基于三角网的等值线填充算法研究

李 强[1],李 超[2],甘建红[3]
LI Qiang[1], LI Chao[2], GAN Jianhong[3]

1.成都信息工程学院 计算机学院,成都 610225
2.成都信息工程学院 发展规划处,成都 610225
3.成都信息工程学院 软件学院,成都 610225
1.College of Computer Science and Technology, Chengdu University of Information and Technology, Chengdu 610225, China
2.Development and Planning Department, Chengdu University of Information and Technology, Chengdu 610225, China
3.College of Software, Chengdu University of Information and Technology, Chengdu 610225, China

LI Qiang, LI Chao, GAN Jianhong. Study on algorithm of isoline filling based on triangle mesh. Computer Engineering and Applications, 2013, 49(5):185-189.

Abstract: With deep research on isoline filling algorithm, a new cover algorithm of isoline filling is proposed in this paper. This algorithm classifies isolines as Convex Hull Closed Isoline(CHCL) and Self Closed Isoline(SCL). Taking full advantage of isoline's characteristic, the judgement of topological relation and filling color is simplified mostly. For CHCL isoline, tracing convex hull isoline-points and convex points are employed to determine filling area, and a multi-branches tree which is established by topological relation is adopted to determine the order of filling for SCL isoline. By repeating cover filling isoline-area orderly, all the isoline-area filling is accomplished. For smoothing isoline, a new method which can decrease the precision loss of isoline smoothing is adopted. Algorithm model is realized using client side programming language-ActionScript3.0, and it shows that this algorithm is easy to implement and the time cost can satisfy the business demands.
Key words: Delaunay triangle mesh; isoline tracing; isoline filling; contour plot

摘 要:通过对现有等值线填充算法的深入研究,提出了一种覆盖填充等值线的算法。该算法把等值线分类为凸包边封闭等值线(CHCL)和自封闭等值线(SCL),充分利用等值线的特点,使对等值线拓扑关系及填充颜色的判定达到最简化。对于CHCL采用对凸包边等值点及凸包点追踪来确定填充区域,对于SCL则根据拓扑关系以最外层的SCL为根节点构建一棵多叉树以确定填充顺序.通过对等值区域依次反复覆盖填充,最终完成所有等值线的填充。采用了一种穿过原始离散点平滑等值线的算法,减小了等值线平滑造成的精度损失。并运用客户端语言AS3.0(ActionScript3.0)实现了算法模型。实验结果表明,该算法简单易于实现,而且运算速度能满足业务需要。
关键词: Delaunay三角网;等值线追踪;等值线填充;色斑图
文献标志码:A **中图分类号:**TP391.7 doi:10.3778/j.issn.1002-8331.1203-0485

1 引言

等值线能够以图形的方式表示离散数据,而色斑图则是由等值线辅以分层着色,以进一步提高等值线图的直观效果。由于色斑图(如温度,降水色斑图等)看起来形象直观,它在GIS、气象学、地质,石油勘探等许多学科中有着广泛的应用。色斑图绘制过程一般分为离散数据三角化及插值[1],等值线追踪[2-3],等值线平滑[4-5],等值线填充[6-11]及裁剪几个步骤。

等值线填充就是根据等值线的拓扑关系,对等值线区域进行快速、准确的填充。等值线填充需要解决的两个关键问题是:(1)如何确定两条相邻等值线之间的区域;(2)确定这个区域应该用何种颜色填充。在图形学中有许多不同的算法能对等值线进行填充,如格网法[7]、扫描线填充[8]、Voronoi图环评[9]等。格网法需要制定基于单个三角形的填充,来拼凑任意形状的填充,如果多边形太复杂势必影响效率。扫描线填充,对填充颜色的确定比较复杂。而Voronoi图环评用Voronoi图来构建拓扑二叉树,虽然算法新颖,但算法不易编程实现。而且这些算法对等值线的特点都没

