

Delaunay 三角化中特征约束细分嵌入算法

田峰敏¹ 徐定杰¹ 李 宁¹

(1 哈尔滨工程大学自动化学院, 哈尔滨市南通大街 145 号, 150001)

摘要:在 Delaunay 三角化中, 以在特征约束上插入附加点的方式对特征约束细分, 可使剖分网格在表达特征约束的同时仍满足 Delaunay 三角网格特性。本文提出了一种改进的细分嵌入算法——端点外接圆法 (ETCM), 该算法具有线性时间复杂度。以剖分结果的稳定性、附加点个数、运算耗时和网格质量为评价标准, 对 ETCM 和现有的细分嵌入算法进行了评估。实验结果表明, 在运算耗时方面, ETCM 受数据分布的影响最小, 且耗时最少; 在其他方面, 与现有算法的最好表现持平。

关键词: Delaunay 三角化; 特征约束; 细分嵌入

中图分类号: P208

Delaunay 三角化 (Delaunay triangulation, DT) 是离散数据集划分的有效方法, 在工程领域中有着广泛的应用。在很多情况下, 离散数据之间存在约束关系, 表现为离散数据带有若干条折线和封闭多边形, 在剖分结果中须包含这些特征。解决这类问题的方法有约束 Delaunay 三角化^[1,2] 和协调 Delaunay 三角化^[3,4]。约束 Delaunay 三角化是 DT 的近似, 嵌入特征约束后, 网格单元的 DT 特性通常不能保持, 网格质量下降。而在某些应用中, 如有限元数值计算的稳定性和收敛性会受网格质量的影响。协调 Delaunay 三角化可以保证网格的 DT 特性, 网格质量明显优于约束 Delaunay 三角化。受 Sapidis^[5]、易法令^[6] 和梅承力算法^[7] 的启发, 本文提出了一种改进的特征约束细分嵌入算法——端点三角形外接圆法 (endpoint triangle's circumcircle method, ETCM), 保证了剖分结果的稳定性, 提高了插入附加点的速度。

1 端点三角形外接圆法 (ETCM)

1.1 ETCM 算法思想

定义 特征线段一个端点的端点三角形是以该端点为顶点、一条三角形边与特征线段相交的三角形。

设 $s[P_h P_e]$ 是一条没有被 DT 网格包含的特征

线段, Δ_h 是 P_h 的端点三角形, Δ_e 是 P_e 的端点三角形, $\text{Cir}(\Delta_h)$ 和 $\text{Cir}(\Delta_e)$ 分别为 Δ_h 和 Δ_e 的外接圆。

ETCM 的基本思想是: 设 $J_h = \text{Cir}(\Delta_h) \cap s[P_h P_e]$, $J_e = \text{Cir}(\Delta_e) \cap s[P_h P_e]$, 如果 $J_h \cap J_e = \emptyset$, 则把 J_h 和 J_e 中较长的一段作为可嵌入的部分, 它所对应的交点为附加点, 较短的一段作为特征线段剩余的部分, 对它继续使用上面的方法, 直到 $J_{h'} \cap J_{e'} \neq \emptyset$ 。此时, 取 $J_{h'} \cap J_{e'}$ 的中点为附加点。若当前处理的特征线段影响已嵌入的特征线段, 则把受影响的特征线段按上面方法重新嵌入。

1.2 ETCM 描述

输入: 无约束 DT 三角网、特征线段。

输出: 嵌入特征约束后符合 DT 特性的三角网。创建一个堆栈 FS_remain 存放待处理的特征线段, 一个双向链表 FS_finished 存放已处理的特征线段。

```
While (FS_remain ≠ 空) {
    弹出一条特征线段  $s[P_i P_j]$ ;
    if ( $P_i$  邻接三角形的顶点中有  $P_j$ )
        continue;
    else {
        1) 初始准备
        ① 通过三角网络的拓扑关系找出与  $s[P_i P_j]$  相交的三角形, 组成  $P_i P_j$  的影响域 crosstri;
        ② 对线段端点赋值:  $P_h = P_i, P_e = P_j$ ;
        ③  $\text{Cir}(\Delta_h)$  交  $s[P_h P_e]$  于  $Q_1, \text{Cir}(\Delta_e)$  交  $s[P_h P_e]$  于
```

