通用点线面集Delaunay三角剖分与动态编辑

丁圣陶[①,②], 王磊[②], 殷勇[①], 李成名[①]

(① 中国测绘科学研究院, 北京 100039; ② 中国矿业大学, 徐州 221000)

摘要: 总结并提出了一种通用点线面集Delaunay三角剖分与动态编辑的统一算法。可以实现离散点的Delaunay三角剖分, 约束线、面的Delaunay三角剖分, 任意多边形内带特征约束(包括点、线、面)的三角剖分, 一般Delaunay三角剖分的外边界都是其离散点集的凸包, 且内岛屿一般没有挖掉, 本算法实现了Delaunay三角剖分时内、外边界的保界处理。

关键词: Delaunay; 三角剖分; 编辑; 保界处理

doi: 10.3969/j.issn.1000-3177.2011.03.020

中图分类号: P208 文献标识码: A 文章编号: 1000-3177(2011)115-0108-04

Universal Point, Line and Polygon Delaunay Triangulation and Dynamic Editing

DING Sheng-tao[①,②], WANG Lei[②], YIN Yong[①], LI Cheng-ming[①]

(① *Chinese Academy of Surveying and Mapping, Beijing* 100039;
② *China University of Mining and Technology, Xuzhou* 221000)

Abstract: This paper summarizes and presents a kind of universal algorithm of generic points, lines and polygon Delaunay triangulation and dynamic editing. Discrete points, constrained line, polygon, polygon features with zone constraints (including point, line, polygon) Delaunay triangulation can be achieved. The outer boundary of Delaunay triangulation in general is the convex bumps of discrete points, and the inner islands generally do not dig out. The algorithm in process of the Delaunay triangulation, realized the inner and outer boundary processing.

Key words: Delaunay; triangulation; edit; security sector to tackle

1 引 言

目前Delaunay不规则三角网(Delaunay Triangular Irregular Network, D-TIN)广泛用来实现二维离散数据域的剖分与建模。D-TIN构建算法主要分为两个研究内容: 无约束D-TIN和约束D-TIN(CD-TIN)。前者主要算法有三角网生长法、逐点插入法、分治算法、凸包算法等[1~3]; 后者主要有约束图法、分割-合并算法、加密点算法、三角形生长算法和两步法等[4~5]。通过近年来众多学者的努力, 目前对D-TIN研究已经趋于成熟。然而对带有岛屿的约束数据域三角剖分, 对任意数据域外边界的保持, 对任意数据域Delaunay三角网进行实时动态编辑, 对Delaunay三角网中形态比有更高要求的细化算法, 目前都只是对其中部分进行了研究[6~8], 还没有进行很好的系统总结与研究。因此, 本文总结并提出了适用于点线面集约束的Delaunay三角剖分与动态编辑的统一算法, 解决一般Delaunay三角剖分的外边界都是其离散点集的凸包, 且内岛屿一般没有挖掉的情况, 同时将其应用到地形三角网的构建和编辑、典型地物的快速建模、地形分析、多边形骨架化、基于DSM数据屋脊线提取和屋顶面剖分重建等领域, 取得了良好的应用效果。

收稿日期: 2010-05-05 修订日期: 2010-09-02

基金项目: 中国测绘科学研究院基本科研业务经费(7771010)

作者简介: 丁圣陶(1985~), 男, 江苏徐州人, 硕士, 主要从事GIS及应用方面的研究。

E-mail: dingshengtao@126.com

2 任意离散点 Delaunay 三角剖分

2.1 数据预处理

二维三角剖分是对二维平面的没有重叠的剖分,要求一个位置上只能有一个离散点。点的重复必然导致构网异常。因此,必须在数据导入到构网程序之前进行严格的检查。对于不同的应用,外部数据结构可能不同,如果在构网程序外部进行处理,将会造成工作的重复,也不利于质量的保障。解决这一问题采取的方法就是引入智能指针,实现动态、透明地增加和删除点。对于约束边,我们允许交叉,且在交叉的情况下会对约束边进行自动分裂,防止程序出现异常。

