**存档编号**

**华北水利水电大学**

## North China University of Water Resources and Electric Power

**毕 业 设 计**

**题目 基于变分网格的曲面三角网格简化**

**学 院 信息工程学院**

**专 业 计算机科学与技术**

**姓 名 马可**

**学 号 201214402**

**指导教师 郑作勇**

**完成时间 2016.5.20**

教务处制

独立完成与诚信声明

本人郑重声明：所提交的毕业设计（论文）是本人在指导教师的指导下，独立工作所取得的成果并撰写完成的，郑重确认没有剽窃、抄袭等违反学术道德、学术规范的侵权行为。文中除已经标注引用的内容外，不包含其他人或集体已经发表或撰写过的研究成果。对本文的研究做出重要贡献的个人和集体，均已在文中作了明确的说明并表示了谢意。本人完全意识到本声明的法律后果由本人承担。

毕业设计（论文）作者签名： 指导导师签名：

签字日期： 签字日期：

毕业设计（论文）版权使用授权书

本人完全了解华北水利水电大学有关保管、使用毕业设计（论文）的规定。特授权华北水利水电大学可以将毕业设计（论文）的全部或部分内容公开和编入有关数据库提供检索，并采用影印、缩印或扫描等复制手段复制、保存、汇编以供查阅和借阅。同意学校向国家有关部门或机构送交毕业设计（论文）原件或复印件和电子文档（涉密的成果在解密后应遵守此规定）。

毕业设计（论文）作者签名： 导师签名：

签字日期： 签字日期：

**目 录**

[摘要 I](#_Toc10930)

[Abstract II](#_Toc30385)

[第1章 绪论 1](#_Toc22882)

[1.1研究背景 2](#_Toc15476)

[1.2 国内外研究现状 2](#_Toc5182)

[1.2.1网格划分 4](#_Toc25236)

[1.2.2全局优化 4](#_Toc11362)

[1.2.3各向异性 4](#_Toc20147)

[1.3 解决的主要问题 5](#_Toc19970)

[1.4 本文的主要工作 5](#_Toc11524)

[1.5 论文的组织结构 5](#_Toc2591)

[第2章 变分网格逼近问题 7](#_Toc14137)

[2.1 三维模型的构成元素 7](#_Toc2271)

[2.2 变分网格逼近方法的原理与思想 7](#_Toc10314)

[第3章 算法的设计 9](#_Toc14625)

[3.1 误差度量的离散及简化 9](#_Toc29894)

[3.2 局部贪心算法 10](#_Toc1295)

[3.3 初始化 11](#_Toc8568)

[第4章 算法的具体实现 13](#_Toc24480)

[4.1 开发环境 13](#_Toc14709)

[4.2 obj文件详解及读写 13](#_Toc17340)

[4.3三角面片的面积计算 17](#_Toc8063)

[4.4顶点高斯曲率的计算 17](#_Toc5788)

[4.4.1 voronoi域面积的计算 17](#_Toc25476)

[4.4.2 顶点周围顶角和的计算 19](#_Toc12736)

[4.4.3 顶点及面片高斯曲率的计算 20](#_Toc1315)

[4.5分簇的初始化 20](#_Toc31202)

[4.5.1 面片邻域的赋值 20](#_Toc4962)

[4.5.2 分簇初值的初始化 20](#_Toc21084)

[4.6进一步优化 22](#_Toc23282)

[4.7分簇边界的提取 23](#_Toc30931)

[4.8分簇边界的简化 24](#_Toc8118)

[4.9实验结果的读取 25](#_Toc29481)

[第5章 总结 27](#_Toc876)

[参考文献 29](#_Toc2723)

[致谢 31](#_Toc18943)

[附录 33](#_Toc23633)

[英文原文 33](#_Toc31335)

[中文译文 37](#_Toc15605)

[毕业设计任务书 37](#_Toc23626)

[华北水利水电学院本科生毕业设计开题报告 43](#_Toc20734)

基于变分网格的曲面三角网格简化

摘要

在计算机图形学中，三维几何形体通常用多边形网格来表示；随着相关技术的进步，得到高精度的复杂网格数据的难度也随之大大降低，甚至拥有数以万计、乃至数以亿计的庞大网格数量的三维模型也不再稀有，不仅外形美观，且保持了较高的质量。但这些复杂的数据也对计算机的处理能力提出了较高的要求，提升了网格存储、读取、传输、绘制等操作的难度。如何权衡网格质量和处理速度之间的关系成为研究所考虑的重点之一，如何得到质量满足要求的网格，并具备较为可观的处理速度是问题的关键所在。

而针对不同的应用目的，所要求的三维模型的精度也有所不同；在很多情况下，我们所需要的往往并非高精度的三维模型，一些粗精度的模型即可以解决应用的需求。在这一背景下，如何将高精度模型简化得到一个粗精度的简化模型成为了一个新的研究方向。而在多边形网格中，三角形网格是结构最为简单、应用最为广泛的，它拥有简单的数学表示，并且也是绝大多数与图形图像有关的软硬件直接支持的最基本的网格单元，而其他多边形的网格亦可对其三角化，转化为三角形网格进行表示。因此本文依据变分网格逼近的思想，使用相关的集合与数学知识，利用局部贪心算法，提出一个对三角形网格简化的解决方案。

**关键词**：三维模型；网格简化；变分网格逼近；局部贪心算法

**中图分类号**：TP399

**A Method For Surface Simplification Of Triangle Mesh Based On Variational Shape Approxiamation**

**Abstract**

In computer graphics,3D objects are always represented by polygonal meshes. With the development of computer technology, the difficulty of getting high-quality and complex mesh data has been reduced a lot, these high-quality objects which have reasonable appearance composed of millions and even billions of meshes are not rare anymore. But the numerous data will cause computer system overloaded in storage, writing, transmission, rendering and etc. While loading these complex mesh,we often need to wait for a long time with its low speed and inefficiency.

For different application purposes, what we need are not always those high-quality meshes but some lower, which could meet our requirements. Therefore, based on the background mentioned above, model decimation techniques become a new research direction, which soon required in many fields. And in polygonal meshes, the triangle mesh has simple structure and wide application. Other polygonal meshes also can be translated to triangle meshes.This paper presents a method with the greedy algorithm to triangle mesh simplification based on the variational mesh approximation.

**Keyword:** Geometric Modeling; Mesh Simplification; Variational Mesh Approximation; Greedy Algorithm

# 

# 第1章 绪论

在涉及到几何设计与塑形的领域中，如建筑、机械、医疗、军事、商业、工业设计等，三维模型的可视化操作极其处理、渲染和传输都是许多应用所考虑的诸多重点之一。虽然在各种各样的领域中，三维模型的表示形式或其特有的存储格式都是其所在的应用系统或应用软件中所必须具备的一项，但由于三角形网格具有最为简单的数学表达、快捷的运算操作、方便的显示与简易的造型特征等特点，许多用以显示三维图像的硬件组成都将三角形网格作为其直接支持的基本显示单元，基于此，三角形网格表示仍然是当前计算机图形学中用以对三维模型表示的主要的较为主流的方法之一。随着图形学及相关领域应用技术的飞速发展，所表示的三维模型的精细程度也在渐渐提高，主要体现在大量网格模型的三角形面片数量的增加，对图形硬件交互显示、线上传输能力提出更高要求，加剧了软硬件的负荷程度。尤其在几何扫描、医学影象、工业设计、地理信息等领域中，涉及到三维重构的部分中，三角形面片的数量数以万计甚至数以千万计，此时我们既想保留模型的高精度的细节、与现实场景更为贴近的逼真表示，又想提升显示、渲染与传输等多个方面的速度，如何对这两方面进行合理有效的协调成为我们考虑的、亟待解决的一个重点问题。

为了解决以上冲突，我们对精度较高的复杂三角网格模型进行简化来满足以上两者的共同需求。简化复杂网格模型的算法及其相关技术的研究一直是计算机图形学领域的热点之一，网格模型简化的基本原理可以描述为：通过对模型进行删除或修改来构成良好的视觉效果，选取对整体影响较小以至于可以忽略的部分面片信息（包括顶点、边、面片及其他相关信息）来减少构成三维模型的基本网格数量，以提高其传输和显示等处理速率，节省传输开销，降低复杂网格对硬件的要求。随着网格简化技术的普及，多分辨率模型表示和模型简化的思想也应运而生，被运用到各种方向的研究。

近年来，国内外许多学者对模型简化算法进行了广泛、深入的研究，许多针对三维网格模型的简化算法被一一提出，这些算法可以按不同的简化思想对其进行分类。其中，基于高几何保真度和基于最少多边形数目原则是对其误差度量的不同进行的划分；观察网格的拓扑结构，按照简化前后是否保持拓扑关系不变，分为保持拓扑结构算法和非保持拓扑结构的算法；动态简化和静态简化算法是按照观察者位置与视点所建立的关系所进行的划分；采样方法、网格自适应方法、网格删减方法、顶点聚类等方法是基于简化操作的基本过程的划分。以上各个分类不仅局限于自身方法的研究，可以产生一定的交叉，目前与网格简化的相关的文献中，其综述大多围绕一定的分类方法或算法的发展历程、国内外研究背景进行介绍，通过分析与网格模型简化算法相关的技术和方法，以这些方法和技术为做延展，对多种网格模型简化算法进行综述。

随着计算机图形学的发展，为大型多边形网格提供快速而准确的简化算法是图形学中一个重要的研究方向，对其多方面都提出了极高的要求。产生与输入几何模型相对应、保持其基本几何特征的另一个相对粗糙的新的逼近模型，是网格简化的主要目的。新产生的模型可以在简化数据量的前提下满足预期的需求，而不必处理过于庞大的数据量，极大程度上提升了工作的效率。而任意的多边形网格都可以看作是由多个三角形网格组成的，所以对三角形网格简化的研究是具有一般意义的基础操作，多边形网格的简化操作亦可先转移为三角形网格再进行简化。

## 1.1研究背景

为一个曲面找到一个简洁的、但仍具备几何特征的数学表示，是许多图形学研究主题的核心。鉴于许多三维数据集的过度冗长（尤其是对网格的扫描），降低网格元素（三角形、四边形或其他多边形）的数量同时保持原始三维模型的几何保真度是几何处理的关键之处。在理想的情况下，通过拉伸曲面来适应我们期望的形状区域来使每个元素尽可能的高效，同时尽量减少几何近似的误差。这种对于几何效率的追求，自然会引发以下问题：给定三维曲面、面元的目标数量以及一个误差度量，在这样的情况下，该对象的最佳的几何逼近是什么？或者简单来说，给定失真公差，能否找到一个比该失真公差更小的最小的多边形网格逼近？尽管这是这个问题最基本的一些方面，但它的NP难度的性质已导致大多数研究者回避寻求最佳的网格表示。在本文中，我们提出一种方法，以形状近似逼近的方法解决一个离散的、变划分的问题。

