

分类号 TP391

密级

学校代码 10542

学号 052100068

三维网格模型的变形算法研究

Research on Deformation Algorithm of 3D Mesh Models

研究生姓名 周世哲

指导教师姓名、职称 满家巨 副教授

学科专业 计算机应用技术

研究方向 计算机图形学与数字图像处理

湖南师范大学学位评定委员会办公室

二零零八年三月

摘 要

三维网格模型是计算机辅助工业品外形设计，计算机动画制作，游戏角色与场景创建的基本素材，也是在计算机上进行微分几何，空间解析几何与拓扑学等数学理论研究所必不可少的研究工具。所以对三维网格模型的研究一直是计算机图形学的热点与重点。

在电子艺术家为电影或是三维电子游戏进行角色动画创作时，或是工业产品设计者用计算机辅助的方法对工业产品的外形进行调整和修改时，都要对三维网格模型进行变形编辑操作。因此三维网格模型的变形算法是网格模型研究中很重要的组成部分。一个好的变形算法，应该具有两个核心的优点：一是在速度上，算法在普通家用电脑的配置上运行的效率必须要达到实时，这样才能使用户在编辑模型时可以实时的看到自己每一步修改后的效果，否则缓慢的形变结果的显示将使编辑的过程十分漫长和繁琐；二是在质量上，一个好的变形算法应该尽可能的保持网格模型的原有的几何细节，同时在大角度的变形过程中维持三维模型的视觉可信度，使得观察者不至于产生模型物体被大尺度变形所“损毁”的感觉。

本文研究并实现了目前国际上流行的比较优秀的几种变形算法，针对他们的在变形效果与算法效率等方面的优缺点进行了分析，并在原有基础上提出了一些改进方案，以程序实现得到了对比结果，接着还在数学上对这些改进方案进行了分析。

关键词：三维网格模型，曲面变形，计算机辅助设计

ABSTRACT

3D mesh models are not only the basic material for shape modeling of industrial products and the making of computer animation, computer game characters and environment, but also an indispensable tool needed in researching of basic theories of differential geometry and topology with computers. So the research of 3D mesh models has been paid great attention in computer graphics for years.

When an artist is designing an animation character for movies or 3D computer games, and when a designer's trying to modify the shape of an industrial product in a computer aided way, they are going to deform and edit the corresponding 3D mesh model. So the deformation algorithm of 3D mesh models is a very important part of mesh researches. A good deformation algorithm must have two kernel advantages: the first one is to have the efficiency of a real-time execution on a regular computer so that the user could watch the results of editing in a real-time way; the second one is that the algorithm should preserve as much geometry details as possible and maintain the visible plausibility of the mesh model in a continuous deformation process to make sure the user will not have such a feeling that

the model has actually been “destroyed” by large deformation.

In the paper we focus on two advanced algorithms which are very popular in the research community, and we analysis the deformation effects and execution speed after having implemented them. Then we proposed some modified step corresponding to their disadvantages. With a lot of experiment results, we showed our improvements in comparison with the original algorithms. Then we did some mathematical analysis of our modifications to show their validity.

Key Words: 3D mesh model, Surface deformation,
Computer aided design

目 录

中文摘要	I
英文摘要	II
1. 引言与研究现状	
1.1 引言	(1)
1.2 研究现状	(5)
2. 快速初始化的基于 laplacian 的网格变形	
2.1 网格表示与微分 laplacian 坐标	(15)
2.2 基于 laplacian 的网格形变算法及原理	(18)
2.3 用变换拟合优化的 laplacian 算法及原理	(22)
2.4 快速初始化的基于 laplacian 的网格曲面变形	(26)
3. 基于线性旋转不变微分坐标的网格变形算法研究	
3.1 曲面的微分结构与离散基本型	(31)
3.2 离散曲面方程	(36)
3.3 利用离散曲面方程与离散基本系数进行网格曲面变形	(38)
3.4 基于线性旋转不变微分坐标的变形流程	(40)
4. 选取主方向的线性旋转不变的曲面变形方法	
4.1 线性旋转不变方法保持几何细节的数学原理	(44)
4.2 网格曲面的主曲率方向	(45)
4.3 选取主曲率方向建立局部标架	(46)
4.4 实验结果与分析	(48)

