

中国科学技术大学

学士学位论文



高亏格曲面切割方法研究

作者姓名： 刘紫檀

学科专业： 计算机科学与技术

导师姓名： 刘利刚 教授

完成时间： 二〇二一年五月二十三日

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



Computing cuts for high genus surfaces

Author: Zitan Liu

Speciality: Computer Science and Technology

Supervisor: Prof. Ligang Liu

Finished time: May 23, 2021

致 谢

时光荏苒，日月如梭。转眼之间，我已经在科大度过了第四个年头，这四年也是我成长最快的时光。在科大，有幸结识了许多良师益友，在与他们的交往中，我不仅学习到了各种知识，还懂得了许多人生道理。

感谢我的导师刘利刚老师。有幸从图形学课程上结识刘利刚老师，刘利刚老师深厚的学术功底，严谨的工作态度使我受益良多。衷心感谢刘利刚老师给予我的悉心教导和热情帮助。

感谢傅孝明老师，与几何处理相关的问题都是在他的帮助下完成的。傅老师无时无刻不在热情拼搏，对研究项目总是充满干劲。他对学术研究的热情令我敬佩，直爽而专注的为人也值得我学习。

感谢叶春阳学长在毕业设计中提供的理论和实现方面的指点和照顾。在毕业设计过程中，学长始终不厌其烦，谆谆教诲，提供了许多帮助，解决了许多实现上的相关疑惑。

同时，也要感谢和我一起讨论问题的同学。潘星越同学在群论和拓扑学基础上给予了懵懂的我很多基本问题上的帮助和指导，王梓越同学在曲面切割问题上的直觉让我受益匪浅，黄业琦同学在听我介绍曲面切割问题后提出的疑问启发了我柄圈计算的实现。

感谢四年里结识的众多好友。感谢舍友和准舍友们，每天叫我去干饭，然后讨论天马行空的稀奇古怪问题；感谢图形和计算几何实验室一起学习的同学们，与他们的讨论让我受益良多；感谢先进计算机体系结构实验室的师兄师姐们，他们指导我完成过不少并行和通信优化的工作。还有很多人想感谢，但是这里空太小，写不下，就让我在心里默默的记下来吧。

最后要感谢我的家人。是您们在二十二年来的陪伴和支持，让我能够安心学习和进步。感谢王志恒同学对我的关心和支持，很高兴能够遇见你。

目 录

中文内容摘要	2
英文内容摘要	3
第一章 绪论	4
第一节 网格切割	4
第二节 相关工作	5
第三节 本文内容	5
第二章 网格切割的基本方法	7
第一节 基本概念	7
第二节 沿割线切割曲面	7
第三节 简单网格切割算法实例	9
第三章 基于柄圈的网格切割算法	10
第一节 简介	10
第二节 单纯同调论	10
第三节 闭曲面分类	12
第四节 一维链, 闭链, 边缘链和同调群的基表示	13
第五节 计算最短同调圈	14
一、利用紧性质缩小搜索空间	15
二、利用标注快速判断同调关系	15
三、算法框架	16
第六节 计算柄圈	16
第四章 算法实现	18
第五章 总结与展望	20
参考文献	21

中文内容摘要

计算机图形学在近年来发展迅速。无论是在工业设计、互动娱乐还是在日常生活中，图形学的应用无处不在。构建图形学大厦的基石之一就是连续曲面的离散表示——网格，而许多几何处理方法依赖网格具有一定的拓扑性质，这其中绝大多数方法又要求网格为圆盘拓扑。考虑到日常我们遇到的网格大多非圆盘拓扑，所以将高亏格的网格在位置不变的情况下，通过连通性的变化变为圆盘拓扑的网格，是数字几何处理应用的一个基本问题。

为了精确的讨论曲面网格概念本身，本文首先介绍了曲面和曲面网格的定义。之后，本文给出了割线集的定义，以及给定割线集情况下的基于 Euler 图的局域网格切割基本方法。作为一种典型方法的说明，本文介绍来自 **Geometry Images** 中的简单网格切割算法。

一些几何处理问题，如参数化和图集生成，需要寻找长度尽可能短的割线集，但寻找最优割线集问题本身是 **NP-Hard** 问题。本文回顾了代数拓扑中单纯同调论的基本知识，总结了割线集优化的相关结果，并且复现了计算在权重函数最短意义下的 \mathbb{Z}_2 域一维同调群基向量组的工作，此工作可以为计算基于柄圈的闭曲面网格切割算法提供帮助。之后，本文介绍了柄圈的形式化定义和基本计算方法，并介绍了基于 **TetGen** 网格剖分工具进行柄圈计算的实现思路。

最后，本文对比了实现的部分方法在不同模型数据上的性能表现，并且探讨了之后的工作可能的发展方向。

关键词：计算机图形学；数字几何处理；割线集构造；高亏格曲面展开；柄圈

Abstract

Computer graphics has been rapidly evolving in recent years. Being widely used in industrial design and interactive entertainment, computer graphics are used everywhere in our daily life. One of the foundations of computer graphics lies on meshes, which are discrete representations of continuous surfaces. Many geometry processing procedures require certain topological characteristics of the input mesh, and in which a topological disk is required most. Considering that most of the meshes we met are not topologically equivalent to a disk, therefore, it's a fundamental problem for digital geometry processing applications to generate a mesh topologically equivalent to a disk from a high genus meshes, with transformations imposed on connectivity instead of geometry only.

This dissertation first introduces the definition of surface and surface meshes. Then, given the cut graph as input, an Euler-circuit based local mesh cutting algorithm is shown. After that, a simple example taken from *Geometry Images* is shown.

Some of the geometry processing applications, such as parameterization and atlas generation, requires a cut graph with its total length as minimal as possible. This problem, however, is NP-Hard in general. This dissertation recalls elementary knowledge of algebraic topology and simplicial homology, summarizes related conclusions related to cut graph optimization. Then, this dissertation gives an implementation of calculating the minimal homology basis under \mathbb{Z}_2 coefficient field with respect to the weight function, which can be used for the cutting algorithm based on handle loops. After that, this dissertation introduces the concept of handle and tunnel loops, how they can be used in mesh cutting algorithms, and an experimental implementation with the help of mesh tetrahedralization utility TetGen.

In the end, this dissertation compares performances on different models of the algorithms implemented, and discusses the possibilities of future studies.

