

中国科学技术大学

学士学位论文



扩展多正方形辅助的平面 区域四边形网格生成

姓 名: 拜重阳

院 系: 数学科学学院信息与计算科学系

学 号: PB12001087

导 师: 刘洋

完成时间: 二〇一六年五月

University of Science and Technology of China
A dissertation for bachelor's degree



Generalized PolyQuad Aided
Planar Quaccdrilateral Mesh
Generation

Author : Chongyang Bai
Department :Information and Computing Science
Student ID : PB12001087
Supervisor : Dr. Yang Liu
Finished Time : May, 2016

致 谢

在中国科学技术大学完成本科学业的四年里，我的学习和研究都得到了系里老师和同学的许多指导和帮助。在此，我要向他们表达诚挚的谢意。

我大学的第四年是在微软亚洲研究院度过的，这里的主管研究员刘洋博士指导我完成了毕业设计和其他的研究工作。你细致的指导和亲切的关怀让我非常感动，你严谨的研究态度和敏锐的科学洞察力使我受益良多。还要感谢在这里指导我、帮助我的童欣老师，以及傅孝明师兄。

感谢中国科学技术大学和微软亚洲研究院，感谢一路陪伴我的老师和同学们。最后，还要谢谢我亲爱的家人，你们一直以来的支持和关怀是我不断向前的动力。

拜重阳

2016 年 5 月 28 日

目 录

致 谢	I
目 录	III
摘 要	V
ABSTRACT	VII
第一章 绪论	1
1.1 扩展多正方形结构	1
1.2 平面四边形网格生成技术	1
1.2.1 四边形网格的应用	1
1.2.2 四边形网格的质量	2
1.2.3 相关工作	4
1.3 计算无翻转映射	5
1.4 本文的贡献及组织	6
第二章 扩展多正方形结构生成	9
2.1 法向光滑和对齐的网格变形	9
2.1.1 法向对齐能量	9
2.1.2 刚性形变能量	10
2.1.3 降低法向对齐能量的预旋转	10
2.2 扩展多正方形结构的提取与网格压平	11
2.2.1 扩展多正方形结构的提取	11
2.2.2 网格压平	13
2.3 网格变形算法和优化	13
第三章 基于中轴线分割的平面四边形网格生成	15
3.1 中轴线的概念	15
3.2 基于中轴线的简单子区域的产生	15
3.2.1 分割区域去除凹角	15
3.2.2 基于中轴线的区域细分	17
3.2.3 讨论与比较	18
3.3 整数规划生成四边形网格	19
3.3.1 合理的 3、4、5 边形区域的产生	19
3.3.2 整数规划生成四边形网格	20

第四章 计算原区域的四边形网格	25
第五章 2 维三角形或四边形网格的优化	27
5.1 目标函数设计	27
5.1.1 同时解决翻转网格和提高网格质量的度量	27
5.1.2 以直角为目标的四边形网格度量	28
5.2 网格优化算法	29
5.2.1 扩展多正方形结构三角网格的优化	29
5.2.2 四边形网格的优化	29
第六章 实验与结论	31
6.1 实验与比较	31
6.1.1 可控的奇异性	31
6.1.2 高质量且奇异点较少的四边形网格	31
6.2 总结与展望	32
参考文献	35

摘要

有限元方法是一项基本的数值技术，它在工程中有着广泛的应用。很多工程问题归结为解不规则区域上的偏微分方程，由于难以找到解析解，常常利用有限元方法在离散化的不规则区域上用数值解去逼近真实的解。近年来，我们已经有了高效的求解算法和计算能力强大的计算机，但仍然要花费大量的时间、甚至手动地把复杂的区域离散化为有效的有限元网格。在工业应用中，三角形网格和四边形网格都可以作为二维有限元分析的数据输入，但一般而言使用四边形网格进行计算具有更高的精度和更快的速度，然而，想要全自动地生成规则的四边形网格也要困难的多。

近年来，研究者们已经提出了一些平面四边形网格的生成方法，但对于复杂且不规则的平面区域，它们要么不够鲁棒或不能全自动化，要么会产生过多的奇异点。因此复杂的边界是对算法的鲁棒性、高效性和产生网格的质量的巨大挑战。为此，本文的思想是先对输入的区域做出简化，把它变形为扩展多正方形结构并且得到两者之间的一个无翻转映射，再利用现有的算法在简化的区域上生成高质量的四边形网格，最后把网格映射回原始的区域。另外，本文还采用了先进的优化算法在确定拓扑结构后提高四边形网格的质量。我们采用基于中轴线的几何分割方法生成四边形网格，验证了这一思想，能快速、鲁棒地产生奇异性可控的2维四边形网格。

关键词：平面四边形网格生成 中轴线 扩展多正方形结构 无翻转映射 变形 网格质量提高

ABSTRACT

The finite elements method (FEM) is a powerful numerical technique which is widely applied in engineering. Many technical problems are abstracted as solving partial differential equations on irregular domains. Because it is usually hard to find analytical solutions, FEM is employed to calculate numerical solutions in discretized irregular domains. Although there are efficient computational algorithms and powerful computers over the past years, discretizing complicated geometry to a valid finite element mesh is a time-consuming work, and may need complex user interaction. Triangular and quadrilateral mesh are commonly used as data preparation in FEM, but in general the use of a quadrilateral mesh has better quality and converges faster than a triangle mesh. However, it is more difficult to generate a quadrilateral mesh automatically.

Recently, various planar quadrilateral mesh generation methods have appeared. However, they are either not fully automatic and robust enough, or will produce too many singularities for complex and irregular planar domains. So, complicated boundary is a great challenge to the robustness, efficiency and quality of quadrilateral mesh of all these methods. To solve this problem, our idea is simplifying the input first, i.e. deforming it to a generalized PolyQuad, computing a inversion-free mapping between them at the same time, then using existed algorithms to generate a quadrilateral mesh of good quality, finally mapping the mesh back to the initial domain. In addition, an advanced mesh optimizing algorithm is employed to improve mesh quality after the topology is determined. A medial axis based geometry decomposition method is used to prove our algorithm, which generates singularity controllable planar quadrilateral meshes efficiently and robustly.

Keywords: 2D quadrilateral mesh generation medial axis generalized PolyCube deformation inversion-free mapping deformation mesh quality improvement

第一章 绪论

本章首先定义了扩展多正方形结构，接着回顾了平面四边形网格的生成及应用，并介绍了无翻转映射这一概念。本文的主要思想是把无翻转的映射作为输入区域和简化区域——扩展多正方形结构之间的桥梁，在扩展多正方形结构上生成四边形网格后，再映射回原始区域得到需要的四边形网格。

1.1 扩展多正方形结构

平面区域的多正方形结构和空间区域的多立方体结构 (PolyCube) 的边界的法向严格与坐标轴对齐(平面中有 4 个方向，空间中有 6 个)。由于具有良好的垂直性质和规则的形态，它们可以作为任意的平面和空间形状的简化和近似。但有时仅用 4 个或 6 个方向逼近原始区域是不够的，我们希望加入更多的方向，让这样的简化结构具有更丰富的表达能力。首先给出

定义 1.1.1. 扩展多正方形结构 M 是指这样一种平面区域， $\forall P \in \partial M, \mathbf{n}(P) \in \{(\cos \frac{2\pi i}{N}, \sin \frac{2\pi i}{N})^T | i = 0, \dots, N - 1\}$ 。

由于我们希望逼近任意的平面区域，所以把多正方形的四个法向自然地扩展为任意均匀分布的法向集合。显然， $N=4$ 时它就是多正方形结构， N 越大越能逼近复杂的区域。

1.2 平面四边形网格生成技术

1.2.1 四边形网格的应用

工程上的许多分析和模拟都归结为在连续的物理系统中求解偏微分方程。比如模拟飞机在不同的气流环境中的飞行表现、计算集成电路中的电场分布、利用波动方程分析油气储集层的地质情况等。由于很多偏微分方程难以找到解析解，人们常常利用计算机模拟这些系统，先在离散化的定义域上建立差分方程模型，再求出满足精度的数值解。为了在这些离散的点组成的平面、曲面或空间中利用点的邻域近似地计算导数，我们需要把这些连续的定义域离散化为网格。图1.1展示了平面和空间中常见的网格模型。

有限元方法是求解不规则区域上的偏微分方程数值解的有力技术。作为它的输入，平面和曲面上的区域被离散化为尽量规则的三角形或四边形网格，而空间区域则被离散化为四面体或六面体网格。一般而言，平面区域上的四边形网格比三角形网格具有更高的计算精度和更快的收敛速度。随着计算机的迅猛发展，飞速提升的运算速度使得计算数值解越来越快，但生成复杂区域上的高质量四边形网格仍然要花费大量的时间，常常需要进行繁琐而艰辛的用户交互。因此，自动、快速、鲁棒地生成高质量的四边形网格是进行有限元分析的重要前提。

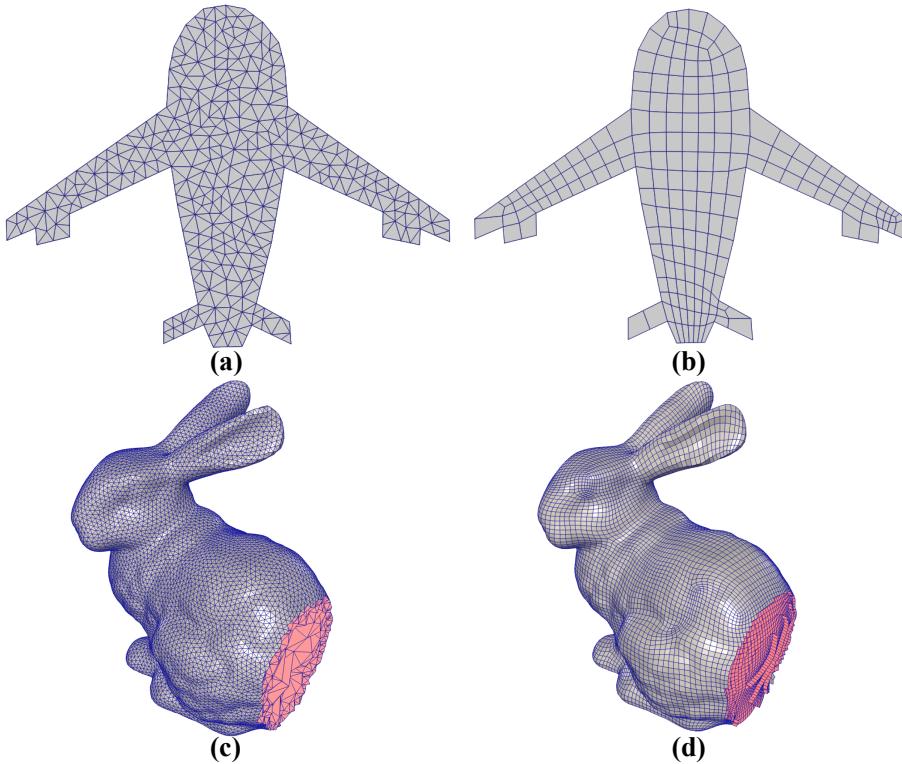


图 1.1 平面和空间中常见的网格模型。(a),(b) 分别为平面三角形网格, (c),(d) 分别为三维的四面体和六面体网格, 它们的表面分别形成了三维的三角形和四边形网格, 粉色部分给出了内部四面体和六面体的截面。

1.2.2 四边形网格的质量

满足如下条件的四边形网格是有限元方法的良好输入:

1. 不含翻转的四边形: 无效的翻转单元会严重影响数值解的精度以及收敛性。我们给出

定义 1.2.1. 如图 1.2, 把四边形的顶点按逆时针顺序记为 v_1, v_2, v_3, v_4 , 考虑三角形 $\triangle v_1v_2v_3, \triangle v_2v_3v_4, \triangle v_3v_4v_1, \triangle v_4v_1v_2$, 若它们的有向面积有至少一个为负, 则称四边形是翻转的, 我们称没有翻转四边形的网格是有效网格。

2. 接近原始边界: 离散化的四边形网格在边界上要和连续的边界曲线尽量接近, 否则会导致数值解的误差过大而不能满足需求。
3. 规则的四边形单元: 越接近正方形或矩形的四边形单元越规则, 规则的四边形单元会提高有限元方法的计算精度和速度。因此, 通过考察四边形单元的角度是否接近直角, 我们可以衡量四边形网格的质量好坏。
4. 较少的奇异点:

定义 1.2.2. 四边形网格的内部点中, 度为 4 的称为规则点, 度不为 4 的称为奇异点。

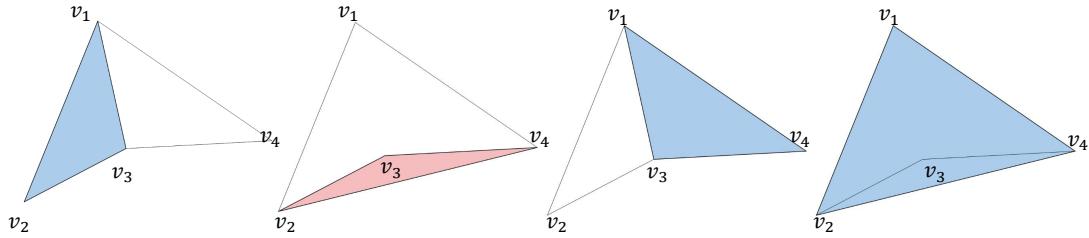


图 1.2 翻转的四边形单元 $v_1v_2v_3v_4$ 。蓝色三角形的有向面积为正，红色的为负。

四边形网格中的奇异点会减慢解方程的速度，并导致数值解的不稳定性，但在一些几何特征(如分叉、凸起)附近的奇异点又能有效地控制网格的扭曲，[1] 中针对具体的应用做了详细解释。因此，生成四边形网格时应避免不必要的奇异点。

