

# 带断层线的等值线生成方法<sup>\*</sup>

陈学工, 张 坤

(中南大学 信息科学与工程学院, 长沙 410083)

**摘 要:** 结合三角网的拓扑结构提出了一种新的分区算法。该算法首先把与断层线重合的三角形的边标志为边界边;然后根据组成区域边界的边类型,搜索出所有合理区域并插值端点和特殊点高程值;最后得到带断层线的等值线。用大量的数据进行了测试,实践证明,该算法能高效处理各种情况,实现简单。

**关键词:** 三角网; 断层线; 特殊点; 边界边; 非扩展三角形; 等值线追踪

中图分类号: TP391

文献标志码: A

文章编号: 1001-3695(2010)08-3144-03

doi:10.3969/j.issn.1001-3695.2010.08.90

## Algorithm of isoline tracing with fault lines

CHEN Xue-gong, ZHANG Kun

(School of Information Science & Engineering, Central South University, Changsha 410083, China)

**Abstract:** This paper proposed an algorithm combined with topological structure of triangulation. Firstly, the algorithm regarded edges coincided with fault lines as boundary edges. Secondly searched all reasonable regions, internal points and interpolated special points and endpoints of fault lines by judging the edge type consisting of region boundary. Lastly could get isoline with fault lines. This algorithm can deal with all kinds of special situation effectively and realize simply after testing with a lot of date.

**Key words:** triangulations; fault lines; special points; boundary edges; no-extended triangles; isoline tracing

## 0 引言

地质构造等值线图在石油勘探、煤矿和科研等领域中有着广泛的应用。由于断层的存在,绘制带断层线的等值线图要事先估计出断层线的端点和特殊点两侧的高程值,比绘制一般的等值线图困难,为此很多学者进行了大量研究。文献[1]通过确定断层线上点的圆形搜索域,找出插值点估计断层线上点两侧的高程,可以处理简单的断层线,当断层线复杂或点稀疏时,将搜索不到要插值的点或出现搜索域交叉。文献[2]提出了通过分区的方法估计断层线上点的高程值,但没有提出具体的分区方法。文献[3,4]中的算法成功实现了分区,但文献[3]中算法要求划分区域的线段至少与边界有一个交点,无法处理在边界内部的线段。文献[4]中算法由于断层线复杂使建树过程繁琐,且由于在每个端点处进行水平扫描建立梯形区域,划分太细,使断层线上某些点无法插值,分区不够理想。文献[3,4]在插值时还要判断点在区域多边形的内外,运算量大。基于以上分析,本文结合三角网的拓扑结构提出了一种新的分区算法来解决带断层线的等值线生成。算法首先通过最优凸壳技术<sup>[5]</sup>生成 Delaunay 三角网;然后嵌入断层线,生成约束 Delaunay 三角网;最后通过判断组成区域边的类型进行合理分区。在分区的同时,实时保存区域内部点,避免了判断点在区域多边形的内外,优化了分区。

## 1 相关定义

**定义 1** 对一个给定的平面点集 SP, 包含 SP 的最小凸多

边形称为 SP 的凸壳。最小凸多边形的边和与断层线重合的三角形的边称为边界边。

**定义 2** 断层线的端点即断层线的起点和终点,如图 1 中的点 A 和 F。

**定义 3** 断层线与三角形的交点称为特殊点,特殊点不包括断层线的端点。如图 1 所示,嵌入断层线 AF 后,交点 G、H、J 为特殊点。

**定义 4** 断层线延伸线穿过的三角形定义为非扩展三角形。如图 2 所示,嵌入断层线 DH 后,断层线延伸线穿过的三角形 HMJ、JMB、JBC、CBS,和 BSQ 被定义为非扩展三角形。

## 2 带断层线等值线生成方法

### 2.1 约束 Delaunay 三角化

读入离散点和断层线端点数据,采用文献[5]的基于最优凸壳技术的三角网生成方法建立三角网,并进行三角网优化使之成为 Delaunay 三角网;然后嵌入断层线生成约束 Delaunay 三角网。文献[6]采用对角线交换方法嵌入约束线,由于断层线的特殊性,如果采用该方法,虽然解决了断层线与三角形相交的情况,但生成的等值线将延伸不到断层线。为了提高精度,本文将断层线与三角形的交点看做插入点,局部更新与断层线相交的三角形。如图 1 所示,将交点 G 看做插入点,连接 G 所在边左右侧的三角形的对点即连接 GA、GD,形成四个三角形和八条边,维护新建立的三角形和边的拓扑结构。用同样的方法处理交点 H 和 J,处理后如图 3 所示。这样新生成的三角

