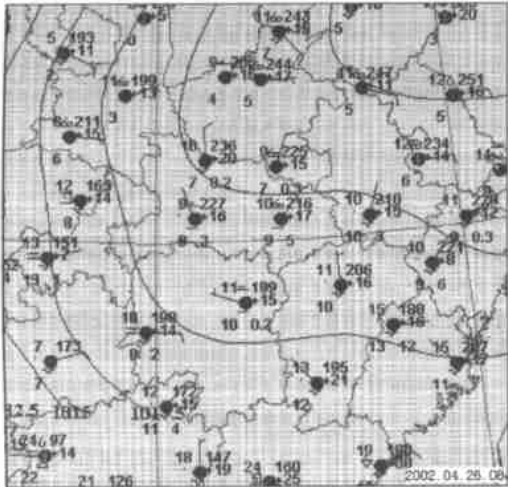


图 3 MICAPS 小区域地面图
Fig. 3 The surface weather chart
drawed by MICAPS

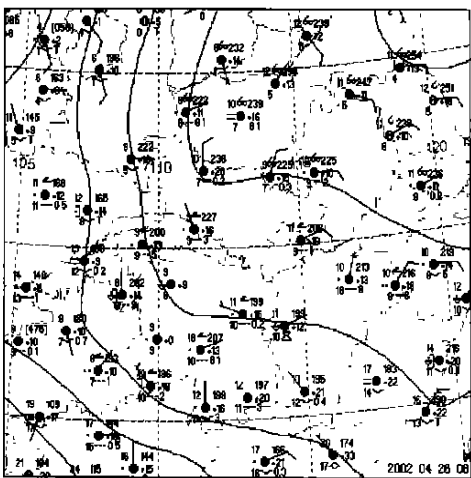


图 4 本文方法小区域地面图
Fig. 4 The surface weather chart
drawed by the method in this paper

4 结 语

使用本文方法开发的软件系统在湖南省气象台近两年的业务运行中表明, 天气图的显示和等值线的分析都可达到业务运行的要求。

利用 Visual C++ 6.0 软件开发环境, 制作位图天气符号等方法填制的地面天气图, 屏幕显示效果好。规则三角形网方法分析等值线, 思路清晰, 程序容易实现, 源程序代码也容易移植。绘制地面天气图的方法可以直接用来绘制高空天气图。下一步将考虑地面锋线、高空槽线和切变线识别, 实现天气图中所有内容的自动化分析。

参考文献:

- [1] Scott Stanfield, Ralph Arveian. Visual C++ 4 开发人员指南[M]. 华译工作室译. 北京: 机械工业出版社, 1997.
- [2] 蔡明志. Borland C++ 4.0 使用与编程指南[M]. 北京: 清华大学出版社, 1994.
- [3] 中国气象局监测网络司. 地面气象电码手册[M]. 北京: 气象出版社, 1999.
- [4] 黄淼云, 张学忠, 辛冬根. 计算机图形技术[M]. 北京: 人民交通出版社, 1996.
- [5] 匡方毅, 尹新怀. MITAS 气象要素场等值线分析方法[J]. 湖南气象, 1998, 15(3): 51-54.
- [6] 李兰友, 庄国瑜, 秦卫光. Visual Basic 绘图与图像处理[M]. 北京: 人民邮电出版社, 1999.
- [7] 王继志, 汤桂生. 气象图形显示原理与方法[M]. 北京: 科学出版社, 1991.

A Method for Drawing Surface Weather Chart and Contours of Meteorological Elements

DAI Ze-jun¹, MIAO Chun-sheng¹,
YU Wei², CHENG Jian-ping¹

(1. Department of Atmospheric Sciences, NIM, Nanjing 210044, China;

2. Hunan Meteorological Observatory, Changsha 410007, China)

Abstract: A method using bitmap as weather symbol to draw surface weather chart and using regular triangle net to draw the contours of meteorological elements are introduced in this paper. The operational results of the programme developed under the Visual C++ 6.0 software environment show that the surface charts drawn by the method are clear, and the contours are accurate and smooth. It can meet daily routine requirement of meteorological observatory.

Key words: surface weather chart; bitmap; weather symbol; regular triangle net; isopleth