一种提高四边形网格质量的方法是保持它的拓扑结构，优化点的位置。为此，人们设计了很多代数度量，把四边形的质量表达成点坐标的函数从而进行优化。显然，这些度量需要是光滑的，具有全局的旋转、平移、伸缩不变性，并且当且仅当四边形为正方形或矩形时达到最大，四边形退化时达到最小。常用的代数度量都是基于网格单元与目标单元之间仿射变换的雅可比矩阵来定义的。

首先考虑任意 $\triangle v_0v_1v_2$ ，对应地，记它在网格中的坐标为 $\mathbf{x}_i = (x_i, y_i)^T$ ，它的目标单元顶点坐标为 \mathbf{x}_i^g ，它的参考单元顶点坐标为 $\mathbf{x}_i^r, i = 0, 1, 2$ ，三者的变换关系见图 1.3。[1.1](#),[1.2](#),[1.3](#) 式分别给出了参考单元到网格单元，参考单元到目标单元，目标单元到网格单元的变换。

$$\mathbf{x} = \mathbf{A}\mathbf{x}^r + \mathbf{x}_1 \quad (1.1)$$

$$\mathbf{x}^g = \mathbf{W}\mathbf{x}^r \quad (1.2)$$

$$\mathbf{x} = \mathbf{S}\mathbf{x}^g + \mathbf{x}_1 \quad (1.3)$$

容易验证，目标单元到网格单元仿射变换的雅可比矩阵，即 \mathbf{S} 为

$$\mathbf{S} = \mathbf{A}\mathbf{W}^{-1} \quad (1.4)$$

其中

$$\mathbf{A} = \begin{pmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \end{pmatrix} \quad (1.5)$$

\mathbf{S} 与参考点的选取无关。若四边形网格以正方形为目标单元，对应的按照[定义 1.2.1](#) 分成的三角形的目标单元为等腰直角三角形，此时目标单元即为参考单元，即 $\mathbf{W} = \mathbf{I}, \mathbf{S} = \mathbf{A}$ 。对于三角形网格，常把目标单元取为正三角形。

我们选用雅可比矩阵 \mathbf{S} 的条件数作为网格单元质量的度量。即

$$\kappa(\mathbf{x}) = \frac{\|\mathbf{S}\|_F \cdot \|\mathbf{S}^{-1}\|_F}{2} \quad (1.6)$$

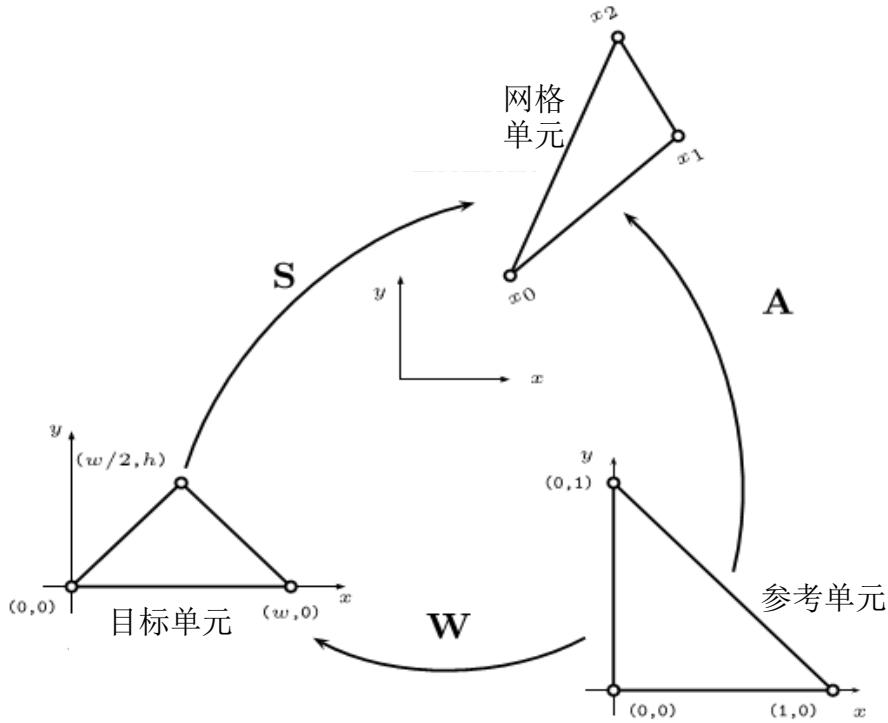


图 1.3 与参考点选取无关的仿射变换

其中 $\|\cdot\|_F$ 为矩阵的 Frobenius 范数，分母 2 为归一化因子。整个网格质量的度量定义为

$$|K_\kappa|_p(\mathbf{x}) = \left[\sum_{m=1}^M (\kappa_m)^p(\mathbf{x}) \right]^{\frac{1}{p}} \quad (1.7)$$

1.2.3 相关工作

由于全自动地生成高质量的四边形网格已经成为工业界的重大需求，近年来研究者们提出了各种各样的网格生成算法。

1. **基于三角网格的转化:** 由于高质量的三角网格较容易生成，一个最简单的想法是直接对每个三角形细分得到四边形网格，但这样产生了大量冗余的四边形单元以及奇异点，而且网格质量较差。这种方法通常被用于输入是四边形为主的网格情形。另一类方法的基本思想是把两个有公共边的三角形合并为一个四边形，如 [2],[3] 的工作。这类方法产生的四边形网格质量虽然高于前者，但仍然会产生较多的奇异点。总之，这类方法由于受到三角网格的点的分布和连接关系的限制，难以产生奇异点较少的高质量四边形网格。
2. **几何分解:** 这类方法先根据输入的平面区域的几何特征把它分解为简单的凸区域，再进一步把它们四边形化。[4],[5] 和 [6] 等使用了递归分解的算

法, [7] 基于递归分解的思想, 有效的提高了最差四边形单元的质量。[8] 使用了四叉树分解的算法。[9],[10] 等利用中轴线结构分解了平面区域。这类算法能在产生相对少奇异点的同时生成高质量的全四边形网格, 但对于复杂的边界变化较为敏感, 过多的算法细节使得算法不够鲁棒且难以自动化。

3. **前沿推进:** 这类方法由边界出发并向内推进, 进而产生四边形网格。比如 [11] 提出了 Paving 的方法, [12] 做出了改进。这类方法能对边界复杂的区域产生高质量的网格, 但在不同方向上推进的单元相遇时可能会发生自交, 对网格的有效性没有保证, 并且当边界比较复杂时会在内部产生较多的奇异点。
4. **网格模板法:** 这类方法先产生特定的四边形网格模板, 再把它应用于目标区域。如 [13] 首先提出用格点自然形成的正方形网格结构作为目标区域内部的四边形, 再对边界附近的正方形单元作出调整以适应不规则的边界。[14] 忽略接近平角的拐角点, 形成逻辑上 3、4 边的原始区域, 再利用无限映射 (Transfinite Mapping) 法把正三角形、正方形上的四边形网格模板映到原始区域上。这类方法能在简单边界的区域产生高质量的网格, 因此人们常常先用几何分解方法把复杂区域分解为简单区域, 再利用合适的网格模板进行四边形化。
5. **由场指导的方法:** 这类方法先在区域上产生定向场 (Orientation Field), 再由它指导产生分布接近这个场的四边形网格。它的优势在于因为场的奇异性直接决定四边形网格的奇异性, 这使得我们可以根据想要的四边形网格设计场。[15] 根据中轴角设计出区域内的定向场并由此产生了少量孤立的奇异点,[16] 先在产生符合边界几何特征的场, 再通过解偏微分方程传播到内部, [17] 提出的快速行进 (Fast Marching) 算法可以把用户给定的稀疏场传播到整个区域。由场的奇异点出发的追踪 (Tracing) 法可以产生合理的分割, 得到容易四边形化的简单区域。

1.3 计算无翻转映射

映射和变形在几何处理、形状建模以及动画等领域中是至关重要的。我们首先给出

定义 1.3.1. 称映射 $f : \mathbb{R}^d \rightarrow \mathbb{R}^d$ 是无翻转的, 是指映射的雅克比矩阵 $J(f)$ 的行列式 $\det(J(f)) > 0$

在计算机图形学中, 我们常常需要构造连续的分片线性映射, 比如网格的变形、表面网格到平面的参数化、网格质量提高等等。这些问题要求映射得到的网格与输入网格具有相同的连接关系, 并在无翻转的基础上最小化二者之间的某种几何度量的形变。由于任何真实的材料无法被压缩为零甚至是负的体积或面

积，翻转的单元无法通过物理变形而得到，所以无翻转的映射是一个基本而广泛的要求，尽管如此，保证这一点却很困难。

计算任意形体之间的无翻转映射 f 的一个常用方法如下：先设计一个能量方程 $E(f)$ 来度量对映射的某种需要，比如共形和面积形变，而 f 通常是点坐标 \mathbf{x} 的函数，然后在无翻转和其它特定约束下（如变形时用户指定的目标位置，网格质量提高时对边界的固定）计算最小化 $E(f)$ 的目标位置 $f(\mathbf{x})$ ，这是一个非线性带约束的优化问题。受到 [18] 的启发，许多工作通过设计一个能量函数，使其在 $J(f)$ 接近 0 时趋于无穷大，把问题转化为无约束优化，如 MIPS 能量 (Most-Isometric Parameterizations)([19]) 及其扩展 ([20],[21],[22])，并且致力于提出高效和收敛性好的优化算法。

在本文中有两处需要计算无翻转映射。一是把输入的平面三角网格变形为扩展多正方形结构的三角网格，这是对原始区域的简化。由于我们要利用变形的映射把扩展多正方形结构上生成的四边形网格映回原始区域，若映射含有较多翻转单元，最后会得到质量很差的四边形网格，我们采用了 [22] 的能量防止翻转。另外一个需要无翻转映射的是四边形网格质量的提高，若待优化的四边形网格是有效的，则我们把它看作是由正方形网格变形得到的结果，固定它的边界，以最小化 $J(f)$ 的条件数为目标（定义见 1.7 式）计算无翻转映射，即可得到不含翻转单元的高质量四边形网格。

1.4 本文的贡献及组织

基于 1.2.2 节对平面四边形网格的要求和 1.2.3 节介绍的相关工作，我们提出了扩展多正方形结构辅助的平面四边形网格生成算法。本文的主要算法及写作组织如下：

1. 第二章计算了由平面三角网格到扩展多正方形结构之间的无翻转映射；
2. 第三章利用基于中轴线的分割方法在扩展多正方形结构上生成四边形网格；
3. 第四章借助 1 的映射把扩展多正方形结构上的四边形网格点映回原始区域；
4. 第五章的网格优化算法通过改变点的位置消除了翻转单元并提高了网格质量；
5. 第六章展示了用扩展多正方形结构辅助中轴线分割法和 Looping 法 ([7]) 的实验结果，并比较和讨论了算法的优缺点，最后提出了把我们的算法扩展到 3 维情形的展望。

图 1.4 给出了算法流程的示例。本文做出了如下的贡献：

1. 通过输入不同的参数，我们的算法把原始网格变形为不同的扩展多正方形结构，以达到对原始区域不同程度地近似，进而控制最终生成的四边形网格奇异点的多少。我们计算的无翻转映射可以有效地保证四边形网格的质量。
2. 本文提出了把简单区域(定义见3.2.2)全部转化为3、4、5-边界简单区域的分割算法，并结合[9]和[23]实现了完整的四边形网格生成的算法。借助扩展多正方形结构的辅助，我们得到的中轴线结构不仅对复杂边界不敏感，而且能体现平面区域基本的几何特征并有效减少奇异点的数目，对任意的输入都能鲁棒地生成高质量、奇异点可控的四边形网格。
3. 任何对边界敏感的算法(如前沿推进法和其他几何分割方法)都可以通过扩展多正方形结构的辅助得到更好的结果。经过实验，我们的算法显著减少了Looping([7])法生成网格的奇异点数目。

c

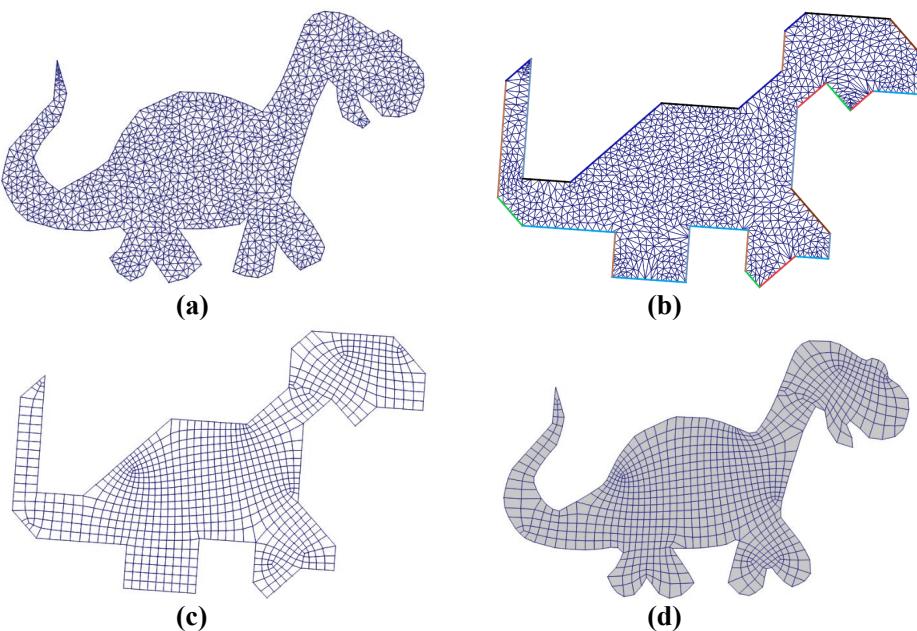


图 1.4 本文算法流程。输入平面区域产生的三角网格(a)，先把它变形为扩展多正方形结构(b)，再利用基于中轴线分割的四边形网格生成算法得到(c)，最后映射回原始区域(d)。