结语.....	(51)
参考文献.....	(52)
附录.....	(55)
后记.....	(56)

第一章 引言与研究现状

1.1 引言

三维模型编辑中的变形技术是一门新兴的学科，其雏形源自于计算机辅助几何设计 CAGD (Computer Aided Geometric Design: 简称 CAGD)中的自由曲线曲面造型技术。曲线曲面造型技术是一门综合了代数曲线曲面的形状调配，重建，变形和简化以及散乱数据点差值等多种领域的学科，它主要研究在计算机系统环境下对曲面信息的表示与光滑性分析。与之不同的是，本文中研究的模型变形研究的对象是离散的网格模型，其基本构成元素是点，线以及形成三角分割的三角形面片。这些相互离散的点，线，面是对其所对应的有着复杂表达式参数解析曲面的一种简化与逼近，由于这种研究对象性质上的差异，造成三维网格模型的各种研究与其肇源于的 CAGD 在研究目的和研究方法上都有很大的不同。

在研究目的上，主要是为了三维动画造型和电子艺术设计。随着近年来国内外电影，电视，商业广告等多媒体作品，以及三维电脑游戏的大量制作与展示，作为其中核心的电脑动画技术得到了广阔的发展空间和长足的进步。在动画设计人员进行动画创作时，希望虚拟角色的动态形象能在电脑模型上得到生动逼真的展现，这就促使人们从原来的效果生硬呆板的角色动画技术中不断创新，从而于最近几年逐渐开创出网格变形与编辑(mesh deformation and editing)的这一新领

域。

在研究方法上，变形技术的焦点集中在对网格曲面的一些几何不变量的保持上，这些几何量类似于微分几何中的主曲率，主方向以及三类基本量等几何不变量，虽然与微分几何中的正则而且连续的情形略有区别，但仍然有着与坐标系无关的特征，可以很好的反应网格模型的内在几何性质。同时，算法的设计者必须别具匠心，自行定义一些独特的不变量来更好的保持网格细节。在对这些量进行数学建模之后，将遇到如何在连续的变形过程中不失偏颇而又较为迅速的恢复这些几何量的问题，现在较为流行的方法是采用线性优化，非迭代式的快速求解，已获得实时的编辑效果；当然应该提到在其中也出现了一些非线性的迭代的方法，这些方法在速度上当然不能与线性的相比，但在变形效果上还是有某些可取之处的。

1.1.1 网格变形方法的研究需求与基本特点

如前所述，网格模型是一切三维动画制作的素材。如果抛开网格点之间的连接关系，网格模型就是一种点云模型，也即由大量的带有三维空间坐标的相互孤立的点所组成的集合。这些点数据的获取曾经是一个比较困难的问题，但是随着激光扫描技术的发展，从对真实物体的三维扫描中获取网格曲面模型的代价已经可以接受了。这些从扫描仪上获取的数据量很庞大，所描述出来的物体表面也是很光滑的。这两个特性使得三维网格模型在多重尺度的范围内包含了丰富的表面几何信息，同时也让许多传统的曲面变形技术难以应付这样的网

格模型，比如传统方法需要的适当的参数化就无法在这些不规则的三角形网格上得到实现。

基于三维网格模型的这些特点，出现了许多有针对性的研究。本文主要对那一类线性的，基于曲面的方法进行研究和总结。区别于传统的CAGD中的自由形态变形(Free-Form Deformation, 简称FFD)这种基于控制点的方法，以及空间/子空间变形(Space/SubSpace Deformation, 简称SPD)这种基于模型空间的方法，基于曲面的变形技术专注于对网格曲面的几何信息的表达与恢复，从而能很好的绕过网格模型数据散乱不规则的障碍。而所谓线性的方法，是指算法的主体部分是在形成一个求解网格顶点坐标值的线性系统方程组并进行求解，这个线性系统形成的目的则是为了保持所定义的几何不变量在最大程度上与原系统保持一致。在许多情况下，这个线性系统是一个全局的二次泛函最小问题所导出的欧拉-拉格朗日方程。

