研究报告

等值线生成与图形填充算法①

孙桂茹[1], 马 亮[1], 路登平[2], 赵国瑞[3], 郝嘉林[4]

(1. 南开大学计算机与系统科学系, 天津 300071; 2. 天津大学材料工程学院, 天津 300072;
3. 天津大学电子信息工程学院, 天津 300072; 4 天津大学计算机信息网络中心, 天津 300072)

摘 要: 等值线图是一种应用广泛的图形 针对已有等值线生成方法进行了改进, 提出了一种简单实用的等值线生成方法 并对等值线图的填充问题进行了研究, 提出了一个易于实现的等值线图填充算法

关键词: 等值线; 填充; 计算机图形学

中图分类号: TP391. 41　　**文献标识码**: A　　**文章编号**: 0493-2137(2000)06-0816-03

等值线图是一种重要图形, 它广泛应用于各种不同的领域 等值线图的绘制过程一般分为离散数据网格化、等值线生成及等值线图填充三个步骤

传统的利用网格点数据绘制等值线的方法有两种[1]. 一是直接在网格边上做线性插值得到等值点, 然后再按一定的方位判别法连接各等值点得到等值线 这种方法实现简单, 但网格较大时等值线以折线的形式出现, 而且不易进行图形填充运算 第二种方法是利用已有的网格点数据再对每个网格拟合一个曲面函数, 然后将网格细分为若干单元, 根据曲面函数的值逐网格逐单元地追踪等值线 这种方法虽然可以得到连续光滑的等值线, 但实现起来极为复杂 本文将在综合上述方法的基础之上提出一种简明易行的等值线生成算法

等值线图的填充就是用不同的颜色填充两条等值线之间的区域, 以便对区域的整体情况进行把握 现有的文献很少提及 因此, 本文所介绍的算法具有较高的参考价值

1 等值线生成算法

算法基本思想是从绘图区域边界开始, 利用线性插值逐网格追踪每条等值线, 得到各条等值线在其所穿过的网格边上等值点的坐标, 对这些等值点进行曲线平滑处理, 将结果(新的坐标点)存放在一个数组或链表中, 连接这些坐标点即可绘出连续光滑的等值线

1. 1 等值线追踪原理

等值线可分为从边界出发到边界结束的等值线和内部封闭的等值线两种情况 追踪法的原理是首先从绘图区域边界或内部网格的边上求得一个等值点(等值线与网格边的交点), 然后由该点出发, 判断下一个等值点的坐标, 直到下一个等值点落在绘图区域边界上或与起点重合, 则对该条等值线的追踪就算完成了.

由一个等值点追踪下一个等值点, 实际上是一个求网格内等值线连接的问题 网格内等值线的连接方法有八种, 如图 1 所示

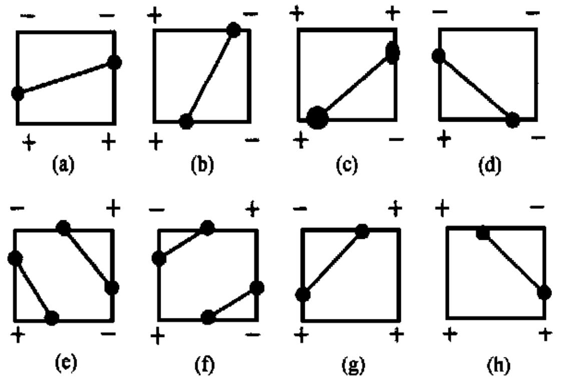

图 1　网格内等值线连接方法

Fig. 1　Isoline connections in meshes

图中+ 号表示网格节点的值大于等值线的值, - 号表示网格节点的值小于等值线的值 在已知网格边上一个等值点的前提下, 可以根据上面的八种情况得

① **收稿日期**: 1999-07-02; **修回日期**: 2000-03-02
作者简介: 孙桂茹(1945-), 女, 副教授

到另一个等值点的坐标

1.2　等值线平滑方法

曲线平滑的方法有 T-N 方法、Bezier 方法、B 样条方法、三次样条方法以及最小二乘等多种[4], 实践证明, 二次B 样条曲线拟合是一种好方法, 对弯曲程度很大的折线及封闭折线都可以得到很好的平滑效果

