

大幅面地图的三维地形重建¹⁾

胡卫明^{1,2} 吴兵² 刘巍²

¹⁾(中国科学院自动化研究所模式识别国家重点实验室 北京 100080)

²⁾(北京大学文字信息处理技术国家重点实验室 北京 100871)

(E-mail: wmhu@nlpr.ia.ac.cn)

摘要 研究完成了一个实用的可用于大幅面像素地图的三维地形重建系统. 提出一个实用的等高线内插算法, 算法克服了网格绘制等值线方法和三角网绘制等值线方法因只考虑了点的位置属性、未考虑等高线的线属性而使插出的等高线常常会与母线相交的弱点; 同时提出了一个实用的等高线高程识别算法和一个可用于大幅面地图的三维规则数据场建立算法, 算法充分利用了等高线的先验知识, 不仅建立的数据场的质量很高, 而且算法的时间复杂性与数据点数(m)和网格点数(n)成线性关系 $O(m+n)$, 因此计算速度很快. 最后利用三维规则数据场生成真实感图形.

关键词 大幅面地图, 等高线自动解释, 三维地形重建

中图分类号 TP391

3-D SURFACE RECONSTRUCTION FROM VERY LARGE TOPOGRAPHIC MAP

HU Wei-Ming¹ WU Bing² LIU Wei²

¹⁾(*The National Laboratory of Pattern Recognition, Institute of Automation
Chinese Academy of Sciences, Beijing 100080*)

²⁾(*The National Key Laboratory of Text Processing Technology, Peking University, Beijing 100871*)
(E-mail: wmhu@nlpr.ia.ac.cn)

Abstract We introduce our practical system that is applicable to very large topographic map. We present a practical algorithm for contour line interpolation, which can get rid of some fatal defects often found in the grid and triangular net approaches, known as “the interpolated line’s intersection with mother contour lines”. We propose an algorithm for contour line elevation interpretation. A practical algorithm for three-dimensional regular data field is presented to handle a very large map, and the quality of the calculated grid elevation is high due to the sufficient exploitation of prior knowledge. The run-time complexity of the approach for establishing the 3-D regular data field is $O(m+n)$, linear to the number of data points (m) and the number of grid

1) 中国科学院自动化研究所创新基金(1M01J02)和国家“九五”重中之重科技攻关基金资助.

points(n), therefore, the processing speed is very high. Finally, a real-time graph is created with the three-dimensional regular data field.

Key words Very large topographic map, contour line interpretation, 3-D surface reconstruction

1 引言

等高线地图是用二维平面表示三维地形的重要工具,但是读懂等高线地图需要一定的专业知识.近几年来,对等高线地图进行三维地形重建成为地图学和地理信息系统的研究热点,其目的就是使地形特征或地理信息表示得更加直观,使地图的三维可视化程度更高,未来的地图应该像一张彩色照片一样直观易懂,即使没有多少专业知识的人也能容易地读懂它.地图的地形重建一般都是先建立好等高线,再对每条等高线逐一赋高程值,然后建立DTM(Data Terrain Model)或TIN(Triangular Irregular Network),并由DTM或TIN生成反映地形的真实感图形.由于等高线是由各点连接而成的,DTM的建立一般是把等高线上的点当一般离散点,由这些离散点生成三维规则数据场,通常有移动曲面拟合内插法、多面函数内插法、配置法、有限元插法等^[1].TIN的建立则应基于最佳三角形的条件,一般有角度判断法,泰森多边形与狄洛尼三角网法,对于等高线地图则应该在相邻等高线之间建立三角网^[2~4].文献[5]则将等高线内插与三维地形重建相结合,生成的三维规则数据场能产生光滑的表面.尽管已有不少方法可以建立基于地图的三维规则数据场,但仍然有一些问题需要解决.例如:大幅面地图有很大的数据量需要处理,要求三维规则数据场建立算法既有较高的质量又有较快的运行速度.

近年来,有人开展了直接从像素地图重建三维地形的研究.直接从像素地图重建地形需要对等高线作自动解释,包括等高线的矢量化和高程识别.由于等高线拓扑的总体约束,从图像地图重建等高线是一个很困难的事情;过去的研究表明,等高线的局部几何特征不足以对等高线的自动重建提供足够的依据.相关文献都采用用户干涉的方法绕过这一不足,其中人的作用大多是解决一些算法无法作出正确判断的情况或者纠正一些不正确的重建结果^[6,7].文献[8]提出了一个学习策略来减少交互次数.文献[9]提出了另一个等高线半自动重建方法,该方法建立了一个矢量跟踪Agent,并提出一个Agent之间的协作环境,以使矢量化时能通过等高线的不规则部分,在此方法中用户的角色是初始化Agent,因为Agent的初始化很精细,所以对用户的要求很高,文献[10]将重建问题形式化描述为一个二着色问题,即用与等高线尽量吻合的两个高程创建一个DTM,尽管这种方法在绝大多数情况下有较好的结果,但是当等高线间断太多并且太大时该方法就失效了.文献[11,12]利用已知的中间DTM判断所有等高线的方向,按照某种规则计算出各等高线“5着色”的概率,然后判断出各等高线高程.这固然是一种新的研究思路,但这种方法求出的等高线高程可靠性得不到保证,需要进行一致性检查,并由用户纠正.文献[13]则从像素地图中抽取高程符号、高程字符和等高线,生成DTM时把高程点作高程约束、把等高线仅作切向约束,这种方法的问题是重建的三维地形很不精确,特别在高程点很稀疏时.

目前直接从像素地图重建三维地形的研究中都存在着两个问题:一个是不够实用;另一

个是未对大幅面地图的三维地形重建作特殊的研究。

本文完成了一个实用的可用于大幅面像素地图的三维地形重建系统. 本系统将图像地图中等高线矢量化, 并自动(半自动)识别出等高线的高程, 再建立三维规则数据场, 从而生成表示地形的真实感图形。

2 系统框架

图 1 是系统框架图. 系统包括如下五个部分。

第一部分是对像素地图等高线的矢量化. 本文采用非细化方式对黑白地图或彩色地图的等高线进行矢量化。

第二部分是等高线内插. 等高线内插是矢量化的辅助工具. 本文提出一个实用的等高线内插算法, 算法克服了网格绘制等值线方法和三角网绘制等值线方法因只考虑了点的位置属性、未考虑等高线的线属性而使插出的等高线常常会与母线相交的弱点。

第三部分是高程识别. 本文提出的等高线高程识别算法不仅可靠, 而且大大减少了用户对等高线赋高程的工作量。

第四部分是建立三维规则数据场. 本文提出了一个可用于大幅面地图的三维规则数据场建立算法, 在该算法中, 网格点的高程是通过与该网格点相邻的等高线高程按距离加权得到的, 网格点的相邻等高线则是通过等高线上的离散点与网格点之间的拓扑关系求得; 求得的三维规则数据场还要进行一次修改, 使山顶和谷底等地形表示得更加自然; 算法充分利用了等高线的先验知识, 不仅建立的数据场的质量很高, 而且算法的时间复杂性与数据点数(m)和网格点数(n)成线性关系 $O(m+n)$, 因此计算速度很快。

