

doi: 10.3969/j.issn.1006-1576.2011.04.027

基于 VC++ 的等值线追踪与填充算法

郭新奇, 严建钢, 杨士锋, 沈培志

(海军航空工程学院 指挥系, 山东 烟台 264001)

摘要: 针对传统绘制等值线算法用于追踪生成及颜色填充等值线图效率不高的问题, 对传统网格追踪及区域填充算法进行改进, 提出优化的追踪算法及基于边界点追踪的区域填充算法。通过描述等值线的数据结构、选取等值点实现了优化的追踪算法。等值线区域填充算法则针对闭合区域和连通区域采用不同的方法填充。并运用 C++ 编程语言在 VC++6.0 集成开发环境实现了算法模型。实例结果表明, 该追踪算法及填充算法能有效地解决等值线生成及填充问题, 且具有算法简单、易于实现的特点。

关键词: 等值线; 追踪; 填充

中图分类号: TP301.6 **文献标志码:** A

Algorithm of Tracing and Filling Isoline Based on VC++

Guo Xinqi, Yan Jiangang, Yang Shifeng, Shen Peizhi

(Dept. of Command, Naval Aeronautical & Astronautical University, Yantai 264001, China)

Abstract: Aiming at the ineffective problem of traditional algorithm which traces, creates and fills the isoline with color. By improving the traditional grid tracing and area filling color algorithm, put forward the optimal tracing algorithm and area filling color algorithm based on boundary point. Through describing isoline data structure, and selecting isoline point, the optimal tracing algorithm is realized. The isoline area filling algorithm uses different method to fill the close area and linkage area. Realize algorithm model by using C++ program language in VC++6.0 integrated developing environment. Experiment results show that this method can effectively resolve isoline creation and filling problem. And the method is simple and easy to realize.

Keyword: isoline; tracing; filling

0 引言

等值线图是一种重要图形, 是现代科学研究中一种常用和有效的重要表示方法, 它广泛应用于地理科学、气象分析、辅助决策和计算等领域, 如地理信息系统 (GIS) 中的地形等高线图、温度场中的等温线图、有限元分析过程中等效应力应变场等值线图等等^[1]。

当前, 绘制等值线算法主要有网格法追踪法^[2]、区间树索引提取法^[3-4]、预测跟踪法^[5]等。通过一定的算法绘制出等值线, 将数据图形化, 有利于工作人员对数据作进一步的分析。如果能把不同等值线围成的不同区域进行颜色上的区分填充, 将使等值线图的数值变化看起来更加直观。

等值线填充, 即在等值线图中所有等值线形成的, 不同区域中用不同的颜色加以填充, 形成彩色等值线图。在图形学有许多不同的算法能进行区域填充, 比如种子法、栅格法, 还有人提出过曲线积分算法等, 但这些算法都没有充分利用等值线绘制过程的一些特点。

笔者结合等值线的一些特点, 在传统网格追踪算法基础上, 对网格遍历原则、等高线走向等问题进行了优化。在区域填充方面, 根据等值线生成特点, 提出了基于边界点追踪的区域填充算法, 并在 VC++ 6.0 集成开发环境中使用 C++ 语言予以实现。

1 等值线追踪算法

1.1 等值线追踪算法原理

算法的基本原理是: 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, 将坐标结果存放在动态数组中, 连接这些坐标点即可绘出连续光滑的等值线。

第 1 步是从绘图区域边界或内部网格的边上求得一个值为 Value 的等值点 (等值线与网格边的交点), 把该点作为该条等值线的起点; 第 2 步是由该点出发, 判断下一个等值点的坐标, 直至下一个等值点落在绘图区域边界上或与起点重合, 依据一定的法则将这些点连接成一条等值线。

收稿日期: 2010-12-02; 修回日期: 2011-01-02

基金项目: 航空科学基金“航空反潜火控技术研究”(20095184005)