第二章 扩展多正方形结构生成

输入一个平面三角形网格 \mathbf{M} , 本章的目标是把 \mathbf{M} 变形为扩展多正方形结构, 并得到一个低形变的无翻转映射。如果输入的是边界曲线, 我们在边界上采点后使用 CGAL 库 ([24]) 的 Delaunay 三角化得到三角网格。[2.1](#) 节通过优化网格边界外法向的光滑和对齐的能量驱动网格变形, 得到近似的扩展多正方形结构。[2.2](#) 节通过给边界打标记得到边界上正确的拓扑结构, 并以此为目标进行变形, 最后压平边界得到严格的扩展多正方形结构, 越大的 N 得到的结构越接近原始区域。[2.3](#) 节给出了总的变形算法和优化策略。

2.1 法向光滑和对齐的网格变形

以扩展多正方形结构的法向为目标, 本节的目标是让输入网格边界的法向通过变形逐渐地与其最接近的目标法向对齐, 并且使变形后的形状与输入尽可能地接近。我们采用法向对齐能量和刚性形变能量 [22] 驱动网格的变形 ([2.1.1](#) 节), 并在变形的每一次迭代之前计算一个全局的旋转 ([2.1.2](#)), 使网格处在合适的坐标系中, 尽量减小初始的法向对齐能量。算法输入的核宽度 k 可以控制结果的奇异性。经过变形, 与输入最接近的扩展多正方形结构会大致成型。

2.1.1 法向对齐能量

设输入的 2 维网格为 \mathbf{M} , 包含的三角形集合为 $\mathcal{T} = \{\mathbf{t}_1, \dots, \mathbf{t}_l\}$, 顶点集合为 $\{\mathbf{v}_1^0, \dots, \mathbf{v}_m^0\}$, 它的边界集合是 \mathcal{B} 。 \mathcal{B} 包含 n 个三角形的边 $\{\mathbf{b}_1, \dots, \mathbf{b}_n\}$, 它们的法向是 $\{\mathbf{n}_1, \dots, \mathbf{n}_n\}$ 。以此为目标法向, 并定义算子 $\mathbb{A}(\cdot)$, 它将任意的二维向量映射到离它最近的目标法向上。因此, 扩展多正方形结构的三角网格满足 $\mathbf{n}_i = \mathbb{A}(\mathbf{n}_i), \forall i$ 。在三维情形下, 若目标方向为空间中的六个轴向, 则生成的四面体网格为多立方体结构 (PolyCube)。因为 $\mathbb{A}(\mathbf{n}_i)$ 算子是作用在每条边上的局部算子, 所以对于复杂的边界, 直接对原法向作用此算子可能会在使边界被过度地分割, 产生不必要的细节。因此我们先对法向做高斯平滑, 再驱动网格变形。我们定义如下的法向对齐能量:

$$E_{align} = \frac{1}{n} \sum_{i=1}^n (\mathbf{b}_i \cdot \mathbb{A}(G_\sigma(\mathbf{n}_i))) 2. \quad (2.1)$$

其中 $G_\sigma(\cdot)$ 为高斯函数:

$$G_\sigma(\mathbf{n}_i) = \sum_{\mathbf{b}_j \in \mathcal{K}_i} \|\mathbf{b}_j\| \exp \left(-\frac{dist(\mathbf{c}_{\mathbf{b}_i}, \mathbf{c}_{\mathbf{b}_j})}{2\sigma_s^2} \right) \cdot \mathbf{n}_j. \quad (2.2)$$

其中 $\mathbf{c}_{\mathbf{b}_i}$ 是边 \mathbf{b}_j 的中点, $dist(\mathbf{c}_{\mathbf{b}_i} - \mathbf{c}_{\mathbf{b}_j})$ 是从点 $\mathbf{c}_{\mathbf{b}_i}$ 到 $\mathbf{c}_{\mathbf{b}_j}$ 沿着边界的距离。设

$$\sigma_s = k\bar{l}. \quad (2.3)$$

其中 k 为算法输入的核宽度, \bar{l} 为三角网格边界的平均长度。因此输入不同的 k , 从而调整高斯函数的标准差, 就可以控制结果的奇异性。

2.1.2 刚性形变能量

为了让变形后的扩展多正方形结构接近输入的网格形状, 我们考虑映射的刚性形变, 用 [22] 定义的刚性能量惩罚大的形变, 并且避免变形过程中三角形的退化甚至翻转。设 $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ 为一个分片线性函数, 是输入网格到输出网格之间的映射, 考虑任意三角形 $\mathbf{t} \in \mathbf{M}$, $f_{\mathbf{t}}(\mathbf{x}) = \mathbf{J}_{\mathbf{t}}\mathbf{x} + \mathbf{b}_{\mathbf{t}}$ 是作用在 \mathbf{t} 上的仿射变换, 其中 $\mathbf{J}_{\mathbf{t}}$ 是 2×2 仿射矩阵, 也是 $f_{\mathbf{t}}$ 的雅可比矩阵, 根据 1.2.2 节的知识, 对于任意输入三角形 $\Delta \mathbf{v}_p^0 \mathbf{v}_q^0 \mathbf{v}_r^0$ 和对应的输出三角形 $\Delta \mathbf{v}_p \mathbf{v}_q \mathbf{v}_r$, 我们有

$$\mathbf{J}_{\mathbf{t}} = [\mathbf{v}_p - \mathbf{v}_q, \mathbf{v}_p - \mathbf{v}_r] \cdot [\mathbf{v}_p^0 - \mathbf{v}_q^0, \mathbf{v}_p^0 - \mathbf{v}_r^0]^{-1}. \quad (2.4)$$

设 σ_1, σ_2 为 $\mathbf{J}_{\mathbf{t}}$ 的奇异值, 则共性形变定义为:

$$\delta_{conf, \mathbf{t}} = \frac{\sigma_1}{\sigma_2} + \frac{\sigma_2}{\sigma_1} = \frac{\text{trace}(\mathbf{J}_{\mathbf{t}} \mathbf{J}_{\mathbf{t}}^T)}{\det \mathbf{J}_{\mathbf{t}}}. \quad (2.5)$$

面积形变定义为:

$$\delta_{area, \mathbf{t}} = (\det \mathbf{J}_{\mathbf{t}} + (\det \mathbf{J}_{\mathbf{t}})^{-1}). \quad (2.6)$$

最后得到刚性形变能量:

$$E_{iso} = \sum_{i=1}^l \exp(\alpha \delta_{conf} + (1 - \alpha) \delta_{vol}). \quad (2.7)$$

由于奇异值 σ_1, σ_2 表示映射在两个主轴方向上长度的缩放, 而雅可比矩阵的行列式 $\det \mathbf{J}$ 表示映射在面积上的缩放, 故当 $f_{\mathbf{t}}$ 是一个共形变换时, 两个形变均达到最小值 1。另一方面, 若任意一个三角形发生退化 ($\det(\mathbf{J}_{\mathbf{t}})$), 则 E_{iso} 会趋于正无穷。因此, E_{iso} 能很好地表示映射的刚性形变的程度, 并有效地惩罚较大的形变。若输入的三角形网格是高质量的, 则它能保证产生无退化和翻转的三角形网格。

结合 2.1 和 2.7, 我们得到网格变形的总能量:

$$E_{deform} = E_{iso} + \lambda E_{align} \quad (2.8)$$

其中 λ 为权衡两个能量的权重, 它的选取将在 2.3 节说明。 E_{iso} 的变量为网格所有点的坐标。

2.1.3 降低法向对齐能量的预旋转

显然, 如果网格的边界法向不沿着目标法向分布, 法向对齐能量可能会很大。为了尽量减小法向对齐能量, 我们希望先对网格进行全局地旋转, 使得旋转后的网格边界的法向与它们的目标法向尽量接近。即:

$$\theta^* = \arg \min_{\theta} \sum_{i=1}^n ((\mathbf{R}(\theta) \cdot \mathbf{b}_i) \cdot \mathbb{A}(\mathbf{n}_i))^2. \quad (2.9)$$

其中 θ 为旋转角, $\mathbf{R}(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$ 为旋转矩阵。

2.2 扩展多正方形结构的提取与网格压平

经过法向光滑和对齐的变形后, 边界的法向已充分地和目标法向对齐, 因此我们可以通过边界的目标法向提取出扩展多正方形结构。类似 [25],(2.2.1节) 采用两个策略对边界的目标法向做微小的修改, 以得到正确合理的扩展多正方形结构。修改后再对网格进行变形, 直到边界法向充分地与目标法向齐, 最后 (??节) 对扩展多正方形结构的每条边加上严格的直线约束, 压平网格, 就可以得到最终的结构。

2.2.1 扩展多正方形结构的提取

为了描述扩展多正方形结构, 我们首先定义三角形网格的段 (Segment) 和拐点 (Corner)。段是指扩展多正方形结构的边, 拐点是指扩展多正方形结构的顶点。于是我们有:

定义 2.2.1. 三角形网格边界上彼此相连并且目标法向相同的一个边集称为一个段, 记为 \mathbf{S} . 这个段的标记定义为它包含边的目标法向, 记为 \mathbf{l} . 长度定义为边的个数, 记为 d . 所有段的集合记为 \mathcal{S}

定义 2.2.2. 两个相邻段的公共点称为一个拐点, 记为 \mathbf{c} 。

这样, 扩展多正方形结构网格定义为:

定义 2.2.3. 网格的所有边界边的法向等于其所在段的标记, 即
 $\forall i, \mathbf{S}_i \in \mathcal{S}, \forall j, \mathbf{b}(i, j) \in \mathbf{S}_i, \mathbf{n}_i = \mathbf{l}_i$.

显然, 扩展多正方形结构拓扑上的充分必要条件是相邻的段不能有相反的标记。另外, 为了达到边界简化的目的, 避免在四边形网格生成时产生过多的奇异点, 我们不希望产生太短的段。于是我们要求:

1. $\forall \mathbf{S}_i \in \mathcal{S}, l_i < k$.
2. $\forall \mathbf{S}_i \in \mathcal{S}, \mathbf{l}_i \cdot \mathbf{l}_{i+1} \neq -1$

注 在不引起混淆的前提下, $\mathcal{S}_{i-1}, \mathcal{S}_{i+1}$ 表示 \mathcal{S}_i 的两个相邻的段, \mathbf{c}_i 所在的两个段为 $\mathcal{S}_{i-1}, \mathcal{S}_i$.

为了使扩展多正方形结构网格满足上述条件, 我们采用如下策略修改边界的目标法向 (图2.1):

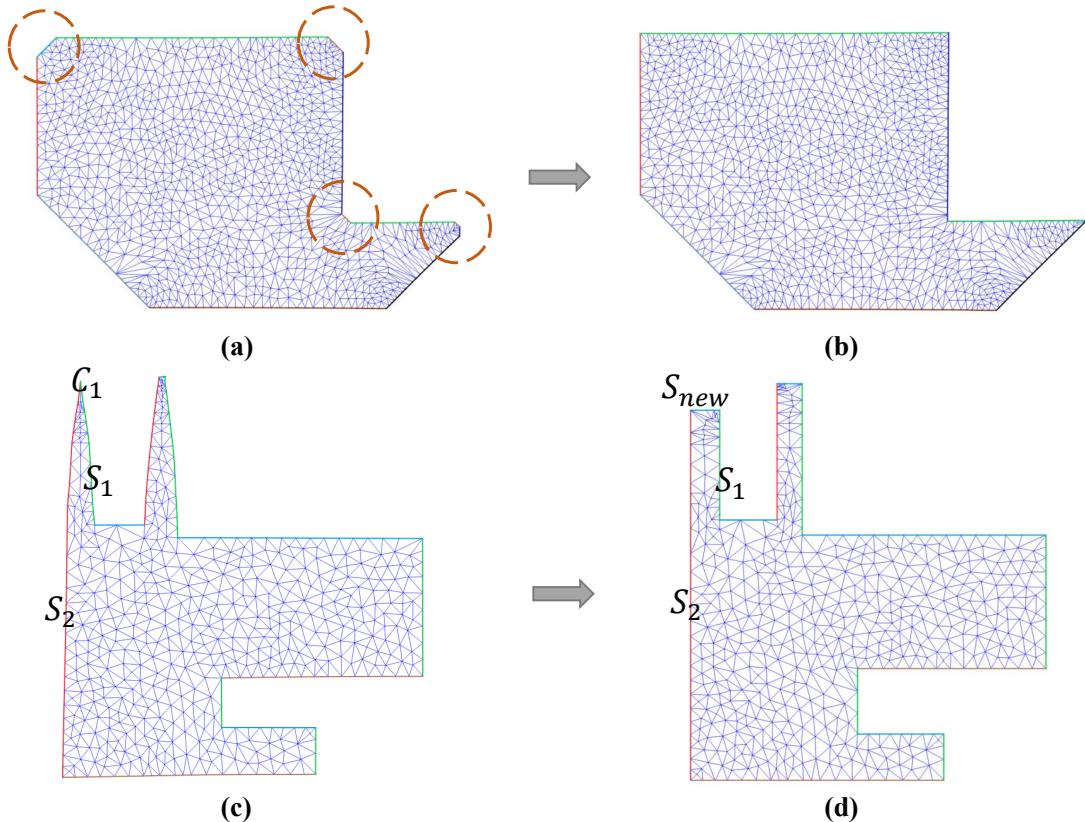


图 2.1 修改标记的两个策略。策略 1 吞掉了 (a) 中较短的段, 得到 (b); 策略 2 消除 (c) 相邻且标记相反的段, 得到 (d)。

1. 对长度小于 k 的段, 用与它相邻的两个段通过泛洪填充 (flood filling) 的策略吞掉。即从段两端的边开始, 依次向中间把边的目标法向修改为相邻段的标记, 直到两条被修改的边相遇。
2. 对相邻且标记相反的段, 从二者公共的拐点开始在较长的段上截取一部分, 形成一个新的段, 以打断相反的标记。若较长段的长度 d 大于 $2k$, 则新段的长度取为 k , 否则取为 $\lfloor \frac{d}{2} \rfloor$ 。通过新段的两个拐点形成向量 \mathbf{v} , 从目标法向中选取不是原有的两个标记且与 \mathbf{v} 的法向最接近的作为该新段的标记。

