

三维约束 Delaunay 三角化的实现*

徐永安^{1,2}, 杨 钦³, 吴壮志³, 陈其明³, 谭建荣²

¹(青岛海信集团技术中心, 山东 青岛 266071);

²(浙江大学 机械系, 浙江 杭州 310027);

³(北京航空航天大学 计算机科学与工程系, 北京 100083)

E-mail: xuyongan@hisense.qd.sd.cn; visual@buaa.edu.cn

http://www.hisense.com.cn

摘要: 分析了约束 Delaunay 三角化中存在的边界一致性问题, 给出了约束 Delaunay 三角化的理论依据, 重点探讨了三维约束 Delaunay 三角化的可行性条件和范围, 同时, 给出了三维有限域约束 Delaunay 三角化的实现方法及其在石油地质勘探数据和机械零件方面的网格剖分实例. 这种算法在复杂对象的科学计算和工程分析中发挥了重要作用.

关键词: 三角化; Delaunay 三角化; 约束 Delaunay 三角化

中图分类号: TP391 **文献标识码:** A

早在 1981 年, Bowyer 算法^[1]和 Watson 算法^[2]就已实现了 n 维空间凸包的 Delaunay 三角化. 随着计算机科学的飞速发展, 科学计算和工程分析的对象越来越复杂, 80 年代中后期任意域 Delaunay 三角化成为网格剖分技术研究的热点问题^[3~7], 至今尚未彻底解决, 因此, 在很大程度上制约了复杂对象的分析和显示, 本文基于边界边、边界面片在 Delaunay 三角化中的存在条件, 在可行的范围内研究了三维约束 Delaunay 三角化的实现算法.

1 约束 Delaunay 三角化中边界一致性问题

Delaunay 三角化是基于点集的优化几何构型, 它只能保证点在 Delaunay 三角化中的存在性, 并不能保证边界边、边界面在 Delaunay 三角化中的存在性. 以如图 1 所示的二维域为例, 不论域内是否有点, 都无法实现 AB 为其中一条边的 Delaunay 三角化, 即以 AB 为边的三角形的外接圆总是包含其他点, 这就是二维任意域约束 Delaunay 三角化问题. 三维任意域约束 Delaunay 三角化除了边界边的约束以外, 还有边界面的约束, 如图 2 所示. 即使边界面片 $\triangle ABC$ 的 3 条边在 Delaunay 三角化中存在, 但由于四面体 1234 穿过 $\triangle ABC$, 面片 $\triangle ABC$ 在 Delaunay 三角化中不存在.

目前的研究情况表明^[7], 二维任意域在不添加辅助点的情况下, 不一定能够实现 Delaunay 三角化; 而在添加辅助点的情况下, 能够实现 Delaunay 三角化. 三维任意域在不添加辅助点的情况下, 非 Delaunay 三角化和 Delaunay 三角化均不一定能够实现; 在添加辅助点的情况下, 能否实现 Delaunay 三角化, 尚无定论. 以往的研究工作大多在剖分算法方面考虑解决方法, 本文基于 Delaunay

* 收稿日期: 1998-12-21; 修改日期: 1999-09-29

基金项目: 国家自然科学基金资助项目(69878038); 浙江省自然科学基金资助项目(696045)

作者简介: 徐永安(1965-), 男, 江苏兴化人, 博士, 助教, 主要研究领域为计算机图形学, 计算机辅助设计, 可视化; 杨钦(1968-), 男, 浙江新安江人, 讲师, 主要研究领域为计算机图形学, 可视化; 吴壮志(1969-), 男, 湖南益阳人, 讲师, 主要研究领域为计算机图形学, 软件设计方法; 陈其明(1932-), 男, 重庆人, 教授, 主要研究领域为计算机图形学, 可视化; 谭建荣(1954-), 男, 浙江湖州人, 教授, 博士生导师, 主要研究领域为计算机辅助设计, 可视化.

三角化理论,对边界边、边界面片在 Delaunay 三角化中的存在条件进行了探讨,并据此设计了三角化算法.