B 样条曲线方法对折线的平滑处理, 主要是对每相邻的 $k+1$ 个点都构造一个 k 次B 样条函数, 称为基函数, 由这些函数进行局部逼近, 得到若干条平滑的B 样条曲线, 并且保证相邻曲线段之间的连续性 定义B 样条曲线是先假定基函数的构造形式, 然后根据连续条件等确定基函数的具体结构, 并用待定系数法求出基函数中各系数

设折线上各点为 $P_0, P_1, \cdots, P_n$, 可以从 P_0 出发对每连续的 $k+1$ 个点构造一个 k 次B 样条曲线, 这样的曲线段共有 $Q_0(t), Q_1(t), \cdots, Q_{n-k}(t)$ 条, 其中, 第 i 条曲线段的定义为

$$Q_i(t) = \sum_{j=i}^{i+k} B_{k,j}(t) P_j \quad 0 \leq t \leq 1$$

其中, $B_{k,j}(t)$ 为对应某个顶点 P_j 的基函数, 一共有 $k+1$ 个基函数 令每个基函数为关于 t 的 r 次多项式, 即 $B_{k,j}(t) = \sum_{j=0}^{r} a_{jr} t^r, 0 \leq t \leq 1$, 这样一共可以得到 $(k+1) \times (r+1)$ 个待定系数 这些待定系数可以利用施加于该曲线的约束条件来确定[3]. 本文仅给出 $k = 2$, 即二次B 样条曲线的具体表达式为

$$Q(t) = \frac{1}{2}(t-1)^2 P_0 + \frac{1}{2}(-2t^2 + 2t + 1) P_1 + \frac{1}{2} t^2 P_2 = \frac{1}{2}[t^2 \quad t \quad 1] \begin{bmatrix} 1 & -2 & 1 \\ -1 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \end{bmatrix} \quad 0 \leq t \leq 1$$

当 $t=0$ 时, $Q(0) = (P_0 + P_1)/2$, $t=1$ 时, $Q(1) = (P_1 + P_2)/2$, 曲线段的两端是每条折线的中点, 如图 2a 所示, 平滑效果见图 2b.

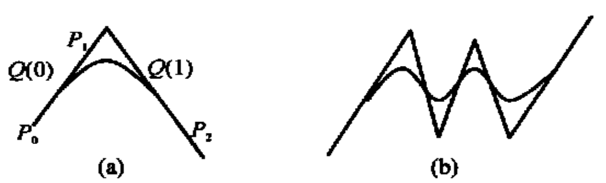

图 2　二次 B 样条曲线

Fig. 2　Quadratic B-spline curve

对每条等值线中所有相邻的三点都利用上述公式拟合出若干个新的数据点, 用新的数据点取代旧的数据点 t 值的递增量可以根据网格的大小来取, 网格大时可取得小一些, 比如 0.1、0.2, 网格小时可取得大一些, 比如 0.25 和 0.5.

1.3　等值线生成算法描述

为了实现算法, 需要定义一种适当的数据结构存放等值线 可以用动态数组存放所有等值线, 它按照追踪的顺序存放各条等值线上所有等值点的坐标 也可以用双向链表实现同样的功能 VC 提供了模板类 CArray 可用于实现动态数组, 等值线数据结构可定义为:

```
struct InterPoint {float x, y; };
struct AIsoline {float z; //等值线的值
CArray< InterPoint, &InterPoint> PointSet; };
CArray< AIsoline, &AIsoline> IsolineSet;
```

依据 1.1 和 1.2 所述的原理, 下面给出一个等值线生成算法的轮廓性描述

```
MakeIsoline(){
    for(依次按区域左边界、上边界、右边界、下边界)
    for(对于每条边界上所有网格)
     if(存在等值点 P 且 P 不在 IsolineSet 中)
     TraceIsoline(P);
    for(不包括区域边界的所有网格的边)
    //生成内部封闭等值线
     if (存在等值点 P 且 P 不在 IsolineSet 中)
       TraceIsoline(P);
    for(IsolineSet 中的每条等值线) 利用二次B 样条拟合法进行光滑处理; }
```