第五部分是利用三维规则数据场生成真实感图形。

系统中等高线矢量化、等高线内插、加粗等高线赋高程值和设置真实感图形生成参数, 需要较多的用户交互; 而其它部分只需用户作一些简单的控制即可。

3 等高线矢量化

栅格数据到矢量数据的转换通常分栅格数据的细化(亦称轴化)和矢量化(亦称线性化)两步进行. 本系统采用一种非细化的矢量化追踪算法. 对于黑白图像不先进行细化预处理, 直接进行矢量化. 算法的思想是使用一个比线条宽度略大的窗口套合线条, 并沿着线条进行追踪. 追踪过程中得到线条的中心轨迹, 在碰到线条的端点或分枝点的时候对窗口内部的信息进行分析, 以确定当前所处的位置并作出相应处理. 该方法的矢量化追踪直接取线条的中

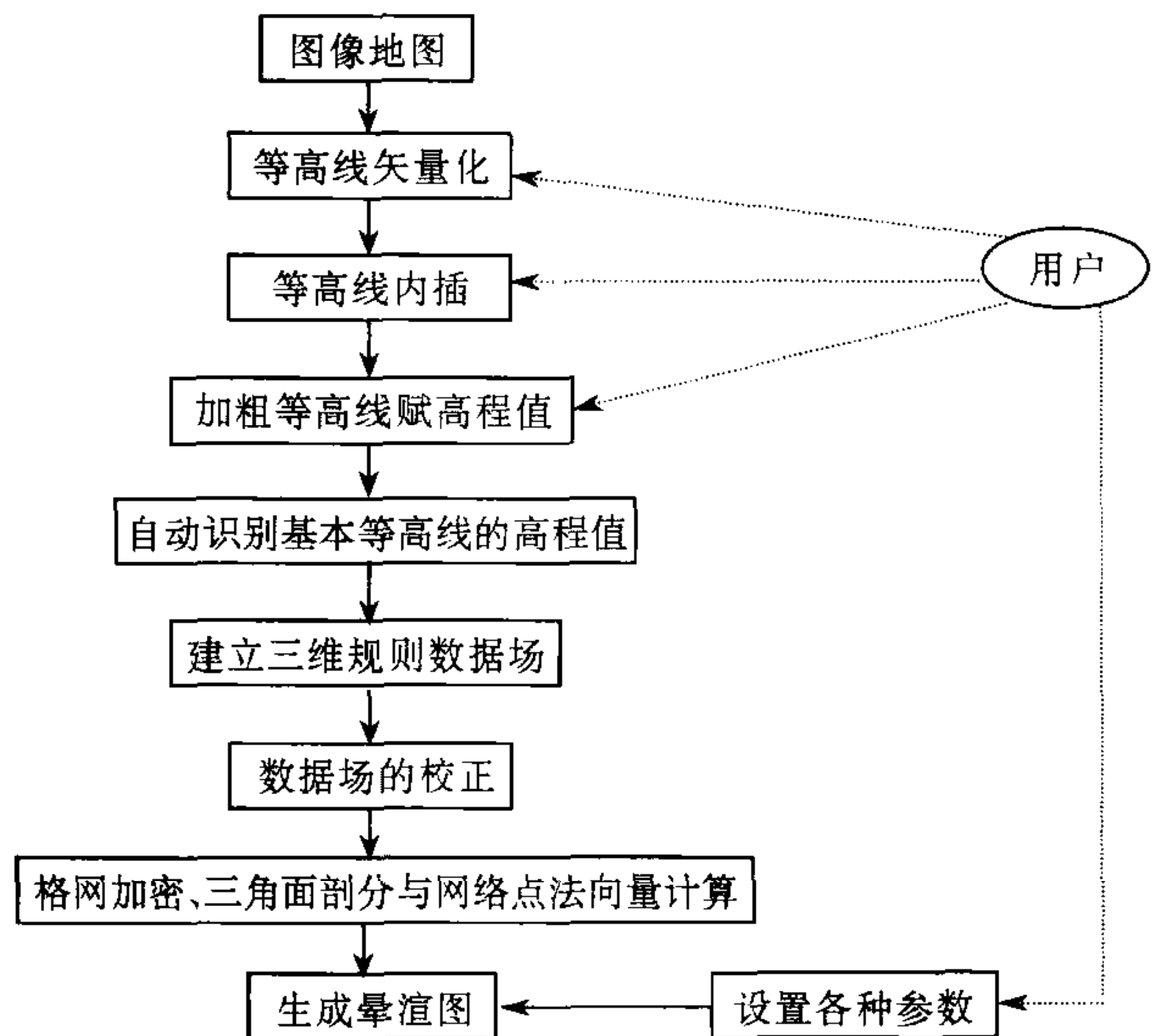


图 1 系统框架

心线,偏差小;线条受毛刺、孔洞的影响小.彩色图像中颜色数较多,图像文件大,且有多种颜色的像素混叠在一起,在矢量化时如何区分哪些是需要提取出来的线条信息,哪些是应该忽略的信息?本系统使用了一种称为动态颜色判定的方法,即并不是绝对划分线条的前景背景,相同颜色的像素在不同的情况下既有可能是需要识别的线条,也可能是背景图像的一部分.而这一切都是在矢量化的追踪过程中动态判定的.实践表明可以较好地彩色图像中识别线条,并且既可以识别浅色背景中的深色线条,也可以识别深色背景中的浅色线条,作者已在文献[14]作过详细介绍.

4 等高线内插

等高线内插是指在已有等高线的基础上自动内插出与已知等高线相一致的若干条等高线.等高线自动内插可以认为是等高线自动解释的一个研究内容.有时像素地图中某些局部等高线很密,自动矢量化时很困难,需要用户的频繁交互;而手工绘这些等高线也是根据已知等高线人为插出的,并非实测,此时可以只矢量化部分等高线(如加粗等高线),而已矢量化等高线之间的等高线(如基本等高线)直接通过等高线自动内插生成.网格法和三角网法都因只考虑了点的位置属性,未考虑等高线的线属性,而使插出的等高线常常会与母线相交.另外,这两种方法的时间复杂性较高.本文提出一个等高线在线内插算法,算法充分利用了等高线的先验知识,速度快,插出的等高线质量高.另外,算法能够在等高线局部弯曲很大的地方内插出很好的等高线,还能解决地图编辑中实际会遇到的等高线延伸方向的判别、封闭等高线和马鞍形地形等问题,从而形成了一个能处理各种复杂情况的等高线内插算法.算法描述如下.