Q_2 ;

2) 分段处理

While ($|P_h Q_1| + |Q_2 P_e| < |P_h P_e|$) {

if ($|P_h Q_1| \geq |Q_2 P_e|$) {

以 Q_1 作为附加点: $P_a = Q_1$; 把 $s[P_h Q_1]$ 链入 FS_finished; 以 P_a 作为特征线段剩余部分的头, P_e 仍为特征线段剩余部分的尾; }

else {

以 Q_2 作为附加点: $P_a = Q_2$; 把 $s[Q_2 P_e]$ 链入 FS_finished; 以 P_a 作为特征线段剩余部分的尾, P_h 仍为特征线段剩余部分的头; }

插入附加点: InsertNewNode(crosstri, P_a);

更新 Q_1, Q_2 : $\text{Cir}(\Delta_h)$ 交 $s[P_h P_e]$ 于 Q_1 , $\text{Cir}(\Delta_e)$ 交 $s[P_h P_e]$ 于 Q_2 ;

}

3) 整体处理

① 以 $Q_1 Q_2$ 的中点作为附加点 P_a ; 把 $s[P_h P_a]$ 和 $s[P_a P_e]$ 链入 FS_finished; 特征线段的剩余部分为零;

② InsertNewNode(crosstri, P_a);

}

InsertNewNode(crosstri, P_a) {

i. 在 crosstri 所标记的三角形中寻找 P_a 所在的三角形或四边形, 然后以一个迭代的过程向外寻找外接圆包含 P_a 的三角形, 组成 P_a 的影响域 efftri, 根据三角形面积坐标的正负关系判断点是否在三角形内;

ii. 如果 efftri 所标记三角形的公共边为已处理过的特征线段, 则把该边从 FS_finished 断链, 压入 FS_remain;

iii. P_a 与其影响域的顶点连线形成新的三角形, 先替换 efftri 所记录的三角形, 余者再添加到三角形数组尾部;

iv. 更新 crosstri:

crosstri = crosstri - (crosstri \cap efftri);

if ($s[P_h P_e]$ 与 P_a 影响域相邻两个顶点 C_k, C_{k+1} 所组成的线段 $s[C_k C_{k+1}]$ 相交)

{

$\Delta_h = \Delta P_a C_k C_{k+1}$; 把 Δ_h 加到 crosstri 的头; }

else

{

$\Delta_e = \Delta P_a C_k C_{k+1}$; 把 Δ_e 加到 crosstri 的尾; }

}

v. 按文献[8]提出的方法动态更新三角形的拓扑关系;

}

ETCM 具有以下优点: ① 不论以线段的哪个端点为起点, 附加点的位置是惟一的; ② 在插入新点的过程中, 迅速找到包含附加点的三角形或四边形是减少计算量的关键。鉴于附加点在特征线段的影响域内, ETCM 一次性找到特征线段的影响域, 然后在影响域中寻找, 不必每次都要在全

局进行寻找, 减少了查找范围, 可以快速完成附加点的插入。相对 Sapidis 算法中需要寻找特征线段剩余部分的端点三角形, 一次性找到特征线段的影响域并没有增加计算量。

1.3 ETCM 附加点个数收敛性说明

在分段处理阶段, 从特征线段两个端点三角形外接圆中选择截取特征线段较长者所对应的交点作为附加点, 此点是使以截取线段为公共边的两个三角形都符合 DT 特性, 且使截取部分占整个特征线段比例最大的点。对特征线段的剩余部分按以上方法处理, 直到两条截取线段的公有部分不为空, 显然可以使单条特征线段上附加点的个数最少。但 ETCM 并不能保证全局附加点个数最少, 因为以已嵌入的特征线段为边的三角形有可能成为后插入的附加点影响域的一部分, 这条已嵌入的特征线段就要重新嵌入。