基金项目:成都信息工程学院人才引进项目(No.KYTZ201040)。
作者简介:李强(1986—),男,硕士,主要研究方向为WEBGIS空间数据分析及WEB3D;李超(1964—),男,教授,主要研究方向为基于网络的计算机应用技术;甘建红(1978—),男,博士,主要研究方向为数字图像处理。E-mail:lqlunwen@126.com
收稿日期:2012-03-21 **修回日期:**2012-07-03 **文章编号:**1002-8331(2013)05-0185-05

有进行充分利用,增加了确定填充区域的难度。本文的目的就是从等值线的特点着手,无需判断等值线首尾点在边界上的走向[11],使对等值线拓扑关系及填充颜色的判定达到最简化,减小传统算法的时间空间复杂度并结合AS3.0语言的特点提出一种可靠性高,时间复杂度小,等值线拓扑关系及填充颜色判定简单且适合在web环境中运行的算法。

2　等值线填充算法的基本思路

本文以气象实时业务系统中基于WEBGIS绘制色斑图为研究背景,以中国气象科学数据共享服务网提供的194个气象基准站的经纬度、温度以及全国行政边界的shp文件为数据源。气象实时业务的汛期(4~9月)数据访问集中,分析业务集中而频繁,希望色斑图绘制分析等业务不负载于服务器,而是使用良好的终端资源,正是基于这样的需求,结合Flash 实现气象实时业务分析系统,其核心是要实现色斑图的绘制。本文结合Flash AS3.0语言优势直接在客户端读取shp文件,对气象站点基于Delaunay快速构建三角网,插值,追踪,生成等值线,然后对等值线进行平滑及填充,最终生成等值线色斑图。

3　等值线精度分析

影响等值线的精度的因素有:离散数据网格化,网格点插值以及等值线平滑,而等值线填充不会影响等值线的精度。离散数据网格化可分为矩形网法和三角网法。相对于矩形网格,三角网法的特点[12]有:(1)三角网法不需要将离散点经插值转换为矩形网格点,减少了计算时间且三角形网格点数据就是离散点数据本身,最大程度上保留了原始数据,其精度显然高于经插值得到的矩形网格点精度。(2)能通过任意形状的凸包区域生成等值线,而矩形网法的边界则只能是矩形。(3)三角网法则对特征点部位任意小的等值线都能绘出,而矩形网格法可能会丢掉很小的闭合等值线。很明显三角网法在精度上优于矩形网法,所以本文采用已经非常成熟的Delaunay三角形来对气象站点进行构网,但是由于站点分布不均匀,可能会导致Delaunay三角网大小差别较大,比如产生狭长三角形或面积较小的三角形,最终绘制出来的等值线精度低,可视化效果差。这时可以对初始delaunay网格进行加密[13],使其成为相对均匀分布的网格,并对新插入的点采用克里金(kriging)或者距离加权反比法(Inverse Distance Weighting,IDW)插值获得数据值,最后绘制等值线,这样绘制出来的等值线可靠性更高。注意由于侧重点不一样本文没有对初始delaunay进行加密,具体算法可以参考文献[13],本文主要讨论等值线平滑对精度的损失以及等值线追踪填充算法。

绘制出来的等值线是一条一条的折线,为了使等值线连续平滑,还需要对其进行平滑处理。由于气象数据是高度离散的,如果偏差一点,可能就会造成很大的精度损失,为了确保空间分析的精度,平滑过后的曲线应该尽量通过原始离散点。常用的等值线平滑算法有二次、三次贝塞尔曲线和二次、三次B样条曲线等,其中以二次B样条曲线最为常用。但这些算法平滑过后的曲线都没有通过原始离散点,对等值线精度造成了一定的损失。三次贝塞尔曲线有两个固定点(起点和终点),另加两个决定曲线走向的控制点(CP),本文采用Maxim Shemanarev提出的一种改进的三次贝塞尔曲线(cubic Bezier)算法[14](以下简称MS Bezier算法),使平滑过后的等值线仍然穿过离散点,如图1所示。其大致思路就是先计算出相邻原始点的中点,再把相邻中点连接起来的线段平移到对应的原始点,以平移后的线段所在直线上的点作为控制点,并可以使用一个与控制点和顶点距离相关的系数K,用来沿直线移动控制点,控制点离顶点越近,拐点看起来就越趋于锐利。然后以相邻原始点为起始点画三次贝塞尔曲线,这样就得到了一条穿过所有原始点的光滑曲线,减小了等值线平滑造成的精度损失。其最终实例效果如4.3节图5所示。