## 1.2 国内外研究现状

许多研究者设计了特别的应用技术来开拓研究对象自身的平面性、对称性和功能性以用来优化其几何表示。但大多数简化方法都试图提供一种相对于各种度量的近似，只有少数是对于在一个给定的线性网格单元的约束下指定一个最小的失真误差。

当今已有的多边形网格简化技术主要分为以下几种:

网格删减:一种基于边缩减的网格简化算法，由Garland等人提出。该方法采用网格的顶点到其相关平面的二次距离作为度量,迭代缩减边，当边的数量减至目标网格边数目时，停止该操作。由Hoppe等、Klein等、Garland等基于顶点坐标值、颜色以及材质、纹理坐标作为误差度量的标准,之后与Garland相类似迭代缩减网格边对网格进行简化.通过网格删减这一方法得到的网格简化效果较为显著,但同样地，网格中边度数极高的顶点数目也非常多。由于以上操作涉及到网格的拓扑结构，较为复杂，并且花费时间较长。

网格精炼: 该方法由Eck等人提出，同时Delingette等、Lee等也提出了同样地方法。首先建立一个粗精度的网格，使用该网格逼近表示原始网格模型,然后使用某种策略对该粗网格进行迭代地细化精炼来逐渐逼近原始网格的形状.但这种方法由于最初就是粗精度的，因此误差较大。

网格重构:网格重构方法最初是使用控制目标网格顶点数目的策略，由Alliez等,Gu等提出。但是由于网格的参数化特性，这些网格重构的方法都受到限制,依然具有巨大的计算量和不稳定的数值.之后Valette等人使用局部的贪心算法构造近似的顶点的Voronoi域值,然后构造voroni图的对偶图，使用该对偶图来简化目标网格,又因为Voronoi域的对偶图是Delaunay三角剖分,所以通过网格重构简化得到的网格质量极高,并且这一方法与上述受制于参数化操作的网格重构不同，无需对网格进行参数化操作.

全局优化:Hoppe等人提出将网格简化问题视作全局优化问题.将能表征模型几何特征的几何元素作为网格能量，定义一个相关的能量函数来度量原始网格,该能量函数则通过控制网格的顶点数,顶点坐标,和拓扑连接关系来进行优化,可以得到具备高保真度的、具有与原始网格集合较为接近的几何特征与相似曲率的简化效果.此种方法同时也被推广到其他图像处理的领域中使用与图像有关的度量来设计能量函数,Lindstorm提出，能量函数不只是和模型的几何性质有关，也可以使用与图像有关的度量定义能量函数。基于上述全局优化的思想，变分网格逼近方法由Cohen-Steiner等人提出，将原始网格分为目标数量的近似平面簇集,使用与法向有关的能量来度量平面簇集,以Lloyd算法来得到优化的网格分簇,最后每一片分簇以一个多边形表示以得到最终的简化网格.该方法直观有效,简化后的网格模型能够有效保持原始网格的细节,且该算法由于不涉及网格拓扑的修改,使得算法简洁并且有着较好的稳定性.Cohen-Steiner等提出的变分网格逼近方法有比较理想的视觉效果,但Lloyd算法的迭代次数决定了它的运算速度，当迭代次数过高时运算速度较慢,而与Lloyd算法相同，均无法保证得到一个全局最优的结果,并且需要后期处理(合并法向几乎一致的平面簇集)以达到理想的效果.

### 1.2.1网格划分

使用贪心算法来尽可能多地聚集几何元素，从而创建一个由这些几何元素聚集而成的原始对象的区域，是一个进行网格简化的有效方案。网格抽取提供了一种简练的划分方法，即通过贪心算法和反复合并网格元素来完成划分。然而，尽管一些用于聚合的误差度量已被证实为接近最优（也就是无穷小的三角形），如L2度量，但贪心算法的本质经常使得这些网格并不如预期理想。此外，还有一种方法通过将特定的面片聚集在一组特征区域中，来提供一个简洁的、高阶的几何描述，与L2度量的表现相类似。即使不断地反复这个过程来改善结果，但仍没有在最小化一个相适应的几何误差上做任何尝试。

### 1.2.2全局优化

与从前的贪心算法技术形成鲜明对比，Hoppe等人提出将铸造网格简化作为最优问题。通过偏离输入网格的一个能量泛函进行度量，它们优化了顶点数目、位置及其连结，捕获了原始几何曲率的变化及特征。虽然该度量函数只是近似于计算一个点对面的欧几里德距离，但却展现了较好的简化效果。这种方法被扩展以后也被广泛用于图像处理上，不仅通过几何方法来处理网格，而且还利用其质地和法线进行优化。尽管弹性限制使得结果呈现各向异性，这样的优化技术往往造成不规则但几何特征特别优质的网格（许多面片都需要捕捉其几何特征）。而其他的方法使用某种形式的局部网格优化，这一课题仍待学习与研究，主要是因为待搜索空间的大小妨碍其效率。

### 1.2.3各向异性

网格重划分技术[Turk 1992; Lee et al. 1998;Kobbelt et al. 1999; Guskov et al. 2000]往往对于近似方法效率的关注远远少于对于网格元素质量的关注。例如，新的顶点常常被遗留在原始的流形中，从而导致了在极值处的简化呈现出较差的视觉效果。然而，当需要一个简洁、准确的几何表示时，网个元素的整体布局和宽高比是至关重要的，并且就网格尺寸与几何精度的比率来说，会致使一个显著、醒目的结果[Simpson 1994; Borouchaki and Frey 1998]。这是由多数曲面本身的各向异性所产生的。如在最近才被摆上台面的图形学工作中，一个原本被过度采样的网格可以从曲率张量的观察视场中提取出来，这是它最主要的特点之一。通过网格重构沿着特征线来对齐网格边缘来表征一个曲面的几何对称性及光照效果是一种高效的方法。尽管这一方略通过限制L2度量的优化状况提高了网格所具有的能量，但它并没有任何理论支撑以保证在粗精度情况下的效率；此外，对于不同曲率在离散网格上的近似仍值得商榷，这些方法也极易出现不理想的状况。

## 1.3 解决的主要问题

本文主要是学习与研究变分网格的逼近算法，并结合贪心算法分析、理解简化过程，通过相关图像处理工具深入算法的核心思想，以达到在可行范围内对网格模型进行优化的目的，并尝试对这一过程进行优化。

## 1.4 本文的主要工作

本文首先阐述变分网格简化的基本原理和思想，介绍变分网格逼近的基本知识，如三维模型的数学表示，误差度量函数等。最后我们详细讲述了网格简化的方法——变分网格逼近简化，详细讲述了其原理，思想以及实现过程，并对其进行编程实现，分析其性能。­­­­­­­­­­­­­

## 1.5 论文的组织结构

第一章绪论，主要对研究背景进行详细的介绍，对国内外研究现状进行描述，介绍目前网格简化的研究情况和发展方向；本文主要解决问题；本文主要工作。

第二章主要介绍变分网格的逼近问题，以及三维模型的构成元素及表示方法。

第三章变分网格逼近问题算法的原理和思想，主要介绍网格简化的概念和一般实现步骤，网格简化的基本思想和原理。

第四章详细介绍如何实现变分网格逼近简化。介绍其原理和思想，讲述实现过程并分析其性能。

第五章结论，总结性地对上文进行论述，提出其中出现的问题，并简单介绍解决问题的方法等。

第2章 变分网格逼近问题

## 2.1 三维模型的构成元素

三维模型是对象的多边形表示，通常用支持图像显示的硬件设备对其进行显示。空间中的三维模型主要由点、线、面三元素构成，此外还包括曲线、表面、外部修剪循环、内部整修循环、连接、成组、显示及渲染属性等，但最基本的仍是点、线、面三元素。多边形网格的表示中，三角网格应用最为广泛，易于推广，本文基于三角网格的表示对算法进行描述。在三角网格的表述中，点元素由三维坐标*x,y,z*组成，边元素由两个点元组成（或带有方向），面元素由三个点元组成（或带有法向）。

三角形网格*M(Mesh)*通常由{*V(Vertex),E(Edge),F(Face)*}三个元素表示:



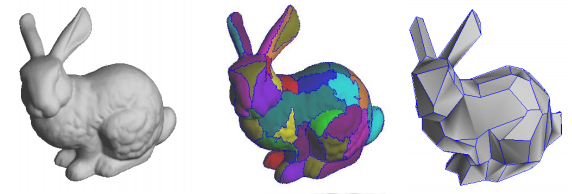
(2-1)

其中,、、,分别表示点、线、面的数量。



## 2.2 变分网格逼近方法的原理与思想

变分网格逼近理论处理将原始网格*M*的网格面片*F*划分为边连通的平面簇集,的问题。其中，.其中用来控制网格简化的比例，每个簇集中的原始网格面片相互关于边连通。当分簇完成后，每一个簇集可以用一个有簇集边缘组成的多边形来表示，进而可以经过简化得到一个逼近原始网格的多边形简化网格。

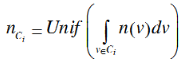


初始状态 分簇完成后的状态 逼近简化后的状态

图2-1 三种状态

通过上述过程我们可以知道，选择适应性较好的网格分簇方法是变分网格逼近的关键所在。Cohen-Steiner提出，使用与法向量相关的全局能量作为网格分簇策略中的度量标准，使得简化后的网格能够更好地逼近原始网格。

对于每一片簇集，定义簇集最接近的平面的法向量为，也称作簇集的特征法向量；ρ为点的单位法向量，指对向量的单位化操作运算。可通过下述公式计算得出：



(2-2)

分簇方法应使得每一个分簇中各个点的单位法向量与其所在的簇集的特征法向量最为接近，这样每个曲面簇集都能够较接近于平面。

现给出一个误差度量：



(2-3)

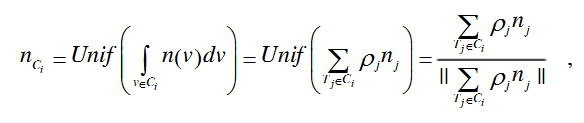
对于给定的所输入的网格*M、*要达到的目标分簇数*NC*，目标是获得尽可能最小化的误差度量。

第3章 算法的设计

对于给定网格*M、*目标分簇数、误差度量，希望得到一个最优的分簇方案使得取得最小值。在2.2中给出的公式均以连续积分的形式呈现误差度量的方案，但在实际操作中，给定的三维模型都是由有限个网格组成的，因此最终希望得到的并不是连续的，而是离散的三角形网格*M*的划分。所以，首先给出表示每个分簇的特征法向量以及误差度量的离散形式，再给出详细的算法描述。