Key Words: Computer Graphics; Digital Geometry Processing; Cut Graph Generation; High Genus Surface Expansion; Handle Loop

第一章 绪论

第一节 网格切割

曲面网格，又经常被简称为网格，是曲面的分片线性表示，是计算机图形学的基石。栩栩如生的 3D 游戏和 CG 电影，以及在工程上广泛应用的有限元物理仿真方法，其背后离不开高精度的网格。

为了让网格本身适应各种场景下的应用，或是为了获得某些网格的变体，研究者们发明了许多网格的处理和变换过程，这些过程又被称为网格的几何处理。

参数化，如图 1.1 所示，即寻找网格到另一个网格或平面的映射，并且使其满足一些特定的要求，是典型的几何处理问题之一^[1]。以纹理映射要求下的参数化应用为例，用户希望获得一个网格到 $[0, 1] \times [0, 1]$ 的映射，并且将模型的颜色、法线细节等绘制于这个映射到的平面区域中，从而在节约计算资源的同时增强模型的真实感和表现力^[2]。

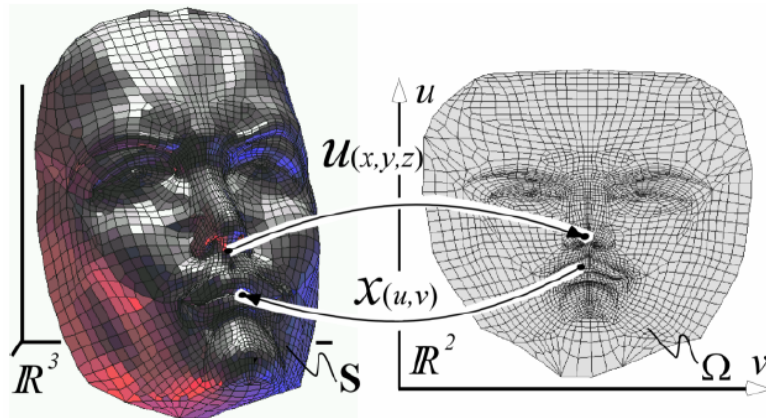


图 1.1 参数化示意图，两个方向的映射分别用 u 和 x 表示

图片源自互联网

曲面网格的许多几何处理过程，如参数化，都依赖网格的拓扑信息。在某些场景下，我们希望能将带有复杂拓扑的网格，在保持几何信息近似不变的情形下变为圆盘拓扑，这就是曲面的切割问题。

以纹理映射为例，我们希望获得一个网格曲面到 $[0, 1] \times [0, 1]$ 的双射，而此映射存在的必要条件是两个拓扑空间同胚。这时，唯一的方法就是将该网格曲面切割，使结果拓扑同胚于圆盘。^[3] 不过，我们注意到对个别情况来说，纹理会被嵌入到二维球 S^2 上，这时候我们需要同胚于球面的网格^[4]。

对于不同的几何处理过程，对三维网格的要求各不相同，切割时应加以考

虑。对于纹理映射来说，一个常见的要求是寻找使参数化后三角形形变最小的切割方案^{[1][5]}。另一个常见的要求是令割线尽可能短，这在提高纹理映射后图集生成的空间利用率上很有帮助^[6]。

第二节 相关工作

由于网格切割在计算机图形学中的重要地位，切割和割线优化问题被研究者们广泛研究。寻找可以将网格切割为圆盘的割线是快速的，Dey et al.^[7]说明只需要对偶图进行广度优先搜索，就可以在 $O(n)$ 的时间复杂度下得到将网格切割为圆盘拓扑的割线；Gu et al.^[5] 中给出了一个实现上快速且简便的网格切割算法；De Verdière et al.^[8] 给出了基于同伦等价的最短圈集合的网格切割算法。

但是，优化割线长度是比较困难的。Erickson et al.^[3] 证明了，对于二维流形网格来说，计算其总长度或总段数意义下的最短割线是 NP-Hard 问题，并且提出了 $O(g^2 n \log n)$ 时间复杂度的最小切割图近似算法；Dey et al.^[9] 定义了柄圈，并给出了计算曲面上较短柄圈的实现^{[10][11]}；Chai et al.^[12] 指出柄圈可能是优化割线长度的曲面切割问题的好的启发。

对于网格参数化过程而言，除了割缝长度外还存在其它优化目标。传统上来说，参数化由两个问题构成：寻找割缝切割曲面^{[13][14]} 和最小化扭曲^{[15][14]}。早期的工作大多只考虑了一个问题，或者将两个问题顺序优化。之后，也有同时考虑两方面问题的工作出现^[16]。

除此之外，许多工作也侧重于割缝生成的其它方面。Poranne et al.^[17] 提供了交互式的割缝生成和参数化展开框架，满足了参数化时低扭曲和对割缝位置的两方面要求；Bruno et al.^[18] 实现了改进的 ABF++ 方法来实现交互式割缝生成和参数化迭代；Teimury et al.^[19] 基于图神经网络 (GNN) 实现了支持语义的割缝生成系统，从而将扭曲和割缝长度之外的因素考虑进来。

第三节 本文内容

本文首先介绍了曲面和曲面网格的定义，并给出了给定割线集情况下的基于 Euler 图的局域网格切割基本方法。之后，本文给出了来自 Geometry Images 中的简单网格切割算法实例。

一些几何处理问题，如参数化和图集生成，需要寻找长度尽可能短的割线

集。本文回顾了代数拓扑中单纯同调论的基本知识,总结了割线集优化的相关结果,并且复现了计算在权重函数最短意义下的 \mathbb{Z}_2 域一维同调群基向量组的工作,此工作可以为计算基于把手链的闭曲面网格切割算法提供帮助。之后,本文介绍了柄圈的形式化定义和基本计算方法,并给出了基于 TetGen 网格剖分工具的实验性实现。

最后,本文给出了部分算法的实现。

第二章 网格切割的基本方法

第一节 基本概念

本文研究的网格是指由顶点、边和三角面片组成的（紧）曲面网格。形式上来说，此处定义的网格可以看作一个满足如下约束的集合 \mathcal{M} ，其中：

- 集合中的元素为 0-单形（点），1-单形（直线段）或 2-单形（三角面）
 - 集合中任意两个元素的交集仍在集合中，如两个 1-单形的交集仍在网格上
- 这样的网格记作 $\mathcal{M}(V, E, F)$ ，其中 V 为顶点子集， E 为边子集， F 为面子集。

自然的，网格可以看作光滑曲面的分片近似，所以我们可以将曲面的相关概念也拓展到离散情形中。我们知道，二维流形是在每个局部与 \mathbb{R}^2 同构的曲面，则定义任意一条边相邻的三角面片数量不超过两个的网格为流形网格。

常见的几何处理方法都需要流形网格本身是可定向的，这主要是因为无边界的二维流形无法在无自交的情形下嵌入到 \mathbb{R}^3 中。

网格的切割，直观上说就是“沿着缝隙把曲面裁开”的过程，如图 2.1 所示。给定一个边集合 C 满足 $|C| > 1$ ，切割过程可以定义为一个边和网格的作用 $\mathcal{M} \times E \rightarrow \mathcal{M}'$ ，其中 \mathcal{M}' 满足如下性质：

