基于球填充法的STL模型曲面自适应网格生成

本文详细地给出了识别STL模型的线曲率、面曲率和邻近特征的具体算法。采用八叉树做背景网格来建立尺寸场信息，叶子节点存储的尺寸大小不是节点长度，而是包含节点长度的尺寸区间，以此来使尺寸场过渡更加光滑。利用球填充算法在STL曲面生成网格节点。通过基本的添加节点和删除节点操作，用新产生的节点替换初始的STL网格节点，然后通过边翻转技术生成质量较高的网格。该算法不需要点面投影、前沿判交等复杂计算，能高效的生成STL曲面自适应网格。

关键词：STL 自适应 曲面网格生成 网格尺寸 球填充

The paper presents detailed algorithms to recognize curvatures of edges, curvatures of surfaces and proximity features of stereolithography (STL) model. And a size filed is built by octree. This field is smoothed more by … New mesh nodes are obtained by ball-packing method, and then the new nodes replace the initial nodes of STL model through two basic operations: adding node and deleting node. Finally, through element-swapping and node-smoothing, a conforming size-adaptive mesh obtained. This algorithm can generate a good conforming size-adaptive mesh efficiently without complex calculation of projection of nodes and intersection of fronts.

Key words: STL Adaptive Surface mesh generation Mesh size Ball-packing

0前言

三角形网格在计算机图形学、几何造型和有限元分析建模等方面具有广泛的应用。网格生成一般以CAD软件导出的模型为输入，STL格式文件是最通用的格式之一。STL格式产生于快速成型技术，用一系列三角面片表示模型，可以方便地用于模型渲染和文件交换。CAD软件直接产生的STL文件是一种质量较差的网格，包含很多形状细长的三角面片，不能够直接用于计算分析，必须对其重新剖分，以生成高质量的曲面网格。

很多模型都包含复杂的几何特征，如高曲率特征，几何微小邻近特征。为了保证网格的质量和有限元计算的精度，往往要求在模型复杂特征区域采用较小的单元尺寸。均匀的小尺寸单元虽然可以保证满足网格的质量和计算的精度，但会在过渡平坦的区域产生过多的单元，从而极大地降低程序的时空效率。因此，生成几何自适应曲面网格就十分必要。

几何自适应曲面网格生成的关键在于自动识别模型的几何特征，产生单元尺寸信息，并对各部分尺寸信息综合优化，从而产生过渡良好的尺寸控制场信息。常考虑的几何特征包括曲率特征和区域形状特征[]。曲率特征的自动识别相对简单，只需要计算曲面上任意点的曲率。区域形状特征主要是指区域形状尺寸相对于整体尺寸特别小，具体体现为组成该区域的几何元素彼此靠得过近。区域形状特征的自动识别则较为复杂，主要有两种方法。第一种方法利用模型实体的中轴到形体边界的距离识别区域形状特征。这种方法可以利用计算几何中关于中轴的许多理论和算法。Persson P O[1]和Quadros W R[2]等基于中轴转换思想实现了邻近特征的快速准确识别。梁义等[3]针对曲面中轴难以求得的问题引入黎曼度量，实现参数曲面上的约束Delaunay三角化，获得了曲面形体的离散中轴，将其用于曲面邻近特征的识别。第二种是直接计算几何模型边界各种元素之间的距离。这种方法是通过计算形体边界的几何元素的距离进行检查。Cunha A等[4] 和Zhu J 等[5]。

尺寸控制场信息一般采用曲面三角网格或者八叉树为背景网格来存储。陈建军等[6]以邻近特征计算时建立的边界采样点集的约束Delaunay三角为背景网格，建立了覆盖整个曲面的单元尺寸场。同时采用拉普拉斯法光滑尺寸场。Quadros W R[2]采用八叉树来存储尺寸场信息，再通过插值的方式是尺寸场光滑。