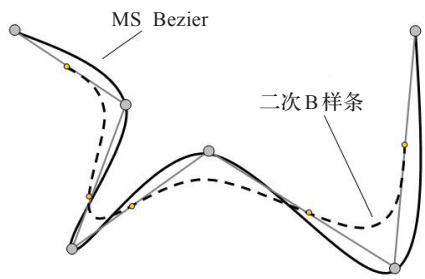


图1　MS Bezier与二次B样条平滑效果示意图

4　三角网构建与等值线追踪

为了便于阐述,整个算法实现过程的数据结构见图2。

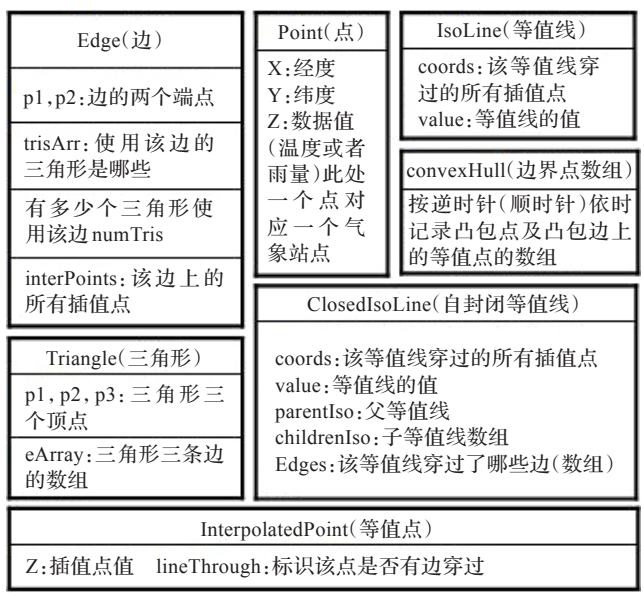


图2　算法数据结构说明

4.1　构建三角网及插值

本文采用Delaunay三角形对气象站点构建三角网。其算法基本思路是由地图边界盒先初始化两个三角形,$\Delta C1C4C2$和$\Delta C2C4C3$如图3所示。然后不断向三角网中添加新的点,在添加点的同时,采用Lawson提出的LOP方法对其进行优化,使优化后的三角形满足空外接圆,最小

角最大化的属性[1-2]。在站点三角化之后，在三角形的边上进行插值，求得等值点的坐标。在所有三角形边中找出凸包边（此处即为$C1C2$，$C2C3$，$C3C4$，$C4C1$），其判断标识为该边只被一个三角形使用。如三角形一条边为E，则该边为凸包边的标识为：E.tris.length==1。并按逆时针（或顺时针）依次记录凸包点（凸包边端点）及凸包边上的等值点（插值点），存入数组convexHull中，使其连接形成一个首尾闭合的凸包多边形，如图3所示。

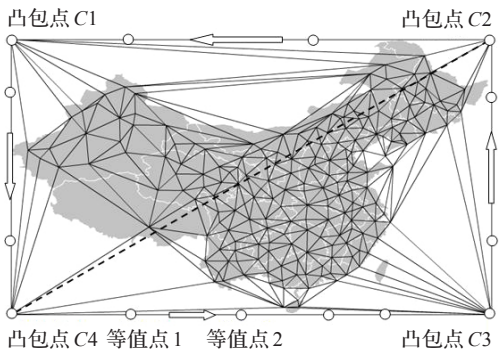


图3　站点三角网及凸包示意图

4.2　等值线追踪

等值线追踪就是在已经插值过的三角形边上找出值相同的插值点，即等值点，然后连接成等值线。如图4虚线部分为由气象站点构建的三角网，从边AB开始到EF结束的曲线为一条凸包边封闭等值线（Convex Hull Closed Isoline，CHCL），而图中较粗的环状曲线则为一条自封闭等值线（Self Closed Isoline，SCL），为了方便后面等值线的填充，把CHCL和SCL区分开，首先追踪CHCL，然后再追踪SCL。只要从凸包边上某点开始追踪，那么等值线的尾点必然落在一条凸包边上，得到的必然是一条CHCL。则遍历convexHull数组顺着凸包边依次追踪每一条CHCL，等值线追踪算法如下：