注: 执行策略 1 时, 我们每次从所有段中选取最短的段; 另外, 为了不与策略 2 冲突, 若与待吞掉的段相邻的两个段标记相反, 则不对此段做处理。对于目标法向为奇数的扩展多正方形结构, 不存在相反的法向, 故不需要执行策略 2。

所有的段都满足上述两个要求后, 固定边界的目标法向, 以 (2.8) 为目标函数进行变形, 其中 (2.1) 改为:

$$E_{align} = \frac{1}{n} \sum_{\mathbf{S}_i \in \mathcal{S}} \sum_{\mathbf{b}_j \in \mathbf{S}_i} (\mathbf{b}_j \cdot \mathbf{l}_i)^2. \quad (2.10)$$

当 $\mathbb{A}(n_j) = \mathbf{l}_i, \forall \mathbf{b}_j \in \mathbf{S}_i$ 时, 再进行五次变形的迭代使网格充分接近正确合理的扩展多正方形结构, 然后进入下面的网格压平。

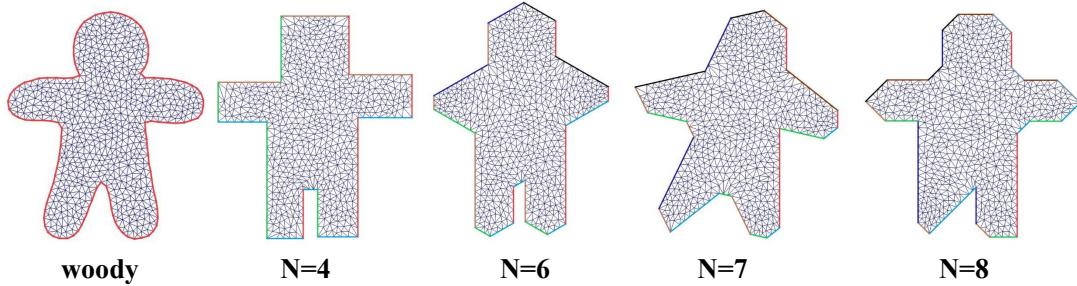


图 2.2 输入不同的 N 对 woody 模型变形得到的结果

2.2.2 网格压平

经过2.2.1节及变形后，每个段2.2.1上所有边的法向已经与该段的标记充分接近，因此只要强制每个段上的边界点都处在一条与该段的标记方向垂直的线段上，并让段的两个拐点成为线段的端点，就可以压平网格从而得到最终的扩展多正方形结构网格。为此，我们通过求解如下带线性约束的二次规划问题来确定拐点的坐标以及每个段对应的线段：

$$\begin{aligned} \min & \sum_{i=1}^m \sum_{j \in S_i} (y_j - l_i(x_j))^2 \\ \text{s.t. } & \mathbf{c}_i = (x_i, y_i) = l_i \cap l_{i-1} \end{aligned} \quad (2.11)$$

其中 \mathbf{c}_i 是段 S_i, S_{i-1} 的公共拐点， $l_i(\cdot)$ 是段 S_i 所在直线，其方向由标记 \mathbf{l}_i 唯一确定， j 遍历所有边界点。因此这个二次规划的变量为 m 条直线的截距、 m 个拐点的坐标。

用拉格朗日乘子法解得拐点坐标及段的位置后，我们直接更新当前拐点的坐标，并固定不是拐点的边界点的横坐标，带入其所在段的直线方程更新纵坐标即可。用拉格朗日乘子解得拐点坐标及段的位置后，我们直接更新当前拐点的坐标，并固定不是拐点的边界点的横坐标，带入其所在段的直线方程更新纵坐标即可。为了解决压平可能带来的三角形的退化甚至翻转，得到输入和输出之间在三角形上的一一映射，压平后我们再固定网格的拐点，利用五章的网格质量提升算法优化其他点的位置。

2.3 网格变形算法和优化

本节给出从输入的三角网格到输出扩展多正方形结构网格的完整算法，并给出优化这些能量的策略，图2.2给出了输入不同的 N 产生的结果。极小化能量 E_{deform} (式2.8)之前，我们先极小化2.9，求出全局的旋转矩阵，然后作用在以减小初始的 E_{deform} 。 E_{deform} 是高度非线性的网格点坐标的函数，我们使用线回溯搜索步长的梯度法进行优化，并借助 HLBFGS 库 [26] 实现，优化停止的条件是能量收敛或达到最大迭代次数。2.8中 λ 表示 E_{align} (2.1) 所占比重，因为初始网

格的边界往往离目标法向差距较大，为了避免产生较大的形变或产生退化或翻转的三角形，初始时我们让 E_{iso} (2.7) 占主导，取 λ 为 1，我们执行 5 次法向光滑和对齐的网格变形，每次使 λ 扩大 10 倍，这样就得到了边界和目标法向充分对齐的网格，然后按照2.2.1的方法提取正确的扩展多正方形结构，根据段2.2.1的标记，通过最小化2.8(其中 E_{align} 修改为2.10) 对不经过光滑的网格直接变形，直到优化停止，此时边界法向和段的标记已经充分接近。最后我们借助 MOSEK 库 [27]，使用拉格朗日乘子法求解2.11，从而压平网格，再利用第五章的算法提高网格质量。由此给出

```

input : triangle mesh  $M$  with vertices coordinates vector  $X$ ,Gaussian
          kernel  $k,N$ 
output: triangle mesh with regular polygon structure, $M_p$ 
1  $\lambda = 1;$ 
2 for  $iter \leftarrow 1$  to 5 do
3    $\theta^* \leftarrow \arg \min_{\theta} E_{rot};$ 
4    $X \leftarrow \mathbf{R}(\theta^*) \cdot X;$ 
5    $\min_X E_{iso} + \lambda E_{Salign};$ 
6    $\lambda \leftarrow 10 \cdot \lambda;$ 
7 end
8 extract segments and corners;
9 swallow short segments;
10 if  $N \bmod 2$  is 0 then
11   | swallow opposite segments;
12 end
13  $\min_X E_{iso} + \lambda E_{align};$ 
14 solve least square problem with linear constraints;
15 flatten mesh;
16 untangling and smooth;

```

算法 2.1: 多正 N 边形网格变形算法

第三章 基于中轴线分割的平面四边形网格生成

输入一个 2 维的边界不自交的有界区域，本章的目标是自动地在此区域上生成质量较高的四边形网格。若区域的边界是曲线，我们在边界上采点，用简单多边形逼近原边界。四边形网格生成分为两步：3.2 节用 [9] 的方法，基于区域的中轴线把它分割成充分简单、可直接产生四边形网格的子区域，3.3 节先把子区域分割 3、4、5 边形结构，再利用 [23] 整数规划的方法在每个子区域上产生四边形网格，并使得子区域网格之间彼此相容。

3.1 中轴线的概念

定义 3.1.1. 考虑 2 维区域 A 以及它的边界 ∂A ，对任意点 $P \in A$ ，记圆 $U_r(P)$ 是以 P 为圆心在 A 中的最大圆。所有 $U_r(P)$ 与 ∂A 交点至少为两个的点 P 构成的集合 $M(A)$ 称为的中轴线。其中 $U_r(P)$ 交于相同的两条边界的点 P 形成一条中轴边，中轴边的端点称为中轴点。

中轴边和中轴点可被看成一个图的边和点，这个图就像 A 的骨架，代表了 A 的拓扑结构，因此可以作为区域分割的指导。

定义 3.1.2. 与 (ref fem2d) 等价地，我们把中轴点分为如下几类：(1) J 类：至少三条中轴边的交点；

- (2) I 类：两条中轴边的交点；
- (3) C 类：中轴边与区域边界的交点；
- (4) T 类：拓扑冗余，中轴边的终点，与边界无交点；

图3.1中给出了各类中轴点的示例。(ref fem2d) 通过 Delaunay 三角化用三角形的外心连线逼近区域的中轴线，并根据三角形的邻接关系定义了与本文对应的几类三角形。本文利用 CGAL 库 ([24]) 精确地计算了中轴线。

3.2 基于中轴线的简单子区域的产生

本节复现了 (ref 2d fem mesh generation) 的方法，3.2.1 先把区域分割为凸子区域的并，3.2.2 再根据凸子区域的中轴线结构进一步把它们细分为充分简单区域的并。3.2.3 对这种方法进行了讨论。

3.2.1 分割区域去除凹角

我们希望最终产生的每个四边形元素都接近矩形，且 2 维区域边界的拐角必然成为四边形网格的点，因此首先根据角度对它们进行分类，以在附近产生最

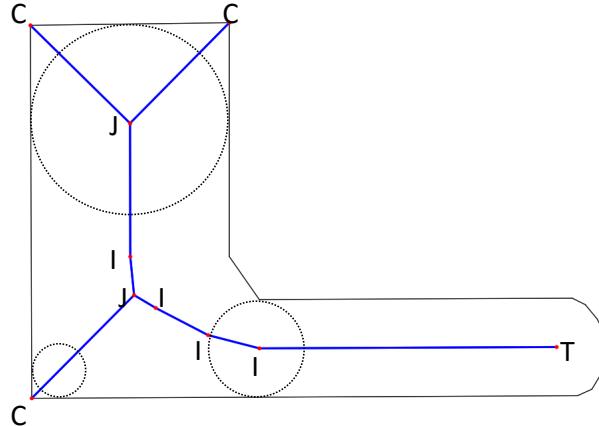


图 3.1 中轴线及中轴点示意。

优的四边形分布。对每个角 θ , 提出如下的准则:

$$\begin{aligned} & \sum_{i=1}^n \left(\theta_i - \frac{\pi}{2} \right)^2 \\ s.t. \quad & \sum_{i=1}^n \theta_i = \theta \end{aligned} \tag{3.1}$$

即我们希望合理地分割 θ 使得附近的四边形尽量好, 由这个准则可以把 θ 分为 3 类:

1. $\theta < 120^\circ, n = 0$, 度为 1;
2. $120^\circ \leq \theta < 216^\circ, n = 1$, 度为 2;
3. $216^\circ \leq \theta, n = 2$, 度为 3;

ref(2d fem) 称第 3 类角是凹的, 即

定义 3.2.1. 角 θ 是凹的, 是指 $\theta \geq 216^\circ$

显然第一类角可直接作为一个四边形的角, 而第二类角可被视为平角, 让四边形直接通过附近的边界, 3.3.2 节将利用整数规划直接处理。但我们必须先合理地分割区域, 使产生的子区域都只有第一、二类的角, 根据3.1式, 它们应尽量接近直角的整数倍。

[9] 先在边界细致地采点, 再以它们为约束对区域进行 Delaunay 三角化, 然后从凹角点所在的内部边中选出最优的分割。选取的准则为

$$\min \sum_{i=1}^4 \left(\theta_i - \frac{n_i \pi}{2} \right)^2 + \mu \| \mathbf{v} - \mathbf{c} \|^2 \tag{3.2}$$

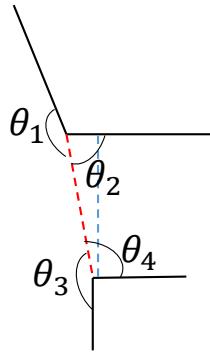


图 3.2 去除凹角的分割。由于我们加入了对分割点和拐点间距离的惩罚，最优分割由蓝色虚线变为红色虚线。

其中 n_i 是整数， θ_i 如图3.2所示。与 [9] 不同的是，我们加入了对另一个点 v 与其最近拐点 c 的距离的惩罚。因为这个点和附近拐点之间的边会成为至少一个四边形的边界，为了避免生成的网格过密，如果角度最优的分割点离另一个拐点很近，我们选择次优的角度，直接连接两个拐角点。我们不断选取边进行分割，并更新子区域的 Delaunay 三角化，直到所有的子区域都没有凹角为止。

3.2.2 基于中轴线的区域细分

给定一个所有拐点都不是凹角的 2 维区域，其中轴线就像该区域的“骨架”，并且很好地代表了区域的拓扑结构。根据3.1.2对中轴点的分类以及它们的连接关系，我们采取如下的一系列方法把这些子区域归结或进一步细分为简单区域 (ref def simple)。