收稿日期: 2010-01-04; 修回日期: 2010-02-08 基金项目: 国家“863”计划资助项目(2006AA06Z114)

作者简介: 陈学工(1965-),男,湖南长沙人,副教授,博士,主要研究方向为地理信息系统等;张坤(1983-),女,湖南长沙人,硕士,主要研究方向为地理信息系统(kunzhang686@163.com)。

形均位于断层线的两侧,估计出特殊点和端点的高程值后,可以得到延伸到断层线的等值线。

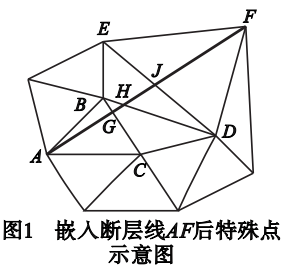


图1 嵌入断层线AF后特殊点示意图

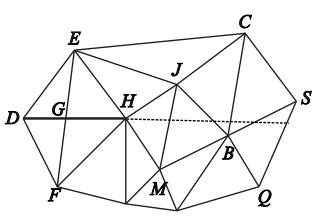


图2 非扩展三角形示意图

2.2 分区思想和算法描述

2.2.1 数据结构

本文的算法用到以下数据结构:  
a)有序点表。该表保存断层线端点、离散点集、排序后的点和断层线与三角形的交点,表中每个元素的结构为

```
typedef struct {
    double x,y,z,w; //点的坐标
    int LeftOrder; //点的区域值
    int RightOrder;
} Point;
```

b)有向边表。保存生成边的信息,表中每个元素的结构为

```
typedef struct {
    long triNo[2]; //边左右三角形序号
    long sOrder; //边的起点序号
    long eOrder; //边的终点序号
    bool bFlag; //边的边界边标志
    bool crossFlag; //断层线延伸线穿过的边标志
} Edge;
```

c)三角形表。保存生成的三角形信息。表中每个元素的结构为

```
typedef struct {
    long eNo[3]; //三角形三条边的序号
    bool used; //三角形的使用标志
    bool cross; //非扩展三角形标志
    int rCode; //三角形的区域标志
    int crossT; //断层线延长线穿过次数
} Triangle;
```

d)凸壳边表。保存点集生成的凸壳边在边集中的位置和顶点的序号,表中每个元素的结构为

```
typedef struct {
    long eNo;
    long sOrder;
    Long eOrder;
} Convex;
```

2.2.2 分区思想

嵌入断层线局部更新与断层线相交的三角形后,首先将凸壳边与断层线重合的三角形的边标志为边界边,同时保存每条断层线左右侧一个三角形到数组 ProArray;然后判断断层线的端点是否在三角网的边界边或另一条断层线(若断层线是折线段,所有折线段看做同一条断层线)。如果没有,则延伸断层线到边界边或另一条断层线,即将断层线延伸线穿过的三角形标志为非扩展三角形,同时标记延伸线穿过的边。如图 4 所示,嵌入断层线 DH,延伸 DH 到断层线 AC,即将 DH 延伸线穿过的三角形 HMJ、JMN 和 JNZ 标志为非扩展三角形,标记边 JM、JN 的属性 crossFlag 为 true。如果延伸过程中穿过的三角形已被标志为非扩展三角形,则停止延伸,同时将该非扩展三

角形的穿过值加 1,统计该三角形中断层线延伸线通过的条数。这样避免了断层线延伸线相交,从而实现合理分区。分区思想如下:

ProArray 数组保存每条断层线左右侧的一个三角形,RegionArray 保存每个区域的三角形。

a)从数组 ProArray 中取一个三角形 Tri,判断该三角形的使用属性。

b)如果为真,说明该三角形所在的区域已被搜索过,从 ProArray 中删除 Tri,重复 a)。

c)如果为假,保存三角形 Tri 到数组 RegionArray,置使用属性为真,同时赋予三角形 Tri 的三个顶点相同区域值;然后通过三角形 Tri 向外扩展。如果三角形 Tri 未标记为非扩展三角形,扩展方法为:遍历 Tri 的三条边,如果某一条边不是边界边,保存该边相邻的三角形 TriNext 到数组 RegionArray,标记三角形 TriNext 的顶点区域值与三角形 Tri 顶点的区域值相同;如果某一条边是边界边,则不扩展。如果三角形 Tri 标记为非扩展的三角形且穿过值为 1,根据相同区域的点在断层线同侧的性质判断三角形 Tri 的三个顶点中归属该区域的点,置相同的区域值,然后添加该三角形中与断层线延伸线相交的非边界边的相邻三角形到数组 RegionArray;如果穿过值大于 1,则不扩展该三角形。遍历完毕后,从数组 RegionArray 中删除 Tri,然后取 RegionArray 数组中的下个三角形 TriNext,用上述扩展方法扩展,直到 RegionArray 为空,说明没有扩展的三角形,至此搜索出一个区域,从数组 ProArray 中删除 Tri。

对数组 ProArray 中每个三角形重复上述 a)~c)操作,直到 ProArray 数组为空,这样就搜索出了所有区域。向数组 RegionArray 中添加三角形的过程中要避免重复添加同一个三角形。图 4 为分区示意图,嵌入断层线 AC、DH、RJ 后,首先保存三角形 DFG、DEG、ANM、ABN、RSE、RSC 到数组 RegionArray,延伸断层线 DH,标志三角形 HMJ、JMN、JNZ 为非扩展三角形,穿过值赋为 1;同时标记边 JM、JN 的 crossFlag 属性为 true。延伸断层线 RJ,置三角形 HJM 的穿过值为 2。延伸断层线 AC,标记三角形 RCW 为非扩展三角形,穿过值赋为 1。然后取三角形 DFG,置三个顶点 D、F、G 的区域值为 1,因 DFG 未标记为非扩展三角形,遍历三角形 DFG 的三条边,边 DF、DG 为边界边,不扩展三角形,边 GF 不是边界边,保存相邻三角形 GFH 到数组 RegionArray,置三角形 GFH 的顶点 H 区域值为 1,同时从数组中删除三角形 DFG。用同样的方法扩展三角形 GHF,直到扩展到三角形 HMJ 和 MNJ。因 HMJ 是非扩展三角形,首先赋值此三角形在该区域中的点,然后判断 HMJ 的穿过值是否大于 1,因其穿过值为 2,所以不进行扩展。从数组中删除三角形 HMJ。同理 MNJ 也为非扩展三角形,首先赋值此三角形中在该区域中的点,因穿过值为 1,保存边 JN 相邻的三角形 JNZ 到数组 RegionArray,然后从数组中删除三角形 JMN。扩展到三角形 JNZ 时,因与断层线相交的边 NZ 为边界边,则停止扩展,至此搜索出一个区域,如图 4 中的区域 DZAF。用上述方法扩展三角形 DEG、ANM、ABN、RSE、RSC,可以搜索出所有区域和各个区域内部所有的点。

2.2.3 估计高程

特殊点有两个性质:a)特殊点作为三角形的顶点,在等值线追踪时,其上的值必须是已知的,而它们又不是真正的数据点,因此这些值必须事先估计出来;b)特殊点上的值有两个,

它们的差就是断层在此处的落差。

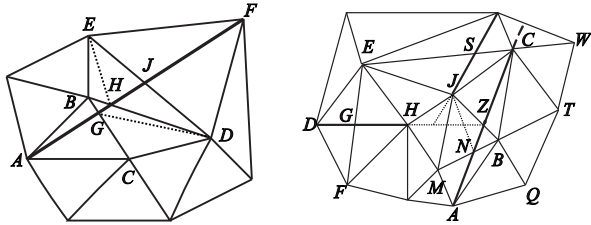


图3 嵌入断层线AF后局部更新

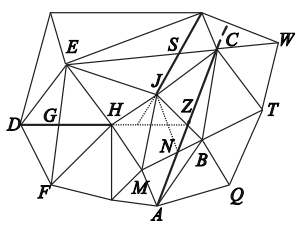


图4 分区示意图(AC、DH和RJ为嵌入的断层线)