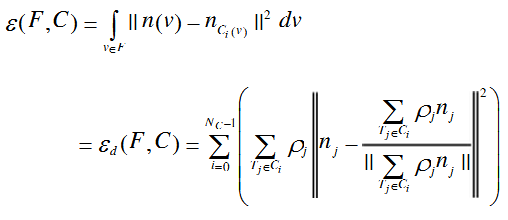
## 3.1 误差度量的离散及简化

每个构成三维模型的三角片的法向量和三角片上三点的法向量是一致的，因此可以将点法向量视作面法向量。以表示簇集所包含的三角片，表示三角片的法向量，2.1中所提及的特征法向量的离散形式可由下式给出：



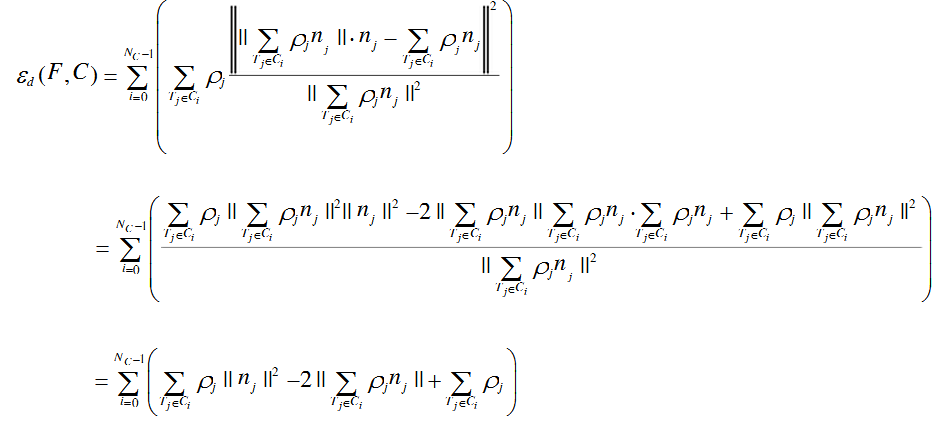
(3-1)

误差度量的离散形式可由下式给出：



(3-2)

我们所期望的算法是基于误差度量来实现的，通过迭代更新每个簇集边界以优化误差度量使之尽可能的减小。所以在簇集边界发生变化时（此时簇集所包含的三角片成员一定发生了变化，但簇集成员变化时簇集边界可能并不会随之改变），保存变化前后误差度量的值并进行比较极为必要。设簇集边界相邻的簇集所包含的面片数量为n,则上式的时间复杂度为O(n \* n).当这些簇集的特征向量变化时，误差度量也会随之改变，时间开销较大。通过分析上述公式进行简化，我们得到了下面的公式：



(3-3)

简化后的公式维度得以降低，只和当前簇集内的三角片成员有关，设成员数量为，则时间复杂度为.对于每一个簇集，只需计算每个成员的法向量以及即可，最后保存标量与向量，计算出误差度量的值。这一过程的时间复杂度为O(1).

## 3.2 局部贪心算法

基于3.1中最终所简化的误差度量公式，我们对每个簇集边界进行更新操作。设三角面片的边为T,边的集合为C,则对于每一条簇集边，设其相邻的三角片为，所属簇集分别为，考虑以下三种变更情况：

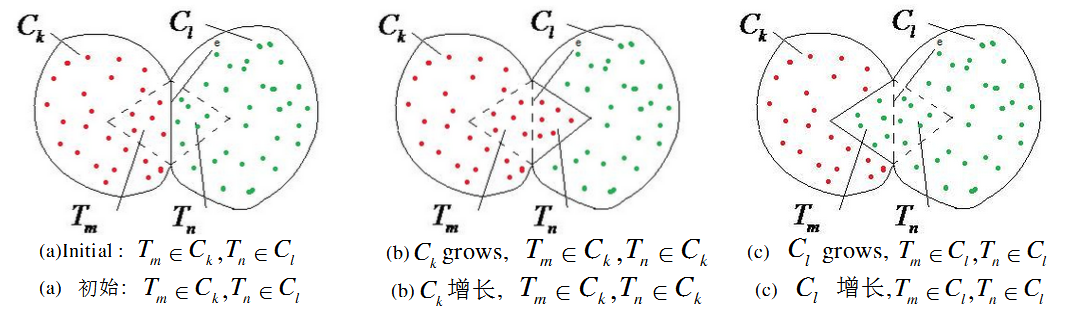
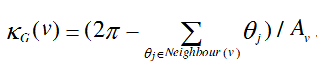


图3-1 三种变更情况

比较这三种情况所得到的误差度量的大小，并保存误差度量最小的情况作为最后的变更结果。如果出现多个最小值，则进行随机处理。对于簇集边界进行迭代更新以尽可能减小误差度量的值，直到所有簇集边界都不再改变时停止变更。由于每一步变更都会使误差度量减小，所以可以保证算法的收敛性。由于簇集边界相邻的两个三角面片通过该簇集边相连接，因此簇集的连通性亦可得以保证。

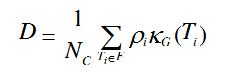
## 3.3 初始化

网格曲率越高的部分，其凸出的部分也较多，应该用较多的多边形来表示，而曲率较低的部分则相应的应该用较少的多边形去逼近表示。而法向量近似、或几乎相同的三角片应属于同一簇集。所以如果在初始划分就将这些三角片归到同一簇集中，在再划分时可以提高算法的效率。对于每个顶点，设其voronoi域面积为,邻域为Neighbour(v),顶角为v，其高斯曲率可通过如下公式计算得出：



(3-4)

三角面片的高斯曲率可近似取为三顶点曲率绝对值的平均值。网格高斯曲率的平均值可通过如下公式计算得出：



(3-5)

在理想的情况下，每一个簇集的高斯曲率总值应该低于网格总体高斯曲率的均值。在初始化时，若当前簇集的高斯曲率大于或等于网格高斯曲率均值D，则停止向该簇集中加入三角片。

在建立新的簇集时，随机选取一个尚未被归于任何簇集的三角片作为当前簇集的种子三角片，将和种子面片法向近似且与簇集边界相连的三角片归于同一簇集中。在加入簇集的过程中，设要加入的三角片为,计算的值，选择最小的值加入簇集中，直到当前簇集的高斯曲率大于或等于网格高斯曲率均值停止并入面片。循环以上建立簇集的方法直到簇集数量达到目标簇集数。此时可能所有面片都被并入簇集中，实际分簇数目可能小于或等于目标分簇数，此时可以将目标分簇数的数量变更为实际分簇数目。

如果簇集数量已达到目标分簇数，但尚有未被归于任何簇集的面片，如果其相邻三角片已加入簇集中，则加入相邻三角片所在簇集；若相邻三角片也没有加入任何簇集中，那么判断其相邻的三角片是否已加入任何簇集中，重复迭代这一过程直至所有面片都被加入簇集中。然后基于3.2中所叙述的局部贪心算法，通过比较簇集边界两侧的面片对误差度量的贡献大小来更新簇集边界，直到所有的簇集边都不再变化时停止。

# 第4章 算法的具体实现

## 4.1 开发环境

硬件开发环境：一般普通配置即可满足要求。

软件开发环境：软件开发环境需要良好地结合三维模型文件进行快速读取。本文所采用的软件配置如下：

操作系统：Windows10

编程工具及环境：Microsoft Visual Studio 2013，C++语言

软件：MeshLab，记事本

Microsoft Visual Studio 2013是一个基本完整的开发工具集，它包括了整个软件生命周期中所需要的大部分工具，所写的目标代码适用于微软支持的所有平台。本文所实现的算法本质上即是对文本数据进行的最基本操作，无需进行繁杂的编程操作，就可以完成Windows环境下对数据的操作、代码编译、测试和细化等工作。

## 4.2 obj文件详解及读写

常见的3D文件主要有obj、off、max、3ds、vtk、dwg等，本文采用的是obj模型文件。Obj模型文件实质上就是内容符合一定规则的文本文件，它包含的元素有许多，本文所用到的obj文件只包含所需要的点、面数据，其他所需要的数据则由代码实现。其中点用字母v及x、y、z坐标来表示，面用三个顶点索引表示，顶点索引从1开始，索引顺序与obj文件中点所出现的顺序一致；点的位置决定法线方向，当逆时针表示时法线向内，顺时针表示时法线向外。

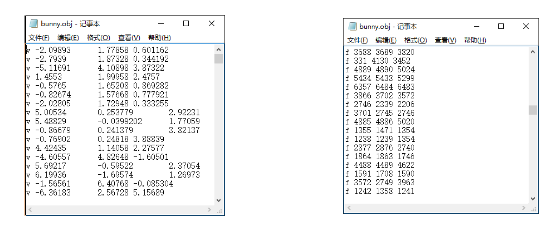


图4-1 .obj文件的数据存储方式

接下来需要利用文件的输入输出流将这些数据元引入到源程序中，使用c++的vector库函数读入mesh结构。通过vector对数据操作较为简便，其自带的函数可以计算出自身所包含数据的长度，因此使用vector可以直接求出输入网格模型所包含点的个数及面的个数。

表4-1 数据结构的建立

|  |  |
| --- | --- |
| **数据结构** | **描述** |
| Float3(class) | 点元数据，以x,y,z三维空间坐标（float型）表示 |
| Edge(class) | 边元数据，以两点坐标索引(int型）表示 |
| TriangleFace(struct) | 面元数据，以三点坐标索引(int型）表示 |
| TriangleMesh(struct) | 网格数据，包含面元数据与点元数据 |
| TriangleGroup(struct) | 面片分簇数据，包含若干面片成员 |

为方便数据的存取，建立五个数据类型：float3，edge，TriangleMesh，TriangleFace，TriangleGroup。其中float3表示点，edge表示边，TriangleFace表示面，TriangleMesh表示网格结构，TriangleGroup表示面片的划分。为方便运算，float3及edge均使用类的定义，同时对float3类型重载+、-、×、÷、=等部分基本的运算符，对edge类型重载=运算符；其他几个数据类型均使用结构体的定义。以上数据类型均使用线性表中的顺序表的数据结构进行操作。

表4-2 点元素的数据成员

|  |  |
| --- | --- |
| **数据成员** | **描述** |
| x | 横坐标，float型 |
| y | 纵坐标，float型 |
| z | 竖坐标，float型 |
| voronoi | Voronoi域面积，float型 |
| theta | 顶点对应的角度总和，float型 |
| KG | 顶点对应的高斯曲率值，float型 |
| + | +重载，两点坐标对应相加，float3型 |
| - | -重载，两点坐标对应相减，float3型 |
| \* | \*重载，float3型，又分为三种重载方式：  ①点坐标的数乘运算，三坐标乘上同一给定数字；  ②点坐标的叉乘运算，实质上为向量坐标的叉乘运算。 |
| / | /重载，float3型，三坐标除以同一给定数字，给定数字不能为0 |
| = | =重载，float3型，对三坐标进行赋值操作 |