- $f = f'$ （面的数量和位置保持不变）
- $f \in \mathcal{M}$ 且 $\partial f \cap C = \emptyset$ ，则 $\partial f = \partial f'$ （不在割缝上的边保持不变）
- $f \in \mathcal{M}$ 且 $\partial f \cap C \neq \emptyset$ ，则对使得 $e_p \in \partial f_m$ 且 $e_p \in \partial f \cap C$ 的 e_p ，有 $e_p \notin \partial f' \cap f'_m$ （在割缝上的边所相邻的两个三角面现在不相邻）

注意到满足上述定义的过程有很多种，但是我们通常会要求切割后的网格增加的边数最少，以及切割后网格中面的几何位置不改变。

我们关心的展开成拓扑圆盘的切割，即是寻找一些特定的 C ，使得其进行切割作用后，剩余的网格构成圆盘拓扑。

第二节 沿割线切割曲面

对于已知割线集合 C 的情形，本节探讨如何从 C 得到切割后的曲面 \mathcal{M}' 。

从割线的定义出发，可以得到一个较为朴素的办法：我们可以将所有三角面分割成为一个个的孤岛，如果两三角面片相邻的边不属于割线，则将两三角面片

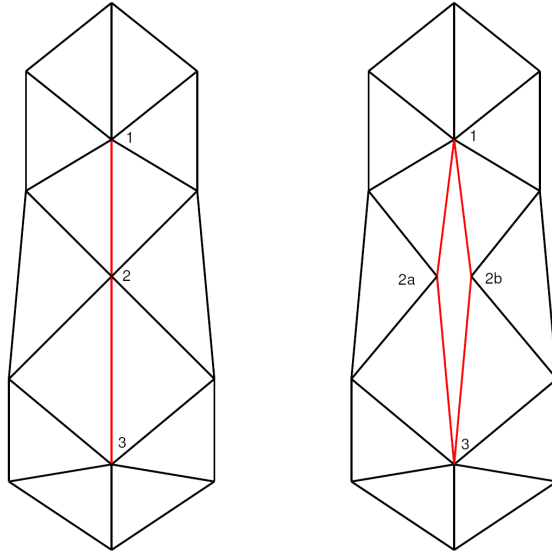


图 2.1 曲面切割示意图，红色为割线

“岛屿”相连。可以观察到，此方法的时间复杂度正比于 $|E| + |F|$ 。

不过，曲面切割本质上是一个局域操作，上述操作的复杂度过高。考虑到割线作为边集合，如果加上在边上的点，就可构成无向图 $G(V_C, C)$ 。此时我们可以进行如下操作：

1. 对图 G 的每个连通片 G_i 的边 $v_i v_j$ ，添加 $e' = v_i v_j$ 使得 $v_i v_j$ 变成重复度为 2 的重边
2. 利用 Fleury 算法计算每个 G_i 的欧拉回路 $v_i v_j v_k \dots v_i$
3. 从起点开始遍历欧拉回路路径，设遍历到 $v_m v_n$ ，令前序顶点引用 v_{pre} 初值为 v_i
 - 如果 $v_n v_m$ 未遍历过，则增加新顶点 v'_n 。构造 $e' = v_{pre} v'_n$ 并将 $v_{pre} v_n$ 右侧的三角面的顶点和边分别替换为 v_{pre} ， v'_n 和 $e = v_{pre} v'_n$ ，同时更新 v_{pre} 引用为 v'_n
 - 如果 $v_n v_m$ 已经遍历过，则更新 v_{pre} 引用为 v_m

不过，上面的算法只对无边界的网格适用，对于有边界的网格，应用上面的做法可能会将流形网格切为非流形网格，如图 2.2 所示。此时，我们应该对边界点的情况进行单独处理，即在第三步遍历时，如果起点或终点的某个点为边界点，则将其完全断开。

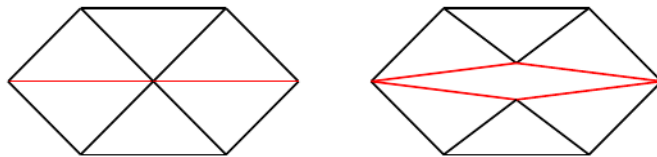


图 2.2 有边界网格情况示意，红色为割线

第三节 简单网格切割算法实例

本节以 Gu et al.^[5] 中提出的网格切割算法为例, 通过实例介绍一个简单的网格切割算法是如何构造的。

Gu et al.^[5] 中提出的网格切割算法可以由一个简单的想法产生: 三角面片本身是圆盘拓扑。那么, 如果我们假设初始时选择一个三角面, 在每次迭代时向集合 A 加入这样的三角面: 他只与某个在 A 中的三角面有一条边相接, 如果有更多边相接, 就将剩余的边切开。

这样一来, 每次迭代结束时就都是圆盘拓扑, 直到 A 中有所有的三角面。对于连通的网格来说, 这种方法就可以找到一个割线集合, 且切割后为圆盘拓扑。

同时, 我们可以观察到, 割线集作为边集合构成的无向图 G , 可以分解成为许多圈和树边的并, 而在网格内部给割线集增加树边, 并不会改变切割的拓扑。这样, 我们可以得到该网格切割算法的核心思想。

算法由两阶段组合而成。第一阶段时, 算法不停的寻找一个不在边界上, 且只与一个三角面 t 相邻的边 e , 并且将 e 和 t 移除。在此阶段结束时, 就可以得到一个已经可以将网格切割为圆盘拓扑的割线集。第二阶段时, 算法不停的寻找割线集的树边并删除, 从而化简割线集。

在第一阶段, 为了尽量得到“最小半径”的切割, 在选择移除的三角面时可以使用种子三角面的测地线距离进行选择。

算法 2.1 来自 Geometry Images^[5] 的网格切割算法

Data: 要切割的网格

Result: 割线集 C

```

1 移除种子三角面;
2 while 有只和一个三角面  $t$  相邻的边  $e \notin B$  do
3   | 移除  $e$  和  $t$ ;
4 end
5 while 有只有一条相邻的边  $e$  的  $v$  do
6   | 移除  $v$  和  $e$ ;
7 end
```

第三章 基于柄圈的网格切割算法

第一节 简介

Dey et al.^[9] 基于几何直观, 发现曲面网格的“把手”和“通道”等特征对于拓扑简化和特征检测都很有帮助, 因此提出了柄圈和其形式化定义。在此基础上, Chai et al.^[12] 认为柄圈是最短割缝问题的一个好的启发, 并且实现了基于柄圈进行曲面切割的方法。

直观上, 柄圈和洞圈描述了曲面的一些几何特征, 这些特征可以用来切割曲面。图 3.1 展示了一个 CAD 工件上的一些柄圈和洞圈。

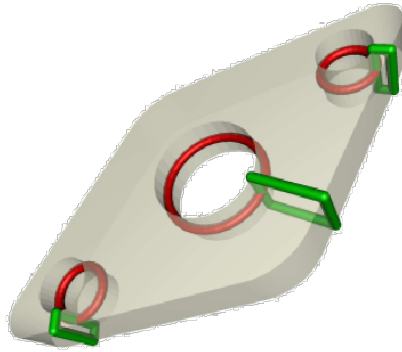


图 3.1 CAD 工件: 红色圈为洞圈, 绿色圈为柄圈

图片源自 Dey et al.^[9]