STL模型的自适应网格生成研究相对较少。生成方法可以分为映射法和直接法。映射法的主要思想是恢复STL模型的参数信息，在曲面的参数域内生成网格，再将其映射回三维空间得到曲面网格。映射法的效率较高，不足是存在映射畸变。基于直接法的网格生成直接在三维空间进行，避免了映射法所存在的映射畸变问题。如Béchet E 等[7]结合Delaunay准则和边分裂操作生成曲面网格。Wang D 等[8]则综合考虑了利用边分裂、边交换和边折叠生成曲面网格。这些方法都是在原来STL三角网格的基础上操作，新生成的网格质量极大地受限于初始的STL网格。效果较好的直接法大都采用前沿推进的技术，如文献[9-10]。这些方法都需要做点到曲面的投影计算以及前沿的判交等操作。虽然文献[11-13]对前沿判交做了加速策略，这种方法相较于映射法仍然非常耗时。

球填充法[14]是刘剑飞提出的一种前沿推进法，它假想小球按照形体模型顶点、边和面的顺序依次填充形体表面，填充完毕后，球心处即节点的位置，最后在连接相邻球的球心就可以得到最终的网格。本文基于球填充法提出一种处理STL模型的曲面自适应网格生成方法。这种方法首先识别曲率特征及区域形状特征，接着根据特征构造尺寸场并采用八叉树背景网格存储尺寸场，最终采用球填充法生成自适应网格。

1. 球填充法概述

为了给读者一个总体印象, 本节先介绍球填充法进行网格剖分的基本框架。



图1 球填充法布顺序：（ 球填充法布顺序：（ 球填充法布顺序：（ 球填充法布顺序：（ 球填充法布顺序：（ a）三维几何模型；（ ）三维几何模型；（ ）三维几何模型；（ ）三维几何模型；（ ）三维几何模型；（ b）从模型顶点开始布球；（ ）从模型顶点开始布球；（ ）从模型顶点开始布球；（ ）从模型顶点开始布球；（ ）从模型顶点开始布球；（ c）继而在 ）继而在 模型边缘布球；（ 模型边缘布球；（ 模型边缘布球；（ d）将球布满模型表面。 ）将球布满模型表面。 ）将球布满模型表面。 ）

1算法流程

本文在球填充法的框架里嵌入几何特征识别及尺寸场存储，提出一种曲面网格自适应算法，其具体算法流程如下：

1.拓扑重建。读取STL文件，构件模型缺失的拓扑信息（如面片相邻关系）。

2.几何重建。识别模型的几何信息（如边，面），按照面将模型分为多个子域。

3.尺寸场的建立。计算模型曲率，识别邻近特征，生成尺寸场信息，建立八叉树背景网格。

4.网格生成。

(1).利用球填充方法生成网格节点。

(2).将网格节点添加到初始STL网格中。

(3).删除初始STL网格节点。

(4).利用边翻转生成最终网格。

2拓扑信息、几何信息的重建

STL文件记录每个三角片顶点的坐标信息，存在大量的冗余，却缺少拓扑和几何信息。剖分STL模型需要改变三角片的存储结构，恢复缺失的拓扑和几何信息。

2.1拓扑信息的恢复

STL文件几何信息的恢复以及几何特征的计算都需要完善的拓扑信息，需要恢复的拓扑信息包括：

2.1.1面片相邻关系

设T={Ti. i=1，Ne}为三角片的集合，Tij（j=1，2，3）为第i个三角片与第j个顶点相对邻接的三角片。当没有三角片与第j个顶点相对时，记Tij=-1。恢复面片邻接关系就是寻找所有三角片T的相邻三角片Tij，算法的基本思想基于相邻的三角片共用相同的线段，线段可以被两端点唯一确定。Lo在文献15给出了具体的算法，在此不再赘述。

2.1.2点线段相邻关系

恢复点线段相邻关系比较简单，只需要遍历所有的线段，将每条线段记录到两端点所对应的相邻线段集合Sp中。

2.2几何信息的恢复

STL网格生成需要模型的几何边和几何面的信息。几何边由相关的线段集合Se表示，几何面由相关的三角片集合Tf表示。

2.2.1几何边的恢复

（1）首先利用Béchet E 等[7]的建议，遍历每一条线段，计算共用该线段的三角片的二面角的大小，将二面角θ大于20的线段记为线段。

（2）恢复某一条几何边的具体步骤如下：

（a）从标记线段中选择一个种子线段，加入到线段集Se。将种子线段的两端点加入到端点集V。

（b）从V选取一个端点，根据点线段相邻关系，判断与当前端点相邻的其他标记线段的个数，若为1，则选取与当前端点相邻的其他标记线段加入到Se，将标记线段的另一端点加入到V。