表4-3边元素的数据成员

|  |  |
| --- | --- |
| **数据成员** | **描述** |
| v[2] | 两点索引值，int型，从1开始 |
| = | =重载，edge型，对两点索引值赋值 |

表4-4面元素的数据成员

|  |  |
| --- | --- |
| **数据成员** | **描述** |
| v[3] | 三点索引值，int型，从1开始 |
| Face\_num | 面片索引值,int型，从0开始 |
| Group\_num | 所属分簇索引值，int型，从1开始 |
| Edges[3] | 三边的点索引表示，edge型 |
| E[3] | 三边的向量表示，float3型 |
| n | 法向量，float3型 |
| s | 面积，float型 |
| KG | 面片高斯曲率值，float型 |
| Neighbor[3] | 邻域面片索引值，int型，最多为3个 |

表4-5 网格元素的数据成员

|  |  |
| --- | --- |
| **数据成员** | **描述** |
| Verts | 所有点数据，float3型，使用vector容器类 |
| Faces | 所有面数据，TriangleFace型，使用vector容器类 |
| D | 网格总体平均高斯曲率，float型 |

4-6 分簇元素的数据成员

|  |  |
| --- | --- |
| **数据成员** | **描述** |
| Size | 分簇长度，即包含面片的个数，int型 |
| Faces[] | 面片成员，TriangleFace型 |
| e | 特征法向量，float3型 |
| KG | 分簇总体的平均高斯曲率值，float型 |
| s | 分簇总面积，float型 |
| n | 法向量，float3型 |
| s | 面积，float型 |
| Ks | 面片的高斯曲率与面积的乘积之和，float型 |
| Boundaries[] | 分簇边界面片，TriangleFace型 |
| All\_edges[] | 分簇中所有所含边，edge型 |
| Edges[] | 分簇边界边，edge型 |
| Ver[] | 分簇边界点索引值，int型 |
| Ver\_v[] | 简化后分簇边界索引值，int型 |
| f\_size | 分簇边界面片数量，int型 |
| ae\_size | 分簇中所含边的数量，int型 |
| e\_size | 分簇边界边的数量，int型 |
| v\_size | 分簇边界点的数量，int型 |

在C++中，vector是一个极有用的容器，在头文件处声明即可使用该容器类型。Vector的元素不仅仅可以使用int,double,string等已有的数据类型，还可以是全局的结构体。在TriangleMesh数据结构中，将全局结构体float3、TriangleFace作为vector的数据元素以存储.obj文件中的数据，并利用vector的函数size计算出点和面的数量，操作较为简便。

在上述过程中，需要添加如下的头文件：

#include<fstream>

#include<vector>

在建立点、线、面的关系时需要注意，面的表示中所使用的点的索引值是从1开始的，而在读入数据时，点实际的索引值是从0开始的。因此在建立它们之间的联系时，在索引的处理上需要一个减一操作。在边的操作上，首先要保存其端点索引，其次要将其向量化以便于计算法向量的值，只需要将三角面片任意两边的向量表示进行数乘就可以得到该面片的法向量。

## 4.3三角面片的面积计算

对三角形的面积的计算有多种方法，如最常见的底乘高除以二等，在本文中我们使用海伦公式。设三角面片的边长分别为a,b,c,周长的1/2为p，则三角形的面积s为：



(4-1)

利用以上公式计算，并将结果保存在TriangleFace的数据成员s中。

## 4.4顶点高斯曲率的计算

### 4.4.1 voronoi域面积的计算

Voronoi图，也称为泰森多边形或Dirichlet图，是计算几何中一种基于距离的平面划分方法。若在平面上有n个不重合的种子点，把平面分为n个区域，他所在区域的种子点的距离比到其他区域的种子点的距离更近，每个区域都被称作该种子点的voronoi区域。它由一组由连接两邻点直线的垂直平分线而组成的连续多边形组成。由于给定模型中的三角面片一般都非常小，因此每个顶点的Voronoi域可视为其所在三角形的所在边与所在边的垂直平分线所围成的区域。设目标顶点为，所在三角形为，所在边的向量表示分别为***a,b***，对应的垂直平分线的向量表示分别为,，面片的法向量为***n***,则可给出下列方程：

***a· = 0***

***b· = 0***

***a·n = 0***

***b·n = 0***

(4-2)

由于三点坐标、面片的法向量已知，易求出两边及其垂直平分线的向量表；由于我们要求voronoi域的面积，就要求两边垂直平分线的交点。令其垂直平分线的两个端点分别表示为每条边中点与垂直平分线的交点，在上述四等式中任取三式联立方程即可求出交点坐标。

解三元一次方程组的过程可以转化为解二元一次方程组的过程。在解二元一次方程组的过程中，需要考虑分母为0的情况。设二元一次方程组：



(4-3)

可通过如下方法得出方程的解：

，

(4-4)

因此要排除以及为0的情况。当出现上述情况时，方程的解需要用其他方式求出：



，

(4-5)

对于三元一次方程组：



(4-6)

则可转化为解如下二元一次方程组：



(4-7)

根据二元一次方程组的解法，解得*x,y*的值。类似的，需要避开其他分母可能为0的情况。当时：



(4-8)

当时：



(4-9)

当时：



(4-10)

将已知数据代入三元一次方程组中求得*x,y,z*的值，即目标顶点所在两边的中垂线交点坐标。之后可以将voronoi图的面积转化为以目标顶点与目标顶点所在两边的垂直平分线交点的连线为界的相邻两三角形的面积之和。三角形面积可由4.3节中所描述的方法求得，亦可由底乘高的一半求得。在这里由于垂直平分线围成区域的特殊性，我们选择使用后者来求三角形的面积，在这种情况下计算更为简单。由此计算出目标顶点在当前三角面片中的voronoi域的面积。目标顶点的voronoi域的面积等于顶点所在所有三角面片的voronoi域的面积之和。

在对每个目标顶点的voronoi域面积进行赋值时，如果直接对点元数据进行操作，需要遍历寻找目标顶点所在的面，较为麻烦，因此我们在这里选择对面元数据进行操作，对面元数据中的每一个点，计算其在当前面的voronoi域面积，根据顶点的索引值定位到点元数据。之后每次均在原始voronoi域面积的基础值上继续累加。

### 4.4.2 顶点周围顶角和的计算

设目标顶点对应的角为*C*，两邻边边长*a,b*，对边边长为*c*。使用我们较为熟悉的余弦公式：



(4-11)

可以计算出目标顶点在当前面片的角度值。在这一操作中我们依然对面元数据进行操作，使用三边的向量表示进行计算。在对边向量的初始化中，是依据面元数据原始的顶点索引顺序顺次对其赋值，首尾顺次相接。这样的处理便于给出一个固定的点和边的顺序关系，便于找到目标顶点的对边与邻边。在求得cosC之后，再利用反三角函数求出C的值。与voronoi域的赋值相类似，之后每次均在顶角和的基础上累加。

### 4.4.3 顶点及面片高斯曲率的计算

已知顶点的顶角和、voronoi域值，根据第3章中给出的公式可以计算出顶点的高斯曲率值。每个面片的高斯曲率值可近似等于面片内三点高斯曲率的平均值。对于网格总体而言，其高斯曲率均值等于每个面片高斯曲率值与其面积的乘积的总和。将该值除以目标分簇数目，将所得值作为误差度量的控制标准。

## 4.5分簇的初始化

### 4.5.1 面片邻域的赋值

对于某一固定面片，由于三角形的结构，其最多存在三个相邻面片。一般情况下，不会出现游离在外的孤立三角形，所以至少存在一个相邻面片。对某个目标面片，遍历所有面片，与目标面片相比较，判断两者有没有公共边。当存在且只存在一条公共边时，将其保存为该目标面片的邻域值。

在判断两者有没有公共边时，可以转化为判断两个面片公共点的个数。当两个面片公共点的个数为2时，将该面片作为目标面片的邻域值。当两个面片公共点的个数为3时，则说明这是两个相同面片，不必特殊将这种情况排除在外。若目标面片的邻域值数量已达到3，则说明其邻域值已满，就不必要再继续遍历下去。也有少数面片的邻域值不足3，需要全部遍历完才能确定其邻域值。

设数据文件中共包含n个面片，则该过程的时间复杂度为*O(n \* n).*

### 4.5.2 分簇初值的初始化

首先遍历所有面片，选取一个尚未加入任何簇集的面片作为尚未加入任何成员的分簇的种子面片。在此之前，可以通过读取目标面片的group\_num值来判断其是否已加入某簇集中。该成员初值为0，当其为0时则说明其尚未加入任何簇集中，否则即已加入某簇集中。此时对当前分簇的size赋值为1，将这一种子面片加入簇集中，就退出遍历。计算并保存这一时刻分簇的特征向量的模长。

新建一个int型变量seed\_num用来记录种子面片在该簇集中的坐标索引，初值赋为0.根据第三章中所提到的方法，循环并入面片；当分簇中成员数量已达上限，或该分簇超出误差度量所允许的范围停止继续并入新的面片。在并入面片的过程中，首先可以根据4.5.1中所描述的方法判断新的面片与种子面片是否是边连通的，时间复杂度为O(n)；而我们已经知道，一个面片的邻域与该面片一定是边连通的，而这一过程的时间复杂度为O(1)，选择这种方法更优。然后新建一个度量变量delta，用以存储该面片法向量与分簇特征向量差值的模长。比较种子面片的所有邻域的delta值，选择delta值最小的面片加入簇集中。此时，将簇集中新加入的面片赋值给种子面片。由于加入操作均在尾部执行，因此新加入的面片即簇集中最后一个面片。该循环的控制条件是判断种子面片的所有邻域是否已加入簇集，若是则退出循环，否则继续循环。

根据第三章的描述，我们在此处定义一个函数metric(TriangleGroup &group)用来保存每个分簇的总面积和分簇内每个成员的法向量与面积的成绩的和的差值的二倍作为分簇的一个度量，用以和误差度量进行比较，当其超出误差度量所允许的范围时不再加入新的面片。当簇集所含成员数量已达数组可容纳上限时，退出循环。

需要注意的是，当种子面片的邻域均已并入簇集中时，已经满足了跳出循环的条件，但循环并不因之停止。此时我们将种子面片的索引seed\_num的值从簇集尾部前移，如当前seed\_num的值为group.size-1，先前移至group.size-2，判断group.faces[seed\_num]的邻域是否均已并入面片中，如果存在尚未加入任何簇集的面片，则执行上述添加操作；否则再将seed\_num移至group.size-3，依次类推。如果直至seed\_num=0时其邻域也都已加入簇集，跳出循环。