本节首先介绍柄圈形式化描述所需要的基本知识, 即单纯同调论和 \mathbb{Z}_2 系数域下同调论描述的几个群的基表示, 然后再给出计算 $H_1(K; \mathbb{Z}_2)$ 的最优基的算法。之后, 本节会介绍柄圈的形式化定义, 以及一个基于前述算法的、计算柄圈的算法。

第二节 单纯同调论

单纯同调论 (Simplicial homology) 是描述欧氏空间中具有组合结构的空间的同调论。根据其研究的主要方法, 单纯同调论又被称为组合拓扑学。本节更详细的内容可以参考^[20]。

R_n 中处于一般位置的 $n+1$ 个点 $\{a_0, \dots, a_n\}$ ($n > 0$) 的凸组合被称为一个 n 维单纯形, 简称 n -单形, 记作 (a_0, a_1, \dots, a_n) 。称 a_i 为其顶点。

设 K 为以单形为元素的有限集合, 如果 K 满足

1. K 中任意两个 m -单形交集为空集, 或相交部分为某个在 K 中的 $m-1$ -单形
2. 构成 m -单形顶点组成的 $m-1$ -单形仍在 K 中

则称 K 为单纯复合形, 又称为单复形。复形中所有维度不超过 m 的单形构成 K 的子复形, 称为 K 的 m -骨架, 记作 K^m 。

可以看出, 单复形可以表示欧氏空间的一个子集, 如果有这样的子集 X 可以被单复形中的所有单形的并表示, 则成这样的 X 为多面体。显然, 我们研究的网格就是这样的多面体。

为了构造同调群, 我们还需要引入定向单形的概念。定向单形为取定了定向的单形, 此时单形分为相同和相反的两种定向, 这两种定向在稍后定义的链群中互为逆元。形式上的说, 即考虑 m -单形 (p_0, p_1, \dots, p_m) , 则

$$p_{i_0} p_{i_1} \cdots p_{i_m} = \text{sgn}(P) p_0 p_1 \cdots p_m$$

其中 P 为 $\{0, 1, 2, \dots, m\}$ 到 $\{i_0, i_1, i_2, \dots, i_m\}$ 的一个排列, $\text{sgn}(P)$ 为此排列的符号。

K 中所有 m -单形所对应的 m -定向单形所生成的在 \mathbb{Z} 系数域上的自由阿贝尔群称为 m -链群, 记作 $C_m(K)$ 。 $C_m(K)$ 的元素称为 m -链。为了叙述上的方便, 定义 $m < 0$ 或者 $m > \dim K$ 时 $C_m(K) = 0$ 。

定义边缘算子 $\partial_q : T_q(K) \rightarrow C_{q-1}(K)$ 满足

$$\partial_q(a_0 a_1 \cdots a_q) = \sum_{i=0}^q (-1)^i a_0 \cdots \hat{a}_i \cdots a_q$$

其中 \hat{a}_i 表示删除 a_i , $T_q(K)$ 为 K 的所有 q -定向单形的集合。可以看出, 边缘算子描述了一个单形组合和其对应的“边界”的关系。同时可以证明, ∂_q 在满足 $\partial_q(-s) = -\partial_q(s)$ 的情形下可以唯一的扩张为 $C_q(K) \rightarrow C_{q-1}(K)$ 的同态。为了方便, 此同态仍然记作 ∂_q , 则有

$$\partial_q\left(\sum_i n_i s_i\right) = \sum_i n_i \partial_q(s_i)$$

成立。

在考虑边缘同态的过程中, 我们可以自然的得到 $C_q(K)$ 的两个子群 $Z_q(K) = \ker(\partial_q(K))$ 和 $B_q(K) = \text{Im}(\partial_{q+1}(K))$, 分别称为 q -闭链群和 q -边缘链群。 $Z_q(K)$ 中的元素称为 q -闭链。显然, $Z_q(K)$ 和 $B_q(K)$ 都是自由阿贝尔群。同时可以证明,

$B_q(K)$ 是 $Z_q(K)$ 的子群, 这可以由下面的事实得到 ($c \in C_q(K)$):

$$\begin{aligned}
 \partial_{q-1}(\partial_q(c)) &= \partial_{q-1}\left(\sum_{i=0}^q (-1)^i a_0 a_1 \cdots \hat{a}_i \cdots a_q\right) \\
 &= \sum_{i=1}^q (-1)^i \partial_{q-1}(a_0 a_1 \cdots \hat{a}_i \cdots a_q) \\
 &= \sum_{i=1}^q (-1)^i \left(\sum_{j=1}^{i-1} (-1)^j a_0 \cdots \hat{a}_j \cdots \hat{a}_i \cdots a_q + \sum_{j=i+1}^q (-1)^{j-1} a_0 \cdots \hat{a}_i \cdots \hat{a}_j \cdots a_q \right) \\
 &= \sum_{0 \leq j < i \leq q} (-1)^{i+j} a_0 \cdots \hat{a}_j \cdots \hat{a}_i \cdots a_q - \sum_{0 \leq i < j \leq q} (-1)^{i+j} a_0 \cdots \hat{a}_i \cdots \hat{a}_j \cdots a_q \\
 &= 0
 \end{aligned}$$

既然 $B_q(K)$ 是 $Z_q(K)$ 的子群, 又因为 $B_q(K)$ 是阿贝尔群, 所以 $B_q(K)$ 是正规子群。于是, 我们定义 q -同调群 $H_q(K)$ 为 $Z_q(K)/B_q(K)$, 即 $Z_q(K)$ 对 $B_q(K)$ 的商群。如果 K 中两个闭链 c, c' 属于同一个 $H_q(K)$ 确定的等价类, 则称这两个闭链同调, 记为 $c \sim c'$ 。

由于 $Z_q(K)$ 和 $B_q(K)$ 在 K 为有限集时均为有限生成自由阿贝尔群, 则他们的商群也是有限生成阿贝尔群。由有限生成阿贝尔群分类定理, 可以将群分为自由子群和扭子群的直和:

$$H_q(K) \simeq \underbrace{Z \oplus Z \oplus \cdots \oplus Z}_{b_q(K)} \oplus Z_{i_1} \oplus \cdots \oplus Z_{i_m}$$

其中 $b_q(K)$ 被称为 K 的 q 维 Betti 数。

我们在前面的推导过程中对链群使用的系数域为 \mathbb{Z} 。如果我们将 \mathbb{Z} 换为 \mathbb{Z}_2 或 \mathbb{R} , 由于 \mathbb{Z}_2 和 \mathbb{R} 均没有非平凡的扭子群, 那么 $H_q(K; \mathbb{Z}_2)$ 或 $H_q(K; \mathbb{R})$ 的结构就很容易得到:

$$H_q(K; \mathbb{R}) \simeq H_q(K; \mathbb{Z}_2) \simeq \underbrace{Z \oplus Z \oplus \cdots \oplus Z}_{b_q(K)}$$