设等高线 1 由点 p_1, p_2, \dots, p_m 连接而成;等高线 2 由点 q_1, q_2, \dots, q_n 连接而成,要在等高线 1 和等高线 2 之间均匀地插入 4 条等高线.

步骤 1. 等高线延伸方向的判别.可以用两个方向绘同一条等高线.对等高线 1 和等高线 2,若延伸方向不一致,那么要将其中一条等高线的点序反过来.判别方法如下:对于等高线 1 与等高线 2 距离最近的两个点 p_i, q_j ,如果点 p_{i-1} 和点 q_{j-1} 在线 $p_i q_j$ 的同一侧,并且点 p_{i+1} 和点 q_{j+1} 在线 $p_i q_j$ 的同一侧,那么等高线 1 和等高线 2 延伸方向相同,否则延伸方向相反.

步骤 2. 封闭等高线的处理.起点和终点的距离小于某一个值的等高线就可以认为是封闭等高线.封闭等高线有它独特的地方.此时的处理方法是:对于等高线 1 与等高线 2 距离最近的两个点 p_i 和 q_j ,等高线 1 的延伸方向不变,起点改为 p_i ,终点改为 p_{i-1} ,等高线 2 的延伸方向也不变,起点改为 q_j ,终点改为 q_{j-1} .

步骤 3. 对于等高线 1 与等高线 2 距离最近的两个点 p_i, p_j ,在线段 $p_i p_j$ 上内插出等值点 r_k ,在等高线 1 上,从点 p_{i+1} 到点 p_m 中求出点 p_{flagI} ,满足:1)以直线 AB 为界 p_{flagI} 与点 r_{k-1} 不在同一侧;2)线段 Bp_{flagI} 与线段 Ap_{i+1} 不相交;3)线段 Ap_{flagI} 与线段 Bq_{j+1} 不相交;4) $\angle Ap_{flagI}B = \max_p \angle ApB$, 其中点 p 是满足 1)~3)的点.

同样,在等高线 2 上,从点 q_{j+1} 到点 q_n 中求出点 q_{flagJ} ,满足:1)以直线 AB 为界点 q_{flagJ} ,与点 r_k 不在同一侧;2)线段 Aq_{flagJ} 与线段 Bq_{j+1} 不相交;3)线段 Bq_{flagJ} 与线段 Ap_{i+1} 不相

交; 4) $\angle Aq_{flagj}B = \max_q \angle AqB$, 其中点 q 是满足 1)~3) 的点.

再作如下处理: 如果 $\angle Ap_{flagl}B > \angle Aq_{flagj}B$ 在线段, 则在线段 $p_{flagl}p_j$ 上内插出等值点, 否则在线段 p_iq_{flagj} 上内插出等值点, ..., 如此下去, 就可以用等高线上的点 p_i 到点 p_m 的线段, 和等高线 2 上的点 q_j 到点 q_n 的线段内插出等高线线段 $contour1$. 用同样的方法可以在等高线 1 上的点 p_i 到点 p_1 的线段, 和等高线 2 上的点 q_j 到点 q_1 的线段内插出等高线线段 $contour2$, 再将 $contour2$ 倒向, 后接 $contour1$ 即为所需要内插的等高线.

步骤 3 中, 点 p_{i+1} 到 $p_{flagl-1}$ 点中的点仍然有选择地与点 q_j 内插出等值点, 方法是: 对于点 p_{i+1} 到 $p_{flagl-1}$ 点中的任意点 p , 如果线段 pq_j 与等高线 1 不相交, 那么在线段 pq_j 上内插出等值点. 这可以在等高线局部弯曲很大的地方内插出很好的等高线. 在这一局部, 三角网等值线绘制方法将形成等高三角形而不插任何等值点, 这样插出的等高线质量不高.

以上算法的时间复杂性为 $(m+n)\log_2^{(m+n)}$, 能达到在线生成等高线.

对于马鞍型地形, 一条等高线内会派生出多条同一高程的等高线, 称之为“一”对“多”的关系. 这种情况可以用“搭桥法”解决, 即将多条同一高程的等高线用线连接起来, 组成一条等高线, 再用上述两条等高线内插等高线的算法插入所需要的等高线. “多”条同一高程的等高线间搭桥的原则是: 1) 等高线间距离最近的两个点相联; 2) 内部“多”条同一高程的等高线的延伸方向应改为与外面的“一”等高线一致; 3) 搭桥后形成的等高线的延伸方向也要与外面的“一”等高线一致. “多”中的等高线可能不封闭(这种情况会出现于地图的边部), 可以将这条等高线首尾相联形成封闭等高线参加搭桥.

5 高程识别

等高线解释的另一个重要问题是等高线高程的识别. 等高线高程识别是一个比矢量化困难得多的问题. 等高线高程的全自动识别是不可靠的, 实用化算法一般都采用用户交互的半自动方式. 地图等高线分为两类, 一类为加粗等高线, 即计曲线; 另一类为基本等高线, 即首曲线. 地图中每隔 4 条基本等高线加绘一条加粗等高线. 本文采用的方法分如下两步. 第一步: 给用户提供一些方便的手段, 由用户给定加粗等高线的高程, 如用户给定起始高程、等高线的高差, 用拉线法给该条线所穿过的加粗等高线一次性赋高程值; 第二步: 由加粗等高线的高程识别出基本等高线的高程. 这种方法不仅可靠, 而且大大减少了用户对等高线赋高程的工作量. 下面是基本等高线高程识别算法的描述.

设地图的左下角坐标为 (X_{left}, Y_{Bottom}) 、右上角坐标为 (X_{right}, Y_{Top}) . 从左到右作 m 条水平直线 $HLine_0, HLine_1, \dots, HLine_{m-1}$, 它们的纵坐标分别为 y_0, y_1, \dots, y_{m-1} , 其中 $y_0 = Y_{bottom}, y_{i+1} - y_i = (Y_{Top} - Y_{Bottom}) / (m - 1), i = 0, 1, \dots, m - 1$, 这 m 条水平线把地图分成 $m - 1$ 等分, 每条水平线会与地图上的一些等高线相交. 设水平线 $HLine_i$ 与地图中的等高线有 n 个交点, 交点从左到右分别为 p_0, p_1, \dots, p_{n-1} . 对水平线 $HLine_i$, 从左到右第 g 个属于加粗等高线的交点记为 p_g , 第 $g + 1$ 个属于加粗等高线的交点记为 p_{g+1} , 它们的高程分别为 $Elev_g$ 和 $Elev_{g+1}$; 设该平行线上交点 p_g 和交点 p_{g+1} 之间有 k 个交点, 记为 p_0, p_1, \dots, p_{k-1} . 如果 $Elev_g \neq Elev_{g+1}$, 用如下算法对点 p_0, p_1, \dots, p_{k-1} 所在的等高线赋高程值 ($Elev_g = Elev_{g+1}$ 的情况下一步处理).

```

int contour [4];
contour[0]=点  $p_0$  所在等高线的序号; count=1;
for (x=1; x<k; x++) { if (点  $p_x$  所在的等高线的序号  $\notin$  {contour[0], ..., contour
    [count-1]}) {
        contour[count]=点  $p_x$  所在等高线的序号;
        count++;
    }
}
if (count==4)
    for (x=0; x<4; x++)
        Elev[contour[x]]=Elevg+0.2(x+1)(Elevg+1-Elevg).