作者简介: 郭新奇 (1975—), 男, 山东人, 博士, 讲师, 从事海军兵种战术学数理分析、军事运筹研究。

1.2 等值线追踪算法描述

1) 等值线数据结构

为了实现算法, 需要定义一种适当的数据结构存放等值线。笔者利用 VC 提供的模板类 CArray 及 CTypedPtrList 来定义按照追踪顺序存放各条等值线上所有等值点的坐标, 其数据结构定义如下:

```

struct CContourPoint
{
float x;      //x: 横坐标
float y;      //y: 纵坐标
int iBorder; //1: 下边界 2: 右边界 3: 上边界 4: 左边界 5: 封闭
};

```

};

```

一条等值线曲线的点集的数据结构可定义为:
CArray< CContourPoint, CContourPoint &>CCurve;
//一条等值线曲线的点集的坐标

```

值为 ValueHight 所有等值线集的数据结构可定义为:

```

CTypedPtrList<CPtrList, CCurve*> CCurveList;
//等值线曲线链表(给定某值的等值线不只一条)

```

2) 前进方向的确定

由于各等值点均位于网格边上, 所以等值线通过相邻网格的走向有自下而上、自左而右、自上而下、自右而左 4 种可能, 如图 1。

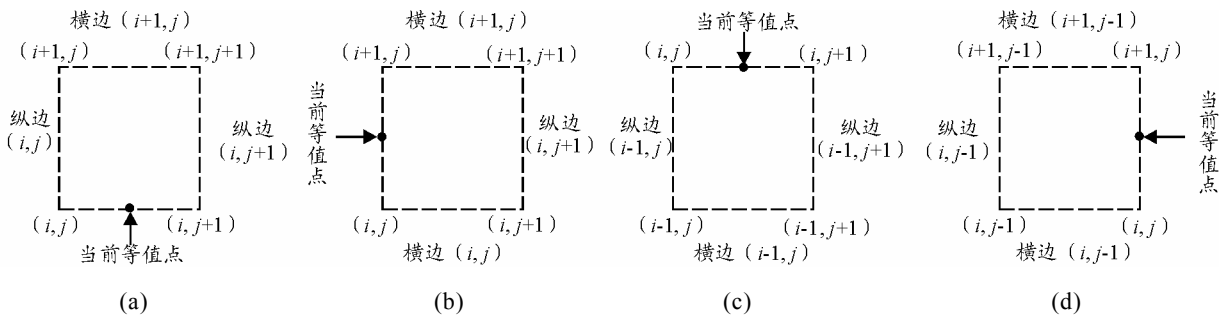


图 1 追踪前进走向图

设前一点为 PP、当前点为 CP 和下一点为 NP, 等值线追踪前进方向判定如下:

```

if(CP.行号 > PP.行号 )
{
前进方向: 下---->上
}
else if( CP.列号 > PP.列号 )
{
前进方向: 左---->右
}
else if( CP 在横边上 )
{
前进方向: 上---->下
}
else
{
前进方向: 右---->左
}

```

3) 等值点的选取方法

以自下向上追踪为例, 给出等值点的选取方法。当前等值点 CP 在网格单元的底边上, 则下一等值点 NP 所在的位置有 3 种情况: ① 在纵边(i,j)上; ② 在纵边(i,j+1)上; ③ 在横边(i+1,j)上。但实际追

踪时只能选择其中之一。

假设纵边(i,j)上存在等值点 NP1, 纵边(i,j+1)上存在等值点 NP3, 横边(i+1,j)上存在等值点 NP2, 选择的次序如下:

- ① 当 NP1、NP3 都存在时, 选择靠近网格底边者为 NP;
- ② 若 NP1、NP3 靠近底边的距离相同, 则选择与 CP 点距离近者为 NP;
- ③ 当 NP1、NP3 中只有一个存在时, 则存在点即为 NP;
- ④ 当无 NP1、NP3 存在时, 对边必定存在 NP2 作为 NP。