所以 $b_q(K)$ 又可以写作 $\dim H_q(K; \mathbb{R})$ 。由此, 我们也可以得到在 \mathbb{Z}_2 下 $C_q(K)$, $Z_q(K)$, $B_q(K)$ 和 $H_q(K)$ 均为线性空间, 有维度和基的概念。

第三节 闭曲面分类

闭曲面, 即没有边界点的紧致连通曲面。我们处理的网格, 由于其分片线性表示的特点而天然具有紧致性, 故代数拓扑中研究的闭曲面的拓扑类型对于网

格处理中拓扑类型的讨论有很大帮助。

我们称环柄为环面上挖一个洞后形成的空间, $\{nT^2\}$ 即球面上安 n 个环柄所得空间。在球面上挖一个洞, 并在洞口粘结 Mobius 带, 这种操作称为在球面上安交叉帽。 $\{mP^2\}$ 即安了 m 个交叉帽的球面。可以证明, $\{nT^2\}$ 为亏格为 n 的可定向闭曲面, 而 $\{mP^2\}$ 为亏格为 m 的不可定向闭曲面。

事实上, 我们有以下定理:

定理 3.1 (闭曲面分类定理) $\{nT^2\}$ 和 $\{mP^2\}$ 不重复地列出了闭曲面的所有拓扑类型。

这个定理因为其重要性, 历史上有很多证明, 一个简单的证明可以参考^[20]。这样, 我们讨论 nT^2 类型的拓扑空间时, 大多数时候只要证明其在 n -环面的某个三角剖分上成立即可。

第四节 一维链, 闭链, 边缘链和同调群的基表示

下面的讨论均在 \mathbb{Z}_2 域下进行。值得注意的是, \mathbb{Z}_2 下 $-1 = 1$, 所以单形的定向在此空间下只有一种, 圈、路径等 $C_m(K)$ 中的元素也不需要指明定向。

一维链群在 \mathbb{Z}_2 下的线性空间表示很好得到, 其中 c_i 为边 e_i 对应的系数。

$$c = \begin{bmatrix} c_1 \\ c_2 \\ \dots \\ c_{|E|} \end{bmatrix} \in C_1(K) \quad (c_i \in \{0, 1\})$$

$C_1(K)$ 的 (自然) 基为 $(e_1, \dots, e_{|E|})$, 其中 e_i 为 K 的 1-骨架 K^1 的第 i 条边。

考虑 K^1 的某个生成树 T , 则有 $|V| - 1$ 条边在生成树上, $|E| - |V| + 1$ 条边不在生成树上。对于不在生成树上的边 e_{m_i} , 生成树提供了从边的一个端点到另一个端点的路径, 这个路径连同 e_{m_i} 形成一个圈, 记为 $\gamma(T, e_{m_i})$ 。

这样, 我们得到了 $\{\gamma(T, e_{m_1}), \dots, \gamma(T, e_{m_k})\}$, 其中 $k = |E| - |V| + 1$ 。 $\gamma(T, e_{n_i})$ 和 $\gamma(T, e_{n_j})$ 间线性无关, 因为 $e_{n_j} \in \gamma(T, e_{n_i})$ 当且仅当 $i = j$ 。

对于一个任意圈 z , 固定树 T 上的某个顶点 s , 记 $T[s, u]$ 为树上 s 到 u 的唯一路径, 那么

$$z = \sum_{e=uv \in z} e = \sum_{e=uv \in z} (T[s, u] + e + T[s, v]) = \sum_{e \in z} \gamma(T, e)$$

其中第三个等号成立是因为所有在树上的边的 $T[s, u] + e + T[s, v] = 0$ 。由

此, 我们可以看出, $\{\gamma(T, e_{m_1}), \dots, \gamma(T, e_{m_k})\}$ 构成 $Z_1(K)$ 的一组基, 矩阵表示如下:

$$Z = \left[\gamma(T, e_{m_1}) \mid \dots \mid \gamma(T, e_{m_k}) \right]$$

为了方便, 我们引入优先基的概念。给定秩为 r 的矩阵 A , 由 A 的部分列向量组成的 $B_{opt} = \{a_{i_1}, \dots, a_{i_r}\}$ 被称为优先基, 如果 $\{i_1, \dots, i_r\}$ 是让 $\text{rank}(B_{opt}) = \text{rank}(A)$ 的, 在标号顺序上最小的一组指标集。记 $\text{EarliestBasis}(A)$ 为 A 的优先基 B_{opt} 。

那么, $B_1(K)$ 的基可以用如下表示, 记为 B :

$$B = \text{EarliestBasis}(\left[\partial_1(f_1) \mid \dots \mid \partial_1(f_{|F|}) \right])$$

考虑到 $H_1(K) = Z_1(K)/B_1(K)$, 那么

$$\left[B \mid H \right] = \text{EarliestBasis}(\left[\partial_1(f_1) \quad \dots \quad \partial_1(f_{|F|}) \mid Z \right])$$

这是因为 $\dim H_1 = \dim Z_1 - \dim B_1$, 而 $\dim B_1 = \text{rank}(B)$, 所以剩下的 $\text{rank}(Z) - \text{rank}(B)$ 个列向量就是 H 的基了。

H 的维数在闭曲面且为 nT^2 类型时也被称为亏格, 用 g 表示。此时, g 和欧拉特征数 $\chi(K) = |V| - |E| + |F|$ 满足关系 $\chi(K) = 2 - 2g$, 此关系是 Euler-Poincare 公式的自然结论。

有了 H 的基, 我们就可以为每个圈都计算对应的坐标, 即同调向量 (homology vector) 了。

第五节 计算最短同调圈

Busaryev et al.^[21] 提出了利用标注 (annotation) 在 $O(n^\omega + n^2 g^{\omega-1})$ 时间复杂度下计算 $H_1(K; \mathbb{Z}_2)$ 的最优基的算法。

所谓最优基, 就是对任意 $c = \sum_i c_i s_i \in Z_1(K)$, 定义 $w(c) = \sum_i c_i w(s_i)$ 为闭链 c 的权, 并且求一组 $H_1(K; \mathbb{Z}_2)$ 的基使得 $\sum_c w(c)$ 最小。

算法的主要思路是, 在曲面上按权重从短到长排序的圈中, 挑选出前 $2g$ 个彼此互相不同调的圈构成的基, 就是最优基。为了高效的计算, 这里需要解决两个问题:

1. 如何减少需要考虑的圈的数量?
2. 如何快速判断两个圈彼此是否同调?