2.2 Delaunay 三角网构建

为充分实现 Delaunay 三角网的快速、高效生成,本工作在常规分割- 合并算法的基础上,采用自适应格网划分方法分割点集构建 Delaunay 三角网[2]。

基于每个子集包含 2 或 3 个点的原则,首先按照 x 方向、然后 y 方向进行点集排序、分割,如此反复进行,直至所有子集都满足条件为止,这样较好地解决了双向划分、四叉树划分导致的各子集数据不均的情况,并且点集分割、三角形生成、三角网合并速度快。程序流程示意图如图 1 所示。

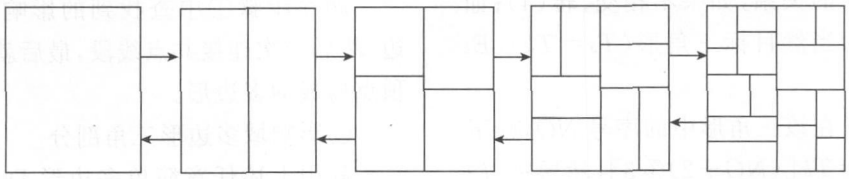


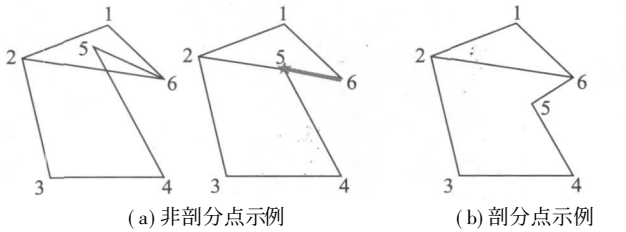
图 1 基于自适应格网划分方法构建 Delaunay 三角网的流程示意图

3 任意简单多边形 Delaunay 三角剖分

3.1 剖分的基本思想

首先不管多边形的凹凸性,从原始多边形中寻找一个可剖分顶点,对原始的多边形进行划分,从而分割出一个三角形同时产生一个新多边形,然后对新生成的多边形判断其凸凹性,若为凸多边形,则顺序连接多边形各点生成三角形网,算法结束;否则对新生成的凹多边形进行递归操作直到原始的多边形划分成一系列三角形和一个凸多边形。

可剖分顶点:①该点为凸点;②以该点及其临近两点组成的三角形不包含多边形上其他点。如图 2(a),1 点为凸点,左图中由 1 点临近两点组成的三角形 162 包含多边形上的点 5,右图中点 5 在三角形 162 上,两种情况都不构成可剖分顶点,图 2(b)中的 1 点为可剖分顶点。



(a) 非剖分点示例 (b) 剖分点示例

图 2

3.2 剖分结果的局部优化

①从剖分结果三角形表形成边表。

②对边表中的每一条非边界边,做步骤(3)。

③取出以次边为邻接条件的 2 个相邻三角形,构成一个四边形。如果这个四边形是凹的,则不做任何处理。如果这个四边形是凸的,则计算 2 个三角形的最小内角 A,交换对角线,计算出 2 个新的三角形的最小内角 B。如果 $B > A$,则以交换对角线所得到得的 2 个新三角形替换原来的 2 个三角形,并按新的三角形修改剖分结果三角形表和边表,并令交换边计数变量 IE 加 1(初始值为 0)。

4 点编辑

4.1 最速方向点定位算法

最速方向点定位算法首先需要分块,建立网格管理,由点- 网格- 三角形的映射关系得到首三角形(如图 3)。首三角形内任取一点 S(一般取三角形的重心)与目标点 O 就可以组成一条方向线,该方向线穿越的三角形是唯一的。

从首三角形内点 S 出发,沿最速线 SO 搜索至目标点 O 所在的三角形。其实现过程如下:

①首三角形 $T1$ 作为当前目标三角形 Ts,判断当前目标三角形 Ts 的边 Ei 与方向线 SO 的关系,如果相交(如果在一个三角形中有两条边同时满足与方向线相交,取先出现的边,后续的相切也同样只取先出现的边),则下一个三角形就是与 Ts 共边 Ei 的且与 Ts 相邻的三角形 Tj,把 Tj 作为当前目标三角形(即 $Ts = Tj$),转(2);如果相切(一个线段的顶点在另一线段上),临时记录下切点 Pi 即边 Ei 落

在方向线 SO 上的端点, 转(3)。

②如果点 P 在当前目标三角形 Ts 中, 返回三角形 Ts, 结束算法; 否则, 在目标三角形 Ts 中, 判断三角形 Ts 的边 Ej (除边 Ei 之外的两条边) 与方向线 SO 的关系, 如果相交, 则下一个三角形就是与 Ts 共边 Ej 的且与之相邻的三角形 Tj, 把 Tj 作为当前目标三角形(即 $Ts = Tj$), 将边 Ej 赋给 Ei, 即 $Ei = Ej$, 重复(2)。

③以切点 Pi 为顶点, 以当前目标三角形 Ts 为出发三角形, 逆时针方向搜索共顶点 Pi 的三角形 Tk, 如果点 P 在三角形 Tk 中, 返回三角形 Tk, 结束算法; 否则, 在三角形 Tk 中, 判断与顶点 Pi 对应的边 Ek 与方向线 SO 的关系, 如果不相交, 转(4); 如果相交, 把 Tk 作为当然目标三角形($Ts = Tk$), $Ei = Ek$, 转(2)。

④根据顶点 Pi 在该三角形中的序号 NO, 则下一个三角形为 $Tj = Tk[(NO + 2)\%3]$, 然后把 Tj 作为当前目标三角形(即 $Ts = Ti$), 转(3)。

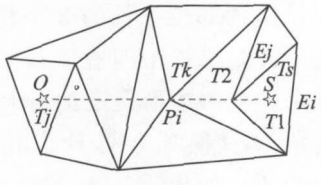


图 3 最速方向点定位

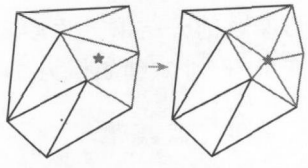


图 4 增量局部重构

4.2 约束点插入的算法

在用最速方向定位算法找到点所在的三角形后, 就需利用局部优化的方法来实现约束点的增量重构从而避免全局重构(如图 4 所示), 其算法步骤是:

①将约束点所在三角形 3 个顶点与该点相连, 形成 3 条有向边, 并将这 3 条边送至全局边链表。方向设置为从 3 个顶点指向约束点。

②设置 3 条新边的拓扑结构, 以右手定则, 4 个手指的方向为边的方向, 大拇指的方向为左三角形, 反之为右三角形。

③由于点加入后, 生成 3 个面, 而原先的三角形所在面必然被删除。为了免于在面链表中进行删除该面而产生的繁琐遍历运算, 故而将原三角形面的指针指向其中一个新生成的三角形, 这样只需新增两个面即可。

④对原始三角形的 3 条边按规则进行边优化, 实施对角线交换。

4.3 约束点删除的算法

①查找共同顶点的三角形

先找到与之共点的第一个三角形, 由任一包含共点三角形的任一边起根据拓扑关系, 寻找其邻接三角形, 当找到的边是首三角形的另一条边时, 则查找结束, 将所查找到的三角形保存起来。这样算法设计就免去了对每一个三角形进行顶点判断, 大大提高了算法执行效率。

②重构点的影响域多边形

删除步骤①中查找到的影响域三角形的公共边, 然后依次连接共点线段, 最后就形成影响三角形顶点构成的多边形。

③影响域多边形三角剖分

利用上述任意简单多边形 Delaunay 的三角剖分算法对影响域多边形进行三角剖分, 即得删除点后的三角网。

5 线编辑

5.1 添加约束线的算法

如图 5, 在 Delaunay 三角网中添加一条约束线段, 我们采用目前使用较多的两步法实现。设添加线段为 AB, 其实现步骤为:

①分别将约束线上两点按约束点动态插入算法添加到 Delaunay 三角网中, 同时更新三角网几何和拓扑信息。

②找到并删除三角网中被约束线(AB) 所"切割"的影响域三角形集合, 再将这些三角形的各顶点以 AB 为界分为左右两个点集 Pleft、Pright, 它们与 AB 构成了 2 个未三角化的多边形区域。

③对约束线的左右影响域 Pleft、Pright 进行三角剖分。

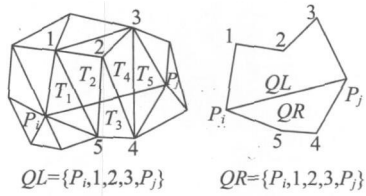


图 5 约束线段的影响三角形及影响多边形

5.2 删除约束线的算法

如图 6, 在 CDT 中删除一条约束线段, 主要是去掉散点间的约束关系, 这是添加约束线的逆过程。设删除线段为 $PiPj$, 其实现步骤为:

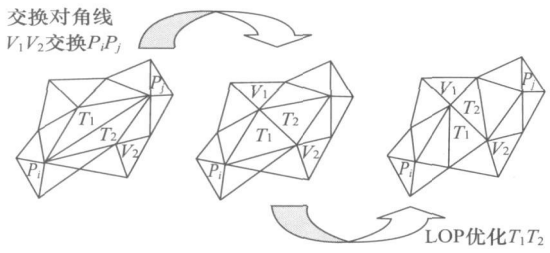


图 6 约束线段的删除

①确定 Pi(或 Pj) 所在的三角形, 如果 Pi 和 Pj 在一个三角形中, 则其已经满足空外接圆准则, 退出, 反之则进入步骤②;

②查找共用 $PiPj$ 的三角形 $T1$, $T2$;

③交换对角线。在 $T1$、$T2$ 中, 找出除公共顶点 Pi、Pj 外的顶点 $V1$、$V2$, 用 $V1V2$ 交换 $PiPj$, 交换对角线后的三角形 $T1$ 为 $PiV2V1$, $T2$ 为 $PjV1V2$;

④LOP 优化三角形。用 LOP 算法优化三角形 $T1$、$T2$。

6 内外边界的保界处理

外边界的保界处理就是移去外边界以外的三角形集, 而内边界的保界处理就是挖掉内边界之内的三角形集。如图 7 所示, 红线和红点都是约束线和约束点, 首先完成约束域 Delaunay 三角剖分, 然后经过外边界的保界处理移去三角形 543 和 710, 内边界的保界处理挖掉三角形 8911 和 91011。

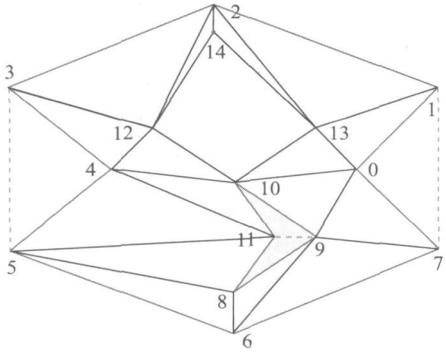


图 7 通用点线面集保界 Delaunay 三角剖分

6.1 内边界的保界处理

①新建两个数组 T 和 E(栈数组), T 中存贮内边界多边形内的三角形, E 中存贮 T 中的非边界边; 首先找到内边界多边形的一个可剖分点, 以可剖分点及其临近两点组成的三角形就是内边界多边形内的首三角形, 并将其加入 T 中, 同时更新 E。

②以步骤①中找到的首三角形为种子三角形, 根据三角网中点、边、三角形的拓扑关系搜索与种子三角形非边界边共边的三角形, 将其加入到 T 中, 同时更新 E。

③遍历 E, 搜索三角网中与 E 共边的三角形, 同时更新 E 和 T。

④栈 E 中无元素时算法结束, 否则执行步骤③。

6.2 外边界的保界处理

由于 Delaunay 三角剖分后的外边界是点、线、面集的凸包, 所以外边界的保界处理就是挖去凸包与外边界之间的三角形集。下面是算法描述:

首先将凸包与外边界的起始点以及顺序置为相同, 再顺序遍历凸包的每一条边 Li, 同时比较在外边界中是否有与 Li 相同的边, 若有, 继续下一条边; 若没有, 则记录外边界中边 Li 的两端点之间的点序列 S, 用内边界保界处理算法挖掉 S 中的三角形; 继续下一条边, 直到凸包中的边遍历完毕。

7 应用与总结

根据笔者提出的算法, 以建筑物与地形的匹配应用为例(如图 8), 未作建筑物与地形匹配之前, 建筑物部分钻入地形以下, 严重影响三维景观可视化效果。按照将建筑物底座作为约束多边形放入地形模型中进行通用约束 Delaunay 三角剖分, 即可实现顾及建筑物与地形的匹配的地形模型重构(如图 9)。

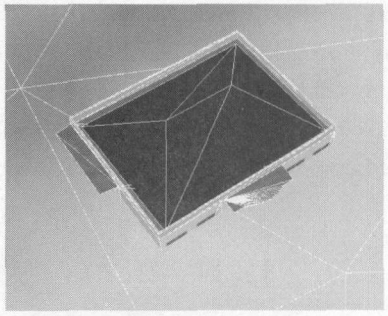


图 8 建筑物未与地形匹配

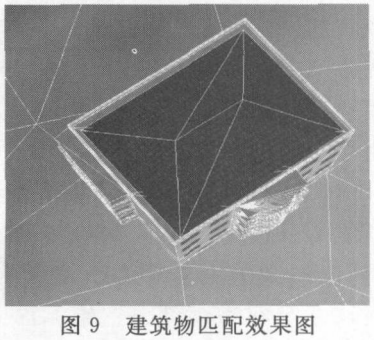


图 9 建筑物匹配效果图

Delaunay 三角剖分在地学可视化、几何造形、有限元网格生成、模式识别等许多领域有着广泛的应用, 本文总结并提出的统一算法, 已经在我们研发的数字城市三维地理空间框架系统的自动建模、地形与地物匹配、三维空间分析中得到了广泛的应用, 实践证明, 统一算法稳定可靠, 对相关领域应用研究具有很好的参考价值。

(下转第 115 页)

3.2 信息标注联动的实现

本系统实现了二三维点、线、面标注的一体化联动标注。为了简化三维立体标注的复杂性和保持数据源的一致性, 标注数据统一采用二维矢量数据方式描述并存放于后台数据库中。在二维系统中进行点、线、面标注时, 三维系统中能同步显示此标注, 同理, 在三维系统进行标注的增加、删除、编辑操作时, 二维系统中的标注也能得到同步更新, 如图 3 所示。