```

上述算法的基本思想是先计算出两条加粗等高线间的 4 条基本等高线,再根据这 4 条基本等高线所处的相对于加粗等高线的位置,计算出各基本等高线的高程.用上述同样的方法处理与该水平线相交的其它等高线,求得其中部分等高线的高程.

对每一条水平线,都作与水平线 $HLine_i$ 相同的处理,求出相应基本等高线的高程;同时也可以从上到下作 n 条垂直直线,用来切割地图,用与水平线相同的处理方法求出未赋高程的等高线的高程.

作了以上处理以后,又反过来处理:在水平线 $HLine_i$ 上,对于从左到右第 g 个属加粗等高线的交点 p_g 和第 $g+1$ 个属于加粗等高线的交点 p_{g+1} ,当这两条加粗等高线的高程相等 ($Elev_g = Elev_{g+1}$) 时的情况.此时交点 p_g 和交点 p_{g+1} 之间的各交点所属等高线的高程基本上已经在其它水平线或垂直线上求出;如果尚未求出,那么就根据第 $g-1$ 个属于加粗等高线的交点 p_{g-1} 和第 $g+2$ 个属于加粗等高线的交点 p_{g+2} 的高程作出判断.不妨设 $Elev_{g-1} < Elev_g$,其它情况类似处理.如果此时交点 p_g 和交点 p_{g+1} 属于同一等高线,那么求出交点 p_g 和交点 p_{g+1} 之间从左到右的 l ($l \leq 4$) 条等高线 $contour[0], contour[1], \dots, contour[l-1]$, 等高线 $contour[j]$ 高程为 $Elev_g + \Delta h * j$, 其中 Δh 为等高线的最小间距.如果此时交点 p_g 和交点 p_{g+1} 不属于同一等高线,求出交点 p_g 和交点 p_{g+1} 之间从左到右的 $2l$ 条等高线 ($l \leq 2$, 这种情况下交点 p_g 和交点 p_{g+1} 之间的等高线必为双数), 对于等高线 $contour[j]$, 当 $j < l$ 时,该等高线的高程为 $Elev_g + \Delta h * j$; 当 $j \geq l$ 时,该等高线的高程为 $Elev_g + \Delta h * (2l - j)$.

6 三维规则数据场的建立

6.1 基本插值公式

如果点 A 与 n 条等高线相邻,这条 n 条等高线的高程分别为 z_1, z_2, \dots, z_n (对于马鞍型地形,点 A 会与多条等高线相邻,此时有 $z_2 = z_3 = \dots = z_n$), 点 A 到这 n 条等高线距离分别为 r_1, r_2, \dots, r_n , 一个点离某一等高线越近就越接近该等高线的高程,点 A 的高程 z 可以用如下公式求得

$$\begin{cases} z = \sum_{i=1}^n c_i z_i \\ c_i = \frac{1/r_i}{\sum_{j=1}^n 1/r_j} = \frac{\prod_{j=1}^n r_j/r_i}{\sum_{i=1}^n (\prod_{j=1}^n r_j/r_i)} \end{cases} \quad (1)$$

其中 $\prod_{j=1}^n r_j = r_1 r_2 \cdots r_n$. 从上式可以看出 $\sum_{i=1}^n c_i = 1$, 这是权系数的必要条件, 权系数与距离成反比, 如果点 A 在等高线 i 上, 则 $c_i = 1$, 其余权系数均为零.

6.2 网格点相邻等高线的判别

地图的等高线是极其错综复杂的, 如何自动判断出与某一点相邻的等高线呢? 这似乎很困难, 其实不然. 事实上, 矢量等高线是由若干个点连接而成的, 这些点可以认为具有 4 维属性, 它除了一般离散点所具备的空间特征(即 x, y, z 坐标)外, 还有是否属于同一条等高线这一属性. 设某一网格点 A 位于 (j, i) 网格坐标交点上, 欲求与点 A 相邻的各等高线, 可以以点 A 为相对原点将地图平面划分为 $l (l \geq 4)$ 个均匀的象限, 在每个象限内找出距离 A 点最近的一个数据点, 这些数据点应该在点 A 相邻的等高线上. 有的象限可能是白化区, 没有等高线通过该象限, 当然在该象限内就没有离 A 最近的点, 这样情况都出现在地图的边部. 设有 N 个象限有离 A 最近的点, 这 N 个点可能有若干点属于同一等高线, 将该 N 个点分成 n 个集合, 每个集合内的点都属于同一等高线, 在每个集合中只取其中离 A 点最近的一个点, 这时可以求出各象限中离 A 点最近的属于不同等高线的 n 个点 (p_1, p_2, \dots, p_n) , 实际上也就求出了与点 A 相邻的 n 条等高线, 可以近似地用点 A 到点 p_1, p_2, \dots, p_n 的距离作为点 A 到各相邻等高线的距离 (r_1, r_2, \dots, r_n) , 再利用公式(1)即可求出点 A 的近似高程.

6.3 算法描述与时间复杂性

设有 m 个数据点、 n 个网格点, 划分 l 个象限, 算法描述如下:

$nearstPoint[n][l]$ 表示 n 个网格点在 l 个象限内离网格点最近的点;

for 每个数据点 p_i

{以 p_i 为中心、以 r 为半径作一个邻域 Np_i ;

for 邻域 Np_i 内的每个网格点 GP_j

{判断点 p_i 位于网格点 GP_j 的第几象限;

不仿设 p_i 位于网格点 GP_j 第 k 象限,

如果 GP_j 与点 p_i 的距离小于 GP_j 与点 $nearst[GP_j][k]$ 的距离,

那么 $nearst[GP_j][k] = p_i$;

}

for 每个网格点 GP_i

{根据每个象限内离 GP_i 最近的点 $nearstPoint[GP_i][l]$, 求出 GP_i 的各相邻等高线;

根据与网格点 GP_i 相邻的各等高线, 由式(1)计算出 GP_i 的高程值. }

该三维规则数据场建立算法的计算时间为 $O(r * m + n)$, 由于 $r \ll m$, 所以该算法的时间复杂性为 $O(m + n)$, 它与数据点数和网格点数成线性关系, 因此速度很快. 又由于本算法是通过网格点的相邻等高线的高程计算网格点高程的, 利用了等高线固有的规律, 因此求得三维规则数据场质量很高.