一、利用紧性质缩小搜索空间

问题一的解决依赖 Erickson et al.^[22] 中提出的最优基的紧性质。当一个圈 l 包含 l 上任意的两点之间的最短路径时, 称 l 具有紧性质。

命题 3.2 (Busaryev et al.^[21]) 每个 $H_1(K; \mathbb{Z}_2)$ 的最优基中的圈都具有紧性质。

这个问题的证明可以参考^[21]。

记 T_s 为从顶点 s 开始的权函数意义下的最短路径树, 即包含从 s 到任意点的最短路径的树。同时, 记候选圈集合为

$$\Pi = \bigcup_{s \in V} \{\gamma(T_s, e) | e \in E \setminus E(T_s)\}$$

我们有如下命题成立^[21]:

命题 3.3 (Busaryev et al.^[21]) 如果按权重升序排列 Π 中的圈, 并且为每个圈计算同调向量, 那么 EarliestBasis(Π) 是 $H_1(K; \mathbb{Z}_2)$ 的最优基。

原文^[21] 将实现细节与证明混合在一起了, 这里给出一个简化的证明:

证明 只要证明最优基中的圈一定在 Π 中。考虑每个圈 $\gamma(T_s, e)$, 其仅在 s 到任意一点 $x \in \gamma(T_s, e)$ 时保留紧性质。事实上, 每个紧的圈 l 都一定在某个 $\gamma(T_s, e)$ 中, 因为一定存在某个 $s_i \in l$, 且 $\{\gamma(T_{s_i}, e_1), \dots, \gamma(T_{s_i}, e_k)\}$ 生成了 1-闭链群, 由于 l 的紧性质, l 一定为其中一个而不是其中某些的线性组合。因为最优基的圈一定是保留紧性质的圈, 所以最优基的圈一定在 Π 中。 ■

通过以上命题, 我们可以将候选的圈数量变为 $O(|V|(|E| - |V| + 1))$ 。

二、利用标注快速判断同调关系

Busaryev et al.^[21] 中提出的标注方法可以帮我们解决问题二。

定义 3.1 一个 p -单形的标注是一个函数 $a : C_p(K) \rightarrow (\mathbb{Z}_2)^g$ 。两个 p -闭链 z 和 z' 彼此同调, 当且仅当

$$\sum_{\sigma \in z} a(\sigma) = \sum_{\sigma \in z'} a(\sigma)$$

此时, 该 p -闭链的标注被定义为 $a(z) = \sum_{\sigma \in z} a(\sigma)$ 。

计算标注的方法如下所示:

1. 计算 $Z_1(K)$ 的基 固定某个 K^1 的生成树 T , 记 $k = |E| - |V| + 1$, $E \setminus E(T)$ 的各边为 e_{m_1}, \dots, e_{m_k} , 则可以构造基 $\{\gamma(T, e_{m_1}), \dots, \gamma(T, e_{m_k})\}$ 。

2. 计算 $H_1(K)$ 的基 通过计算 $\text{EarliestBasis}([B|Z])$ 并移除前 $\text{rank}(B)$ 列得到的剩余列向量, 记为 H , 即为 $H_1(K)$ 的基。同时, 记 $\tilde{Z} = [B|Z]$ 。
3. 计算标注 对每个圈 $z = \gamma(T, e)$, 其坐标是满足 $\tilde{Z}x = z$ 的 x 的最后 $\text{rank}(Z) - \text{rank}(B)$ 个分量, 这些作为该圈的标注。我们令 $E(T)$ 上的边对应的标注为零向量, 令 $E \setminus E(T)$ 上的边对应的标注为对应的圈的标注, 这样就得到了所有边的标注。

有了标注, 我们只需要计算两个圈的标注是否相同, 就可以判断两个圈是否同调了。这样, 我们就解决了问题二。

三、算法框架

综上, 我们可以得到算法的框架如下:

1. 以任意一点为源点, 计算生成树 T , 然后按照第二节中的方法计算每个边的标注向量 $a(z)$ 。
2. 对 K 的每个顶点 s :
 - (a) 计算 s 出发的最短路径树, 得到圈 $\gamma(T_s, e_{m_1}), \dots, \gamma(T_s, e_{m_k})$ 。
 - (b) 计算每个圈的权重和标注, 并加入候选圈集合。
3. 选出以总权重从小到大排列的候选圈的前 $2g$ 个彼此互相不同调的圈, 即为要求的最优基。

第六节 计算柄圈

Dey et al.^[9] 在其文章中引入了柄圈 (handle loop) 和洞圈 (tunnel loop) 的概念。

考虑一个可定向闭曲面 \mathcal{M} , 将其放在一个三维球 S^3 中, 其将 S^3 分为两部分。给定曲面的定向, 我们称 $S^3 \setminus \mathcal{M}$ 的两个连通片分别为内部和外部, 分别记作 I 和 O 。我们定义

$$\mathbb{I} = I \cup \mathcal{M}, \quad \mathbb{O} = O \cup \mathcal{M}$$

那么, 我们可以给出柄圈和洞圈的定义。

定义 3.2 $c \in Z_1(M)$

1. c 是柄圈, 如果 $c \in B_1(\mathbb{I})$ 且 $c \notin B_1(\mathbb{O})$
2. c 是洞圈, 如果 $c \in B_1(\mathbb{O})$ 且 $c \notin B_1(\mathbb{I})$

我们观察到, 如果我们沿着柄圈切割 \mathcal{M} , 并且将每个柄圈切割后剩下的“空

洞”分别用两个圆盘补上，我们就可以“消灭”一个柄圈。这可以由以下命题以及闭曲面 nT^2 的性质得到：

命题 3.4 (Dey et al.^[9]) 对于亏格为 g 的可定向闭曲面 $\mathcal{M} \subset S^3$ ，存在 g 个柄圈 $\{h_i\}_{i=1}^g$ 和 g 个洞圈 $\{g_i\}_{i=1}^g$ ，分别构成 $H_1(\mathbb{O})$ 和 $H_1(\mathbb{I})$ 的基，且 $\{h_i\}_{i=1}^g$ 和 $\{t_i\}_{i=1}^g$ 构成 $H_1(\mathcal{M})$ 的一组基。

这个命题的证明可以参考 Dey et al.^[9]。

考虑本章前述计算最短同调圈的工作，因为两个线性空间的直和的一组基并不一定能够直接拆分为两个线性空间各自的基，所以我们并不可以通过直接求解 $H_1(\mathcal{M})$ 的一组基来计算 $H_1(\mathbb{O})$ 的基。

不过，我们可以将其中的技术迁移到最短柄圈的计算上。我们只要：

1. 对 \mathbb{O} 进行三角剖分，得到 \mathbb{O} 的、有许多 3-单形（四面体）的单复形表示
2. 利用本章前述的技术，计算 $H_1(\mathbb{O})$ 的一组基，并且筛选时只选择在 $H_1(\mathcal{M})$ 上的圈