1.1.2 网格变形的基本概念和一般编辑流程

网格变形是一种对物理现象的计算机模拟，即模仿在真实世界里物体在收到外力的作用下发生形变的过程与结果。一般的，为便于理解可以将模型看成一个可塑性较好的橡胶材质的物体，在人的双手的揉捏下变形，比如图1-1演示了一个类八字形双环状物体在固定住底下一侧，然后在上端一侧受到类似挤压的外力后产生形变的效果。

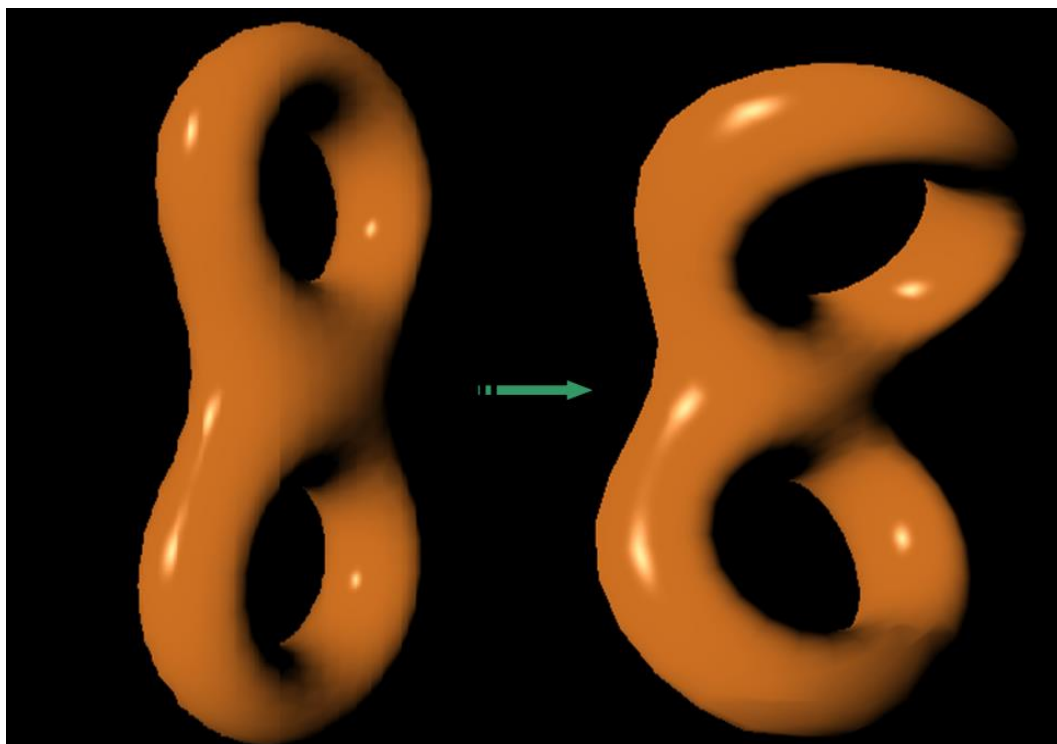


图1-1 简单的网格模型的变形示意图

网格编辑就是一连串的网络变形操作。我们定义初始网格模型为 M , 变形后的网格为 B' , 网格编辑就是一系列受控制约束的网络变形: 如图1-2所示, 用户在编辑网格 M 时, 首先选定该模型中的一个感兴趣ROI(Region Of Interest, 简称ROI)的区域作为编辑区域 E , 即 $E \subset M$, 如图1-2中红色的部分所示; 然后以这个区域内的一部分顶点 H 作为“手柄”, 即 $H \subset E$, 如图1-2中黄色的部分所示; 移动或是旋转这部分点集, 通过变形算法使得整个编辑区域内的所有顶点跟随“手柄”的运动而产生合理的运动, 或者“手柄” H 部分的运动以某种方式外推到编辑区域 E , 从而使网格发生形变。通过了解编辑的流程, 我们知道变形算法的主要任务就是确定编辑区域内顶点的跟随运动过程, 使得编辑的效果尽可能的“逼真”。这里逼真的具体含义是指: 不仅能使得 M 和 M' 在局部的几何细节上尽可能的相似, 同时在大体上看又是按照“手柄” H 的移动完成了合乎直观的形变。