设含 N 个点 M 条边的平面直线图, AB 是其中一条特征边, 设 $\text{circle}(A)$ 是过 A 且内部不包含其他点或与特征边的半径最大的圆, 若 $\text{circle}(A) = \text{circle}(B)$, 则不用对 AB 进行细分; 若 $\text{circle}(A) \neq \text{circle}(B)$, $\text{circle}(A)$ 交 AB 于 A' , $\text{circle}(B)$ 交 AB 于 B' , $A'B'$ 长度为 D , d 为其他点和边到 $A'B'$ 的最短距离。采用如下方法: 把 $A'B'$ 分成 $[D/2d]$ 段, 每段的长度不超过 $2d$, 以每一个分段为直径的圆就是内部不包含其他点的圆。加上 AA', BB' , AB 最多需要插入 $[D/2d] + 1$, 即 $O(1)$ 个附加点, 且每条特征边的细分嵌入都不会影响已嵌入的特征边。因为 $M = O(N)$, 所以附加点的总数 $N_m = O(N)$ 。显然, 这个方法对 $A'B'$ 的细分存在冗余。ETCM 只对受影响的区域进行处理, 所以得到的 $\text{circle}(P_i)$ 的平均半径比上面方法的要大, 附加点个数要少, 所以 ETCM 的附加点个数收敛。

1.4 ETCM 时间复杂度分析

设 T_{P_i} 为计算第 i 个附加点位置所需的时间, T_{in_i} 为插入第 i 个附加点所需的时间。文献[9]指出, 在 DT 网络中, 一个点所属三角形的平均个数为 6 个, 那么平均进行 6 次线段相交判断就能找到特征线段的一个端点三角形, 第二个与特征线段相交的三角形也可同时确定。若特征线段共穿过 n 个三角形, 那么根据网格间的拓扑关系, 平均进行 $6 + (n-2) \times 2$ 次相交判断就能完成特征线段影响域的查找, 而计算三角形外接圆与特征线段的交点是常数时间, 所以 $T_{P_i} = O(1)$ 。在特征线段的影响域内寻找包含附加点的三角形或四边形, 最多需 $n-1$ 次点在三角形内的判断, 而每次

判断也是常数时间。文献[8]指出,更新拓扑关系是一线性过程,所以 $T_{in_i} = O(1)$ 。

设 T_p 为计算所有附加点位置所需的时间, T_{in} 为插入所有附加点所需的时间, $T_{in} = \sum T_{in_i} \{ i \in N_{in} \} \leq N_{in} \cdot \max\{T_{in_i}\}$, 因为 $T_{in_i} = O(1)$, $N_{in} = O(N)$, 所以 $T_{in} = O(N)$ 。同理, $T_p = O(N)$ 。则 ETCM 耗时 = $T_p + T_{in}$ 。在平均意义下, ETCM 具有线性时间复杂度。

2 实验

2.1 性能评价标准

以往的文献中并没有提出细分嵌入算法性能的评价标准。本文以如下 5 个方面作为评价标准:① 剖分稳定性;② 附加点个数(插入尽量少的附加点,可以尽可能小地改变原有数据集,减少处理数据量);③ 运算耗时;④ 插入附加点后网格的质量(平均形态比(AAR)越接近 0.5,表明网格质量越好);⑤ 单点提升平均形态比,即每个附

加点对网格平均形态比提升的贡献,计算公式为:

$$\Delta_{per AAR} = (AAR_{Refine} - AAR_{CDT})/n \quad (1)$$

式中, AAR_{Refine} 是插入附加点后网格的平均形态比; AAR_{CDT} 是 CDT 剖分后网格的平均形态比; n 为附加点个数。

2.2 性能测试与结果分析

实验环境为 P4, 2.8 GHz, 2×256 M RAM, Windows XP, VC6.0, 时间测试工具为 CodeTest 4.0。

使用 15 组纵横坐标在 $[0, 1000]$ 内且满足均匀分布的随机点和条数为点数 15% 的互不交叉的特征线段组成的平面直线图作为测试数据域, 从以上 5 个标准对 Tsai^[10]、卢朝阳^[11]、Sapidis^[5]、梅承力^[7]、ETCM 五种算法进行测试。图 1 为一组测试数据的剖分结果图, 图中粗虚线为特征线段(其条数为点数的 15%), 方形点为插入的附加点。图 2~图 4 为性能测试曲线。剖分稳定性的测试方法是将每条特征线段的两个端点对调, 看剖分结果是否一致。