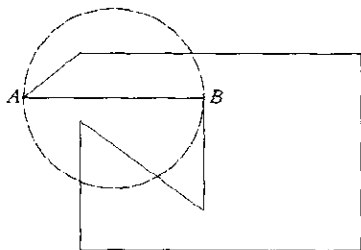


Fig. 1 Circle that passes through boundary edge AB contains the other vertex

图1 过边界边 AB 的圆内包含其他顶点

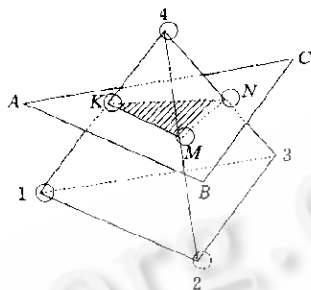


Fig. 2 Boundary patch $\triangle ABC$ penetrated by the tetrahedra

图2 边界面片 $\triangle ABC$ 被四面体穿过

2 边界边、边界面片在 Delaunay 三角化中的存在条件

针对约束 Delaunay 三角化中存在的问题,本文给出如下已证明的命题^[7],为三维有限域约束 Delaunay 三角化的实现提供理论依据.

命题 1. 对于二维空间剖分域上的任意一条边 AB ,若存在一个圆过 A, B 两点且该圆内不包含剖分域内的其他点,则线段 AB 必为域 Delaunay 三角化中的一条边.

命题 2. 对于三维空间剖分域上的任意一条边 PQ ,若存在一个球过 P, Q 两点且该球内不包含剖分域内的其他点,则线段 PQ 必为域 Delaunay 三角化中的一条边.

命题 3. 对于三维空间剖分域中的一个三角面片 $\triangle ABC$,若存在一个球过 A, B, C 三点,且不包含其他点,则面片 $\triangle ABC$ 必为 Delaunay 三角化中四面体的一个面.

依据命题 1,用边界边中点加点法可以实现二维任意域约束 Delaunay 三角化,如图 3 所示,即为中点加点法实现约束 Delaunay 三角化的过程.

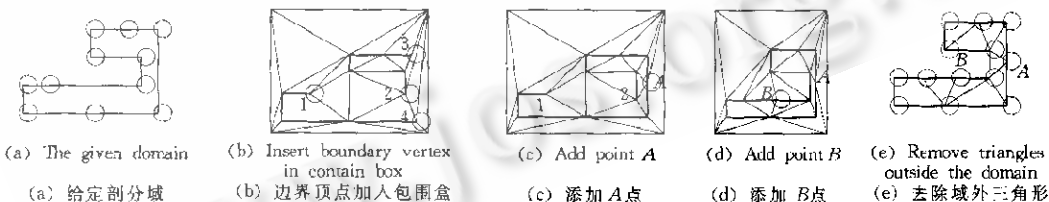


Fig. 3 Delaunay triangulation of a polygon by adding points

图3 加点实现多边形 Delaunay 三角化的过程

命题 1 从理论上说明,在整条边不能满足 Delaunay 空外接圆准则的情况下,通过加点细分,总可以使边界边分段满足空外接圆准则,最终实现边界一致的 Delaunay 三角剖分.在边界边上加点有很多种,在边界边中点处添加辅助点是实际应用中易于操作的方法之一.

命题 2 和命题 3 说明,在三维空间只要给定数据域的边界边和边界面片存在空外接球,就可以实现三维约束 Delaunay 三角化.三维空间任意数据域的边界能否满足命题 2 和命题 3 的条件可以通过“共边的两面”和“共顶点的多面”来分析,如图 4~6 所示,共边三角形面片和共顶点三角形面片能否存在空外接球是任意域约束 Delaunay 三角化的关键.

对于两面相交的情形,在交线两边,三角形有两种相交情形,一种是共边,另一种是共顶点. $\triangle ABC$ 和 $\triangle ABD$ 是边界面 Delaunay 三角化中共边的两个三角形,过 $\triangle ABD$ 的外接圆有无穷多

个球,总存在与 $\triangle ABC$ 的交弧包含于 $\triangle ABC$ 中的球,使 $\triangle ABD$ 满足存在于约束 Delaunay 三角化中的条件.同样, $\triangle ABC$ 也存在这种球. $\triangle EFG$ 和 $\triangle EMN$ 是边界面 Delaunay 三角化中共顶点的两个三角形,过 $\triangle EFG$ 的外接圆有无穷多个球,总存在除 E 点外、与 $\triangle EMN$ 所在面没有其他交点的球,使 $\triangle EFG$ 满足存在于 Delaunay 三角化中的条件.同样, $\triangle EMN$ 也存在这种球.