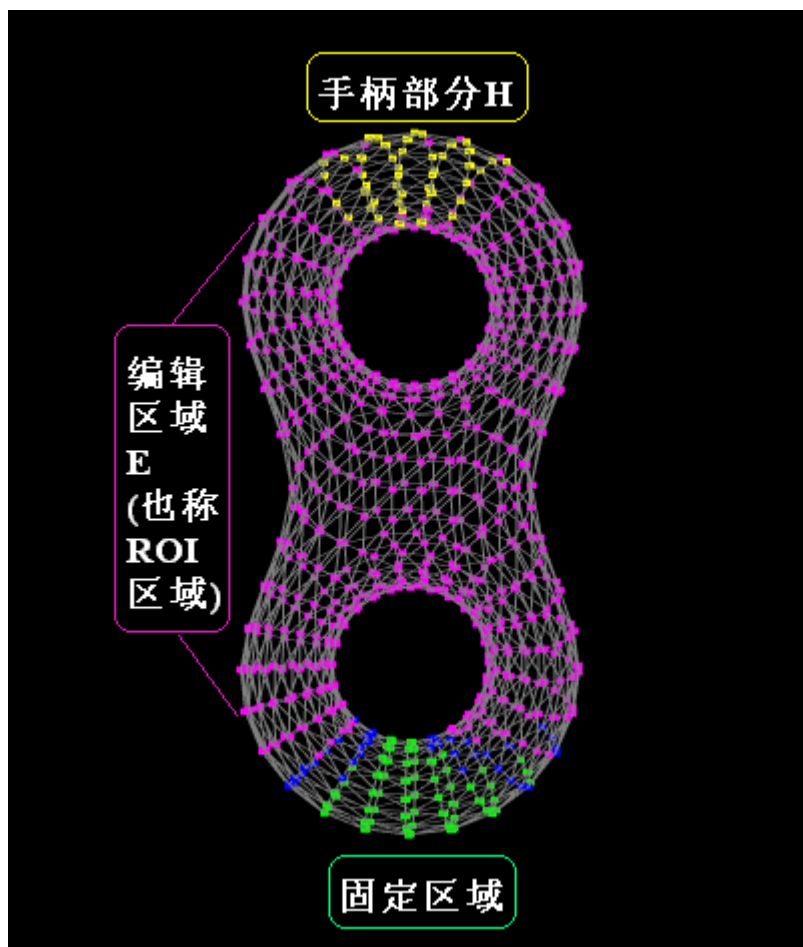


图1.2 网格模型变形用户交互区域选取示意图

1.2 研究现状

按照其所基于的数学原理与基本变形框架,我们可将目前出现的较为流行的变形算法分为三类:基于网格微分表示的变形算法,基于多分辨率的变形算法和基于物理形变能的变形算法。我们着重研究的是第一类算法,即基于网格微分表示的变形算法中的代表者。鉴于时间和精力有限,以及我们没有进行深入的研究与程序实现,所以我们在这里只能对基于多分辨率变形算法框架与物理形变能的变形算法框架进行概括性的描述。

1.2.1 基于曲面微分表示的变形算法

在微分几何学中，使用主曲率，平均曲率等可以反应正则参数曲面在任一点处局部微小区域内的弯曲程度；在基于曲面的网格变形技术中，也定义了一些类似的几何量来反映由离散的三角面片拼合成的曲面的局部几何信息。这就引出了基于曲面微分表示的一类变形算法。这一类算法的共同点是通过定义在网格顶点上的微分几何量来描述网格模型的几何特征，而且这些几何量基本上都可找到它们各自在连续情形下对应的微分几何量。由于是定义在相互独立的网格顶点上，所以这些量反应的是局部的微分属性，如果要获得整个模型的全局光滑自然的变形效果，必须将这些局部量进行整合，而这个过程通常是由对全局形变差异函数取最小二乘意义的极值来完成的，这往往得到一个关于顶点位置的线性系统，因此基于曲面微分表示的一类变形算法大都是线性的算法。由于线性系统求解的相对快速，这就给基于曲面微分表示的变形算法带来了速度上的优势。