（c）重复过程（b），直至V为空时结束。当前几何边由Se中的线段组成。

2.2.2面的恢复

（1）从面片列表中选取一个种子面片，将其加入面片集Tf。

（2）从Tf选取某个三角片做当前面片。根据面片相邻关系选取与当前面片相邻的三角片，若选取的相邻面片与当前面片共用的线段为非标记线段，则将选取的面片加入到Tf。重复该过程，直至Tf为空停止。当前几何面由所有加入过Tf的三角片组成。

3几何自适应尺寸场的建立和光滑

3.1线曲率特征尺寸值计算

本文识别的STL模型几何边采用前后相连的线段表示。除去几何边两端点剩余的线段节点选为线曲率的采样点。采样点*S*的线曲率计算方法如图1。与*S*相连的线段分别为*AS*和*SB*，线段*AS*和*SB*的中垂线相交于点*O*，曲率半径*R*s = |*SO*|。*S*点的曲率特征尺寸值为：

 (1)

式中为用户参数，缺省值为1.0。

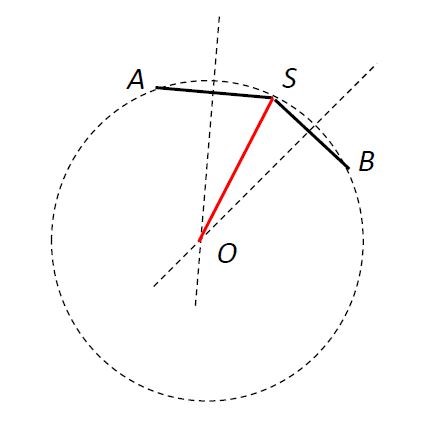
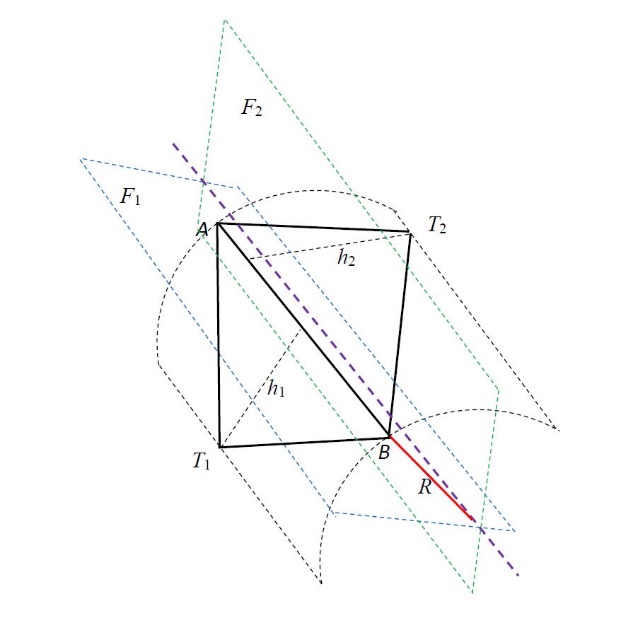
 

图1 线曲率计算示意图 图2 面曲率计算示意图

3.2 面曲率特征尺寸值计算

Béchet E 等[7]根据面片法矢量快速估计曲面曲率，Wang D 等[8]通过构建局部二次曲面计算曲面曲率。本文利用柱曲率来表示面曲率。如图2，*T*1和*T*2为采样三角片，*AB*为它们的共有线段，*h*1为三角片*T*1线段*AB*的高，*h*2为三角片*T*2线段*AB*的高，*F*1为过*h*1中点与*h*1垂直的面，*F*2为过*h*2中点与*h*2垂直的面，线段*EF*为*F*1与*F*2的交线。*AB*与*EF*的距离*R*即为*AB*线段上的面曲率半径。*AB*线段上的面曲率尺寸值为：

 (2)