在该分簇已经不再并入新的面片时，仍可以将与分簇面片基本在一个平面上的分簇值加入簇集中。当目标面片的法向量与分簇特征法向量的差值的模长为一个极小值且分簇成员数量尚未达到上限时，可以继续加入这样的面片。

在所有的簇集都初始化完毕后，可能还有尚未归属于任何簇集的面片，则将其加入邻域所在的簇集中。若邻域尚未归属于任何簇集，则对其邻域进行上述操作。循环这个过程，直至所有面片都有所归属的簇集。此时簇集初始化的工作才真正地完毕。

## 4.6进一步优化

在做完上述分簇初始化的工作后，我们可以基于第三章所描述的方法对分簇进行进一步的优化。在初始化之后，可能存在多个可以近似为一个平面的分簇，但在划分时被划分成了多个，或有一些分簇的效果尚不理想。进一步优化的目的是为了将每个簇集的误差度量尽可能调整至最小，以使得每个簇集的状态调整至最佳，将近似于一个平面的面片并入到同一簇集中。

根据第三章中算法的描述，我们需要对簇集边界两侧的面片进行比较。首先，我们需要保存那些簇集边界上的面片。这里有几种方法来获取簇集边界上的面片：

对某一指定分簇，对所有分簇进行遍历，使用4.5.1中所描述的方法判断两个分簇之间有公共边的面片，并把它们保存为簇集边界上的面片。对于给定分组与当前分组，设其长度分别为n1和n2，则每一次比较的时间复杂度为O(n1 \* n2)；设分簇的平均长度为n，则全部比较完的时间复杂度为O(NC \* n \* n).其中NC为现有的分簇数目。同时要注意的是，有些分簇边界并不和其他分簇有交集，而这些分簇边界面片可能仅有一个或两个邻域。因此还要找出这些分簇边界的另一边为空的分簇边界面片，时间复杂度为O（n）。这种方法较为麻烦，处理上较为复杂，当分簇成员数量较多时时间上就更加不占优势。

由上一种方法延伸开来，我们只需要遍历每个簇集的成员的邻域，当其group\_num和当前成员面片不相等时，将其保存为边界面片；当存在邻域值为-1（即邻域未经初始化）的成员面片时，将其保存为边界面片。设当前簇集所含成员数量为n，则这一过程的时间复杂度为O（n）。相较之下，这种方法显然更加简单易行，效率更高。

因此对于每个分簇，遍历其成员面片，并保存那些不是所有邻域都在当前分簇内或邻域数量少于三个的成员作为边界面片。需要注意的是，邻域数量少于三个的成员也有可能出现不是所有邻域都在当前分组内的情况，因此不要重复并入该成员。

因此我们只需要比较这些相邻的边界成员即可。此时我们需要考虑第三章中所提到的三种情况。对于每个簇集，遍历其边界成员面片，并判断其邻域是否与其属于不同分簇；若属于不同分簇，再判断是否属于边界成员。若属于边界成员，则先将其加入当前面片所在的簇集中，并把这一面片从面片原始的簇集中删去；再计算当前分簇误差度量的总值，并与原始值进行比较。若当前分簇误差度量的总值小于或等于原始值，则维持当前状态不变；若当前分簇误差度量的总值大于原始值，则从当前面片中删去刚才加入的面片，并将该面片恢复到原始位置（即添加至原始簇集中）。由于将新面片添加至簇集的操作均是从尾部进行，所以将其删除时亦可直接对尾部进行操作，将尾部数据赋空，再对分簇所含成员数目执行减一操作。

从理论上分析，我们需要比较前后误差度量总值的变化，也就是每一个簇集误差度量的前后变化；但在每一步操作中，我们仅仅对两个分簇执行了操作，前后产生变化的也仅有这两个分簇，因此我们只需要比较这两个簇集的误差度量之和的前后变化即可。

执行这一循环，直至每个分簇内的成员都不再发生变化停止循环。由于我们仅仅在尾部进行添加删除操作，所以在每一步操作后，我们只需要判断操作后的分簇的最后一个元素和操作前的分簇的最后一个元素是否完全一致即可。如果完全一致，说明成员不再发生变化，退出循环；如果不完全一致，说明成员还在变化，继续循环。

## 4.7分簇边界的提取

前面已经提到，在.obj格式的文件中，一个面是由点所构成的；而点的位置顺序决定了这个面的法线方向。相近的法向量往往有着相似的灯光效果。对于一般的封闭三维几何体来说，法线向外一般表现为正面，法线向内一般表现为反面。构成一个多边形的点是固定的，而多边形的表示并不唯一；我们需要注意到，给出这些固定的点的坐标，可能能够生成多个多边形，生成多边形的形状也并不唯一。所以必须要遵循固定的逆时针或顺时针顺序，才能绘制成固定的多边形。因此，我们在提取分簇边界边的过程中，也应该有序对其进行排列。随机选取簇集内某条边作为起始边，遵循下一条边的起点应作为上一条边的终点的原则进行排列。

在获取簇集边的过程中，我们可以对簇集边界面片进行遍历，对目标簇集边界面片，判断它的三条边是否在这一簇集内；判断方法为找到其处于同一簇集内的邻域，邻域与目标面片的公共边即在这一簇集内的边。对在簇集内的边进行标记，未被标记的边即为簇集边界上的边。这种方法建立在簇集边界面所存在的基础上，看似较为简便，当簇集边界面片数量为n时，时间复杂度为O（n），时间复杂度并不高，但代码的书写较为繁琐。

我们也可以对簇集中所有边进行整理，调整边的数据结构，为其增添两个表示相邻面片的数据成员，而我们所要选取的边就是仅含有一个在簇集内的相邻面片的边。这一过程的时间复杂度为O（n）。但需要注意的是，对每条边的邻域面片进行赋值又是一个繁琐的过程，设所有面片的数目为s，则所有边的数目为3s；其时间复杂度为O（s \* s）。结合这一点考虑，时间复杂度不占优势，因为对边的相邻面片赋值的操作仅是对这个过程起作用，代价较大，且代码书写一样较为繁琐。

最后我们采用的做法是计算分簇中每条边出现的次数，找到那些仅在簇集中出现一次的边作为该簇集的边界边。设簇集边的总数目为n，则这个过程的时间复杂度为O（n \* n）。虽然时间复杂度上要逊于第一种方法，但代码书写十分简洁易懂，可读性好，易于操作。

之后，对分簇中每个成员进行排序，做到每个边都以首尾相接的方式进行保存，并顺次保存边界中出现的点集。在排序时，我们选定当前每个簇集中的第一条簇集边作为初始簇集边，使用for循环遍历簇集边界，接下来使用while循环遍历，判断下一条边的起点（这里视作边成员的第一个顶点索引）与上一条边的终点（这里视作边成员的第二个顶点索引）是否一致；如果一致不再调整，执行下一部循环，如果不一致，则需要再判断下一条边的终点和上一条边的终点是否一致，如果一致，将这一条边的起点和终点位置互换；如果不一致，则继续往下遍历，直到找到满足条件的、一致的边为止，交换这条与之一致的边，将之与当前操作的边对象的下一条边进行交换，直到所有的边都满足上一条边的终点与下一条边的起点相一致。最终排序后的结果除了满足上述条件外，第一条边的起点与最后一条边的终点也应该相一致，这样所有的边才能围成一个闭合的图形区域。

但是要注意到，有一些簇集不止有一个簇集边界，这并不是说该簇集不能闭合或非边连通的区域，而是呈现一个类似环状的图形结构。这种结构大多数是有一个较大的簇集边界以及一个或多个较小的簇集边界。在这种情况下，我们可能遍历到簇集边界的末尾也不能满足上一条边的终点与下一条边的起点相一致的条件。但此时我们可以发现，当前边的终点和第一条边的起点相一致，即已经形成了一个闭合的区域，所以接下来应该还有其他闭合的区域。当遍历到簇集边界末尾也无法满足上述判定条件时，退出while循环；然后继续进行for循环的遍历。在当前簇集的边界排序结束后，顺次不重复地摘取边界上的点。设每个簇集边界的数目为n，这个过程中最好情况的时间复杂度为O（n），最坏的情况为O（n \* n），与在交换过程中所要交换的边在边界集合上的位置有关。

我们还需要注意到，有少数空间多边形无法由点集顺次表示，当组成它的点并不总是顺时针或逆时针排列时，.obj文件无法对其进行绘制。此时我们还要判断到哪个点为止一直是顺时针表示或到哪个点为止一直是逆时针表示，对这个点进行标记。将这个分簇分成两个或多个部分进行表示。

## 4.8分簇边界的简化

遍历每个分簇的边界点集，与其他分簇边界的点集进行比较，若有三个或三个以上的簇集中都出现过这个点，将这个点作为简化后的边界点集保存。通常这些点的数目不会太多，也尽可能使用少量多边形对其进行简化表示。设某个簇集边界的点数目为n，该算法的时间复杂度为O（n \* n）.如4.7中所提到的一些组成多边形的点并不能够简单地顺次表示，在这一阶段我们也仍要对其进行判断。但简化得到的点通常数目较少，基本上都可以较顺利地表示出来。最后我们将这些点输出，得到我们简化后的.obj文件。

## 4.9实验结果的读取

在保存新生成的.obj文件时，将简化后的点顺次输出作为新的面片。实质上，我们此前的操作都是对数据文件的操作，.obj文件也可以看做具有一定格式要求的文本文档，我们需要通过图形界面的窗口来检验我们所做的结果。本文中我们选用了MeshLab软件来实现对.obj的读取操作。

MeshLab是一个开源的、可移植得和可扩展的三维几何处理系统，主要用于交互处理和非结构化编辑三维三角形网格。与传统的3dsmax、maya等三维图像处理软件相比，MeshLab更加轻便且操作简单，对于网格的处理有不错的优势。我们以经典的bunny三维模型为例，最初的原始状态如下：