（1）如图4从等值点a开始追踪，首先判断a点是否已经被等值线穿过（a.lineThrough?=true），已穿过则跳过，否则记录a点为等值线（a到i，记为a-i）的第一个点，即a-i.coords[0]=a。将a的lineThrough属性标识为true，并标识currTri（当前三角形）=$T1$(ΔABC)。

（2）判断三角形$T1$(ΔABC)的其他两条边AB，AC，即lineThrough=false的两条边上是否有与点a值相同的等值点。

（3）通过判断发现边BC上有点b与点a值相同，记录点b为等值线a-i的第二个点，即a-i.coords[1]=b。并将b的lineThrough属性标识为true。

（4）判断点BC是否为凸包边即BC.tris.length是否等于1，是则终止，否则继续。此时使用边BC的有两个三角形$T1$，$T2$，则currTri=[(currTri==BC.tris[0])?1:0]，很明显此时currTri=$T2$则继续判断三角形$T2$(ΔBCD)的其他两条边BD，CD是否有与点b相同的等值点。

（5）重复上面三步，追踪到凸包边EF结束（EF.tris.length==1凸包边一定只被一个三角形使用），这样按顺序记录了所有等值点的一条CHCL就追踪完成了。同时记录下等值线a-i的isolineID为存放等值线的数组的当前索引值，此处为第一条即为0。

（6）然后继续以逆时针方向开始追踪凸包边BE上的下一个等值点j，直到所有CHCL追踪完毕，这样就得到一个按凸包边上等值点依次存放的CHCL数组：chclArr。

（7）最后遍历非凸包边，用同样的算法追踪SCL，只是SCL追踪的结束标识为：首尾点为同一个点，即当lineThrough=true时，即可结束追踪。此时又得到一个存放SCL的数组：sclArr。

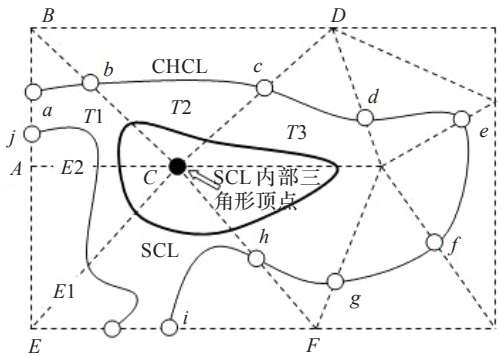


图4　CHCL和SCL等值线示意图

4.3　等值线绘制及平滑

将追踪得到的每条等值线绘制出来其实是由逐个点连接起来的一条折线，为了使后面填充的等值区域看起来连续平滑，本文采用前面提到的MS Bezier算法来对等值线进行平滑，其效果如图5所示。其中底图为经过MS Bezier平滑的等值线，两个小方块中折线为未平滑的等值线，平滑曲线分别为MS Bezier和二次B样条平滑过后的等值线。

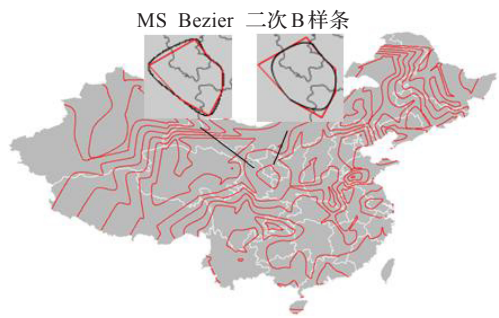


图5　MS Bezier与二次B样条平滑效果实例对比

5　等值线填充

等值线填充算法主要解决如何确定两条等值线之间的区域，以及确定用何种颜色填充的问题。但是由于任意两条等值线之间区域的形状是极不规则的，因此单纯从确定两条等值线间区域的思路出发将会使问题变得极为复杂。先来看看等值线的特征与性质：