6.4 异常处理

基本上各象限内离 A 点最近的点一定在点 A 相邻的等高线上, 但也存在一些异常情况, 如图 2(a) 所示, 如果以点 A 为原点像二维坐标轴那样, 把平面分成四个象限, 那么在第一象限中, 点 A 离点 p_1 的距离比点 A 离点 p_2 的距离要远, 但是与点 A 相邻的等高线是 p_1 所在的等高线 1 而不是 p_2 所在的等高线 2, 这是等高线 1 上的线段 $p_1 b$ 太长造成的, 可以用

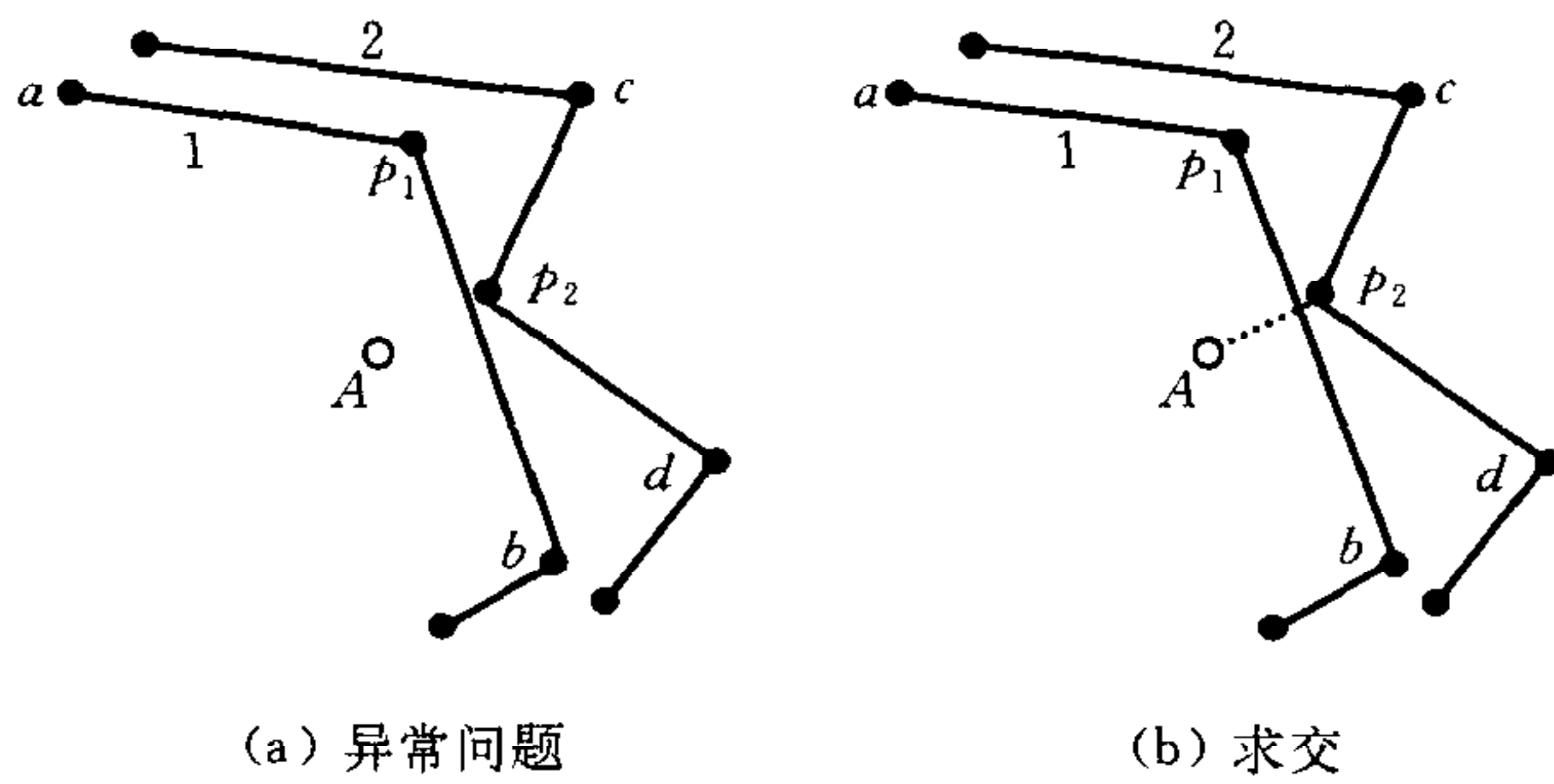


图 2 异常处理示意

如下方法解决这类异常问题:各象限同时求出两个离 A 点最近又属于不同等高线的点 p_1 和 p_2 , 不仿设 $r_1 > r_2$, 连接点 A 和 p_2 , 如果线段 Ap_2 与以点 p_1 为顶点的两条线段中的任何一条线段相交, 那么点 p_1 所在的等高线为点 A 的相邻等高线, 否则点 p_2 所在的等高线为点 A 的相邻等高线. 如图 2(b), 线段 Ap_2 与线段 p_1b 相交, 因此取点 p_1 所在

的等高线为点 A 的相邻等高线.

6.5 数据修正

用以上方法求出的网格点高程值的质量是比较高的, 但有一种情况, 需要说明一下, 即各象限上离网格点 A 最近的点都属于同一等高线, 这时如果用该等高线的高程值作为点 A 的高程值, 那么生成的真实感图形会出现平顶山和平坦谷底等, 这对黄土地貌是合适, 但对山区等地貌有时表示得不一定很自然. 本文提出用曲面拟合来修正该点高程. 方法是在 A 点的每个象限上求出离 A 点最近且属于不同等高线的两个点, 然后用这 2 个点拟合成趋势面, 再求出 A 点的高程值. 这样求出的高程值可以使山顶和谷底等地形表示得更加自然. 趋势面拟合方法见文献[15].

7 真实感图形的生成

7.1 网格加密

为了保证绘出的晕渲图具有连续色调的效果, 要求建立的三维规则数据场的格网边长不应大于 0.25mm. 对于大幅面地图, 如果格网边长仍取 0.25mm, 不但需要过长的计算时间, 而且需要过大的内存空间. 对于大幅面地图, 格网边长 1.0mm 就足以表示地形特征了. 因此, 可以先以格网边长 1.0mm 生成三维规则数据场, 在生成真实感图形时, 在已有三维规则数据场的基础上在线地将网格加密为 0.25mm. 此时需要求出三维规则数据场范围内任意一点 $P(X, Y)$ 的高程. 由于此时已知该点所在的格网各个角点的高程, 因此可利用这些网格点高程拟合一定的曲面, 然后计算该点的高程. 所拟合的曲面一般应满足连续乃至光滑的条件. 双线性多项式内插只能保证相邻区域接边处的连续, 不能保证光滑, 但因其计算量较小, 因而是最常用的方法. 本文采用双三次多项式(三次曲面)内插, 三次多项式内插虽然属于局部函数内插, 即在每一个方格网内拟合有一个三次曲面, 但由于考虑了一阶偏导数与二阶混合导数, 因而它能保证相邻曲面之间的连续与光滑. 具体算法见文献[1].