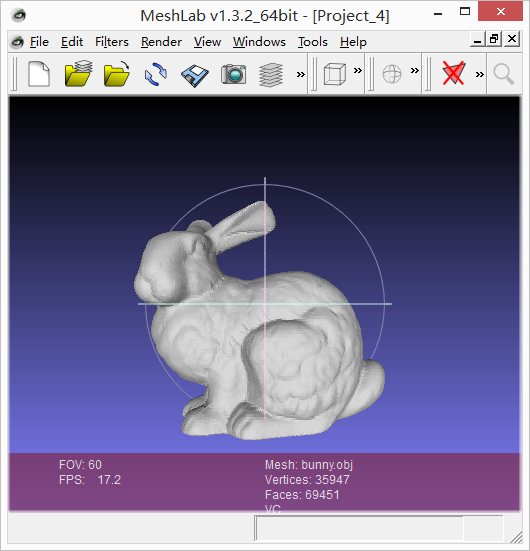


图4-2 原始的三维数据模型bunny

提取簇集边界面片后的结果如下：

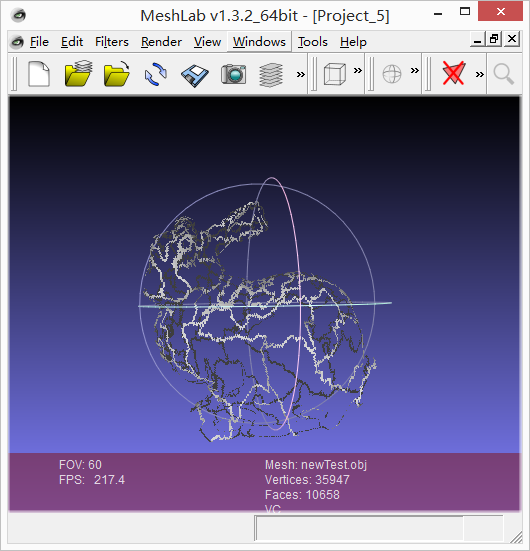


图4-3 提取分簇边界面片的三维数据模型表示

最终简化后的结果：

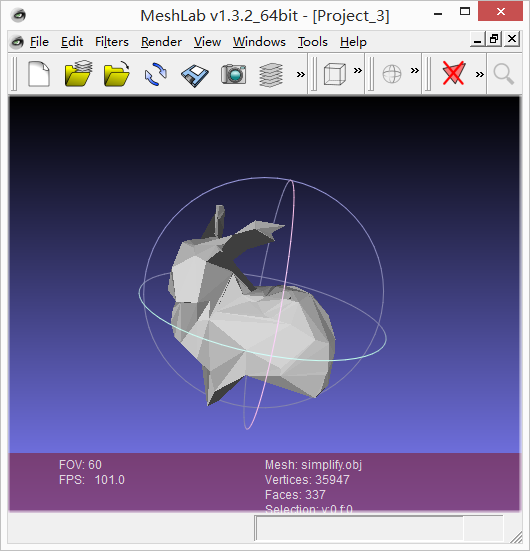


图4-4 经过最终简化的三维数据模型

# 第5章 总结

随着计算机图形学的发展，为大型多边形网格提供快速而准确的简化算法逐渐成为了图形学中一个重要的研究方向，这项技术对其包括网格处理、着色、绘制等诸多方面都提出了极高的要求，而网格简化的技术也不断被更新。本课题是基于变分逼近的思想提出的一种三角网格简化算法，以目标分簇总数量作为控制模型简化比例的参数，通过离散基于法向量的分簇能量，提出自适应的分簇初值选取和局部贪心优化算法以得到理想的结果分簇，在视觉效果上能够得到和原始三维模型特征相近的粗精度的三维模型。本文最终所呈现的结果是以90个面片对bunny模型进行逼近的结果。

本文所使用的算法的分簇结果取决于分簇初值三角形的选取，通过自适应选取的算法，分簇结果有良好的适应性。在实验过程中，效率是一个很大的问题，在Debug模式下需要两小时以上的运行时间，而在Release模式下进行运行，仍需要半小时以上的运行时间，计算量大，效率过低。简化得到的模型仅能模糊地表征原始模型的几何特征，视觉效果较一般，在简化效果上还有待加强。在图形界面方面，需要借助于MeshLab等软件来帮助检查结果的正误性，接下来应该尝试搭建如使用OpenGL搭建的可视化界面，这样更加便于代码每一步的编写与程序的调试。

本文采用法向量作为主要的分簇能量，今后可尝试使用曲率、欧式距离等组为分簇度量进行分簇的划分。

# 

# 参考文献

[1]Garland M, Heckbert PS. Surface Simplification Using Quadric Error Metrics. [J]In: Whitted T, ed. Proceedings of ACM SIGGRAPH. Los Angeles: ACM Press, 1997. 209-216.

[2]Hoppe H.Progressive Meshes.[J] In: Rushmeier H, ed. Proceedings of ACM

SIGGRAPH. New Orleans: Addison-Wesley Professional, 1996. 99-108

[3]Klein R, Liebich G, Straßer W. Mesh Reduction with Error Control. [D]In: Yagel R, Nielson GM, eds. Proceedings of IEEEVisualization. San Francisco: IEEE Computer Society Press, 1996. 311-318.

[4]Garland M, Heckbert PS. Simplifying Surfaces with Color and Texture using

Quadric Error Metrics.[D] In: Ebert DS, Rushmeier H, Hagen H, eds. Proceedings

of IEEE Visulaization. Washington: IEEE Computer Society Press, 1998.

263-269.

[5]ECook R, eds. Proceedings of ACM SIGGRAPH.[J] Los Angeles: ACM Press, 1995.

173-182.

[6]Delingette H, Herbert M, Ikeuchi K. Shape representation and image segmentation using deformable surfaces. Image and Vision Computing, 1992,10(3):132-144.

[7]Lee AWF, Sweldens W, Schröder P, Cowsar L, Dobkin D. Maps: Multiresolution adaptive parameterization of surfaces. [J]In: Machover C, ed. Proceedings of ACM SIGGRAPH. Orlando: ACM Press, 1998. 95-104.

[8]Alliez P, Meyer M, Desbrun M. Interactive Geometry Remeshing. [J]In: Appolloni T, ed. Proceedings of ACM SIGGRAPH. San Antonio: ACM Press, 2002. 355-361.

[9]Alliez P, Cohen-Steiner D, Devillers O, Levy B, Desbrun M. Anisotropic Polygonal Remeshing. [J]In: RockWood AP, ed. Proceedings of ACM SIGGRAPH. San Diego: ACM Press, 2003. 485-493.

[10]Gu X., Gortler S, Hoppe H. Geometry Images.[J] In: Appolloni T, ed. Proceedings of ACM SIGGRAPH. San Antonio: ACM Press, 2002. 363-374.

[11]Valette S, Chassery J-M. Approximated Centroidal Voronoi Diagrams for Uniform Polygonal Mesh Coarsening. [D]Computer Graphics Forum(Proc. Eurographics), 2004,23(3):381-389.

[12]Valette S, Kompatsiaris I, Chassery J-M. Adaptive Polygonal Mesh Simplification With Discrete Centroidal Voronoi Diagrams. [J]In: Lazzari G, Pianesi F, Crowley JL, Mase Kenji, Oviatt SL, eds. Proceedings of ICMI. Trento: ACM Press, 2005. 6 55-662

[13]Hoppe H, DeRose T, Duchamp T, McDonald J, Stuetzle W. Mesh Optimization. [J]In: James TK, ed. Proceedings of ACM SIGGRAPH. Anaheim: ACM Press, 1993. 19-26.

[14]Lindstorm P, Turk G. Image-driven simplification.[J] ACM Transactions on Graphics, 2000,19(3):204-241.

[15]Cohen-Steiner D, Alliez P, Desbru M. Variational Shape Approximation. [J]In: Marks J, ed. Proceedings of ACM SIGGRAPH. LosAngeles: ACM Press, 2004. 905-914.

# 致 谢

在本文即将完成之际，回想本科毕业设计完成的过程，我要向我的指导老师表示衷心的感谢。正是在老师的指导与督促下我才能顺利的完成这篇论文。老师待人诚恳亲切、关心学生，对本人课程学习与研究、论文的选题和撰写均给与了关注和指导，在此致以衷心的感谢!老师对待学术研究的态度和人生道理给我们带来了深刻的影响，这将让我们终身受益。

在这四年的学期中结识的各位生活和学习上的挚友让我得到了人生最大的一笔财富。在此，也对他们表示衷心感谢。

本文参考了大量的文献资料，在此，向各学术界的前辈们致敬！

# 附录

## 英文原文

Web日志文件

用户识别

会话识别

数据清理

路径补充

**Fast** **and** **Robust** **Detection** **of** **Crest** **Lines** **on** **Meshes**

**Abstract**

We propose a fast and robust method for detecting crest lines on surfaces approximated by dense triangle meshes. The crest lines, salient surface features defined via first- and second-order curvature derivatives, are widely used for shape matching and interrogation purposes. Their practical extraction is difficult because it requires good estimation ofhigh-order surface derivatives. Our approach to the crest line detection is based on estimating the curvature tensor and curvature derivatives via local polynomial fitting.

Since the crest lines are not defined in the surface regions where the surface focal set(caustic) degenerates, we introduce a new thresholding scheme which exploits interesting relationships be- tween curvature extrema, the so-called MVS functional of Moreton and Sequin, and Dupin cyclides,

An application ofthe crest lines to adaptive mesh simplification is also considered.

**Introduction**

Developing methods for fast and accurate detection of feature lines on polygonal and point-sampled surfaces is currently a subject of intensive research[Khaneja et al. 1998; Gumhold et al. 2001; Hubeli and Gross2001; Lee and Lee2002; Pauly et al. 2003; Page et al. 2002; Ohtake et al. 2004; Stylianou and Farin2004; Cazals and Pouget2004a]. In this paper, we propose a fast and robust method for detecting surface creases on surfaces approximated by dense triangle meshes.

Surface creases, curves on a surface along which the surface bends sharply can be intuitively defined as loci of sharp variation points of the surface normal. Mathematically the sharp variation points of the surface normals are described via extrema of the sur- face principal curvatures along their corresponding lines of curva- ture. These curvature extrema, called also ridges, have been thor- oughly studied in connection with research on classical differen- tial geometry and singularity theory[Koenderink1990; Porteous 1994; Belyaev et al. 1997; Hallinan et al. 1999; Cazals and Pouget 2004b]. The ridges and their subsets have numerous applications in image and data analysis[Monga et al. 1992] quality control of free-form surfaces[Hosaka1992], human perception[Hoffman and Richards1985], analysis and registration of anatomical struc- tures[Pennec et al. 2000], geomorphology[Little and Shi2001] and non-photorealistic rendering[Interrante et al. 1995; DeCarlo et al. 2003]. See also references therein. The so-called crest lines are formed by the perceptually salient ridge points and consist of the surface points where the magnitude of the largest(in absolute value) principal curvature attains a maximum along its correspond- ing line of curvature.

Practical detection of the crest lines and other types of curvature extrema is a difficult computational task because it requires a high- quality estimation ofthe curvature tensor and curvature derivatives. In general, global fitting methods do a betterjob in estimating high- order surface derivatives and lead to more accurate detection of curvature extrema[Kent et al. 1996; Ohtake et al. 2004] than the local estimation schemes. On the other hand, the local schemes are much faster and often demonstrate a quite satisfactory per- formance[Gue´ziec1993; Stylianou and Farin2004; Cazals and Pouget2004a].