通过上面的分区算法搜索出所有区域和内部点后,对断层线的特殊点和端点采用距离反比插值法进行估计。对于断层线的端点,寻找距该点最近的三个数据点插值,对于特殊点左右侧的高程值寻找距该点最近的且区域值相同的三个数据点插值。由于不同区域点的高程值不同,插值后特殊点左右侧的高程值也不同,从而生成带落差的等值线。在实际应用中,可以根据需要适当增加插值点的个数。

2.3 主要算法步骤

假设未延伸的断层线都已延伸到边界边或另一条断层线,并保存每条断层线左右侧一个三角形到数组 ProArray,设区域序号 rCode 初始值为 0。

分区算法主要步骤如下:

- a)取 ProArray 数组中一个三角形 Tri,判断该三角形的使用属性 used,如果为 true,移除该三角形,转向步骤 f);否则执行步骤 b)。
- b)将 Tri 保存到数组 RegionArray 中,置 used 属性为 true,区域序号 ( rCode) 加 1,将该三角形中顶点的区域值赋值为 rCode。
- c)判断三角形 Tri 是否标记为非扩展三角形。如果否,遍历 Tri 的三条边,如果为非边界边,将该边相邻的三角形保存到数组 RegionArray,置相邻三角形中顶点的区域值为 rCode,如果为边界边不作处理,遍历完毕从数组 RegionArray 中删除该三角形;否则,转向步骤 d)。
- d)判断该扩展三角形的穿过值是否大于 1。如果否,根据相同区域的点在断层线同侧的性质,置该三角形在同一区域的顶点区域号为 rCode,然后将三角形 Tri 中与断层线相交的非边界边的相邻三角形保存到数组 RegionArray,同时将三角形 Tri 从 RegionArray 数组中移除;否则不扩展。
- e)判断 RegionArray 数组是否为空。如果否,取 RegionArray 中的一个三角形,转向步骤 c);如果是,从数组 ProArray 中删除三角形 Tri。
- f)判断 ProArray 数组是否为空。如果是转向步骤 g);否则转向步骤 a)。
- g)结束。

2.4 追踪算法

计算等值线时应特别注意那些有一条边在断层线上的三角形,当等值线到达断层线时,应立即停止,而不是穿过断层线。为了达到这个目的,将这些三角形作为边界三角形处理,其在断层线上的一条边作为三角网的边界即可。估计出断层线端点和特殊点的高程值后,对每条断层线,追踪算法如下:

- a)单元遍历。对所有三角形进行遍历,找出存在等值点的三角形,保存在 m\_TriangleTemp 数组。

- b)线头搜索。扫描凸壳边表,判断凸壳边表中的边是否有等值点通过,如果有则找到线头(采用凸壳技术建立三角网时将生成的边界边保存在凸壳边表),插值得到等值点的位置,记下等值点所在的边和该边相邻的三角形。

- c)等值线追踪。在线头的搜索过程中,同时记录了等值点通过的当前边及该边相邻的三角形,在其相邻的三角形中查找下一个等值点所在的边和该边相邻的三角形,更新当前边和当前三角形,直到下一个等值点所在的边没有相邻的三角形,并记录所有等值点。在追踪的过程中,必须将每次找到的存在等值点的三角形从 m\_TriangleTemp 数组中删除,否则等值点会重复出现。

以上是开曲线等值线的追踪过程。开曲线追踪完毕后,如果 m\_TriangleTemp 数组中还有三角形,则存在闭合等值线。扫描 m\_TriangleTemp 数组,按同样的方法追踪闭合等值线。如果在追踪过程中遇到三角形中某一条边为边界边,则从该等值线第一个等值点所在的边开始,换个方向继续搜索,直到再次遇到边界边,至此等值线追踪完毕;否则一直追踪,直到追踪到同一等值点,从而搜索出所有闭合等值线。