7.2 三角面剖分与法向量计算

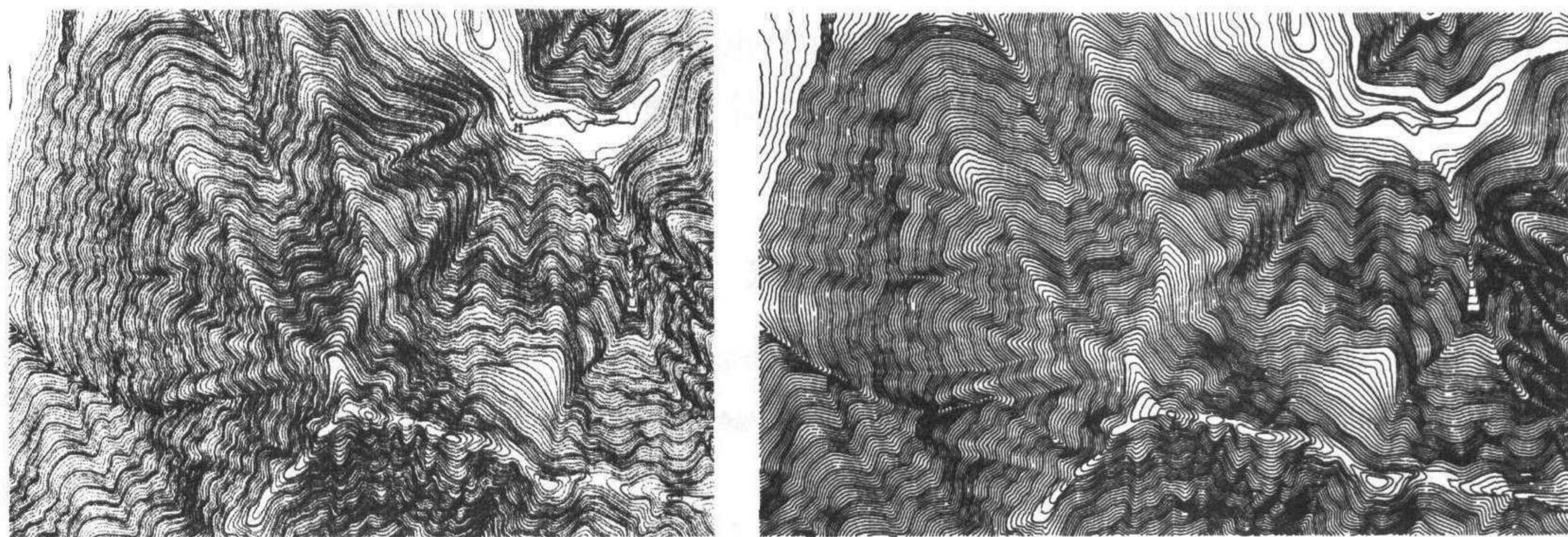
生成真实感图形需要将网格剖分成三角面, 并计算网格点的法向量. 网格剖分三角面的方法不好, 会影响真实感图形的生成质量. 实验表明, 对每个格子延着地形的走向剖分成三角面效果较好. 对于网格点的法向量, 一般是根据格子的三角面剖分方法, 将网格点周围所有三角面的法向量相加, 再规格化处理后得到的向量作为该网格点的法向量. 经过实验表明, 不采用传统的方法, 对每个网格点, 用 45° 和 135° 两个方向上的直线把网格点相邻的四

个格子分成 8 个三角面,将这 8 个三角面的法向量相加后进行规格化得到的向量作为网格点 A 的法向量,这种方法的效果较好,原因是这种方法使得 A 点周围的三角面数目最多,更加充分地反应了点 A 处的地形特征.

根据求出每个网格点的法向量和三角面剖分方法,即可用 OpenGL 原语对三维规则数据场直接进行三角面串处理,从而生成真实感图形.

8 实例

本系统已用 Visual C++6.0 编程实现,已经形成了良好的社会效益和经济效益^[16,17].图 3(a)是一幅实际像素地图中的一小块,图 3(b)是用本系统重建的矢量等高线图.图 4 是等高线内插的实例,图中的细等高线就是由图中粗等高线内插得到的,图中有实际做地图时会遇到的等高线局部高度弯曲、等高线起点、等高线的延伸方向、封闭等高线、马鞍型地形等问题.从图中可以看出内插的等高线质量很高.图 5~7 是本系统生成的晕渲图,已缩小(因版面所限,无法刊出大幅面晕渲地图),其中第一幅为山区地形;第二幅为黄土地貌;第三幅上边为山区地形、下边为平原地形.从图中可以看出晕渲图较好地反映了相应的地形特征.在带 64M 内存的 PII-350 计算机生成第一幅晕渲图仅需 0.4 分钟;生成第二幅晕渲图仅需 1.9 分钟;生成第三幅晕渲图仅需 2.7 分钟.经用户测试,在这种计算机上对 3 个全开(即 3



(a) 实际像素地图(一小块)