1. **T 类** 对 T 类中轴点附近区域的处理取决于沿着形成这段冗余拓扑结构的“圆弧”张成的角度总和，若它接近 π ，则插入一段分割分开这段“圆弧”和剩余的几何区域，一方面此“圆弧”和插入的分割形成了一个 2-边界简单区域，另一方面剩余的几何区域减少了一个 T 类中轴点，并增加了两个 C 类点 (3.3)。若整个区域只有一个中轴点 T ，则这个区域被视为 1-边界简单区域。其他的情况 T 均可被视为3.2.1中第二类拐角点：
2. **F 类** 若 C 类中轴点是??中第二类拐角点，我们重新称它为 F 点，任何与 F 点相连的 J 点被重新定义为 FJ 点，即由 $F - J$ 转化为 FJ ，四边形网格将沿此处连续地通过 (3.3)。 FJ 或 I 点意味着两条边被合并为一条逻辑上的边，因此在所有的操作中，我们忽略任何的 FJ 或 I 点，直接考察与它们相连的 C, F, J 点。
3. **C 类** 若 C 点与 T 或 F 点相连，则此区域为 1-边界简单区域，若与 C 点相连，则此区域为 2-边界简单区域 (3.4)，若与 J 点相连，则我们直接考察：

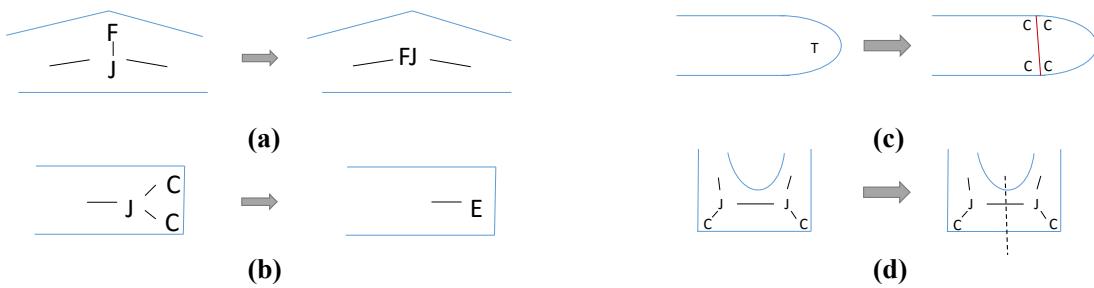


图 3.3 基于中轴线结构的分析示意图。(a) 把 F-J 结构转化为 FJ 结构，并忽略这个中轴点；(b) 分割去除 T 类中轴点，产生 4 个新的 C 类中州点；(c) 把 C-J-C 结构改为 E 结构，代表 4 边界简单区域的两个拐角 (d) 分割去除 J-J 结构。

4. **J 类** 若 J 点与 F 点相连，则更改 J 为 FJ 点，若与三个 C 点相连，则此区域为 3 边界简单区域，若 J 点与两个 C 点相连，则被重新定义为 E 点，并忽略这两个 C 点 (3.3)。产生所有的 E 点后，我们考查与剩余 J 点相邻的 C, E 点个数：若有一个 C ，两个 E ，则此区域为 5 边界简单区域，若有三个 E ，则此区域为 6 边界简单区域。若不满足如上所有情况，则此区域不能直接归为简单区域，将在 (6) 中处理。
5. **E 类** 若与 E 相连的点为 C 点，则此区域为 3 边界简单区域，而 $E - E$ 结构的区域为 4 边界简单区域。
6. **长链分割** 以上操作后若还有 $J - J$ 结构，则需要产生合适的分割，在分割两侧分别产生两个 E 点，分割后产生的两个子区域又可以归结为简单区域。若这条中轴线对应的边界有 F 点，则我们使用准则 3.2 再次从 Delaunay 三角形中选取分割，此时 F 点被当作凹点处理 (3.3)。若无边界点，则以中轴线上的点为圆心，沿着中轴线产生一系列内接圆，并以长度为极值的直径作为分割。
7. **圆环** 最后我们要检查是否有逻辑上的圆环结构存在，若有，则我们类似 (6) 产生两条分割，即可产生两个 $E - E$ 结构，即 4 边界简单区域。

定义 3.2.2. 仅以 C 点为拐角点，根据逻辑上的边界数的不同，定义如图所示的简单区域。

3.3 节将会先进一步处理简单区域，再用整数规划的方法得到四边形网格。

3.2.3 讨论与比较

[9] 的方法先产生最优角度的非凹子区域，再根据子区域中轴线的结构进一步分解为简单区域进而四边形网格化。对任意输入的 2 维有界区域，它都能够生成四边形网格。

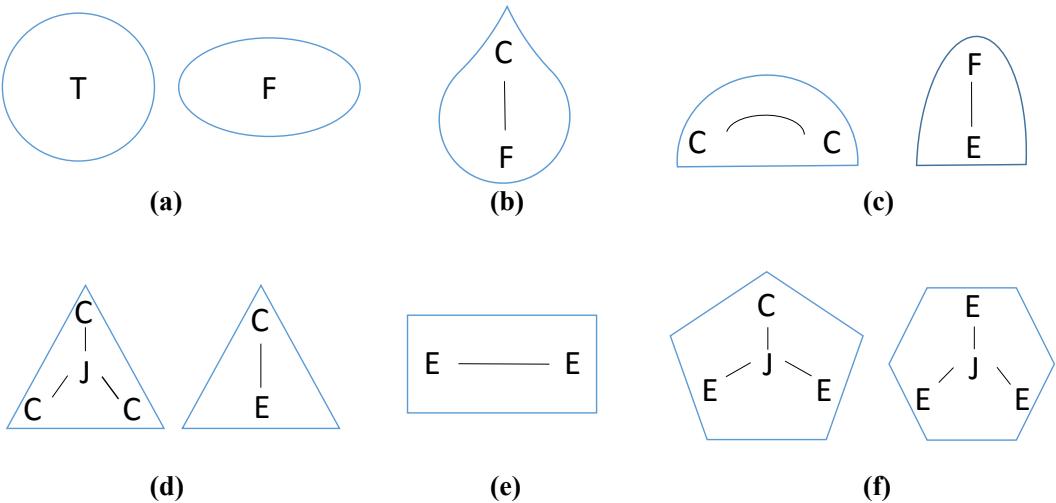


图 3.4 简单区域的定义。(a),(b) 为 1-边界简单区域, (c) 为 2-边界简单区域, (d), (e),(f) 分别为 3, 4, 5-边界简单区域

上节中, 忽略 FJ 点是对中轴线的减支。[3.3.1](#) 节将指出每个三支及以上中轴线的交汇点代表着至少一个奇异点的产生, 因此对于有一定噪音的输入, 忽略 FJ 点能有效地简化中轴线, 避免生成不必要的奇异点。但对于抖动剧烈的噪音, 此方法仍然会产生复杂的中轴线结构。我们先把较复杂的输入简化为多正 N 边形结构, 再在它的多边形边界上做区域分割时, 第一、六、七类的情况不会出现, 奇异点的数目也会减少。

[15] 指出, 由于中轴线对几何特征不够敏感, 图[3.5\(a\)](#) 中两种差异较大的形状具有相同的中轴线结构, 所以 [15] 的方法会把它们归结为 $E - E$ 结构而最终产生没有奇异点但质量较差的四边形网格 (b)。但我们的算法借助扩展多正方形的辅助, 取 $N=4$ 可以在同样没有奇异点的情况下产生质量更高的网格。取 $N=8$ 时两种形状的中轴线结构不同, 因而会产生合适的奇异点, 使得网格的缩放雅可比值更高,。

3.3 整数规划生成四边形网格

上节把 2 维任意有界区域细分为简单区域 ([3.2.2](#)), 本节先对简单区域中 0-边界、1-边界、2-边界, 6-边界区域做合理地分割, 全部归结为 3-边界、4-边界、5-边界简单区域, 再利用 [23] 提出的整数规划方法生成四边形网格。

3.3.1 合理的 3、4、5 边形区域的产生

与[3.2.1](#) 节分割去除凹角的目标类似, 我们要使新产生的角尽量接近直角, 并且希望尽量少地引入新的顶点和短边。因此, 我们先对非 3、4、5-边界简单区域的边界采样做 Delaunay 三角化, 再根据准则[3.2](#)从所有与 F 点相连的内部边中选

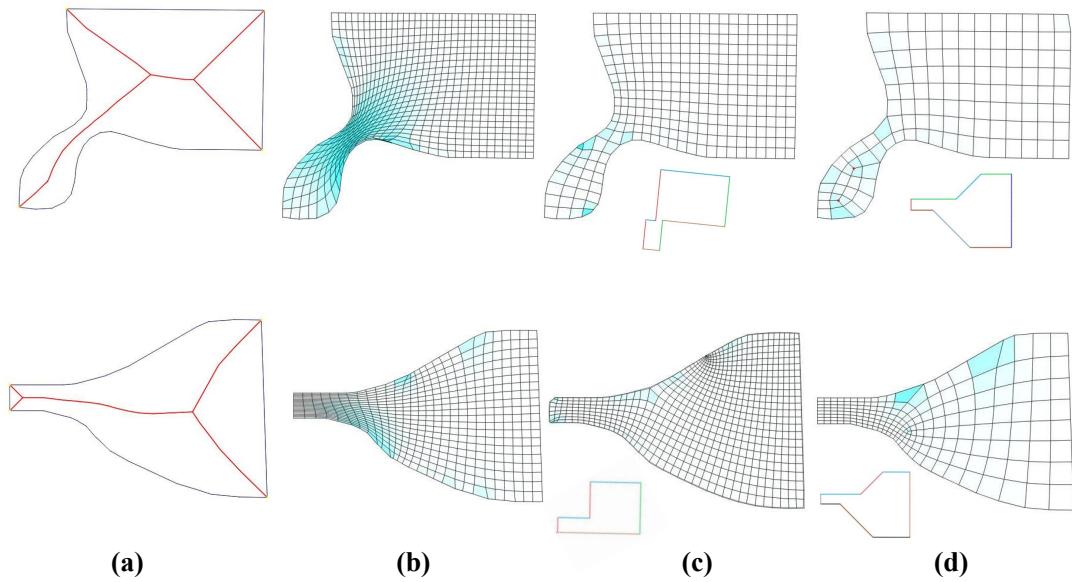


图 3.5 中轴线对几何特征不够敏感。从上到下, (a) 展示的两种图形几何特征差异很大, 但具有相同的中轴线结构。因此 [9] 均检测为 4 边界简单区域, 产生的四边形网格 (b) 的缩放雅可比值 J_{min}/J_{avg} 分别为 0.01/0.86, 0.45/0.94。(c) 为多正方形辅助后产生的结果, J_{min}/J_{avg} 分别为 0.22/0.98, 0.44/0.99。(d) 为取 $N=8$ 产生的结果, J_{min}/J_{avg} 分别为 0.74/0.98, 0.51/0.98, 引入的奇异点数分别为 2 和 1。

出最优的一条作为分割。分割在两个两个新的区域中产生的 4 个中轴点直接记为 C 点。显然此时 3.2 中直角的整数倍 n_i 为 1。3.6 给出了分割示意图。

注 对于 6 边界简单区域, 为了保证一定能分割为 3、4、5 边界区域, 我们直接选取连接角度最大的 C 点和与它相对的两条边界上点的所有边, 在其中选择最优的边作为分割, 这样一定可以分解为 3.6(d),(e),(f) 中的一种。本文中采取的分割办法都是基于在 Delaunay 三角形中搜索最优边, 但我们不需要为了找到最优分割过密地采点, 因为第五章中通过优化的方法可以优化所有的非拐角点的位置来提高四边形网格的质量, 而所有的分割以及整数规划中产生的新的点主要是为了决定四边形网格的拓扑结构, 并使得优化有较好的初始。

3.3.2 整数规划生成四边形网格

给定 3、4、5 边界的简单区域以及它们对应的四边形网格的样式后, 每条边被细分成四边形网格边的数目决定了产生的四边形网格。以简单区域的边的细分数为整数变量, 对不同的网格样式加上特定的约束, 再使相邻区域的公共边界相容, 最后以最小化所求细分数与一组特定值(比如用户给定的细分数, 或基于之前的有限元分析的误差自动计算出的目标细分数之间的差异为目标([28]), 就可以求得一组最优的细分数。给定符合网格样式的细分数后,[23] 通过把简单区域映射为 3、4、5 边形并产生四边形网格, 再映射回原区域, 即可得到质量较高的四边形网格。本节给出了类似 [23] 的目标函数和线性约束, 不同的是, 我们没有利用基于映射的方法决定内部点的位置, 而是直接在简单区域上产生初始的