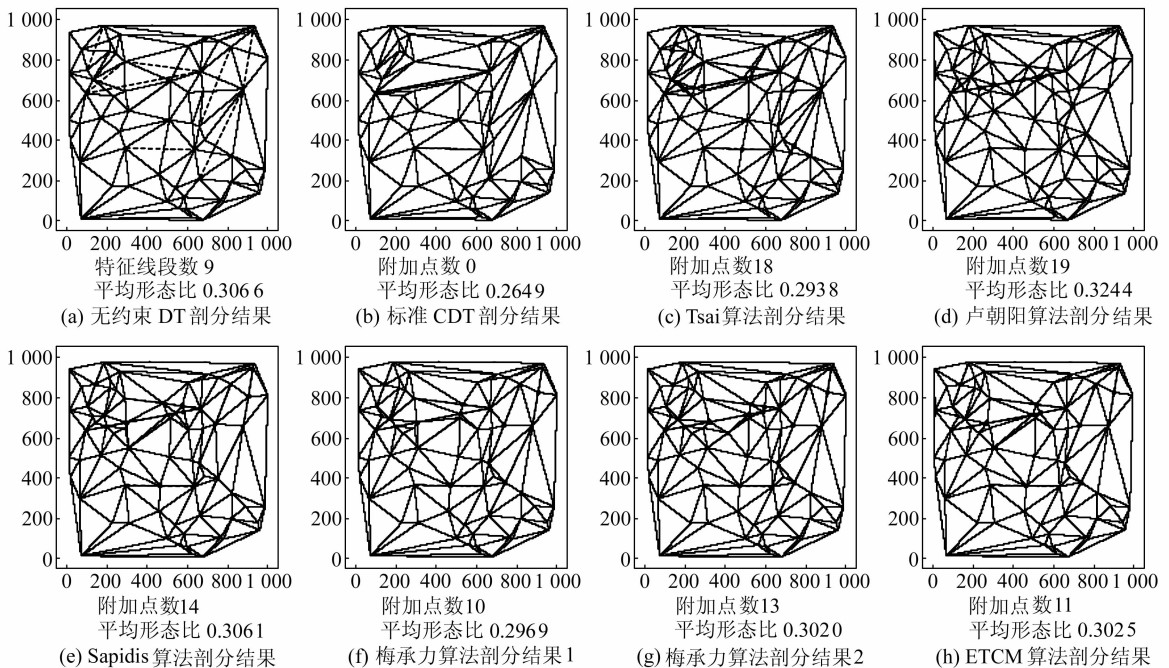


图 1 随机点为 60、特征线段为 9 条时,各种细分嵌入算法的剖分结果

Fig. 1 Triangulation Results of Node Refinement Algorithms, with 60 Nodes and 9 Feature Segments

以图 1 所用数据为例,可见,梅承力算法有两个结果,其他算法都只有一个结果,表明除了梅承力算法,其他算法都是稳定的。

从图 2 可见, Tsai 算法所需的附加点个数最多,卢朝阳算法次之, Sapidis 算法、梅承力算法和 ETCM 所需的附加点个数几乎相同,都较少。

从图 3 可见, Tsai 算法耗时最多, ETCM 耗时

最少,且 ETCM 的测试曲线起伏最小,与 N 大体上呈线性关系,说明 ETCM 的耗时主要受数据量的影响,而受数据分布的影响很小,算法鲁棒性较强。

从图 4 可见,所有细分嵌入算法都可使三角网格的平均形态比恢复到或略高于嵌入特征线段前的水平,其中,卢朝阳算法提高最多, Tsai 算法最小,其他三种算法几乎相同。在单点提升形态