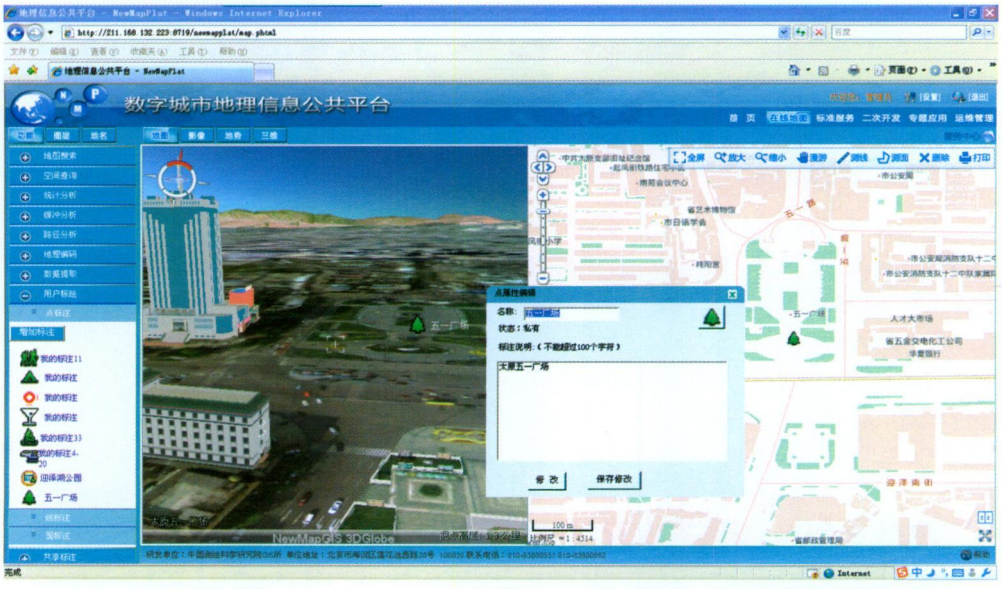


图 3 二三维系统标注联动

4 结束语

本文实现了一种无缝集成的 WebGIS 二三维联动系统解决方案, 对目前应用需求有了初步的实现, 在人们的日常生产生活中发挥着切实的作用。目前此系统还只是对二三维联动系统的一种探索, 更深层次的功能比如三维空间分析、同步编辑更新等, 还需进一步扩展。

参考文献

[1] 万幼川, 边馥苓. 二三维联动的 GIS 系统体系结构构建技术[J]. 地理信息世界, 2008, (2).
[2] 李海洋, 范文义, 李明泽. 二维地图与三维虚拟场景交互技术的研究与应用[J]. 东北林业大学, 2008, 36(11).
[3] 王继周, 李成名, 林宗坚. 三维 GIS 的基本问题与研究进展[J]. 计算机工程与应用, 2003, (24).
[4] 王宏武, 董卜海. 一个与视点相关的动态多分辨率地形模型[J]. 计算机辅助设计与图形学学报, 2000, 12(8): 575- 579.
[5] 周新耿, 刘芬. 二维、三维空间信息系统的结合和应用[J]. 计算机与数字工程, 2005(3).
[6] 刘东琴, 徐文中, 林宗坚. 城市空间二维信息系统与三维虚拟场景一体化研究与应用——BDA 城市三维信息系统的设计与实现[J]. 测绘科学, 2007(1).
[7] 张剑波, 刘修国. 线性四叉树在基于 LOD 的地表模型绘制中的应用[J]. 计算机工程与应用, 2002, (8).

(上接第 111 页)

参考文献

[1] 武晓波, 王世新, 肖春生. 一种生成 Delaunay 三角网的合成算法[J]. 遥感学报, 2000, 4(1): 32- 35.
[2] 胡金星, 潘懋, 马照亭, 吴焕萍. 高效构建 Delaunay 三角网数字地形模型算法研究[J]. 北京大学学报(自然科学版), 2003, 39(5): 736- 741.
[3] Lee DT. Generalized delaunay triangulation for planar graphs[J]. Discreteand Computational Geometry, 1986, 1(1): 201- 217.
[4] 周晓云, 刘慎权. 实现约束 Delaunay 三角剖分的健壮算法[J]. 计算机学报, 1996, 19(8): 615- 624.
[5] 蒲浩, 宋占峰, 詹振炎. 基于约束 Delaunay 三角剖分的道路三维建模方法[J]. 华中科技大学学报(自然科学版), 2005, 33(6): 111- 113.
[6] 刘学军, 龚健雅. 约束数据域的 Delaunay 三角剖分与修改算法[J]. 测绘学报, 2001, 30(1): 82- 88.
[7] 刘少华, 吴东胜, 罗小龙, 陈华军. Delaunay 三角网中点目标快速定位算法研究[J]. 测绘科学, 2007, 32(2): 69- 70, 113.
[8] 贾晓林, 吴立新, 王彦兵. 一种带岛屿数据域的三角网剖分算法研究[J]. 地理与地理信息科学, 2004, 20(5): 28- 31.