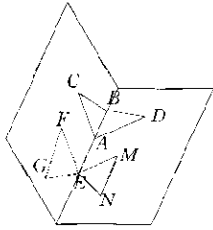


Fig. 4 The intersection case of two planes
图4 两面相交情形

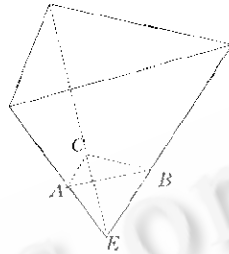


Fig. 5 The intersection case of three planes
图5 三面相交情形

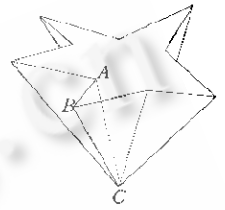


Fig. 6 A folding cone
图6 “折叠”锥

对于三面相交, $\triangle AEC$ 是边界面 Delaunay 三角化中的一个三角形,通过它的外接圆有无穷多个球,其中总存在这样的球,该球与 $\triangle CEB$ 的交弧包含于 $\triangle CEB$,与 $\triangle AEB$ 的交弧包含于 $\triangle AEB$,从而满足 $\triangle AEC$ 在 Delaunay 三角化中存在的条件. $\triangle AEB$ 和 $\triangle CEB$ 存在同样的结论.多面相交形成凸锥的情形与三面相交的结论是相同的.

若多面相交形成如图 6 所示的“折叠”锥,则以锥点为顶点的 $\triangle ABC$ 由于夹在前、后两个面之间,不能满足命题 2 和命题 3 的条件,这种情形不能实现 Deluanay 三角化,可在工程误差范围内去除锥点或锥点附件保留非 Delaunay 三角化.在应用中存在铸造圆角和倒角,很少出现上述情况,本文命题的条件在绝大多数工程实例中是能够满足的.

由于存在上述不能 Delaunay 三角化的“折叠”锥,所以,我们将由本文的算法能够实现的三角化称为三维有限域约束 Delaunay 三角化.

3 约束 Delaunay 三角化算法的实现

算法的总体思路是:用一个包围盒包围待剖分域,将其转化为凸包的三角化,先将剖分域的边界点按凸域点集的三角化方法逐点加入,得到以包围盒为假设边界的 Delaunay 三角化,然后追踪恢复剖分域的原始边界,剔除不需要的四面体,最后插入域内的数据点,并进行网格匀称处理.

3.1 算法步骤

- (1) 原始数据预处理(重点、边界自相交、不封闭、包围盒);
- (2) 剖分域外表面进行二维 Delaunay 三角化(应用命题 1);
- (3) 将边界面数据点加入包围盒中,得到只包含边界数据点的初始三角化;
- (4) 追踪边界边、边界面(应用命题 2 和命题 3);
- (5) 删除多余四面体;
- (6) 插入内部节点,内部节点可以是域剖分生成的节点,也可以是点集剖分给定的数据点;
- (7) 按网格质量、尺度要求进行加点优化;
- (8) 网格 Laplacian 匀称.

算法的核心部分有两点:

- 在已有四面体网格中加入一点后新生成的四面体及其后局部换面优化;

• 如何使边界边、边界面在 Delaunay 三角化中存在.

3.2 插入点的算法过程

```
BOOL Net3D::InsertAnPt(Pt * pPt)
```

```
{
    void * pNode;          //记录点所在的四面体
    Patch * pPatch;       //记录四面体的面片
    int position;        //记录点在四面体上的位置
    PatchStack * pStack;  //保存待交换面的栈空间

    pNode = PtInWhichTetrahedral(pPt, position); //判别点在哪一个四面体内
    switch (position)
    {
    case 在四面体内:      pStack = SplitTetrahedral(pPt, (Tetrahedral *) pNode); //情形 I break;
    case 在四面体面上:   pStack = SplitPatch(pPt, (Patch *) pNode); //情形 II break;
    case 在四面体边上:   pStack = SplitEdge(pPt, (Tetrahedral *) pNode); //情形 III break;
    case 与四面体重顶点: m_pPtSet -> SvsDel(pPt); break;
    default: return FALSE;
    }
    while (pSwapPatchStack != NULL) //记录待交换面的栈不为空
    {
        从栈中弹出一个待交换面;
        进行空外接球测试;
        if(符合优化条件)
            从栈中删除;
        else
            调用换面子函数 swapPatch (pPatch, pStack);
    }
}
```