（1）同一条等值线上的数值一定相等。

（2）相邻的两条等值线数值相差一个等值距，只有在凸包边封闭等值线包含自封闭等值线时有可能相等。

（3）等值线可能自封闭的，也可能是凸包边封闭的，而不封闭等值线的端点必然落在凸包边上。

(4)等值线一定不会相互交错。

(5)对于任意连通区域来说,围成它的所有等值线值最多为2个。

(6)“大于大的,小于小的”原则:就是在两条等值线a,b之间有一条子封闭等值线c。若c的数值与a,b中较小者相等,则c内的数值小于c的数值。若c的数值与a,b中较大者相等,则c内的数值大于c的数值。通过对等值线的特征分析,本文提出了一种快速且容易编程实现的算法。

5.1　凸包边封闭等值线(CHCL)填充

(1)如图6所示,若以a点为开始点的等值线L为chclArr中的第一条等值线,则L=chclArr[a.isolineID]=chclArr[0],从L的线头(L.coords[0])开始,顺着边界逆时针比较凸包边上点(凸包点及等值点)的X,Y值是否等于L.coords[$n-1$](n=L.coords.length)的X,Y值,直到相等时结束,在convexHull中标识L的线头和线尾为已追踪(convexHull[0].isTraced=convexHull[n].isTraced=true),并标记L的线尾为第一条填充等值线的线尾(convexHull[n].isFirstEndpoint=true),得到了第一个封闭区域。

(2)而该封闭区域的颜色则由逆时针顺序的下一个点的值来确定,此处即为<20,16>,如果图例从0开始以4为间距,则该封闭区域应该填充图例中16~20区间的颜色。

(3)继续从下一个点开始追踪,首先判断点的isTraced属性,若为true直接跳过,反之则继续。此时继续追踪左边值为16的等值线,得到第二个封闭区域,此处需要与凸包边端点b进行比较即<16,14>,所以给该封闭区域填充图例中16~12区间的颜色。

(4)直到第一条填充的等值线L的结束点变成了开始点,即点的isFirstEndpoint属性为true,此时等值线(L)的下半部分的所有CHCL等值区域都已填充好,则应顺着边界逆时针比较凸包边上点的X,Y值是否等于L.coords[0]的X,Y值。只对此等值线进行特殊处理,其他等值线都按原来的方法填充。

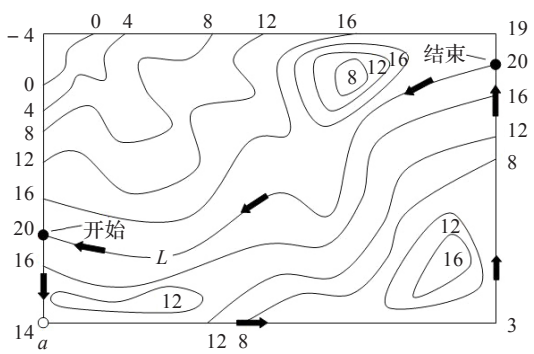
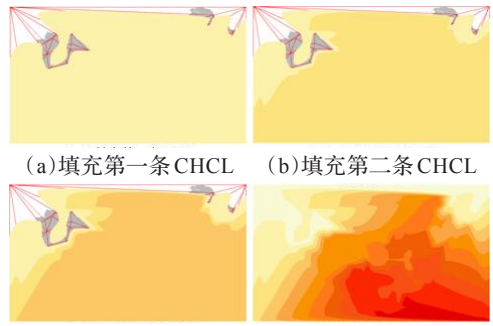


图6　CHCL填充区域追踪示意图

具体的填充过程,如图7。

5.2　自封闭等值线(SCL)填充

对于SCL需要根据等值线之间的包含关系以最外层的SCL为根节点构造为一棵棵的拓扑树,即把所有SCL转换为数据结构中的森林。首先填充外层SCL,然后依次递归填充其子SCL,其填充颜色由其子SCL的值来确定。具