3 算法实现实例

本文实例是在 Visula Studio. NET 下,结合 OpenGL 显示功能,实现了绘制带断层线的等值线程序。实践证明效果良好,图 5 为不带断层线的等值线追踪效果图,图 6 为带断层线的等值线追踪效果图,图中小黑点为离散点。



图5 不带断层线等值线追踪图



图6 带断层线等值线追踪图

本文算法中分区的时间复杂度几乎是线性的,在断层线的条数为 10、等值线级别为 10 的情况下,对以下几组离散数据进行了测试,时间对比结果如表 1 所示。

表 1 本文算法中区域划分和等值线生成时间		
离散点数据	区域划分时间/s	等值线生成时间/s
1 609	0.4	0.98
5 670	1.21	2.5
11 297	2.48	4.7

4 结束语

虽然目前存在一些处理带断层线的等值线生成的方法和一些常用的分区算法,但是这些算法没有与三角网的拓扑结构结合来应用,并且由于断层线的复杂性,常用算法难以实现合理分区,无法很好地解决绘制带断层线的等值线图的困难。基于此,本文结合三角网提出了一种分区算法,使用该方法可以实现合理分区,解决了绘制困难。

参考文献:

[1] 徐刚.带断层的地层空间插值算法研究[J].工程技术,2009,(12):91-92.  
[2] 王家华.克里金地质绘图技术[M].北京:石油工业出版社,1999:268-300.  
(上接第 3164 页)



3 实验结果及分析

算法使用 VC++6.0 和 Intel 开源视觉库 OpenCV 实现。OpenCV 由一系列 C 函数和少量 C++ 类构成了图像处理和计算机视觉方面的很多通用算法。OpenCV 提供了 Camshift 函数和卡尔曼滤波器的数据结构 CvKalman 与函数 CreateKalman、KalmanPredict、KalmanCorrect 和 ReleaseKalman 来完成卡尔曼滤波。

本文利用 OpenCV 实现了对运动目标的实时跟踪。其中实验平台是 Intel P4-2.0 GHz CPU 的微机,操作系统为 Windows XP,彩色运动图像是采用极速 USB 摄像头在室内拍摄的。实验首先在 Visual C++6.0 环境下安装和配置 OpenCV 库,需要配置环境变量和完成一些库文件的设置;然后利用函数 CvCreateCameraCapture 来实现摄像头数据的采集,逐帧读取视频并进行分析;最后通过调用 Camshift 算法和卡尔曼滤波器实现。实验将 Camshift 算法和改进的算法进行了对比,通过模拟半遮挡、全遮挡以及快速运动等状态,来实现运动物体的跟踪,以对比两个算法之间的优劣。实验以瓶盖为运动物体,白色的挡板为遮挡物来进行模拟实验。图 5~12 显示了在遮挡情况下运动物体的跟踪效果。

1) Camshift 算法



图5 正确跟踪 图6 半遮挡 图7 全遮挡 图8 直方图

2) 改进算法



图9 正确跟踪 图10 半遮挡 图11 穿越障碍 图12 正确穿越

在上列图示中,本文截取图 5~7(分别代表第 4、15、31 帧图像),由实验结果可以看出,传统的 Camshift 算法在复杂背景下,难以解决遮挡等问题,通过加入卡尔曼预测后,如果发生遮挡等问题,卡尔曼滤波能够根据其质心的测量和预测方程实现对下一运动物体跟踪框的质心和大小的自动调整,较好地解决了快速遮挡等问题。当发生全遮挡的时候,传统的 Camshift 算法会存在跟踪丢失等现象,然而通过改进预测,能够正确识别出丢失物体。