同理, 可得出自左向右、自上而下、自右而左时等值点的选取方法。

2 等值线区域填充算法

2.1 等值线的一些性质

- 1) 不同高程值的等值线必不相交^[6];
- 2) 所有等值线或者是本身形成闭合区域, 或者与其它等值线以及边界形成连通区域^[6];
- 3) 对于任一连通区域来说, 围成它的所有等值线的等值线值最多可以取 2 个^[6]。

2.2 等值线填充区域搜索算法

对于首尾相接等值线形成的闭合区域, 填充算法比较简单, 利用 VC 中的 CDC 类对封闭多边形进行填充即可, 而连通区域的填充算法, 首先要搜索出区域的边界点, 然后再用 VC 中的 CDC 类的封闭多边形进行填充方法。

连通区域边界点搜索算法为:

1) 以边框左下角顶点为起点 P_0 , 按逆时针 (也可顺时针) 顺序, 将边框上所有的点构建一点集合, 设为 AP ;

2) 按每相邻两点组成一条线段的原则, 由点的集合 AP 得到线段的集合 AL 。并约定每条线段的起点为原点集中序号在前的点。将点集中的最后一个点 P_n 与第一个点 P_0 组成最后一根线段, 这个过程也就是将区域矩形的边界线切割成许多有方向的小线段。

3) 在线段集合 AL 中取出第一条线段, 标记为已使用, 以此线段的终点 P_1 为目标, 在等值线的集合中寻找以此点为端点的等值线。显然存在这样一条等值线, 而且根据等值线不相交的特性, 这样的等值线只有一条, 同时还可以得到这条等值线的另一端点 P_i , 这样可以在线段集合中找到一条以 P_i 为起始点的线段, 这样的线段有且只有一条, 它的终点记为 P_j , 将此线段标记为已使用, 然后寻找下一条以 P_j 为端点的等值线。如此循环, 直到回到起始线段的 P_0 点为止, 形成一个闭合连通区域。

4) 取出下一条线段, 重复步骤 3, 形成一个新区域。不断重复, 直到将所有未做使用标记的线段遍历, 就可找到这个区域范围内的所有连通区域。

2.3 算法描述

1) 填充区数据结构

```

struct CPointStruct //边框点数据结构
{
    float valueContour; //等值线值
    int index; //等值线索引
    int iborder; //点的边框索引
    int iBE; //起始、终止点索引
    float valuePoint; //点的值
    float x,y; //坐标 xy 的值
};
struct CLineStruct //线段数据结构
{
    BOOL bUsed;

```

```

CPointStruct begin;
CPointStruct end;
};
CArray <CPointStruct, CPointStruct>
m_pointcollection //边框点集合
CArray <CLineStruct ,CLineStruct>
m_linecollection; //线段集合

```

2) 搜索算法描述

根据上述搜索算法, 给出搜索核心程序 C++ 代码如下:

```

CCurve *m_lineCurve;
for (int j=0;j<m_linecollection.GetSize();j++)
{
    BOOL bEnd= false;
    CLineStruct lineindex =
m_linecollection.GetAt(j);
    if (!lineindex.bUsed){
        .....
        m_lineCurve = new CCurve;
        while (!bEnd){
            CLineStruct linetemp =
m_linecollection.GetAt(k);
            if (!linetemp.bUsed){
                .....
                linetemp.bUsed = TRUE;
            }
            CPointStruct endpoint =
linetemp.end;
            .....
            if (endpoint==lineindex.begin)
                bEnd = TRUE;
            else {
                if
                (endpoint.valueContour==0.0&&endpoint.index=
                =-1)//非等值线点
                { .....
                    linetemp.bUsed = TRUE;
                    if (endpoint==lineindex.begin)
                        bEnd = TRUE;
                }
                else{
                    CCurve* pCtrLinew =
                    NULL;
                    FindContourByEndPoint(endpoint,m_lineCurve);
                    .....
                }
            }
        }
    }
}