(a)填充第一条CHCL　(b)填充第二条CHCL

(c)填充第三条CHCL　(d)所有CHCL填充完毕

图7　CHCL的填充过程图

体算法实现如下:

(1)如图6右下角两条SCL,12到16之间就应该填充图例中12~16区间的颜色值。而值为16的SCL内部则应该填充图例中16~20区间的颜色值。

(2)而对于图4中左下角的那条SCL,它既没有父节点又没有子节点。虽然直观从等值线图上以“大于大的,小于小的”原则很容易判定该SCL内部应该填充什么颜色,但是让程序去判断包围该SCL的等值线的拓扑结构是难以办到的。通过观察发现SCL内部必然有1个或者多个三角形顶点,可根据其值判断,如图4所示。因此判断标准如下:若$E1$(EC)和$E2$(AC)为该SCL穿过的两条边,$P1$,$P2$分别为其端点,有:

```
If(E1.P1==E2.P1||E1.P1==E2.P2)
    三角形顶点值=E1.P1.Z
Else If(E1.P2==E2.P1||E1.P2==E2.P2)
    三角形顶点值=E1.P2.Z
```

(3)现在需要解决的问题是如何构建这棵拓扑树。根据等值线性质4,等值线不能相互交错,则判断两条自封闭等值线的包含关系即可简化为,判断某条等值线是否包含另一条等值线的任意一个点,即判断点是否在多边形内。判断点在多边形内的算法很多,如:水平交叉点数判别法[15]。其算法为$O(N)$,N为多边形边的数目。

(4)建立拓扑树的伪代码如下:

```
For Each (var ci:ClosedIsoLine in sclArr)
  For Each (var cii:ClosedIsoLine in sclArr)
   If(cii.val!=ci.val+interval&&
     ii.val!=ci.val - interval)
         continue;
     //如果两条等值线相差不是一个等距值直接跳过(等值线性质2)
     //insidePolygon为判断点在多边形内函数
    Else
     If(insidePolygon(ci.coords,cii.coords[0]))
        ci.childrenIso.push(cii);
     Else If(insidePolygon(cii.coords,ci.coords[0]))
        ci.parentIso=cii;
     EndIf
    EndIf
  EndFor
EndFor
```

假如自封闭等值线SCL的条数为M,由于排除了不符合等值线间距规则的等值线,所以两次遍历closedIsos的时间T应该小于等于$M\times M$,再加上insidePolygon的时间复杂度为$O(n)$,则整个算法的时间复杂度小于等于$O(M\times M\times N)$,N为多边形边的数目,即该等值线插值点个数。以全国194站的温度为例,填充好的色斑图如图8所示。

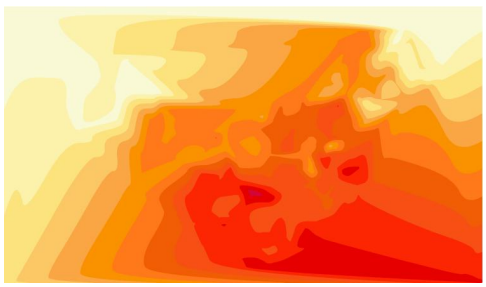
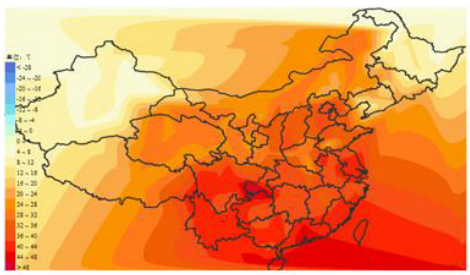


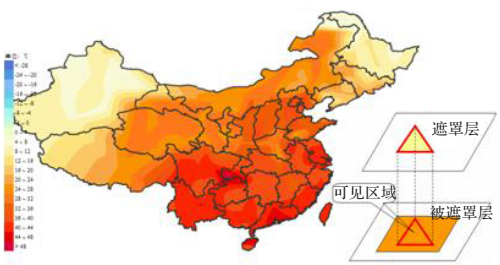
图8　全国194站温度色斑图

5.3　色斑图裁剪

等值线图绘制好后,还需要对超出地区边界区域的部分进行裁剪。本文使用Actionscript3.0内置了蒙板遮罩的函数,其原理就是,遮罩层决定看到的形状,被遮罩层决定显示的内容。那么可以将底图显示对象复制为两份,一份用于画等值线作为被遮罩层,一份用于决定显示形状作为遮罩层。用全国行政边界的数据和内置函数得到裁剪后的色斑图如图9所示。