3) 跟踪误识率分析

针对实际视频监控场景,实验设置遮挡和复杂背景,观测

跟踪丢失的效果。通过设置 100 次情形实现遮挡、快速运动等场景实现。表 1 显示了 Camshift 算法和改进算法的误识率。

表 1 算法误识率

算法比较	正确率/%	误识率/%
Camshift 算法	78	22
改进算法	86	14

4) 实时性分析

对于算法的实时性分析,实验通过 640×480 的 USB 摄像头采集图像数据。为了能够实时显示捕捉的图像,程序中采用从首帧 0 开始抓取视频文件的标号,计算两帧之间的时间,抓取两帧之间的图像进而显示。程序通过函数“cvShowImage(“TRACK”,img);key=cvWaitKey(10);”来显示运动跟踪的图像,速率可以达到 10 fps,能较好地满足实时跟踪的效果。

4 结束语

本文提出一种改进的结合 Camshift 算法和卡尔曼滤波的运动物体实时跟踪方法,在 Camshift 算法的基础上,通过二次搜索和卡尔曼滤波来保障运动物体跟踪的可靠性和实时性。实验表明该方法具有较好的实时性和鲁棒性,在复杂背景以及遮挡等情况下也能够实现较好的跟踪效果。然而本文算法在针对时间较长的遮挡问题时,预测也存在跟踪丢失等现象,有关这方面的研究仍需要进一步进行。

参考文献:

[1] ANDERSON C, BURT V. Change detection and tracking using pyramid transformation technique [C]//Proc of SPIE-intelligent Robots and Computer Vision, 1985:72-78.

[2] WILDES R P, ASMUTH J C, GREEN G L. A system for automated iris recognition [C]//Proc of the 2nd IEEE Workshop on Application of Computer Vision, 1994.

[3] LAI J, GAUCH J, CRISMAN J D. Computing optical flow in color image sequence [J]. Innovation and Technology in Biology and Medicine, 1994, 15(1):76-87.

[4] BASCLE B, DERICHE R. Region tracking through image sequences [C]//Proc of IEEE International Conference on Computer Version, 1995:302-307.

[5] KASS M, WITKIN A, TERZOPOULOS D. Snakes: active contour models [J]. International Journal of Computer Vision, 1988, 1(4):321-331.

[6] BOYLE M. The effects of capture conditions on the CAMSHIFT face tracker, Report 2001- 691- 14 [R]. Alberta, Canada: Department of Computer Science, University of Calgary, 2001.

[7] KALMAN R E, BUCY R S. New methods and results in linear filtering and prediction theory [J]. Trans of ASME, Journal of Basic Engineering, 1961, 83:95-108.

(上接第 3146 页)

[3] 王浩,姚长利,严红平,等. 平面区域几何划分的拓扑算法研究 [J]. 计算机应用与软件, 2008, 25(12):12-14.

[4] 奥罗克. 计算几何 C 语言描述 [M]. 2 版. 北京:机械工业出版社, 2005:288-292.

[5] 陈学工,黄晶晶. 基于最优凸壳技术的 Delaunay 三角剖分 [J]. 计算机工程, 2007, 33(17):93-95.

[6] 李立新,谭建荣. 约束 Delaunay 三角剖分中强行嵌入约束边的多对角线交换算法 [J]. 计算机学报, 1999, 22(10):1114-1119.

[7] 董玉杰,颜伟达,马远乐. 油藏数值模拟中复杂等值线图的绘制方法研究 [J]. 计算机仿真, 2007, 24(7):220-224.

[8] 章孝灿,黄智才,戴企成,等. GIS 中基于拓扑结构和凸壳技术的快速 TIN 生成算法 [J]. 计算机学报, 2002, 25(11):1212-1218.

[9] 孟亮,方金云,贾文涛. 二维 Delaunay 三角网的动态更新 [J]. 计算机应用, 2007, 27(B06):143-145.

[10] 贾晓林,吴立新,王彦兵. 二维 Delaunay 三角网局部更新:点插入与点删除 [J]. 地理与地理信息科学, 2004, 20(5):28-31.

[11] 赵伟,赵卓宁,李五生. 一种有效的离散数据场等值线生成方法 [J]. 成都信息工程学院学报, 2007, 22(1):116-121.

[12] 陈敬周,张和生,王双美. 离散点集等值线生成方法的实现 [J]. 科技情报开发与经济, 2007, 17(12):192-193.