在这类算法的研究上，以色列与德国的研究人员取得了大量的成果：如特拉维夫大学的研究人员Sorkine等提出了最小二乘网格重构法^[1]。Sorkine等后又提出了一种基于laplacian微分坐标的网格表示方法与变形框架^[2]；Lipman等指出了在网格编辑中微分坐标表示与局部交互控制的关系^[3]；Alexa则用基于laplacian框架进行了二维边界图形及三维网格模形变体 (mesh morphing)的研究^[4]。但是laplacian坐标不是旋转及缩放不变的，由于这个缺陷使得变形效果不尽如人意，Lipman等随后提出了一种在编辑交互中修正法向方向的方法，在形变效果上

提高了一些^{[5][6]}；Sorkine等则用隐式优化拟合局部变换的方法改进了原始的基于laplacian的网格变形框架^[2]；这些改进在形变效果上比以前有了较大的提高。

除了使用laplacian算子来定义网格曲面上的离散微分量，人们还想到了使用其他的离散几何算子。其中最著名的梯度域poisson方法就是其一。Poisson方程是数学中较为重要的方程，其在工程应用的很多方面都有应用；最早基于poisson的方法被用来进行图像处理，Pérez等将其成功的用于图像编辑操作^[7]，获得了极大的成功，在poisson的梯度域光滑过渡的思想下，使得不同色调亮度饱和度的图像之间的柔和过度与拼接成为可能，使用与实现起来也很方便。由于这些优点，人们一直希望将poisson方法在网格研究应用，但是由于网格坐标是三维信息以及网格顶点构成的流行曲面的高度不规则性，使得poisson方法不能直接从图像处理移植到三维网格。后来对离散微分几何算子的研究提供了途径，Desbrun等在进行网格广顺方法的研究论文^[8]中提出了一种定义在不规则三角形网格上的梯度算子的方法，从而使得poisson方法有了在网格处理上应用的可能；很快，Yu等在2004年^[9]开发了一种基于poisson方法进行网格表面梯度域操控进行网格变形的框架，使得poisson方程在网格变形中发挥了平滑形变误差保持局部细节的作用；随后Xu等就利用基于poisson的框架进行了网格三维形状插值与变体的研究^[10]。

在网格变形中一个重要的变换操作是旋转，但是基于laplacian的方法不能感知旋转的变化，基于poisson的方法只能对小角度(135度以

内)的旋转进行较好的形变,在大角度多周期的旋转上也无能为力,因此人们希望开发出一种面对旋转变换不变的微分坐标法。Lipman等在2005年就提出了一种新颖的基于线性旋转不变^[11]的方法来同时恢复网格顶点处的标架与重建曲面的几何形状,同时该方法保持了线性求解的框架,所以获得了很好的效果和效率。

Sheffer等为了处理旋转则另想办法定义了一种所谓的“金字塔”坐标来进行^[12]网格变形与变体,这种坐标不仅可以反映旋转对网格的影响,还能在一定程度上体现网格细节的体积量,应该说是一种比较全面的坐标,但是由于这个框架是一个非线性的迭代的框架,速度很慢,因此使用及后续研究较少。这里我们就引出了网格变形中的体积保持的问题,大尺度变形往往使得网格严重的萎缩和扭曲,即在视觉体积上不能很好的保持,这也是网格处理中较难解决的一个问题,其原因在于:体积的计算是一个非线性的过程。许多研究者对如何线性的保持体积进行了研究: Bostch^[13]等首先在多分辨方法的框架下使用模型体内衍生的平行六面体棱柱来间接维持模型的体积,获得了非常不错的效果,其缺点是需要生成大量的棱柱数据,实现较为麻烦;中国的 Zhou等则另辟蹊径^[14],他们在网格内外双向生成所谓“体积图”对原有网格曲面进行扩展使得在大尺度变形下模型仍旧很好保持其体积,而且这种体积图的方法是线性的,所以速度也很快;随后 Huang等又利用子空间梯度域变形的思想解决了大规模点数模型保体积变形的问题^[15]。另外,在线性系统求解方面,为了解决大型模型变形时所生成的超大规模线性系统求解较慢的问题, Shi^[16]等使用了多重网