```

```

pCurveListnew->AddTail(m_lineCurve);
    }
}

```

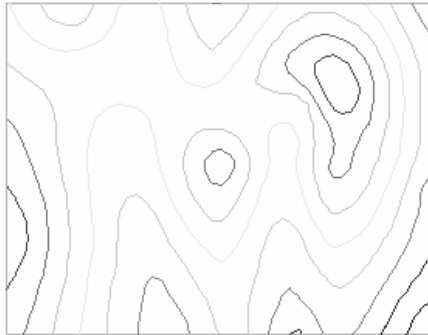


图 2 等值线结果绘制

4 结束语

经实验分析结果表明, 该追踪算法及填充算法能有效地解决等值线生成及填充问题, 且算法简单, 易于实现, 对研究等值线自动综合以及 DEM 精度评估等方面起到了积极的作用。

参考文献:

[1] 张登荣, 等. 值线自动建立拓扑关系算法与快速填充应用[J]. 中国图像图形学报, 2001, 6(3): 264-269.
 [2] 边淑莉, 等. 基于规则格网 DEM 的等高线提取算法的

(上接第 73 页)



图 3 信息显示界面

3.4 显示器的对比度设计

对比度与亮部和暗部的亮度有关。对比度越大, 视敏度越高。不同颜色光的亮度是不同的, 如黄白色亮度高, 而蓝黑色亮度就小, 这也就是白底黑字容易看得清的道理。屏幕上文字信息的前景色和背景色的对比度大一些好。但为了区分多层次的信息, 也可以采用不同的颜色, 不过对特别重要的信息, 对比度要用最醒目的。因此, 将公共信息的显示设计为白底黑字。

3 实验结果

采用 50×50 网格点数据, 实验结果如图 2、图 3。

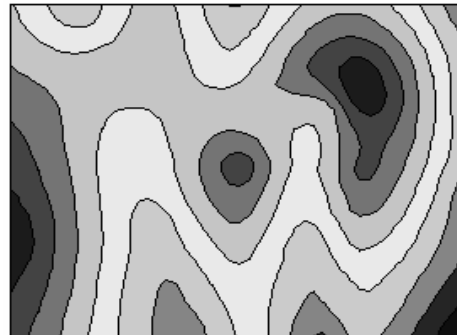


图 3 等值线填充结果绘制图

优化与实现[J]. 测绘科学, 2008, 33(6): 151-153.

[3] 王涛, 毋河海, 刘纪平. 基于区间树索引的等高线提取算法[J]. 武汉大学学报: 信息科学版, 2007, 32(2): 131-134.
 [4] WANG Tao. An Algorithm for Extracting Contour Lines Based on Interval Tree from Grid DEM[J]. Geo-spatial Information Science, 11(2): 103-106.
 [5] 刘平芝. 地图等高线矢量化中的预测跟踪法[J]. 测绘技术装备, 2005, 7(4): 3-4.
 [6] 韩丽娜, 石昊芬, 张群会. 基于边界点追踪的等值线图区域填充算法[J]. 计算机工程与科学, 2006, 28(11): 66-67.

4 结束语

该设计可为研制适应未来战争需要的武器系统提供参考。下一步, 笔者将深入研究战场环境和作战人员的工作环境, 以人机界面学为根据, 科学系统地确定车载武器系统人机界面显示器的模型, 使设计更加人性化、科学化, 使作战人员能够更加舒适、快速地获取相关重要信息, 进一步使武器系统和人相互协调, 提高武器系统的综合作战效能。

参考文献:

[1] 左春桢. 人机工程与造型设计[M]. 北京: 化学工业出版社, 2007.
 [2] 郭伏, 钱省三. 人因工程学[M]. 北京: 机械工业出版社, 2006.
 [3] 曾宪楷, 等. 视觉传达设计[M]. 北京: 北京理工大学出版社, 1991.
 [4] 成大先, 等. 机械设计手册[M]. 北京: 化学工业出版社, 1994.
 [5] 田华. 兵器科学技术总论[M]. 北京: 北京理工大学出版社, 2003.