点加入四面体的 3 种情形如下(如图 7 所示):

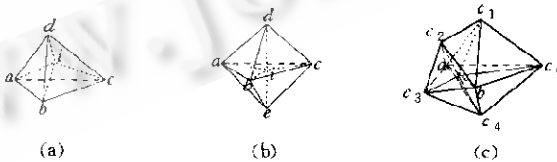


Fig. 7
图7

情形 I (如图 7(a)所示)——插入点位于四面体的内部,从三角化中,去除四面体 $abcd$,加入新产生的四面体: $abde, dbce, acde$,将新生成的面加入优化队列中进行优化测试,不符合优化准则的将进行局部变换.

情形 II (如图 7(b)所示)——插入点位于四面体的面上, abc 为内面,从三角化中删除共享此面的两个四面体 $abcd, abce$,新生成 6 个四面体: $abdi, bcdi, acdi, abei, bcei, acei$,将新生成的面加入优化队列中进行优化测试,不符合优化准则的将进行局部变换.

情形 II (如图 7(c) 所示)——插入点位于四面体的边上. 若插入点 i 位于边 ab 上, 则共享此边的所有四面体 $abc_0c_1, abc_1c_2, abc_2c_3, abc_3c_4, abc_4c_0$ 都将一分为二, 则原四面体从三角化中删除, 新生成的四面体有: $ic_0c_1a, ic_0c_1b, ic_1c_2a, ic_1c_2b, ic_2c_3a, ic_2c_3b, ic_3c_4a, ic_3c_4b, ic_4c_0a, ic_4c_0b$; 将新生成的面加入到优化队列中进行优化测试, 将不符合优化准则的进行局部变换.

3.3 换面优化

插入点后, 新生成的面需经过局部换面优化, 其算法过程如下:

```
SwapPatch(Patch * pPatch, PatchStack * pSwapPatchStack)
{
    Patch * pLeftPatch;
    Patch * pRightPatch;
    Pt * pPt;

    switch(pPatch->GetSwapType(pLeftPatch, pRightPatch, pPt)) //确定变换类型
    {
        case SWAP2TO3: //变换类型 I
            return DoSwap2TO3(pPatch, pStack);
        case SWAP3TO2: //变换类型 II
            return DoSwap3TO2(pPatch, pLeftPatch, pRightPatch, pPt, pStack);
        case SWAP2TO2: // 变换类型 III, 变换类型 IV
            return DoSwap2TO2(pPatch, pLeftPatch, pRightPatch, pPt, pStack);
        default:
            return FALSE;
    }
}
```

下面, 给出 4 种变换类型 (如图 8 所示):

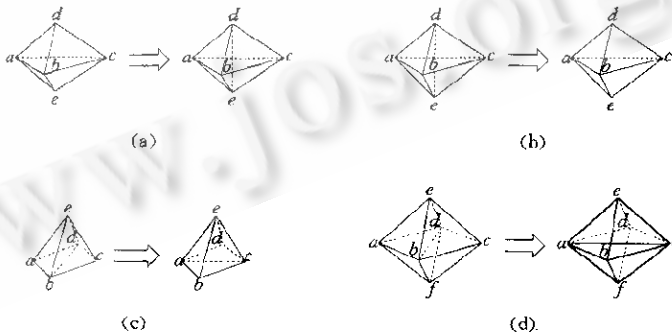


Fig. 8
图 8

变换类型 I (如图 8(a) 所示). de 交 $\triangle abc$ 的内部, 原两个四面体 $abcd$ 和 $abce$ 被 3 个四面体 $abde, bcde$ 和 $acde$ 取代. 内面 $\triangle abc$ 为可换面;

变换类型 II (如图 8(b) 所示). de 交 $\triangle abc$ 的内部, 原 3 个四面体 $abde, bcde$ 和 $acde$ 被两个四面体 $abcd$ 和 $abce$ 取代;

变换类型 III (如图 8(c) 所示). a, b, c, d 共面, 若 $\triangle abd$ 和 $\triangle bcd$ 是边界面, 四面体 $abde, dbce$ 被

四面体 $abce, dace$ 取代. 内面 $\triangle ebd$ 为可换面;

变换类型 IV (如图 8(d) 所示). a, b, c, d 共面, e, f 位于面的两侧, 四面体 $abde, bcde$ 被 $abce, dace$ 替代; 四面体 $abdf, dbcf$ 被 $abcf, dacf$ 取代.