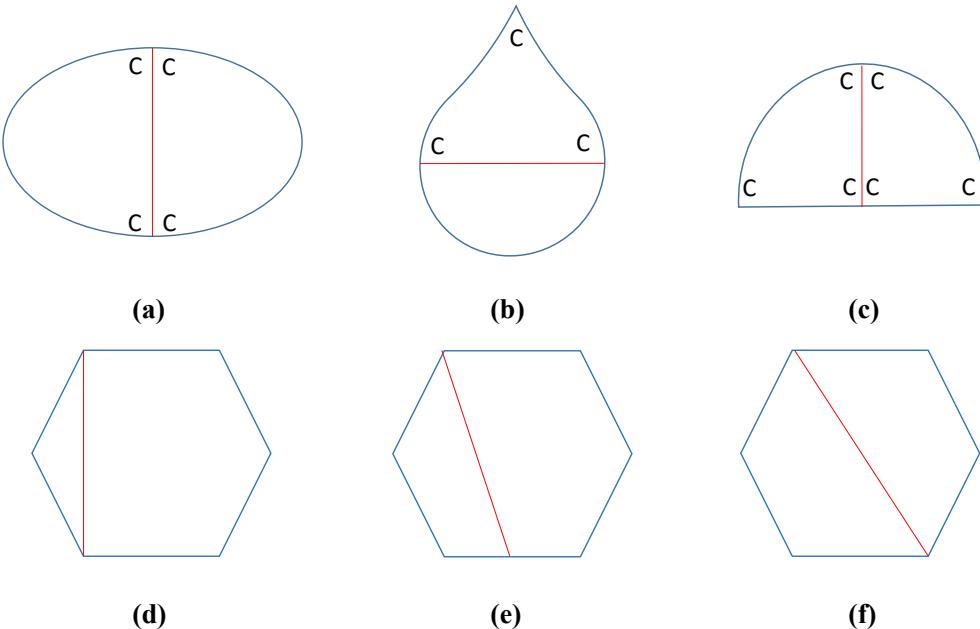


图 3.6 把简单区域分割为 3, 4, 5 边界简单区域: (a) F 类分割为两个 C-C 结构; (b) C-F 结构分割为一个 C-C 结构和一个 3 边界简单区域; (c) C-C 结构分割为两个 3 边界简单区域; (d), (e), (f) 分别为 6 边界简单区域被分割为 3+5, 4+5, 4+4 边界简单区域。

四边形网格, 再显式地以四边形网格质量为目标优化非拐角点的位置, 得到最终的四边形网格。

变量 简单区域的边界上只有 C, F 两种点 (定义见??), 每两个相邻的点形成一条边, 设 x 为它的细分数。每个 C 点和与它最近的 C 点形成一条简单区域的逻辑边, 它由至少一条边形成, 它的总细分数记为 N 。

约束 3、5 边界简单区域生成的四边形网格分别产生度为 3、5 的奇异点, 而 4 边界区域产生规则的四边形网格, 没有奇异点。

3 边界区域的四边形网格样式如 3.7(a) 所示, 故需满足

$$\begin{pmatrix} N_1 \\ N_2 \\ N_3 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix} \quad (3.3)$$

其中 N_i, n_i 分别为外部和内部细分数。实际上, 变量 n_i 是冗余的, 因为上式等价于

$$\begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{pmatrix} \cdot \begin{pmatrix} N_1 \\ N_2 \\ N_3 \end{pmatrix} \quad (3.4)$$

又因为 n_i 是正整数，故上式等价于

$$\begin{cases} N_1 + N_2 > N_3 \\ N_2 + N_3 > N_1 \\ N_1 + N_3 > N_2 \\ N_1 + N_2 + N_3 = 2k(k \geq 1) \end{cases} \quad (3.5)$$

前三个式子保证了内部细分数为正，最后一个式子保证内部细分数为整数。

4-边界区域的约束很简单，只需保证相对的逻辑边细分数相等即可 (3.7(b)), 即

$$\begin{cases} N_1 = N_3 \\ N_2 = N_4 \end{cases} \quad (3.6)$$

5-边界区域的约束与 3-边界类似，根据其四边形网格样式 (3.7(c)), 我们有

$$\begin{pmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \\ N_5 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \\ n_5 \end{pmatrix} \quad (3.7)$$

其中 N_i, n_i 分别为外部和内部细分数，同样地，上式等价于

$$\begin{cases} N_1 + N_2 + N_3 > N_4 + N_5 \\ N_2 + N_3 + N_4 > N_5 + N_1 \\ N_3 + N_4 + N_5 > N_1 + N_2 \\ N_4 + N_5 + N_1 > N_2 + N_3 \\ N_5 + N_1 + N_2 > N_3 + N_4 \\ N_1 + N_2 + N_3 + N_4 + N_5 = 2k(k \geq 1) \end{cases} \quad (3.8)$$

前 4 个式子保证了内部细分数为正，最后一个式子保证内部细分数为整数。若只对每个简单区域加上独立的约束，则可能在公共边上产生不一致的结果因此，相容约束要求每条公共边的细分数必须同时满足它所在两个区域的约束。由于我们在区域的细分过程中始终维持一个半边数据结构，所以只要对每一组方向相反的半边设一个细分数变量，就可自动满足区域间的相容性。

目标函数和其他约束 我们希望产生质量较高且尽量稀疏的四边形网格，故对于每个简单区域，令其最短的边的目标细分数为 1，对其余边长与最短边长的比值向下取整，作为它们的目标细分数(至少为 1)，然后规定解得的细分数不小于目标，并让它们和目标尽量接近。设 x_i 的目标细分数为 d_i ，则目标函数为

$$\min \sum_{i=1}^n x_i \quad (3.9)$$

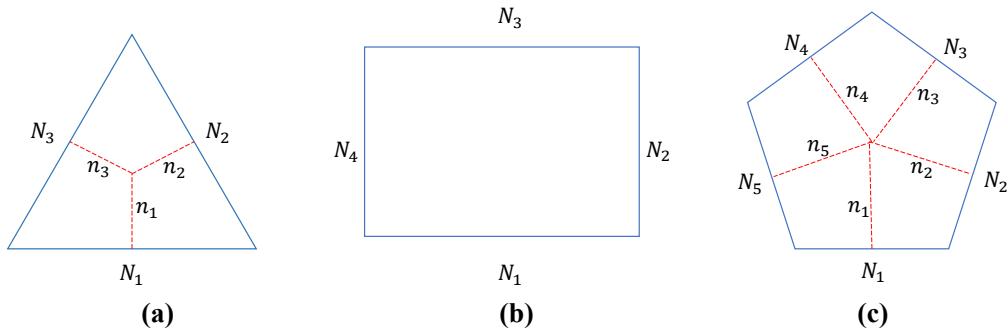


图 3.7 (a), (b), (c) 分别为 3, 4, 5-边界简单区域的细分方法和约束

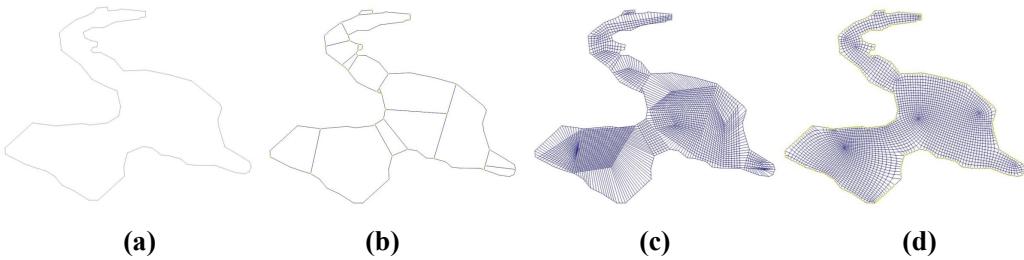


图 3.8 基于中轴线分割的四边形网格生成步骤。(a) 为输入的平面区域, (b) 为分割形成的 3, 4, 5-边界简单区域, (c) 为整数规划生成的四边形网格, (d) 为优化后的四边形网格。c

再增加如下的约束

$$x_i \geq d_i \quad (3.10)$$

这样, 以3.5, 3.8, 3.10为线性约束, 3.9为线性目标函数, 我们用 MOSEK 库 [27] 可以非常快的得到最优解。求出每条边的细分数后, 我们直接在均分边界得到不是拐角点的边界点。对于 4-边界区域, 直接连接对边上的对应点, 并两组对边连线的交点作为内部点, 即得到四边形网格。对于 3、5-边界简单区域, 先利用边界上的 C 点求出它们的中心作为奇异点, 再连接产生内部边, 进而分别分割为 3 个和 5 个 4-边界简单区域, 进而得到四边形网格。这些初始的四边形网格质量较差, 且可能含有翻转单元, 我们再利用5.2.2节的优化算法改变非拐角点的位置, 得到有效且高质量的四边形网格, 示例见图3.8。最后, 我们可以根据需要通过把所有四边形单元一分为四得到更密的网格。

第四章 计算原区域的四边形网格

本章目标是借助点在三角形中的重心坐标，把扩展多正方形结构上生成的四边形网格映回原始区域。记原始三角网格为 \mathbf{M}^0 , 其点集合为 $\mathbf{V}^0 = \{\mathbf{x}_1^0, \dots, \mathbf{x}_n^0\}$, 面集合为 $\mathbf{F}^0 = \{\mathbf{f}_1^0, \dots, \mathbf{f}_m^0\}$, 对应的扩展多正方形结构的三角网格为 \mathbf{M} , 其点集合为 $\mathbf{V} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, 面集合为 $\mathbf{F} = \{\mathbf{f}_1, \dots, \mathbf{f}_m\}$, 在 \mathbf{M} 上生成的四边形网格点集合为 $\mathbf{Q} = \{\mathbf{q}_1, \dots, \mathbf{q}_l\}$ 。

由于任意的四边形网格点必落在 \mathbf{M} 的某个三角形的边界或内部，我们可以用这个三角形的中心坐标表达这个点，即

$$\forall \mathbf{q}_i \in \mathbf{Q}, \exists! \{\mathbf{x}_r, \mathbf{x}_s, \mathbf{x}_t\} = \mathbf{f}_j \in \mathbf{F}, s.t.$$

$$\mathbf{q}_i = r\mathbf{x}_r + s\mathbf{x}_s + t\mathbf{x}_t, r + s + t = 1, r, s, t \geq 0 \quad (4.1)$$

由于上式也是点落在三角形边界或内部的充分条件，故我们直接把重心坐标应用于原网格上的三角形，即计算出

$$\mathbf{q}_i^0 = r\mathbf{x}_r^0 + s\mathbf{x}_s^0 + t\mathbf{x}_t^0 \quad (4.2)$$

其中 \mathbf{q}_i^0 是原始区域上的四边形网格的点，它也落在相应的三角形上，四边形网格的连接关系不变。由此可以看出，若第二章计算的映射形变较大，在原始区域上生成的四边形网格质量就会变差。最后我们可以利用第五章的方法提高网格质量。图4.1给出了由重心坐标得到的四边形网格及优化后的结果。

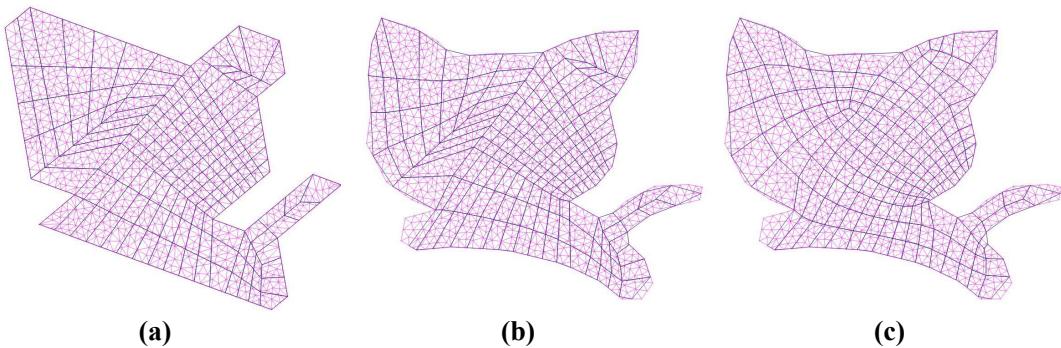


图 4.1 利用重心坐标映回原区域。(a) 为在扩展多正方形结构三角网格(红色)上生成的四边形网格(蓝色); 利用变形得到的映射, 通过重心坐标映回原始区域得到(b); 优化四边形网格后得到(c)

第五章 2维三角形或四边形网格的优化

一种网格优化技术保持网格的连接关系不变，通过移动自由点的位置最优化某种网格质量的度量。本章基于 [29] 的方法，把得到无翻转网格和提高网格质量合并为一个优化过程。通过改变非拐角点的位置，就可以快速地得到无翻转且高质量的网格。

5.1 目标函数设计

设待优化的网格目标顶点的位置坐标记为 \mathbf{x} ，网格的边集为 $\mathcal{E} = \mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_N$ ，考查的三角形为 $\mathcal{T} = t_1, t_2, \dots, t_M$ 。对于四边形网格，我们先按 1.2.2 节所述把每个四边形分成四个三角形。为了提高网格质量，5.1.1 节采用了 [29] 中可以同时惩罚翻转单元和低质量单元的目标函数，5.1.2 节设计了针对四边形网格的以直角为目标的网格度量。

5.1.1 同时解决翻转网格和提高网格质量的度量

我们在 1.2.2 节中定义的度量虽然能表达一个有效网格的质量，但无法处理有翻转三角形的网格。这是因为它们无法连续地定义在整个 \mathbb{R}^2 空间上，特别地，当某个三角形连续地从一个定向变为退化再变为另一个定向时，这些度量函数的变化是不连续的。因此，许多网格优化的技术都需要为此定义不同的目标函数，先得到无翻转的网格，再提高网格质量，而这种方法无疑需要较多的迭代次数并且很容易收敛到较差的局部极小值。

考虑用雅可比矩阵 J （见 1.6 节）的条件数 κ 度量给定三角形与目标三角形的差距，首先将其重写为

$$\kappa = \frac{\|J\|_F \cdot \|J^{-1}\|_F}{2} = \frac{\|J\|_F^2}{2\sqrt{|\sigma|}} \quad (5.1)$$

其中 $\sigma = \det(J)$ 。显然 κ 达到最大值 1 当且仅当这个三角形是等腰目标三角形，但 κ 在 $\sigma = 0$ 处不连续，即它不能表达退化或翻转单元的质量。