Our procedure for detecting the crest lines combines local polynomial fitting based on a modification of the method of [Goldfeather and Interrante2004], a finite difference scheme/test proposed in[Ohtake et al. 2004] and used for curvature max- ima/minima identification, and a careful thresholding based on the MVS functional of Moreton and Sequin[Moreton and Sequin 1992]. Our method is fast since we estimate necessary surface derivatives via local polynomial fitting. For example, for the Igea model consisting more than200*K* triangles it takes only nine sec- onds for estimating the curvature tensor and curvature derivatives and four seconds for detecting crest lines on a standard1.7GHz Pentium4 PC. Our approach is capable of achieving high quality results comparable with those obtained via global fitting procedures [Ohtake et al. 2004]. Fig. 1 shows crest line patterns found on sim- ple and complex geometrical models for various values of a user- specified parameter which controls the strength of detected crest lines.

We also consider applications of the crest lines to adaptive mesh simplification.

**Estimating** **Surface** **Derivatives**

Given a mesh M approximating a smooth surface S, in order to achieve a fast and accurate estimation of the principal curvatures and their derivatives a bivariate polynomial is fitted locally to each mesh vertex. To date, two polynomial fitting strategies are used for estimating surface derivatives at a mesh vertex. According to one strategy, it is assumed that the surface normal at vertex is pre- liminary estimated. It leads to the so-called adjacent-normal cubic approximation method [Goldfeather and Interrante 2004]. The sec- ond strategy [Cazals and Pouget 2003] does not assume that the mesh normal is already given. According to our numerical expe- rience, if the vertex normal is approximated appropriately, the first strategy leads to a better estimation of the surface curvatures and their derivatives at the vertex.

**Crest** **Lines** **and** **Mesh** **Simplification**

In this section, we develop a quadric-based mesh simplification pro- cedure guided by the distance field from crest lines. Our use of crest lines for adaptive mesh simplification purposes is inspired by re- cent work [Kho and Garland 2003]. Since crest lines on a mesh are important shape features, it is natural to simplify the mesh aggres- sively far from the most salient crest lines and preserve the mesh in a vicinity of them.

Given a set offeature lines (crest lines, in our case) on surface S, following [Le´vy et al. 2002] for a surface point p ∈ S we consider d(p) the geodesic distance between p and the closest feature line (crest line) point. Let max(d) be the maximum of the geodesic dis- tances d(p) over all points of S.

**Discussion**

We have presented a fast method for detecting salient curvature extrema on surfaces approximated by dense triangle meshes.

Our method is capable of achieving high quality results in de- tecting salient curvature extrema to compare with schemes based on global fitting procedures. In Fig.9 we give a visual compari- son of our method with that developed in [Ohtake et al. 2004] (the two left images) and with the exact detection of the crest lines on analytical waving surface **r**(*u*,*v*) = [*u*cos*v*,*u*sin*v*,cos*u*] (two right images).1

Our filtering scheme for removing unessential crest lines is based on interesting relationships between Dupin cyclides, focal sets, cur- vature extrema, and variational functionals. We use cyclideness (6) as the main ingredient of our filtering scheme and measure the strength of crest lines by scale-independent quantity (5). Thus long but weak crest lines are preferred to strong but short ones. Of course, different filtering procedures can be also used instead ofthat based on (5). Similar manual thresholding schemes were also used in [Ohtake et al. 2004; Cazals and Pouget 2004a]. Manual filtering is hardly avoidable for complex geometry surfaces, since the crest lines are local surface features while saliency-based thresholding should take into account global surface shape.

Finally we have demonstrated how crest lines can be used for adaptive mesh simplification preserving visually important shape features.

## 中文译文

Web日志文件

用户识别

会话识别

数据清理

路径补充

快速而健壮的网格脊线探测技术

**摘要**

在这一研究中，我们提出了一种快速的、精确度较高方法，通过密集的三角网格的逼近表示来提取曲面的特征线。曲面的特征线是由一阶和二阶的曲率导数所定义的、用以表示曲面显著特征的几何参数，被广泛运用于形状的匹配及目的问询等多个方面。在实际操作中，对它们的提取有一定的难度，因为它需要对其高阶曲率导数有一个良好的估计。我们用来检测特征线的方法是由模型本体的多项式的拟合所决定的建立在对曲率张量以及曲率导数的良好估计上所进行的操作。

由于特征线不是由沿焦散面衰退的表面区域所定义的，我们引入一个新的阈值方案，利用曲率极值之间的微妙关系，也就是所谓的由Moreton,Sequin提出的MVS函数以及Dupin圆纹曲面。

对于自适应网格简化，特征线的应用也被考虑在其中。

**介绍**

如何去获取快速的、准确的对多边形和点采样的表面特征线方法，是目前正在发展的深入研究的主题之一。在本文中，我们提出了用密集的三角形网格近似的表面来探测表面褶皱的亦可快速可靠的方法。表面褶皱痕迹沿着这一表面的弯曲急剧变化，可以较为直观地定义为表面法线的处于尖锐变化的位点所在表面上的曲线。而曲面法线的突变位点则是由曲面弯曲程度对应的表面主曲率的极值所决定的。这些曲率极值，也称作脊点，都与经典查分几何和奇点理论有着密切的联系，在这些理论上有相关的较为深入的研究。脊点和它们在图像和数据分析、自由曲面分析、人类的感知、解剖分析、质量控制、地形地貌和非真实感渲染等众多应用上都有着卓越的贡献。其他的应用另请参阅其中的参考文献。所谓的波峰线，也就是由凸出的脊点形成的、并取其中幅度最高的绝对值点的主曲率，达到沿其对应的线曲率最大的曲面上的点所构成的线。而实际情况下，对于棱线和其他类型的曲率极值的检测是一项艰巨的计算任务，需要对曲率张量以及高阶曲率导数有良好的估计。一般情况下，全局拟合的方法来估计高阶曲率导数所展现出的效果比局部拟合更佳优秀，对于曲率极值有更准确的检测。但另一方面，局部拟合的方法更快，常常表现出相当令人满意的性能。

我们在对Goldfeather和Interrante的方法以及Ohtake所提出的有限差分测试法进行修正的基础上，结合局部多项式的自适应策略来实现对棱线的检测，并用于曲率极大极小值的检测，基于Moreton和Sequin的MVS函数给出一个精确的阈值。由于我们估计通过局部多项式拟合的曲面导数，因此我们的方法足够迅速。例如，在一台标准的1.7GHz的奔腾四电脑商，对于包含了超过二十万个三角形网格的Igea模型，它仅仅需要九分钟来拟合曲率张量和曲率导数；检测棱线的整个过程则仅需要四分钟。我们的方法是通过全局拟合策略来实现的高质量的结果，其结果可与Ohtake的方法得到的效果相媲美。

我们也考虑到波峰线自适应网格简化方面的应用。

**曲面导数估计**

给定一个近似于曲面S的网格M，我们的目标是在一个二元多项式被局部适应每一个网格顶点之后，获得一个对顶点的主曲率和它们的导数的快速、准确的估计。迄今为止，二元多项式拟合策略被广泛运用于网格顶点所在曲面导数的估计。根据某一方法，我们假设该曲面在顶点法线作为初步的一个已知的估量值。由这一方略，我们得到了所谓的相邻的普通立方体近似值的方法。第二个策略没有假定网格法向量是已知的。根据我们以往的数值经验，如果顶点法向量的近似效果的适应性良好，则第一种方法对表面曲率和导数有更好的估计。

**波峰线与网格简化**

在这一部分中，我们开发了一个基于二阶的从波峰线的距离场所确定的网格简化程序。我们所研究的脊线自适应网格简化的方法，其根本是从Kho和Garland最近的工作中汲取的灵感。由于一个网格上的波峰线是重要的形状特征，很自然地从最突出的凸纹线进行简化，并在它们的附近保持网格的特征。

给出一个表面S的一组特征线（在我们的情况中，也就是脊线），根据Levy等人用于表面S上的点P，我们考虑到d(p)，即p和最近特征线（棱线）点之间的测地距离。使max(d)作为S上的所有点的d(p)的最大测量距离的最大值。度量wjQ（TJ）分配给每个三角形网格M的TJ，其中Q（Tj）为标准值。我们设置WJ= 1/ F（DJ），我们设置WJ= 1/ F（DJ）和控制的波峰线影响通过参数η的程度。

**总结**

我们已经提出了一个使用密集的三角网格近似曲面来检测曲面特征线的较为快速的方法。

我们的方法是通过检测突出的曲率极值来与基于全局拟合的方案比较，以实现高质量的效果。在图9中，我们比较了我们的方法与Ohtake（左侧两个图象）的方法、表面波纹分析的脊线特征的精确检测（右侧两个图像）的视觉效果。

我们的波峰线滤波方案是基于Dupin四次圆纹曲面、焦点集合、曲率极值和变函之间的微妙关系实现的。我们使用四次圆纹曲面作为我们滤波方案的主要成分，以及测量比例无关量波峰线的强度。因此长的、但密度较低的脊线比短的、但密度高的效果更佳。当然，不同的过滤程序也都可以使用，不局限于以上所述的方法。类似的手动阈值方案也被用在Ohtake、Cazals、Pouget的方法中。手动阈值过滤难以避免构造复杂的曲面，因为波峰线的局部的表面特征，而基于显著性阈值的处理则应考虑到全局的曲面形状。

最后，我们已经证实脊线是如何用于自适应网格简化来保持视觉上的重要形态特征的。

毕业设计任务书

**设计题目：基于变分网格的曲面三角网格简化**

一、毕业设计的目的

1. 综合运用大学所学知识，培养自己独立分析问题和解决问题的能力。
2. 强化基本知识和技能的理解和掌握，培养收集资料和动手操作能力。
3. 对三维图像处理方面和计算几何、计算机图形学的知识有一定的了解和认识。
4. 对自己掌握知识的广度和深度、动手能力、外语水平的综合考核。
5. 为继续学业做好技术储备。

二、主要设计内容

本课题以C++语言、Visual Studio 2013变成工具和三维图像处理软件MeshLab相结合为背景，重点研究基于变分网格逼近的理论实现网格简化的算法，采用局部贪心算法实现对给定的高精度的复杂三维模型，输出粗精度的简化模型。