比方面(图 5),Tsai 算法最低,卢朝阳算法次之, 其他三种算法几乎相同。

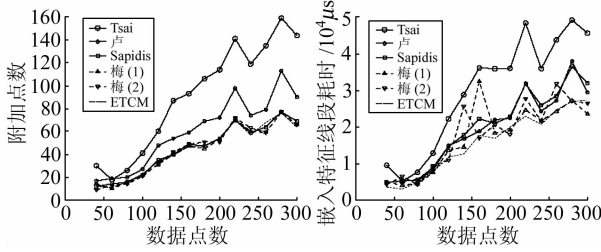


图 2 附加点个数
Fig. 2 Number of Added Nodes

图 3 嵌入特征线段耗时
Fig. 3 Time Cost of Inserting Feature Segment

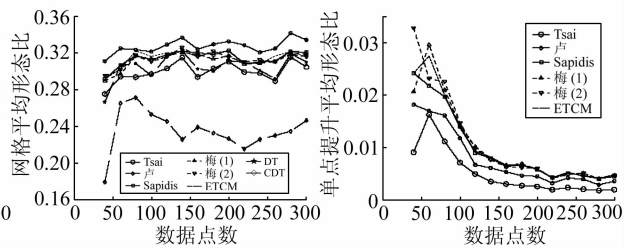


图 4 网格平均形态比
Fig. 4 AAR of TIN

图 5 单点提升平均形态比
Fig. 5 Improved AAR of TIN

3 结 语

本文提出了一种改进的特征约束细分嵌入算法——端点外接圆法,该算法具有线性时间复杂度。实验结果表明,ETCM 在稳定性、附加点个数、网格质量三个方面与现有算法的最好表现持平;在耗时方面,受数据分布的影响最小,耗时最少。附加点个数最少、同时网格质量最优的特征约束细分方式是个最优化问题,现有算法包括 ETCM 都是该问题的决策解法,如何达到更好的效果有待进一步研究。

参 考 文 献

[1] Anglada M V. An Improved Incremental Algorithm for Constructing Restricted Delaunay Triangulation [J]. Computer & Graphics, 1997, 21(2):215-223

[2] 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):1 114-1 118

[3] Hansen A J, Levin P L. On Conforming Delaunay Mesh Generation[J]. Advances in Engineering Software,1992,14(2):129-135

[4] Ruppert J. A Delaunay Refinement Algorithm for Quality 2-Dimensional Mesh Generation[J]. Journal

of Algorithms, 1995,18(3):548-585

[5] Sapidis N, Perucchio R. Delaunay Triangulation of Arbitrarily Shaped Planar Domains[J]. Computer Aided Geometric Design, 1991, 8(6): 421-437

[6] 易法令,韩德志. 带特征线约束的 Delaunay 三角剖分最优算法的研究及实现[J]. 计算机工程,2001, 27(6):32-34

[7] 梅承力,肖高逾,周源华. 一种新的带特征约束的 Delaunay 三角剖分算法[J]. 电子学报,2001,29(7): 895-898

[8] 刘学军,符锌砂. 三角网数字地面模型快速构建算法研究[J]. 中国公路学报,2000,13(2):31-36

[9] O'Rourke J. Computational Geometry in C (Second Edition)[M]. Cambridge England:Cambridge University Press,1998

[10] Tsai V J D. Delaunay Triangulations in TIN Creation: an Overview and Linear Time Algorithm[J]. International Journal of Geographical Information Systems, 1993, 7(6):501-524

[11] 卢朝阳,吴成柯,周幸妮. 满足全局 Delaunay 特性的带特征约束的散乱数据最优三角剖分[J]. 计算机学报,1997,20(2):118-124

第一作者简介:田峰敏,博士生。研究方向为海洋 GIS 和水下地形辅助导航。
E-mail:tianfengmin2000@163.com.cn

A Node Refinement Algorithm for Inserting Feature Constraints in Delaunay Triangulation

TIAN Fengmin¹ XU Dingjie¹ LI Ning¹

(1 College of Automation, Harbin Engineering University, 145 Nantong Street, Harbin 150001, China)

Abstract: An improved algorithm of node refinement scheme called endpoint triangle's circumcircle method (ETCM) is proposed, which has a linear time complexity. Regarding stabi-