就可以获得最短同调圈集 $\{h_i\}_{i=1}^g$ 。

同时，应用柴双明^[23]工作中的基于柄圈的网格切割算法，我们可以得到如下算法：

1. 按前述技术计算最短同调圈集 $\{h_i\}_{i=1}^g$
2. 构造割线集 C 为所有 h_i 之并，同时再加入任意两个相邻的边，应用该割线集进行切割

这样，我们就可以得到多了 $2g$ 个洞的圆盘，我们将多余的洞补上就可以得到拓扑圆盘。

第四章 算法实现

亏格测试数据采用 Blender 软件生成, 通过 Cube 模型的布尔修改器得到, 并导出为 obj 格式。实现代码均位于 <https://github.com/libreliu/GeoProcessing>。

计算最短同调圈的算法采用 Python 语言编写, 使用 OpenMesh 处理网格, 并采用 PyVista 进行网格可视化; 计算柄圈的代码还额外用到了 TetGen^[24] 及其封装 meshpy 库进行剖分, 同时利用 ParaView 进行四面体网格可视化。

图 4.1 展示了计算最短同调圈算法对 Genus2 模型的输出。

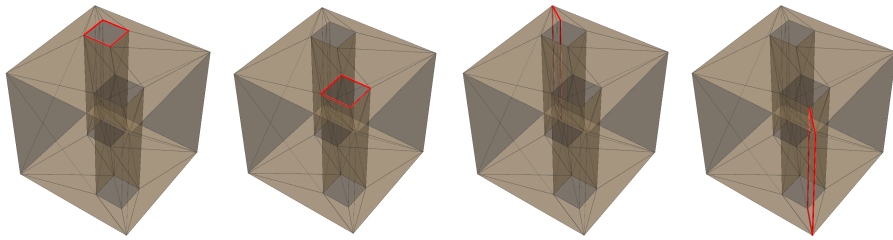


图 4.1 Genus2 模型的最短同调圈, 红色为同调基圈, 共 4 个

图 4.2 展示了计算最短同调圈算法对 Eight 模型的输出。

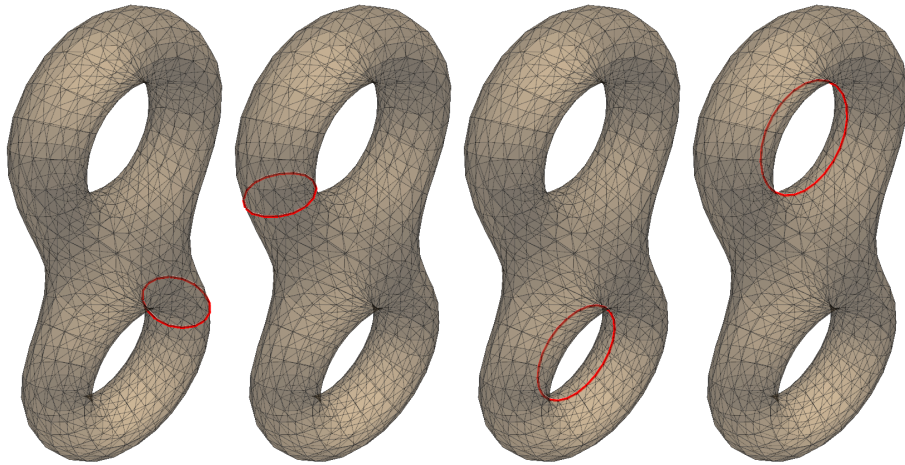


图 4.2 Eight 模型的最短同调圈, 红色为同调基圈, 共 4 个

可以看到, 最短同调圈计算的实现在这两个例子下是正确的。

图 4.3a 和图 4.3b 展示了对亏格为 1 的模型计算柄圈的部分步骤的计算结果。图 4.3a 中没有显示 \odot 中除 \mathcal{M} 之外的四面体的面。

针对部分模型进行的最短同调圈计算性能测试见表 4.1。从表格中可以看到, 最短同调圈的计算速度还不能满足较大模型的计算需求。

Genus1 进行三角剖分后的 (V, E, F) 分别为 (198, 1080, 2500)。当前版本的最短同调圈算法实现中, 解 \mathbb{Z}_2 下线性方程组是通过 numpy.linalg 中的 solve 方法在

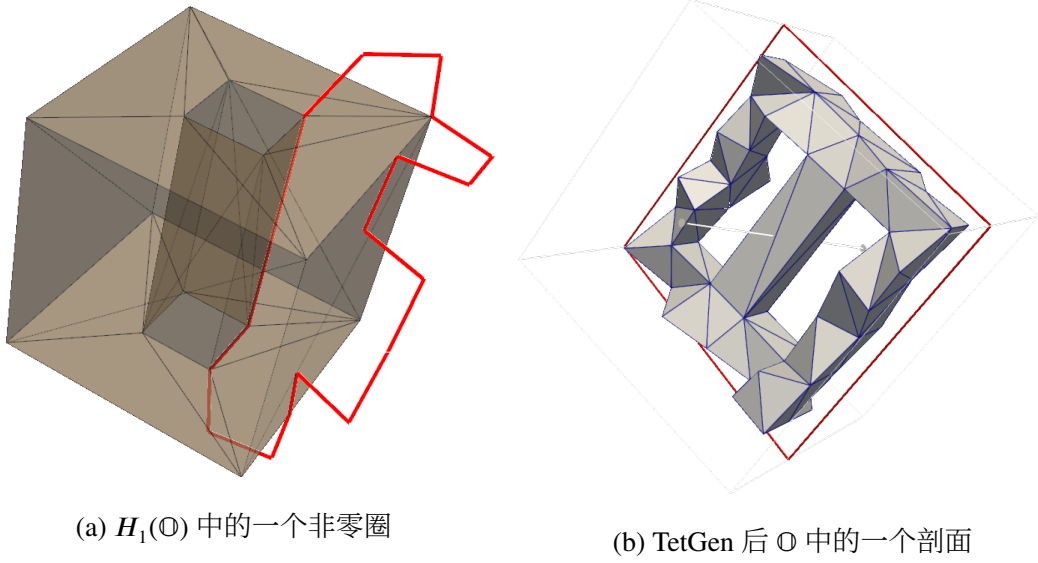


图 4.3 计算亏格为 1 模型的柄圈的部分步骤的结果

表 4.1 部分模型的性能测试

模型名称	V	E	F	亏格	候选圈数	最短同调圈计算用时
Genus1	16	48	32	1	528	0.187s
Genus2	26	84	56	2	1534	0.464s
Chair_reduced	488	1500	1000	7	494344	174.382s
Eight	766	2304	1536	2	1178874	501.112s

\mathbb{R} 上计算, 舍入到整数后再模 2 进行的。此方法受数值精度影响较大, 在该剖分后 \mathcal{O} 的计算结果有误。

纯 Python 实现的 \mathbb{Z}_2 域 Gauss 消元由于 Python 本身的开销, 计算速度较浮点的 solve 方法慢 1 到 2 个数量级, 而测试时间有限, 所以没有得出柄圈计算的性能数据和最终结果。