1. 重点研究问题

重点研究如何编写代码实现变分逼近算法，来实现曲面三角网格的简化。

1. 主要技术指标或主要参数

1、前期查阅相关的资料，了解三维模型在计算机中的数据存储方式，了解相关的数学知识，掌握MeshLab的使用方法；

2、在Visual Studio 2013下使用C++编程实现算法，并进行系统测试。

五、基本要求

通过对本课题的设计，输入一个高精度的三维模型，输出一个对应的保留其几何特征的粗精度的三维模型，具有可观的视觉效果与简化程度。

六、其它

第3周：自行选择设计题目，课题调研、查阅检索相关的文献以及资料，研究课题的可行性，以及课题的研究结果可以带来的效益。

第4周：借阅图书，收集资料，完成毕业设计开题报告，了解网格简化的具体方法及研究背景、研究方向，编制工作计划表。

第5～6周：熟悉变分网格逼近理论及算法中所使用的数学概念与数学公式、几何计算的方法。

第7～9周：进行代码编写。

第10周：调试程序，最后阶段的补充，对各步骤进行连接，使整个简化过程顺利进行。

第11-12周：提交成果，完成毕业论文的编写、修改，为答辩做好准备。

第13-14周：预答辩及正式答辩。

指导教师：郑作勇 刘卫光 2016 年 3 月 7 日

## **华北水利水电学院本科生毕业设计开题报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学生姓名 | 马可 | 学号 | 201214402 | 专业 | 计算机科学与技术 |
| 题目名称 | 基于变分网格的曲面三角网格简化 | | | | |
| 研究概述 | 在计算机图形学中，三维几何形体通常用多边形网格来表示；随着相关技术的进步，得到高精度的复杂网格数据的难度也随之大大降低，甚至拥有数以万计、乃至数以亿计的庞大网格数量的三维模型也不再稀有，不仅外形美观，且保持了较高的质量。但这些复杂的数据也对计算机的处理能力提出了较高的要求，提升了网格存储、读取、传输、绘制等操作的难度。如何权衡网格质量和处理速度之间的关系成为研究所考虑的重点之一，如何得到质量满足要求的网格，并具备较为可观的处理速度是问题的关键所在。  而针对不同的应用目的，所要求的三维模型的精度也有所不同；在很多情况下，我们所需要的往往并非高精度的三维模型，一些粗精度的模型即可以解决应用的需求。在这一背景下，如何将高精度模型简化得到一个粗精度的简化模型成为了一个新的研究方向。而在多边形网格中，三角形网格是结构最为简单、应用最为广泛的，它拥有简单的数学表示，并且也是绝大多数与图形图像有关的软硬件直接支持的最基本的网格单元，而其他多边形的网格亦可对其三角化，转化为三角形网格进行表示。因此本文依据变分网格逼近的思想，使用相关的集合与数学知识，利用局部贪心算法，提出一个对三角形网格简化的解决方案。 | | | | |
| 主  要  内  容 | **一 、课题背景及设计目标**  **1.课题背景**  随着计算机图形学的发展，为大型多边形网格提供快速而准确的简化算法是图形学中一个重要的研究方向，对其多方面都提出了极高的要求。产生与输入几何模型相对应、保持其基本几何特征的另一个相对粗糙的新的逼近模型，是网格简化的主要目的。新产生的模型可以在简化数据量的前提下满足预期的需求，而不必处理过于庞大的数据量，极大程度上提升了工作的效率。而任意的多边形网格都可以看作是由多个三角形网格组成的，所以对三角形网格简化的研究是具有一般意义的基础操作，多边形网格的简化操作亦可先转移为三角形网格再进行简化。  为一个曲面找到一个简洁的、但仍具备几何特征的数学表示，是许多图形学研究主题的核心。鉴于许多三维数据集的过度冗长（尤其是对网格的扫描），降低网格元素（三角形、四边形或其他多边形）的数量同时保持原始三维模型的几何保真度是几何处理的关键之处。在理想的情况下，通过拉伸曲面来适应我们期望的形状区域来使每个元素尽可能的高效，同时尽量减少几何近似的误差。  **2.设计目标**  通过对本课题的设计，用户可以将高精度的大容量的三维模型转化为粗精度的三维简化模型以满足各种需求。这些三维模型可以是由3DMax，AutoCAD等3D绘图软件所绘制的3D图形，或一些给定的三维模型，借助三维图像处理软件MeshLab，获得较好的简化程度及视觉效果。  **二 、网格简化实现的意义**  在计算机图形学中，通常采用多边形模型来绘制单个物体或整个三维环境。平面多边形，尤其是三角面片，能够简洁、高效地进行绘制，所以得到广泛的应用。  而三角面片的简单性在应用过程中也造成了一些困难。通常，三维模型需要上百万的三角面片对其进行渲染，而更复杂的三维场景则需要上亿甚至更多的三角面片来绘制一个平滑的曲面，这使得计算机硬件处理能力已不能满足现在三维计算的要求。在协同设计领域，复杂的模型已经成为异地传输的一个瓶颈。  在很多情况下，可以在模型精确度和硬件处理能力之间进行折衷，即在保持模型几何外观和允许误差范围内，采用适当的简化操作，减少原始模型的几何特征（包括面片数、边数和顶点数），达到用户需要的简化速度和质量，或者二者之间一个极佳的平衡。因此，各种多边形网格模型简化算法应运而生，这些算法的提出使得大型模型的渲染和网络传输成为可能，具有重要意义与应用价值。  **三 、算法实现功能**   1. **根据设计目标，初步对系统模块的划分和功能进行描述如下：**    1. **基于.obj文件进行的文件操作**   OBJ文件是Alias|Wavefront公司为它的一套基于工作站的3D建模和动画软件"Advanced Visualizer"开发的一种标准3D模型文件格式，很适合用于3D软件模型之间的互导，也可以通过Maya读写。比如你在3dsMax或LightWave中建了一个模型，想把它调到Maya里面渲染或动画，导出OBJ文件就是一种很好的选择。目前几乎所有知名的3D软件都支持OBJ文件的读写，不过其中很多需要通过插件才能实现。它仅是一种3F模型文件，不包括动画、材质特性等多余信息，支持三个点以上的面，亦可根据需求对多边形进行三角化操作.   * 1. **通过fstream对给定.obj数据文件进行读写操作**   2. **使用MeshLab对三维数据文件读取和绘制，观察其效果**   **2. 系统设计指导思想**  本课题设计以Microsoft Visual Studio2013作为开发工具，使用线性表的数据结构存储数据，综合面向对象与面向过程的编程思想，基于计算几何的数学知识进行设计。 | | | | |
| 主要参考文献 | [1]Garland M, Heckbert PS. Surface Simplification Using Quadric Error Metrics. In: Whitted T, ed. Proceedings of ACM SIGGRAPH. Los Angeles: ACM Press, 1997. 209-216.  [2]Hoppe H.Progressive Meshes. In: Rushmeier H, ed. Proceedings of ACM  SIGGRAPH. New Orleans: Addison-Wesley Professional, 1996. 99-108  [3]Klein R, Liebich G, Straßer W. Mesh Reduction with Error Control. In: Yagel R, Nielson GM, eds. Proceedings of IEEEVisualization. San Francisco: IEEE Computer Society Press, 1996. 311-318.  [4]Garland M, Heckbert PS. Simplifying Surfaces with Color and Texture using  Quadric Error Metrics. In: Ebert DS, Rushmeier H, Hagen H, eds. Proceedings  of IEEE Visulaization. Washington: IEEE Computer Society Press, 1998.  263-269.  [5]ECook R, eds. Proceedings of ACM SIGGRAPH. Los Angeles: ACM Press, 1995.  173-182.  [6]Delingette H, Herbert M, Ikeuchi K. Shape representation and image segmentation using deformable surfaces. Image and Vision Computing, 1992,10(3):132-144.  [7]Lee AWF, Sweldens W, Schröder P, Cowsar L, Dobkin D. Maps: Multiresolution adaptive parameterization of surfaces. In: Machover C, ed. Proceedings of ACM SIGGRAPH. Orlando: ACM Press, 1998. 95-104.  [8]Alliez P, Meyer M, Desbrun M. Interactive Geometry Remeshing. In: Appolloni T, ed. Proceedings of ACM SIGGRAPH. San Antonio: ACM Press, 2002. 355-361.  [9]Alliez P, Cohen-Steiner D, Devillers O, Levy B, Desbrun M. Anisotropic Polygonal Remeshing. In: RockWood AP, ed. Proceedings of ACM SIGGRAPH. San Diego: ACM Press, 2003. 485-493.  [10]Gu X., Gortler S, Hoppe H. Geometry Images. In: Appolloni T, ed. Proceedings of ACM SIGGRAPH. San Antonio: ACM Press, 2002. 363-374.  [11]Valette S, Chassery J-M. Approximated Centroidal Voronoi Diagrams for Uniform Polygonal Mesh Coarsening. Computer Graphics Forum(Proc. Eurographics), 2004,23(3):381-389.  [12]Valette S, Kompatsiaris I, Chassery J-M. Adaptive Polygonal Mesh Simplification With Discrete Centroidal Voronoi Diagrams. In: Lazzari G, Pianesi F, Crowley JL, Mase Kenji, Oviatt SL, eds. Proceedings of ICMI. Trento: ACM Press, 2005. 6 55-662  [13]Hoppe H, DeRose T, Duchamp T, McDonald J, Stuetzle W. Mesh Optimization. In: James TK, ed. Proceedings of ACM SIGGRAPH. Anaheim: ACM Press, 1993. 19-26.  [14]Lindstorm P, Turk G. Image-driven simplification. ACM Transactions on Graphics, 2000,19(3):204-241.  [15]Cohen-Steiner D, Alliez P, Desbru M. Variational Shape Approximation. In: Marks J, ed. Proceedings of ACM SIGGRAPH. LosAngeles: ACM Press, 2004. 905-914. | | | | |
| 采取的主要技术路线或方法 | **四．研究方法：**  1. 熟悉所需要用到的计算几何公式及具体方法。   1. C++ Code的编写。 2. 使用MeshLab对obj数据文件进行读取操作，观察简化效果。 | | | | |
| 时间安排 | 第3周：课题调研、查阅检索相关的文献以及资料，研究课题的可行性，以及课题的研究结果可以带来的效益。  第4周：借阅图书，收集资料，完成毕业设计开题报告，了解网格简化的具体方法及研究背景、研究方向，编制工作计划表。  第5～6周：熟悉变分网格逼近理论及算法中所使用的数学概念与数学公式、几何计算的方法。  第7～9周：进行代码编写。  第10周：调试程序，最后阶段的补充，对各步骤进行连接，使整个简化过程顺利进行。  第11-13周：提交成果，完成毕业论文的编写、修改，为答辩做好准备。  第14周：正式答辩。 | | | | |
| 指导教师意见 | 签 名：  2016 年 月 日 | | | | |
| 备注 |  | | | | |