(a)未裁剪的温度色斑图



(b)裁剪后的温度色斑图

图9　温度色斑图裁剪前后的对比图

6　算法效率测试

取全国194站的温度,用AS3.0语言在CPU为Core 2.1 GHz,内存1 GB的PC上实现了全部算法。表1中列出了本算法及近年来发表的等值线填充算法执行效率比较,衡量标准为站点数比上运行时间(单位:s)。

表1　几种等值线填充算法效率对比

算法	站点数/运行时间	硬件平台	语言
本文算法	约730	Core 2.1 GHz,1 GB	AS3.0
文献[11]	约630	Core 2.1 GHz,1 GB	AS3.0
文献[9]	约570	Core 2.1 GHz,1 GB	AS3.0
文献[7]	约410	Core 2.1 GHz,1 GB	AS3.0

7　结论

当前,随着气象探测系统的进一步发展与完善,往往一个省的观测站网将达到2 000~3 000个。另一方面,气象实时分析业务所采用的硬件平台(PC)比本文实验中采用的硬件平台(见表1)性能要好得多,因此,利用该算法在终端处理气象实时业务的色斑图分析完全是可行的,从而保证实时业务的稳定性(特别是在每年4~9月汛期)。实践表明,该算法充分结合AS3.0语言的特点及优势,易于编程实现,等值线可信度高,适合在web中运行,且运算速度能满足气象业务需求。

参考文献:

[1] Bourke P.Efficient triangulation algorithm suitable for terrain modelling[C]//Pan Pacific Computer Conference.Beijing:[s.n.],1989.

[2] 郭新奇,严建钢,杨士锋,等.基于VC++的等值线追踪与填充算法[J].兵工自动化,2011,30(4):81-84.

[3] 张顺谦.基于Delaunay三角网的区域等值线绘制关键算法[J].计算机与网络,2005,91(25):39-41.

[4] 蒋瑜,杜斌,卢军,等.基于Delaunay三角网的等值线绘制算法[J].计算机应用研究,2010,27(1):102-103.

[5] Wen Yihong,Liu Yongjiang.An isoline generating algorithm based on Delaunay[C]//International Conference on Computer Engineering and Technology.Chengdu:[s.n.],2010:173-175.

[6] 黄本宇,王家华,王湘波.复杂地质构造的等值线填充及实现[J].阜阳师范学院,2006,23(4):51-52.

[7] 杨清.C#实现基于三角网的等值线追踪及填充算法[EB/OL].(2010-11-29).http://blog.sciencenet.cn/u/moustudio.

[8] 邓飞,王美平.基于扫描线转换的快速等值线填充算法[J].电子技术应用,2006(3):39-40.

[9] 吴培宁,谭建荣.基于Voronoi图的环评等值线快速拓扑填充[J].浙江大学学报:工学版,2009,43(2):322-327.

[10] 吴自银,高金耀.一种基于格网的快速等值线充填算法[J].测绘学报,1999,28(4):351-354.

[11] 汤子东,郑明玺,王思群.一种基于三角网的等值线自动填充算法[J].中国图象图形学报,2009,14(12):2578-2581.

[12] 巫昌海.提高三角网法绘制等值线精度的算法[J].武测科技,1993(2):10-12.

[13] 胡金虎.基于不规则三角网的高精度等值线生成方法[J].工程勘测,2011(2):64-68.

[14] Interpolation with Bezier a very simple method of smoothing polygons[EB/OL].(2006-08-16).http://www.antigrain.com/research/bezier_interpolation/index.html.

[15] Bourke P.Determining if a point lies on the interior of a polygon[EB/OL].(2010-12-05).http://paulbourke.net/geometry/insidepoly/.