在追踪边界面时, 对于未在 Delaunay 三角化存在的边界面片在其最长边中点处添加辅助点, 缩小其外接球, 使之满足命题 3 的条件, 最终实现三维约束 Delaunay 三角化. 不满足条件的“角点”, 保留非 Delaunay 三角化.

4 应用实例

如图 9~12 所示为应用本文算法生成的网格剖分实例. 其中, 图 9 是一组包含层面和断面的石油地质勘探数据. 硬件平台为 PII233 微机、32M 内存.

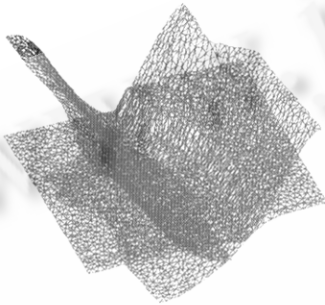
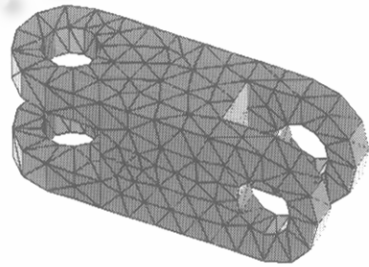


Fig. 9 The triangulation of geological data of oil exploration

图 9 石油地质勘探数据的三角化

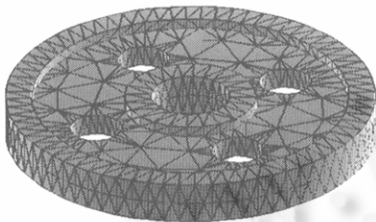


(324 points, 750 tetrahedra, CPU time: 1 minute)

(324 个点, 750 个四面体, 剖分时间: 1 分钟)

Fig. 10 The hidden mesh of a mechanical part

图 10 零件网格消隐图



(997 points, 2 478 tetrahedra, CPU time: 10.5 minutes)

(997 个点, 2 478 个四面体, 剖分时间: 10.5 分钟)

Fig. 11 The hidden mesh of a mechanical part

图 11 零件网格消隐图



(1 805 points, 5 118 tetrahedra, CPU time: 33 minutes)

(1 805 个点, 5 118 个四面体, 剖分时间: 33 分钟)

Fig. 12 The hidden mesh of a mechanical part

图 12 零件网格消隐图

5 算法分析

Delaunay 三角化的算法效率主要取决于寻找包含新加入点的网格单元, 若不加任何策略, 则三维 Delaunay 三角化算法复杂度最坏情形为 $O(n^3)$ (n 是点数). 若采用一般链表查询算法, 则三维三角化的算法效率一般为 $O(n(\log n)^2)$ ^[8]. 本文采用基于四面体侧面法矢的“指南针”快速查询方法, 如图 13 所示, 从任一个四面体开始, P_w 为四面体的重心, P_N 为新插入点, 搜索沿其侧面法矢与

$P_W P_N$ 夹角小的方向搜索,即沿 $\triangle ACD$ 的相邻四面体搜索,采用这种方法可以大幅度地提高算法速度,算法复杂度优于 $O(n^3)$ 和 $O(n(\log n)^2)$.图14即为图10所示的零件从300~1000点的测试时间曲线,曲线两端上升快,中间比较平坦,其原因在于当数据点较少时,体现不出快速查询法的优点;而当数据点太密时(在一定空间内加),每加入一个新点,则较多四面体的外接球包含该点,用于换面优化的时间增多.总体上来讲,快速查询法的效果是明显的.