函数 TraceIsoline 追踪一条等值线, 其描述为

```
TraceIsoline(InterPoint P){
 IsolineSet 增加一个元素, 将 P 插入新增元素的 PointSet 末尾;
 do{依图 1 所示的判别方法得到下一个等值点 Q;
 将Q 插入新增元素的 PointSet 末尾; }
 while(Q 不在区域边界上 &Q 不等于 P); }
```

2　等值线填充算法

一般的开发工具中都提供对区域的填充函数 例如Windows API 中就提供了函数 FillRgn, 可以对任意多边形进行填充 因此, 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, 以及确定用何种亮度进行填充的问题 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 通过对各种等值线进行观察可以发现: 任意一条等值线, 不外乎属于图 3 所示的四种情况之一:

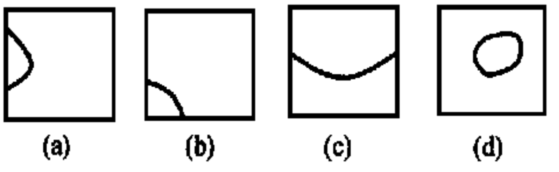

图 3　等值线种类

Fig. 3　Kinds of Isoline

基于任意两条等值线都不相交的前提, 可得到前三种等值线与网格边界所围区域(简称为 a 区、b 区、c 区) 的覆盖关系为: a 区内部决不会出现 b 区, b 区内部决不会出现 c 区 因此, 在填充时只要依据一定的顺序依次填充第三、第二、第一种等值线与网格边界所围的区域, 然后按照由外层向内层的顺序填充第四种等值线所围的区域就可以完成对整个区域的填充

为了实现这一算法, 需要对保存在 IsolineSet 中的等值线按照其起点的位置进行排序 排序的顺序与等值线追踪的顺序一致, 即区域左边界、上边界、右边界、下边界以及内部等值线 内部等值线按照横坐标递增排序 这样, 排好序的等值线按照起点坐标左、上、右、下、内部的顺序存放在 IsolineSet 中 整个算法的基本描述如下:

1) 按起点纵坐标从下至上的顺序对起点在左边界上的等值线排序; 2) 按起点横坐标从左至右的顺序对起点在上边界上的等值线排序; 3) 按起点纵坐标从上至下的顺序对起点在右边界上的等值线排序; 4) 按起点横坐标从右至左的顺序对起点在下边界上的等值线排序; 5) 按起点横坐标从左至右的顺序对内部封闭的等值线排序; 6) 填充第三种等值线与网格下边界或左边界以及起点和终点所在的边界所围的区域 对最后一条等值线, 则还需填充与网格上边界或右边界所围的区域; 7) 填充第二种等值线与起点和终点所在的边界以及这二边界相交的顶点所围的区域; 8) 填充第一种等值线与起点和终点所在的边界的顶点所围的区域; 9) 填充内部封闭等值线所围的区域

3　结　语

经实践证明, 与其它方法比较, 本文提出的等值线生成算法简便易行, 能够快速得到连续光滑的等值线图 填充算法易于实现, 能够有效地解决绝大多数等值线图的填充问题

参考文献:

[1]　李鸿吉, 张菊明. 电子计算机制图方法及应用[M]. 北京: 地质出版社, 1981.

[2]　罗笑南, 王若梅. 计算机图形学[M]. 广州: 中山大学出版社, 1996.

[3]　沈亚飞, 龚卫国. 等值线生成算法研究[J]. 计算机工程, 1994, 2: 11～12.

INVESTIGATION ON THE ALGORITHM OF MAKING AND FILLING ISOLINE

SUN Gui-ru[1], MA Liang[1], LU Deng-ping[2],
ZHAO Guo-rui[3], HAO Jia-lin[4]

(1. Department of computer and system science, Nankai University, Tianjin 300071; China
2. School of Materials Science and Engineering, Tianjin University, Tianjin 300072, China;
3. School of Electronic Information Engineering, Tianjin University, Tianjin 300072, China;
4. Computer Information & network center, Tianjin University, Tianjin 30072, China)

Abstract: An isoline is a figure which has been extensively used in engineering. In this paper we improve on the existing isoline making methods and put forward a method which can be implemented more easily. We also advance a simple and effective algorithm to fill an isoline.

Keywords: Isoline; fill; computer graphics