式中为用户参数，缺省值为1.0。以用户设置的最小尺寸*h*min为间隔分割*AB*，*AB*端点和分割点即为面曲率采样点。

3.3邻近特征尺寸值计算

邻近特征的识别需要计算模型几何边的距离，由于STL模型的几何边由多条相连的线段组成，邻近特征的计算可以转换为计算点与线段的距离。以用户设置的最小尺寸*hmin*为间隔分割几何边线段，几何边端点和分割点即为邻近特征采样点。选取某一采样点*S*，计算其与其他几何边线段的距离。本文我们利用八叉树和设置查找范围来加速搜索速度。设采样点*S*与其他几何边的最短距离为*d*，则采样点*S*的邻近特征尺寸值为：

 (3)

式中为用户参数，缺省值为1.0。

3.4几何自适应尺寸场的光滑

为保证网格的合理过渡，需要根据约束条件对采样点直接产生的尺寸场做光滑操作。本文采用八叉树做背景网格来光滑尺寸场，为了保证相邻叶节点尺寸值不产生2倍的差距，节点存储的网格的尺寸值可为节点长度的0.75-1.25倍，即叶节点所记录的尺寸值小于该节点长度的0.75倍才会分裂，新产生的节点所记录的尺寸值可以为节点长度的1.25倍。为了保证八叉树为强八叉树，本文每设置完某一点p的尺寸值之后，计算某一距离值D= H \*g，式中H为p点处的尺寸值，g为用户设置的网格过度系数，必须大于1.0，缺省值为1.2。选取p点上下左右前后六个方向以距离d延伸p点到pi（i=1，…，6），设置尺寸pi，重复以上过程，直到八叉树不需要分裂为止。

4网格生成算法

（1）分布较好的节点构型往往可以产生高质量的网格；

（2） 很容易通过一定的规则在已有的三角形网格中添加或者删除节点形成新的三角形网格；

（3）可以通过有限次的边翻转优化已有的三角形网格。

本文正是基于以上三个认识发展了STL曲面网格生成算法：利用球填充算法生成新的网格节点（见下文4.1）；通过一定的规则将生成的网格节点添加到初始的STL网格中，再将初始STL网格中的节点删除（见下文4.2）；通过边翻转优化网格，生成最终网格（见下文4.3）。

4.1生成网格节点

Liu[16]提出的球填充算法能够产生较好的网格节点构型，球心位置就是生成的节点位置，本文借鉴了这一算法来生成曲面网格节点。主要分为两个步骤。

4.1.1初始节点的产生

生成几何面内节点之前首先要在几何边上生成初始节点。Liu[16]对初始节点的生成做了详细的研究，我们采用他们的算法，这里不再赘述。生成节点p的半径就是节点p所在的八叉树叶子节点所存储的尺寸值。

4.1.2面内节点的产生

对于曲面自适应网格的生成，填充球的半径会随着球心的位置而改变，确定球心位置的过程是一个迭代的过程。如图3，生成节点S的具体过程如下：

选取已生成的相邻节点S1和S2，最初取S的半径r为（r1+r2）/2，紧邻S1和S2布球，球心p确定为节点S的当前位置。判断由八叉树背景网格确定的点p的半径γ是否与r相等，如果相等，点p所在的位置即为节点S的最终位置；如果不等，以γ代替r，重复上述过程。需要注意的是，当交点p与其他已有的节点距离小于某个阈值（本文设置不可以小于两节点半径之和的0.7倍），则该节点S不可添加。直到该几何面没有可以添加的节点即完成了该几何面的节点生成。