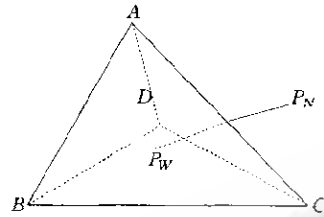


Fig. 13 The searching method based on the normal vector
图13 基于法矢的查询法

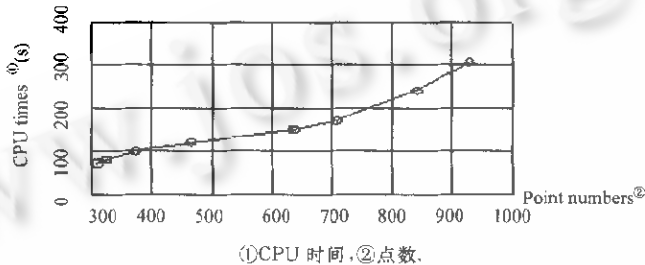


Fig. 14 The CUP time table (P II 233, 32M)

图14 图8所示零件的剖分时间表(P II 233, 32M)

6 结 论

本文基于边界边、边界面片在Delaunay三角化中的存在条件,设计出有限域约束Delaunay三角化的算法,在石油地质勘探数据和机械零件的网格剖分中的应用表明,本文的算法的网格剖分速度和精度达到了实用要求,通过为本文研究的网格剖分系统提供数据交换接口,可以将目前优秀CAD软件的造型功能、有限元软件的分析功能综合在一起,充分发挥现有软件在复杂数据场分析中的作用.

References:

- [1] Bowyer, A. Computing dirichlet tessellations. *The Computer Journal*, 1981, 24(2):162~166.
- [2] Waston, D. F. Computing the n -dimensional Delaunay tessellation with application to voronoi polytopes. *The Computer Journal*, 1981, 24(2):167~172.
- [3] Sloan, S. W. A fast algorithm generating constrained Delaunay triangulations. *Computers and Structures*, 1993, 47(3):441~450.
- [4] Hazlewood, C. Approximating constrained tetrahedrizations. *Computer Aided Geometric Design*, 1993, (10):67~87.
- [5] Weatherill, N. P., Hassan, O. Efficient three-dimensional Delaunay triangulation with automatic point creation and imposed boundary constraints. *International Journal of Numerical Methods in Engineering*, 1994, 37(2):2005~2039.
- [6] Zhou, Xiao-yun, Liu, Shen-quan. Robust algorithm of constrained Delaunay triangulation. *Chinese Journal of Computers*, 1996, 19(8):615~624 (in Chinese).
- [7] Xu Yong-an. Research of the key problems, algorithms and application of constrained Delaunay triangulation [Ph. D. Thesis]. Zhejiang University, 1999 (in Chinese).
- [8] Barty, Joe. Construction of three-dimensional Delaunay triangulations using local transactions. *Computer Aided Geometric Design*, 1991, (8):123~142.

附中文参考文献:

- [6] 周晓云,刘慎权.实现约束Delaunay三角剖分的健壮算法.计算机学报,1996,19(8):615~624.
- [7] 徐永安.约束Delaunay三角化的关键问题研究与算法实现及应用[博士学位论文].浙江大学,1999.

The Algorithm of 3D Constrained Delaunay Triangulation

XU Yong-an^{1,2}, YANG Qin³, WU Zhuang zhi³, CHEN Qi-ming³, TAN Jian-rong²

¹(HiSense Engineering Institute, Qingdao 200671, China);

²(Department of Mechanism, Zhejiang University, Hangzhou 310027, China);

³(Department of Computer Science and Engineering, Beijing University of Aeronautics and Astronautics, Beijing 100083, China)

E-mail: xuyongan@hisense.qd.sd.cn; visual@buaa.edu.cn

<http://www.hisense.com.cn>

Received December 21, 1998; accepted September 29, 1999

Abstract: In this paper, the boundary conformability of 3D constrained Delaunay triangulation is analyzed. Based on the theory of Delaunay triangulation, the feasibility of 3D constrained Delaunay triangulation is discussed emphatically. The algorithm of 3D constrained Delaunay triangulation of finite domain is designed. Applications in oil and geological exploration and mechanical part design are illustrated. The algorithm plays an important role in the computation and analysis of complicated objects.

Key words: triangulation; Delaunay triangulation; constrained Delaunay triangulation