(b) 重建后的矢量等高线

图 3 等高线矢量化示例

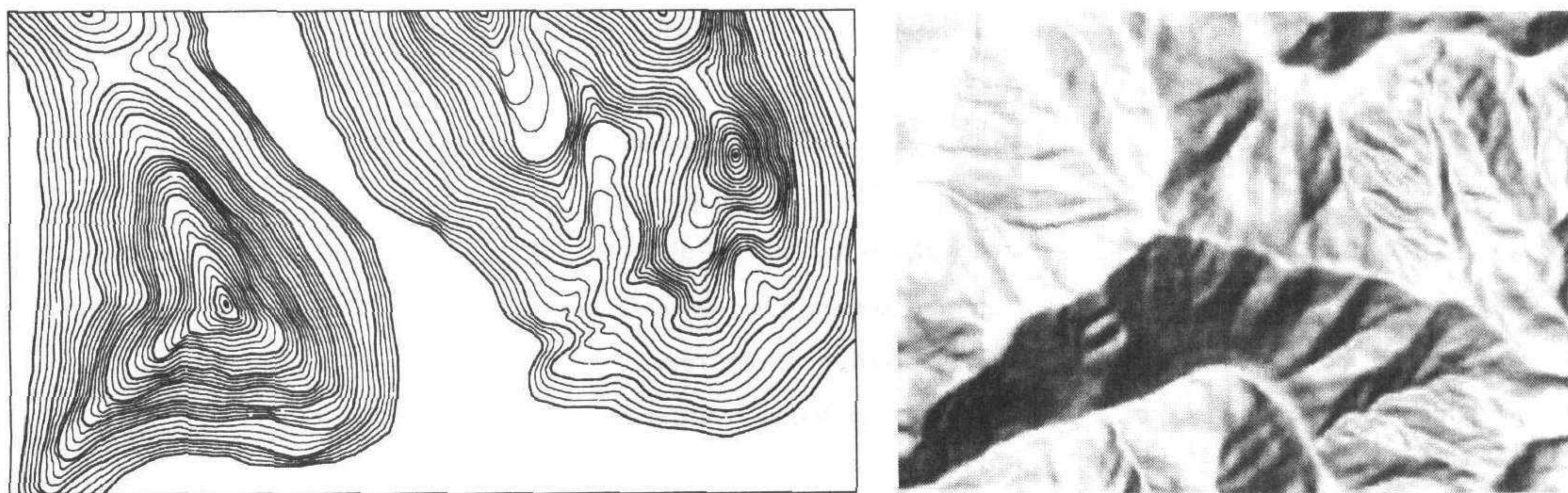


图 4 等高线内插实例

图 5 晕渲图示例 1

个 A0 幅面)的地图作晕渲,生成 300dpi 的图像,30 分钟内能完成.对有 316,796 个数据点的等高线地图建立三维规则数据场,本算法只需 4.2 分钟.