第五章 总结与展望

本文从网格切割问题出发,介绍了网格和网格切割问题的定义,之后分别介绍了几类问题和其部分实现。

第一类问题是给定割线集,如何生成切割后的网格。这个问题可以通过将所有三角面切割为一个个“孤岛”,再通过连接不在割线上来解决。但是,上述算法没有考虑该问题的局域性质。我们在文章中介绍了一个基于 Euler 回路的切割算法。

第二类问题是如何在优化割线长度的意义下,生成可以将网格切割为圆盘拓扑的割线集。Chai et al.^[12]的工作中发现,计算网格的柄圈可能是这种问题的好的启发。这引导着我们将 Busaryev et al.^[21]中的方法应用在计算 $H_1(\mathbb{O})$ 的一组最短基的问题中,即用 TetGen 构造内部空间的单复形表示,再构造链群,闭链群,边缘链群和同调群等线性空间的基表示,用同调坐标标记每个边,最后利用最短路径树构造 \mathcal{M} 上的圈集合,再筛选出分居不同 $H_1(\mathbb{O})$ 的等价类的、长度最短的圈。

本文的方法仍然存在着很多不足。首先,本文描述和实现的算法,只对于较小的模型适用,对于稍大一点的模型,其计算时间便增长到令人难以忍受的地步。另外,虽然柄圈的定义依赖于流形的嵌入,但是使用 TetGen 生成较细致的网格划分仍然产生了很多不必要的冗余信息,去掉这一步骤可能是可行的优化方向之一。

Dey et al.^[11]中的提到的柄圈计算算法,采取用单复形序列逐级构造的办法,只需要进行内部空间的剖分就可以进行计算;之后的文献^[10]则实现了利用 Reeb Graph 来求解近似最短的柄圈集,克服了本文提到的方法需要网格剖分环节的缺陷。本文应该进一步参考研究上述实现。

同时,很多几何处理应用都需要更精细,与具体应用联合优化,或用户可调节的切割方案。未来工作中我也会进一步探索交互式的切割方案生成机制。

参 考 文 献

- [1] Floater M S. Parametrization and smooth approximation of surface triangulations. *Computer Aided Geometric Design*, 1997, 14(3): 231-250. DOI: 10.1016/S0167-8396(96)00031-3.
- [2] Wikipedia. Texture mapping. 2021. https://en.wikipedia.org/wiki/Texture_mapping.
- [3] Erickson J, Har-Peled S. Optimally cutting a surface into a disk//Proceedings of the Annual Symposium on Computational Geometry. 2002: 244-253. DOI: 10.1145/513400.513430.
- [4] Haker S, Angenent S, Tannenbaum A, et al. Conformal surface parameterization for texture mapping. *IEEE Transactions on Visualization and Computer Graphics*, 2000, 6(2): 181-189. DOI: 10.1109/2945.856998.
- [5] Gu X, Gortler S J, Hoppe H. Geometry images. *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '02*, 2002(Figure 1): 355-361. DOI: 10.1145/566570.566589.
- [6] Liu H Y, Fu X M, Ye C, et al. Atlas refinement with bounded packing efficiency. *ACM Trans. Graph.*, 2019, 38(4). <https://doi.org/10.1145/3306346.3323001>.
- [7] Dey T K, Schipper H. A new technique to compute polygonal schema for 2-manifolds with application to null-homotopy detection. *Discrete & Computational Geometry*, 1995, 14(1): 93-110. DOI: 10.1007/BF02570697.
- [8] De Verdière É C, Lazarus F. Optimal system of loops on an orientable surface. *Discrete and Computational Geometry*, 2005, 33(3): 507-534. DOI: 10.1007/s00454-004-1150-2.
- [9] Dey T K, Li K, Sun J. On computing handle and tunnel loops//2007 International Conference on Cyberworlds (CW'07). 2007: 357-366. DOI: 10.1109/CW.2007.12.
- [10] Dey T K, Fan F, Wang Y. An efficient computation of handle and tunnel loops via Reeb graphs. *ACM Transactions on Graphics*, 2013, 32(4). DOI: 10.1145/2461912.2462017.
- [11] Dey T K, Li K, Sun J, et al. Computing geometry-aware handle and tunnel loops

- in 3D models. *ACM Transactions on Graphics*, 2008, 27(3): 1-10. DOI: 10.1145/1360612.1360644.
- [12] Chai S, Fu X M, Hu X, et al. Sphere-based cut construction for planar parameterizations. *Computers and Graphics (Pergamon)*, 2018, 74: 66-75. <https://doi.org/10.1016/j.cag.2018.05.007>.
- [13] Julius D, Kraevoy V, Sheffer A. D-charts: Quasi-developable mesh segmentation. *Computer Graphics Forum*, 2005, 24(3): 581-590. <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2005.00883.x>. DOI: <https://doi.org/10.1111/j.1467-8659.2005.00883.x>.
- [14] Khodakovsky A, Litke N, Schröder P. Globally smooth parameterizations with low distortion. *ACM Trans. Graph.*, 2003, 22(3): 350-357. <https://doi.org/10.1145/882262.882275>.
- [15] Desbrun M, Meyer M, Alliez P. Intrinsic parameterizations of surface meshes. *Computer Graphics Forum*, 2002, 21(3): 209-218. DOI: 10.1111/1467-8659.00580.
- [16] Li M, Kaufman D M, Kim V G, et al. OptCuts: Joint optimization of surface cuts and parameterization. *SIGGRAPH Asia 2018 Technical Papers, SIGGRAPH Asia 2018*, 2018, 37(6). DOI: 10.1145/3272127.3275042.
- [17] Poranne R, Tarini M, Huber S, et al. Autocuts: Simultaneous distortion and cut optimization for UV mapping. *ACM Transactions on Graphics*, 2017, 36(6). DOI: 10.1145/3130800.3130845.
- [18] Bruno V, Levy B. What you seam is what you get. 2009.
- [19] Teimury F, Roy B, Casallas J S, et al. Graphseam: Supervised graph learning framework for semantic uv mapping. 2020.
- [20] 尤承业. 基础拓扑学讲义. 北京: 北京大学出版社, 1997.
- [21] Busaryev O, Cabello S, Chen C, et al. Annotating simplices with a homology basis and its applications. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2012, 7357 LNCS: 189-200. DOI: 10.1007/978-3-642-31155-0_17.
- [22] Erickson J, Whittlesey K. Greedy optimal homotopy and homology generators. *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, 2005: 1038-1046.

- [23] 柴双明. 面向低扭曲参数化的网格切割方法研究. 中国科学技术大学, 2019.
- [24] Si H. TetGen, a delaunay-based quality tetrahedral mesh generator. *ACM Transactions on Mathematical Software*, 2015, 41(2). DOI: 10.1145/2629697.