[29] 对上述度量做了改进，上式中 σ 被替换为一个恒正且单调递增的函数

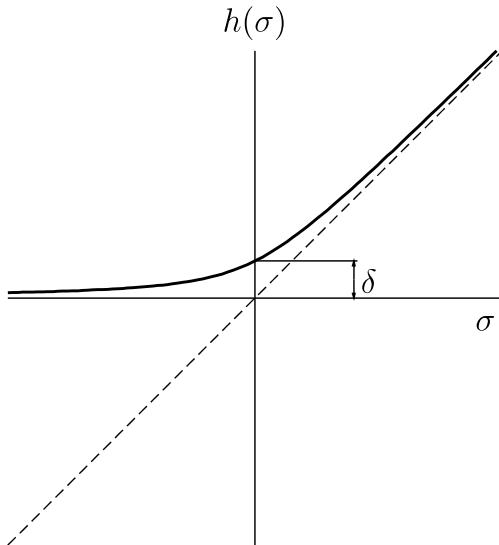
$$h(\sigma) = \frac{1}{2}(\sigma + \sqrt{\sigma^2 + 4\delta^2}) \quad (5.2)$$

因此有 $\delta = h(0)$ 。 $h(\sigma)$ 和 σ 的函数图像见图 5.1。因此对三角形质量的度量更改为

$$\kappa^* = \frac{\|J\|_F^2}{2\sqrt{h(\sigma)}} \quad (5.3)$$

由此给出目标函数

$$E_{sus} = |K_{\kappa}^*|_p(\mathbf{x}) = \left[\sum_{m=1}^M (\kappa_m^*)^p(\mathbf{x}) \right]^{\frac{1}{p}} \quad (5.4)$$

图 5.1 目标函数的修改: $h(\sigma)$

其中 κ_m^* 为改进的对三角形 t_m 的度量。

改进目标函数的合理性 虽然 κ^* 不是 J 的条件数, 但由于 $\lim_{\delta \rightarrow 0} h(\sigma) = \sigma, \forall \delta \geq 0$ 且 $\lim_{\delta \rightarrow 0} h(\sigma) = 0, \forall \sigma \leq 0$, 对于未翻转的网格单元, 总可以选取充分小的 δ 使得 κ^* 与 κ 任意地接近。又由于 $h(\sigma)$ 关于 σ 在 \mathbb{R} 上恒正且单调递增, 所以目标函数可以对翻转单元做出更大的惩罚。另外, 考虑到 $\forall \sigma > 0, \lim_{\delta \rightarrow 0} h'(\sigma) = 1$ 且 $\lim_{\delta \rightarrow 0} h^{(n)}(\sigma) = 0, \forall n \leq 2$, 容易证明修改后的目标函数的各阶导数与原函数有类似的收敛性, 故我们可以使用基于一阶导数(如梯度法)和二阶导数(如牛顿法)的方法优化这个能量。

δ 的选取 在计算机实现中, δ 应取为在不引起任何数值计算问题的前提下尽量小的正数。令 γ 是机器精度 ($0 < \gamma \ll 1$), 为了避免计算目标函数 5.4 时计算机做除 0 运算, 我们选取

$$\delta \geq \delta_{\min} = \begin{cases} \sqrt{\gamma(\gamma - \sigma_{\min})} & \text{if } \sigma_{\min} < \gamma \\ 0 & \text{if } \sigma_{\min} \geq \gamma \end{cases} \quad (5.5)$$

其中 $\sigma_{\min} = \min_{m=1,\dots,M}(\sigma_m)$ 。由于 $h(\cdot)$ 的单调性, 总有 $h(\sigma) \geq h(\sigma_{\min}) \geq \gamma$, 由上式也可以看出若网格无翻转或退化单元, 则 δ 可取为 0, 此时 $h(\sigma) = \sigma$, 式 5.4 即为原始的目标函数 1.7。

总之, 通过选取小的 δ 使用改进的目标函数 5.4, 我们可以同时解决网格的翻转单元并提高网格的质量。

5.1.2 以直角为目标的四边形网格度量

[23] 指出, 四边形网格的角度往往被作为最直接的质量度量, 因此在保证网

格无翻转单元的基础上，我们再显式地加入

$$E_{ver}(\mathbf{x}) = \sum_{\theta} \cos^2(\theta) = \sum_{(\mathbf{e}_i, \mathbf{e}_j)} \left(\frac{\mathbf{e}_i \cdot \mathbf{e}_j}{\|\mathbf{e}_i\| \cdot \|\mathbf{e}_j\|} \right)^2 \quad (5.6)$$

其中 θ 遍历四边形网格的所有角， $(\mathbf{e}_i, \mathbf{e}_j)$ 遍历组成角的两条边的向量。对于无翻转单元的四边形网格， $0 < \theta < \pi, \forall \theta$ 。 θ 越接近直角， E_{ver} 越小。此能量不能处理含有翻转单元的情况。

5.2 网格优化算法

本文有两处需要优化网格的质量。一是2.2.2节对压平的扩展多正方形结构三角网格，二是3.3.2节解整数规划问题后产生的初始四边形网格。

5.2.1 扩展多正方形结构三角网格的优化

由于2.2.2节压平网格时可能在边界处产生较差的网格，我们以正三角形为每个网格三角形的目标三角形。并给出式5.3的梯度：

$$\partial_{\alpha} \kappa^* = 2\kappa^* \left[\frac{(\partial_{\alpha} J, J)}{\|J\|_F^2} - \frac{\partial_{\alpha} \sigma}{4\sqrt{\sigma^2 + 4\delta^2}} \right] \quad (5.7)$$

其中 $(\partial_{\alpha} J, J) = \text{tr}(\partial_{\alpha} J^T J), \alpha = x, y$ 为点的横、纵坐标。

为了保持扩展多正方形结构，我们固定拐角点的位置，并把边界非拐角点约束在边界上。我们使用梯度下降法优化能量 E_{sus} ，每次局部地移动一个点，而固定其它点，使用线回溯搜索的方法确定步长，当前点的局部能量一旦下降，就转向优化下一个点。对于内部点，移动方向即为其梯度方向，对于边界非拐角点，把梯度方向在边界方向上的投影作为移动方向。算法终止的条件是所有点沿着目标方向的移动都不能使能量减小。

讨论 如果以所有点的坐标为变量，整体地优化 E_{sus} ，则使能量下降的方向尤为重要。由于所有的点会同时移动，若任何点的子方向很差，则会导致整体的能量下降缓慢。若使用二阶的方法确定移动方向，则需要求解大规模的线性方程组，且会受到海森矩阵性质的影响，若使用全局的梯度作为近似的下降方向，会导致能量收敛性较差。局部地优化网格时我们不需要求解精确的下降方向，只需简单地用梯度方向近似并搜索出合适的步长。

5.2.2 四边形网格的优化

我们把网格的每个四边形分成四个三角形，然后以等腰直角三角形为目标三角形，此时 W 为单位阵。结合式5.4和5.6，得到总的目标函数

$$E_{quad} = E_{sus} + \lambda E_{ver} \quad (5.8)$$

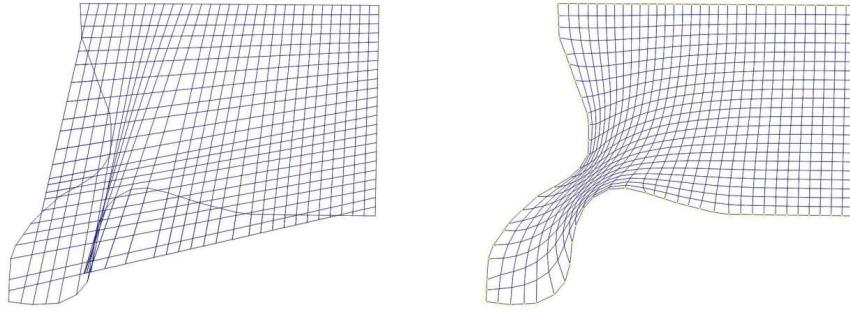


图 5.2 对有翻转单元的四边形网格的优化。左图含有大量翻转单元，右图是固定边界点优化后的网格。

其中 λ 控制以直角为目标的能量所占比重。由于以等腰直角三角形为目标，取 $\lambda = 0$ 也能得到较好的角度。若初始网格含有翻转单元，我们先去掉 E_{ver} ，采用上节所述的优化策略。一旦得到有效的网格，就加上 E_{ver} 继续优化，并在实行梯度下降时保证当前的网格无翻转单元。图3.8(d), 4.1(c) 给出了算法对模型 lake-0hole 网格质量的提高，图5.2消除了翻转单元并提高了网格质量。

第六章 实验与结论

我们在各种不同的平面区域上进行了实验，验证了扩展多正方形结构辅助算法的优越性，把生成的四边形网格和无辅助的结果做了比较和讨论。最后提出了把算法扩展到三维情形生成六面体网格的展望。

6.1 实验与比较

我们的实验在一台 3.4GHz CPU, 8GB RAM 的电脑上运行。[6.1.1](#) 节展示了不同参数下算法产生的扩展多正方形结构以及四边形网格不同的奇异结构。[6.1.2](#) 节比较了本文的方法辅助[第三章](#)介绍的基于中轴线分割的方法（以下简记为 **MA**）和 [[7](#)] 的 **Looping** 方法（以下简记为 **LQ**）产生的四边形网格，后者是用软件 AQUTOMESH-2D 生成的。

6.1.1 可控的奇异性

输入不同的 N 以及高斯核宽度 k ，我们的算法将 2 维网格变形为不同的扩展多正方形结构，并由此产生奇异性可控的四边形网格。由于奇异点的数目和四边形网格的质量往往是相互矛盾的，即：为了更好的逼近连续的边界，较高质量的四边形网格往往需要适当的奇异点，但引入奇异点又会影响有限元分析的解的性态。所以算法对奇异性的控制非常重要。

[图6.1](#)展示了不同的 N 对 tool 模型变形得到的扩展多正方形结构以及用中轴线分割方法产生的四边形网格，高斯核宽度 k 均为 4。tool 模型的法向几乎是均匀分布的，故 N 越大，[第二章](#)的网格变形算法中的目标法向就越多，对原始模型的逼近越好，但产生的奇异点也越多。

[图6.2](#)展示了不同的高斯核宽度 k 产生的结果， N 取 8。 k 越大，变形过程中对网格边界的光滑程度越大，且网格标记时会吞掉越长的边，因此得到的多正八边形结构对原始模型的简化程度越大，奇异点也越少。

6.1.2 高质量且奇异点较少的四边形网格

我们用扩展多正方形结构分别辅助了 **MA** 和 **LQ** 方法，后者是基于循环迭代的几何分割法，虽然能产生很规则的四边形单元，但会产生许多不必要的奇异点。[表6.1](#)统计了算法生成的四边形网格的缩放 Jacobian 最大值 J_{max} 、最小值 J_{min} 、平均值 J_{avg} ，角度的标准差 σ_{angle} 以及奇异点的数量 ($\#Singularity$)。* 表示有扩展多正方形结构的辅助，[图6.3](#)展示了对应的四边形网格，颜色越浅表示缩放 Jacobian 值越接近理想值 1。

通过比较可以看出，扩展多正方形结构的辅助均能有效地减少奇异点，同时产生与原算法结果相当的 J_{avg} 。**MA** 算法对中轴线结构的减支 ([3.2.3](#)) 使得它能忽

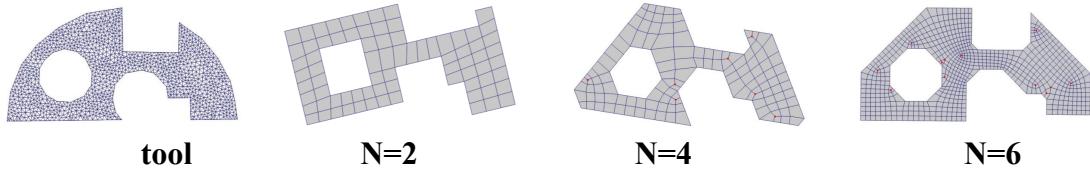
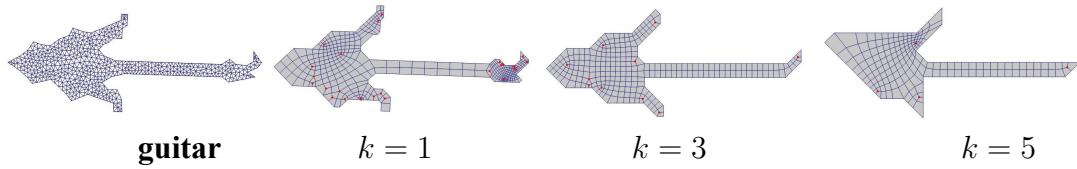


图 6.1 tool 模型产生的不同的扩展多正方形结构以及对应的四边形网格

图 6.2 不同的 k 产生的 guitar 模型的扩展多正方形结构及其四边形网格

略不必要的奇异点，但我们的算法通过简化边界不仅进一步减少了奇异点，而且在 cactus 模型上显著地提高了 J_{avg} 。LQ 算法本身无法控制奇异点的数目，但我们对边界的简化显著地减少了奇异点，如 lake, woody 模型，较为接近的 J_{avg} 说明这些奇异点是不必要的。