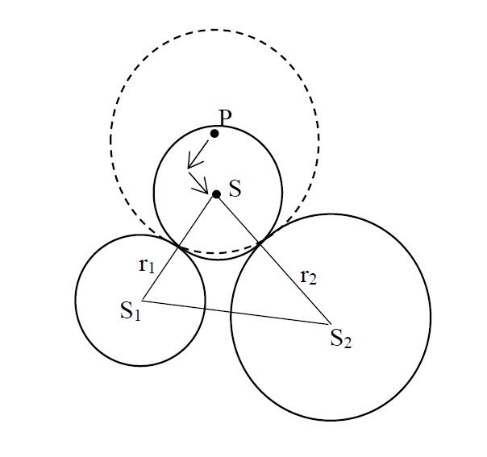


图3 生成面内新节点S过程

4.2添加，删除节点

4.2.1基本操作

（1）一个三角形分成三个三角形

将一点P插入三角形ABC内部，可以三角形ABC分成三角形ABP、PBC和APC。如图【】。

（2）两个三角形分成四个三角形

三角形ABC和三角形BAD共边AB，将一点P插到边AB上，可以将三角形ABC和BAD分成三角形APC、PBC、BPD和PAD。如图【】。

（3）替换节点

一点P插入三角形ABC时正好落在点A附近，三角形ABC被替换为三角形PBC。如图【】。

（4）三个三角形合并成一个三角形

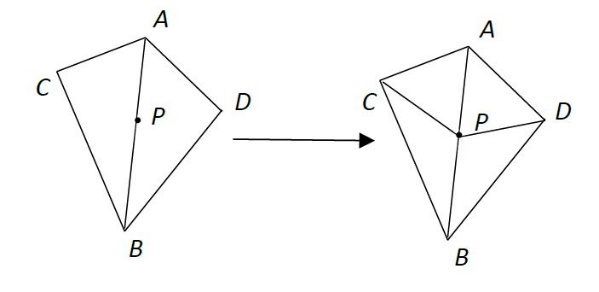
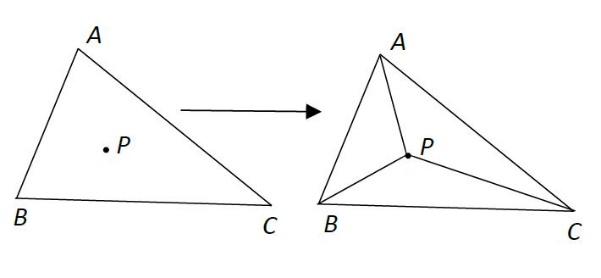
三个三角形包围一点P时，可以将P移除，合并成一个三角形。如图【】。这个操作用于移除网格内部节点。

（5）四个三角形合并成两个三角形

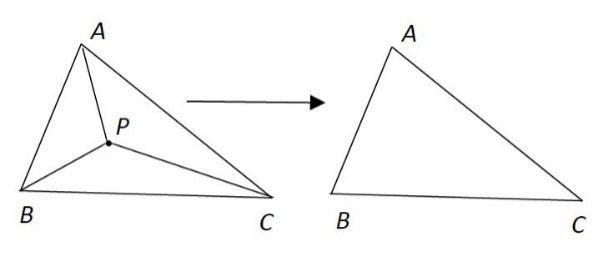
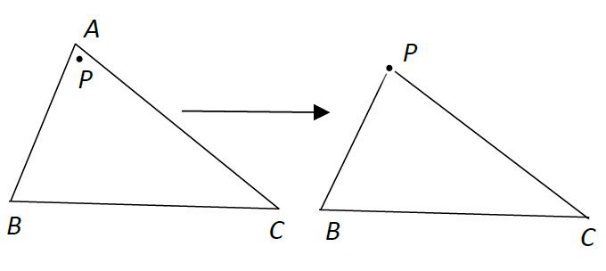
四个三角形包围一点P时，可以将P移除，合并成两个三角形。如图【】。这个操作用于移除边界线段上的节点。

（6）边翻转

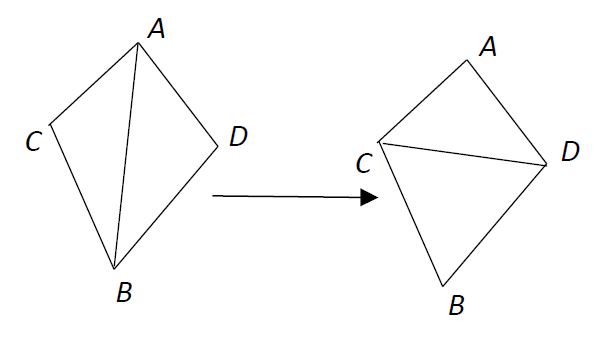
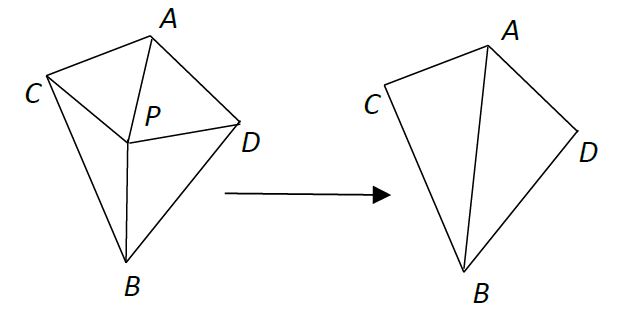
这是一个通过修改网格拓扑结构来改变网格构型的基本操作。如图【】，三角形ABC和三角形BAD共边AB，通过改变对角线AB为CD，可以替换为三角形BCD和三角形ACD。曲面网格的边翻转操作需要满足三个要求：翻转后，<CAD和<CBD小于180度；对角线AB不能为几何边线段；若ABCD不在一个平面内，四面体ABCD体积需要小于某个阈值。基于边翻转的曲面网格优化还需要满足：翻转后三角形BCD和三角形ACD的最小角大于翻转前三角形ABC和三角形BAD的最小角。