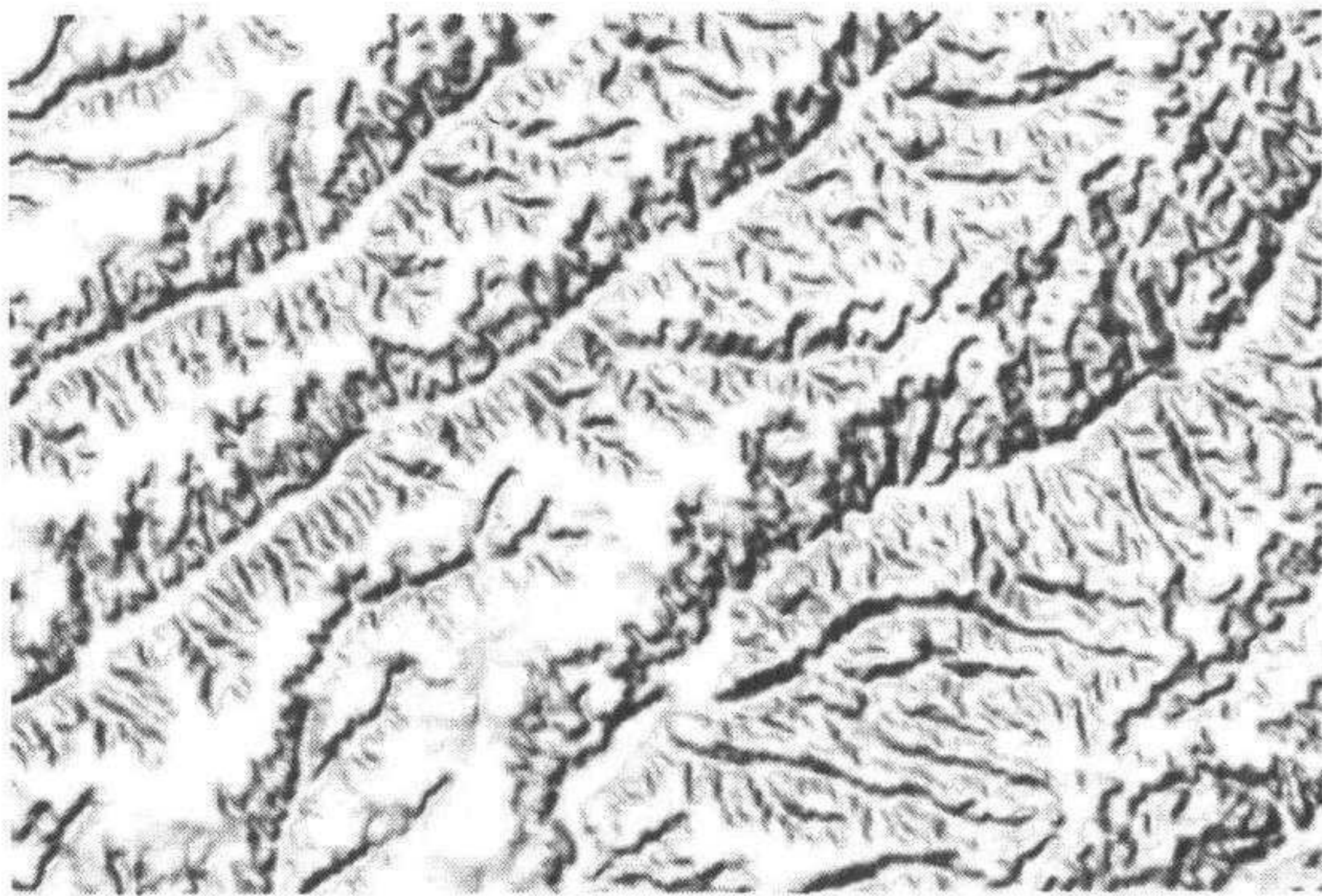


图 6 晕渲图示例 2

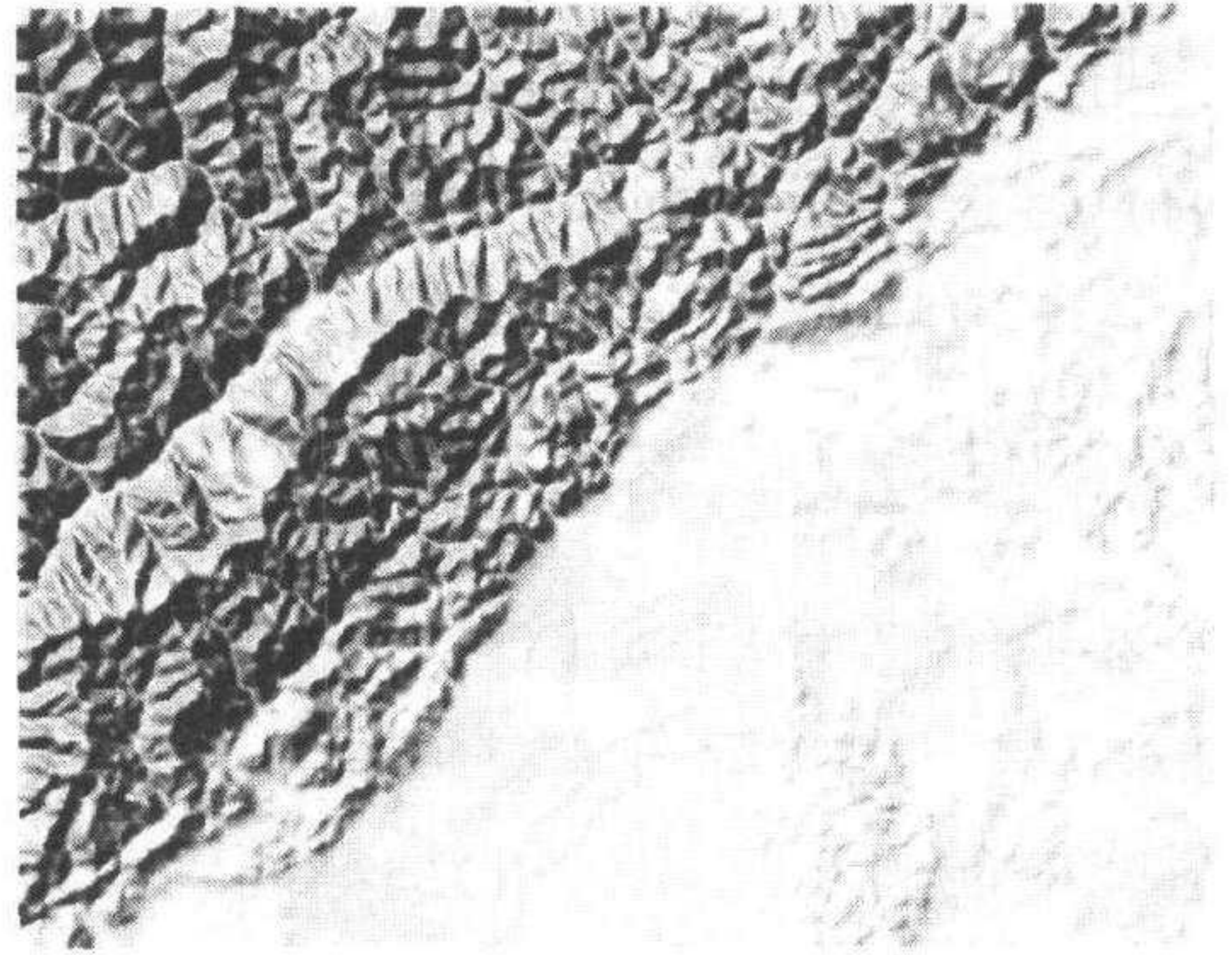


图 7 晕渲图示例 3

9 结论

本文完成了一个实用的可用于大幅面像素地图的三维地形重建系统.系统先对像素地图的等高线进行解释,从而获得具有高程值的矢量等高线,再生成三维规则数据场,然后生成表示地貌特征的三维真实感图像.算法充分利用等高线本身的规律(无规则的离散数据不具备这一特征);具有很快的运行速度;生成的立体图像逼真美观;能对大幅面的地图作地形重建.

参 考 文 献

- 1 张祖勋,张剑清.数字摄影测量学.北京:武汉测绘科技大学出版社,1996.12~65
- 2 Cong G, Parvin B. Surface reconstruction from sparse fringe contours. In: Proceedings of 4th IEEE Workshop on Applications of Computer Vision, 1998. 140~145
- 3 Talele S, Johnson T, Livadas P E. Surface reconstruction in parallel. In: Proceedings of the 4th IEEE Symposium on Parallel and Distributed Processing, 1992. 102~106
- 4 Johnstone J K, Sloan K R. Tensor product surfaces guided by minimal surface area triangulations. In: Proceedings of IEEE Conference on Visualization, 1995. 354~361
- 5 Chai J, Miyoshi T, Nakamae E. Contour interpolation and surface reconstruction of smooth terrain models. In: Proceedings of Visualization, 1998. 27~33
- 6 Eikvil L *et al.* Tools for interactive map conversion and vectorization. In: Proceedings of the 3rd International Conference on Document Analysis and Recognition, 1995. 927~930
- 7 Mecocci A, Lilla M. AI-based technique for tracking chains of discontinuous symbols and its application to the analysis of topographic maps. *Image and Signal Processing for Remote Sensing*, 1994, **2315**:680~691
- 8 Goodson K, Lewis P. A knowledge based line recognition system. *Pattern Recognition Letters*, 1990, **11**:295~304
- 9 Shimada *et al.* Agent-based parallel recognition method of contour lines. In: Proceedings of the 3rd Conference on Document Analysis and Recognition, 1995. 154~157
- 10 Dupont F *et al.* Automatic interpretation of scanned maps; Reconstruction of contour lines. *Lecture Notes in Computer Science*, 1997, **1389**:194~206

- 11 Dupont F, Deseilligny M P, Gondran M. Automatic interpretation of contour lines by using external data. In: Regina Spancer Sipple ed(s), Proceedings of 4th IEEE Workshop on Applications of Computer Vision. Los Alamitos California: IEEE Computer Society Press, 1998. 200~205
- 12 Dupont F, Deseilligny M P, Gondran M. DTM Extraction from topographic maps. In: Bub Werner ed(s), Proceedings of the Fifth International Conference on Document Analysis and Recognition. Los Alamitos California: IEEE Computer Society Press, 1999. 475~478
- 13 Yamamoto K, Yamada H, Muraki S. Symbol recognition and surface reconstruction from topographic map by parallel method. In: Proceedings of the 2nd International Conference on Document Analysis and Recognition. Los Alamitos California: IEEE Computer Society Press, 1993. 914~917
- 14 刘崴,董振宇. 地图数据矢量化方式采集的发展与实现. 见:中国地理信息系统协会年会论文集,深圳:1999. 451~455
- 15 许隆文. 计算机绘图. 北京:机械工业出版社,1989. 356~366
- 16 胡卫明,吴兵. 地图等高线实时内插算法. 计算机学报,2000,23(8):847~851
- 17 胡卫明,凌海滨. 大幅面地图的快速自动地形晕渲. 计算机学报,2000,23(8):852~856

胡卫明 博士、博士后、副研究员. 主要从事地图出版系统、IC-CAD 和计算机视觉等方面的研究工作.

吴兵 博士. 目前主要从事地图出版系统的开发工作.

刘崴 高级软件工程师. 目前主要从事地图出版系统的开发工作.



(上接第 26 页)

4. 中国大陆的作者请将稿件投寄到:

福建省厦门大学自动化系(361005)

李茂清教授

Tel:(0592)2185649

Fax:(0592)2182407

E-mail:mqli@xmu.edu.cn

其他地区和国家的作者请将稿件投寄到:

Professor Lihua Xie

School of EEE,BLK S2

Nanyang Technological University

Singapore 639798

Fax:(65)7920415

E-mail:elhxie@ntu.edu.sg

5. 特邀组织委员会和相关领域的专家报告,请与秦化淑教授联系:

秦化淑教授

北京中关村中国科学院系统科学研究所(100080)

E-mail:qin@iss03.iss.ac.cn

重要日期:

论文摘要和特邀专题提交的截止日期:2002年3月15日

论文录取通知时间:2002年4月15日

论文全文交稿时间:2002年5月15日

会议网址:<http://cies.xmu.edu.cn/icca>(中文网址)

<http://hdd.ece.nus.edu.sg/~icca/>(英文网址)