遗憾的是，我们的算法不能保证最差单元质量，即 J_{min} 的限度。LQ 方法产生的 J_{min} 最高，然而用我们的方法辅助后降低了 J_{min} 。从图 J_{min} 中可以看出我们的算法产生的最差单元通常在边界附近，一方面是由边界附近的形变较大，这导致映射回原区域时这些单元发生了较大的扭曲，另一方面是我们在提升网格质量时，算法会固定拐角点，并限制其他边界点在线段上滑动，这导致边界和拐角点附近单元的质量难以被提高。如果我们能基于原始的连续边界来优化网格，比如让现有的边界点沿着原始边界移动，那么这些单元的质量可能会得到提升。

6.2 总结与展望

本文提出了扩展多正方形结构的生成算法，并在它的辅助下自动地生成了高质量、奇异性可控的四边形网格。算法输入的 N 和高斯核宽度 k 控制了扩展多正方形结构的复杂度，因而我们能够生成不同奇性的四边形网格，尤其可以产生尽量少的奇异点，提高有限元方法解的稳定性和精度。我们完善并实现了 [9] 基于中轴线对区域的分割算法，最终只得到 3, 4, 5 边界的简单区域，将其分块四边形化进而得到整个区域的四边形网格。特别地，我们的算法还克服了中轴线结构对区域的几何特征不够敏感的缺点（见 3.2.3 节）。通过移动四边形网格点的位置，我们能处理翻转的四边形网格并提高网格的质量。

我们的算法还存在一些不足。由于中间域——扩展多正方形结构的引进，最终得到的四边形网格的质量依赖于原始区域与中间域之间映射的形变。在边界附近，较大的形变和网格点较小的自由度导致了较差的网格单元，因此我们不能保证最差单元的质量。根据问题的实际应用，把离散的边界点与原始的连续边界结合起来优化边界附近的网格单元也许能够提高最差的单元，这也是未来需要

表 6.1 几种方法生成四边形网格质量的比较

模型	# 顶点 / # 四边形	J_{min}/J_{avg}	σ_{angle}	#singularity
cactus-LQ	402/323	0.58/0.94	13.83	40
cactus-LQ*	428/344	0.21/0.92	17.64	34
cactus-MA	617/544	0.17/0.88	23.48	9
cactus-MA*	613/520	0.40/0.96	11.26	4
dino-LQ	1151/1024	0.38/0.95	12.8	51
dino-LQ*	1059/948	0.25/0.94	12.69	42
dino-MA	1377/1248	0.31/0.92	18.33	16
dino-MA*	923/825	0.32/0.96	12.16	11
lake-LQ	1009/891	0.51/0.93	14.73	159
lake-LQ*	615/524	0.16/0.95	12.63	50
lake-MA	995/877	0.43/0.93	16.35	16
lake-MA*	655/562	0.31/0.94	14.6	7
shark-LQ	544/473	0.34/0.94	16.06	71
shark-LQ*	606/532	0.24/0.95	12.41	29
shark-MA	599/536	0.3/0.94	14.26	10
shark-MA*	492/423	0.21/0.95	13.31	4
woody-LQ	1204/1111	0.69/0.97	9.39	114
woody-LQ*	1047/964	0.35/0.96	13.36	52
woody-MA	1287/1206	0.31/0.95	14.67	14
woody-MA*	1751/1648	0.29/0.98	9.19	12
guitar-PQ	926/797	0.34/0.95	13.18	84
guitar-LQ	550/458	0.48/0.93	14.87	82
guitar-MA	1053/920	0.15/0.95	13.09	10
guitar-Ours	845/756	0.28/0.94	16.17	4

的探索之一。

在三维情形下，法向为 6 个坐标轴向的封闭形态称为多立方体结构 (Poly-Cube)。[25],[30],[31] 等工作研究了多立方体结构的自动生成算法，并利用它自然的垂直特性得到直接六面体网格，进而映回原始的区域。但由于这样的网格内部完全没有奇异点，也限制了六面体网格的质量以及多立方体结构的其他应用。受到他们的启发，我们的算法一方面提出了用更多法向逼近原始区域的结构，这可以推广到三维以更好地逼近原始形态；另一方面，平面区域的中轴线结构也可以被扩展为空间区域的中轴面结构，我们希望可以在三维的简化结构上鲁棒地计算中轴面结构，并且由它导出内部有少量奇异点的六面体网格。

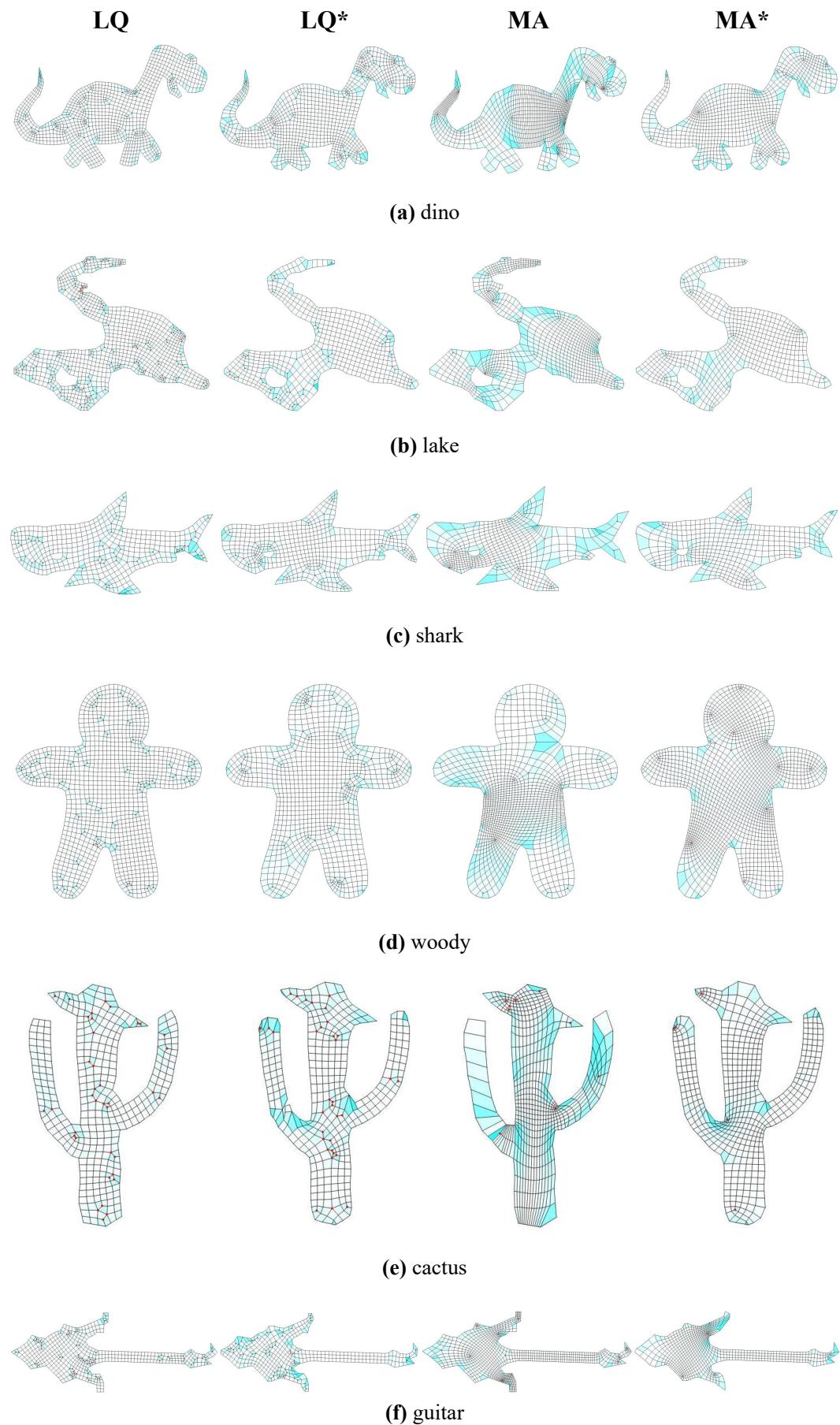


图 6.3 实验结果比较，从左到右依次为 LQ,LQ*,MA,MA* 生成的四边形网格

参考文献

- [1] Armstrong C G, Fogg H J, Tierney C M, et al. Common themes in multi-block structured quad/hex mesh generation[J]. Procedia Engineering. 2015, 124:70–82.
- [2] Johnston B P, Sullivan J M, Kwasnik A. Automatic conversion of triangular finite element meshes to quadrilateral elements[J]. International Journal for Numerical Methods in Engineering. 1991, 31(1):67–84.
- [3] Lee C K, Lo S. A new scheme for the generation of a graded quadrilateral mesh[J]. Computers & structures. 1994, 52(5):847–857.
- [4] Chae S W, Jeong J H. Unstructured surface meshing using operators[C]. Proceedings, 6th International Meshing Roundtable. [S.l.], 1997: 281–291.
- [5] Talbert J A, Parkinson A R. Development of an automatic, two-dimensional finite element mesh generator using quadrilateral elements and bezier curve boundary definition[J]. International Journal for Numerical Methods in Engineering. 1990, 29(7):1551–1567.
- [6] Nowotny D. Quadrilateral mesh generation via geometrically optimized domain decomposition[C]// Citeseer. Proc. 6th Int. Meshing Roundtable. [S.l.]: Citeseer, 1997: 309–320.
- [7] Ma X, Zhao G. An automated approach to quadrilateral mesh generation with complex geometric feature constraints[J]. Engineering with Computers. 2015, 31(2):325–345.
- [8] Baehmann P L, Wittchen S L, Shephard M S, et al. Robust, geometrically based, automatic two-dimensional mesh generation[J]. International Journal for Numerical Methods in Engineering. 1987, 24(6):1043–1078.
- [9] Tam T, Armstrong C G. 2d finite element mesh generation by medial axis subdivision[J]. Advances in engineering software and workstations. 1991, 13(5):313–324.
- [10] Rigby D. Topmaker: a technique for automatic multi-block topology generation using the medial axis[C]// American Society of Mechanical Engineers. ASME/JSME 2003 4th Joint Fluids Summer Engineering Conference. [S.l.]: American Society of Mechanical Engineers, 2003: 1991–1997.
- [11] Blacker T D, Stephenson M B. Paving: A new approach to automated quadrilateral mesh generation[J]. International Journal for Numerical Methods in Engineering. 1991, 32(4):811–847.
- [12] White D R, Kinney P. Redesign of the paving algorithm: Robustness enhancements through element by element meshing[C]. 6th International Meshing Roundtable: volume 10. [S.l.], 1997: 830.
- [13] Thacker W, Gonzalez A, Putland G. A method for automating the construction of irregular

- computational grids for storm surge forecast models[J]. Journal of Computational Physics. 1980, 37(3):371–387.
- [14] Haber R, Shephard M S, Abel J F, et al. A general two-dimensional, graphical finite element preprocessor utilizing discrete transfinite mappings[J]. International Journal for Numerical Methods in Engineering. 1981, 17(7):1015–1044.
- [15] Fogg H J, Armstrong C G, Robinson T T. New techniques for enhanced medial axis based decompositions in 2-d[J]. Procedia Engineering. 2014, 82:162–174.
- [16] Kowalski N, Ledoux F, Frey P. Automatic domain partitioning for quadrilateral meshing with line constraints[J]. Engineering with Computers. 2015, 31(3):405–421.
- [17] Sethian J A. Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science: volume 3[M]. [S.l.]: Cambridge university press, 1999.
- [18] Boyd S, Vandenberghe L. Convex optimization[M]. [S.l.]: Cambridge university press, 2004.
- [19] Hormann K, Greiner G. MIPS: An efficient global parametrization method[R]. [S.l.]: [s.n.], 2000.
- [20] Degener P, Meseth J, Klein R. An adaptable surface parameterization method.[J]. IMR. 2003, 3:201–213.
- [21] Smith J, Schaefer S. Bijective parameterization with free boundaries[J]. ACM Transactions on Graphics (TOG). 2015, 34(4):70.
- [22] Fu X M, Liu Y, Guo B. Computing locally injective mappings by advanced mips[J]. ACM Transactions on Graphics (TOG). 2015, 34(4):71.
- [23] Tam T, Armstrong C G. Finite element mesh control by integer programming[J]. International Journal for Numerical Methods in Engineering. 1993, 36(15):2581–2605.
- [24] Fabri A, Pion S [C]// ACM. Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems. [S.l.]: ACM, 2009: 538–539.
- [25] Gregson J, Sheffer A, Zhang E. All-hex mesh generation via volumetric polycube deformation[C]// Wiley Online Library. Computer graphics forum: volume 30. [S.l.]: Wiley Online Library, 2011: 1407–1416.
- [26] Liu Y. Hlbfgs[J]. <http://research.microsoft.com/en-us/um/people/yangliu/software/HLBFGS>. 2009.
- [27] Mosek A. The mosek optimization software[J]. Online at <http://www.mosek.com>. 2010, 54.
- [28] Zienkiewicz O C, Zhu J Z. A simple error estimator and adaptive procedure for practical engineering analysis[J]. International Journal for Numerical Methods in Engineering. 1987, 24 (2):337–357.
- [29] Escobar J, Rodriguez E, Montenegro R, et al. Simultaneous untangling and smoothing of

- tetrahedral meshes[J]. Computer Methods in Applied Mechanics and Engineering. 2003, 192(25):2775–2787.
- [30] Livesu M, Vining N, Sheffer A, et al. Polycut: monotone graph-cuts for polycube base-complex construction[J]. ACM Transactions on Graphics (TOG). 2013, 32(6):171.
- [31] Huang J, Jiang T, Shi Z, et al. ℓ_1 -based construction of polycube maps from complex shapes[J]. ACM Transactions on Graphics (TOG). 2014, 33(3):25.