安是谁所所多多多多多多多多



安是谁所所多多多多多多多多



安是谁所所多多多多多多多多

新生成网格节点的添加和原始STL网格节点的删除基于上述六个基本操作完成。利用前三个操作可以将新生的网格节点添加到原始的网格中。为了加速后续节点的添加和删除，当前网格需要保持一个良好的构型，因此每添加完一个节点之后，通过操作六对新加入节点周围局部区域做一次优化。所有节点添加完成后，通过操作六使得初始网格节点的度转换为三或者四，然后通过操作四和五将其去除掉。这样就完成了生成的网格节点和初始网格节点的替换。

4.3优化网格

节点替换后的网格并没有达到最优的构型。继续通过操作六对网格进行优化，直到没有对角线可以翻转为止。

5实验结果及分析

结论

参考文献

1.Persson P O. PDE-Based Gradient Limiting for Mesh Size Functions[C]//IMR. 2004: 377-388.

2. Quadros W R, Owen S J, Brewer M L, et al. Finite Element Mesh Sizing for Surfaces Using Skeleton[C]//IMR. 2004: 389-400.

3. 梁义, 陈建军, 陈立岗, 等. 几何自适应参数曲面网格生成[J]. 計算機輔助設計與圖形學學報, 2010, 22(2): 327-335.

4. Cunha A, Canann S, Saigal S. Automatic boundary sizing for 2D and 3D meshes[J]. ASME APPLIED MECHANICS DIVISION-PUBLICATIONS-AMD, 1997, 220: 65-72.

5. Zhu J, Blacker T D, Smith R. Background Overlay Grid Size Functions[C]//IMR. 2002: 65-73.

6. 陈建军, 梁义, 黄争舸, 等. 面向 STL 模型的几何自适应曲面网格生成[J]. 机械工程学报, 2011, 47(7): 128-133.

7. Béchet E, Cuilliere J C, Trochu F. Generation of a finite element MESH from stereolithography (STL) files[J]. Computer-Aided Design, 2002, 34(1): 1-17.

8. Wang D, Hassan O, Morgan K, et al. Enhanced remeshing from STL files with applications to surface grid generation[J]. International Journal for Numerical Methods in Biomedical Engineering, 2007, 23(3): 227-239.

9. Ito Y, Nakahashi K. Surface triangulation for polygonal models based on CAD data[J]. International Journal for Numerical Methods in Fluids, 2002, 39(1): 75-96.

10. Rypl D, Bittnar Z. Generation of computational surface meshes of STL models[J]. Journal of Computational and Applied Mathematics, 2006, 192(1): 148-151.

11. Rassineux A. Generation and optimization of tetrahedral meshes by advancing front technique[J]. International Journal for Numerical Methods in Engineering, 1998, 41(4): 651-674.

12. De Cougny H L, Shephard M S. Parallel volume meshing using face removals and hierarchical repartitioning[J]. Computer methods in applied mechanics and engineering, 1999, 174(3-4): 275-298.

13. De Cougny H L, Shephard M S. Parallel refinement and coarsening of tetrahedral meshes[J]. International journal for numerical methods in engineering, 1999, 46(7): 1101-1125.

14. Liu J F. Automatic triangulation of nD domains[C]//Proceedings of CAD/Graphics. 1991, 91: 238-241.

15.Lo D S H. Finite element mesh generation[M]. CRC Press, 2014.

16. Liu J, Li S, Chen Y. A fast and practical method to pack spheres for mesh generation[J]. Acta Mechanica Sinica, 2008, 24(4): 439-447.