格方法进行快速并行的运算求解线性方程组，同时他们还在多模型同时形变的场景上获得了突破。

1.2.2 基于多分辨率表示的网格变形方法

对一个网格模型，我们在不同的应用场合可以不同的点数进行表示，这就是所谓的模型多分辨率表示，如图1-3所示。

多分辨率变形的主要思想是将网格曲面 S 视为一个“几何信号”，以便将其中的高频信号和低频信号进行分离，如图1-4所示。低频部分可视为一张光滑的基曲面 B ，它表示了模型的全局的形状；高频部分表示了几何细节 D ，它是原网格曲面 S 和基曲面 B 之间的差异，即有 $D = S - B$ ；当然原始网格曲面 S 也可以通过将细节叠加在基本层上重新获得，即有 $S = D + B$ 。多分辨率变形的基本流程就是对剥离了细节的基本层 B 进行变形使成为 B' ，然后通过重建得到 $S' = D + B'$ 。这样做既对全局的大尺度的形状进行了形变，同时又保留了局部小尺度的几何细节，一举两得。

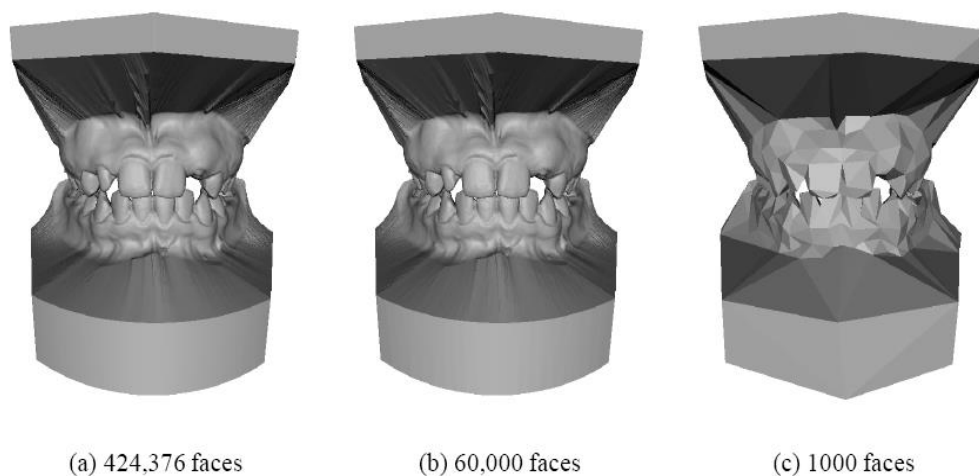


图1-3 以不同多分辨率层次表示的同一个网格模型示意图

为了获得一个比较光滑的同时又能较好的体现原模型外观的基本

层，我们需要一个优秀的网格光顺算法。Guskov等在^[17]，Taubin等在^[18]以及Desbrun等在^[8]中分别各自提出了不同的从网格模型中移除高频噪声分量的光顺算法。与之对应，为了生成在基本层上的细节层，使得网格的局部几何细节能够得到较好的编码存储与还原，研究者Kobbelt等在^[19]以及Zorin等在^[20]中分别各自提出了不同的提取网格细节层分量的算法。在后面的重构方式上面，Bostch等提出了一种法向偏移的细节转移算法来进行基本层与细节层融和^[21]；这之前他们还提出了一种利用法向偏移体计算的多分辨率网格编辑框架^[22]。Marinov等则提出了一种利用曲面细分(surface subdivision)的进行多分辨率分离的方法^[23]，随后Bostch等也利用曲面细分来开发了一种多分辨率网格编辑框架^[24]。

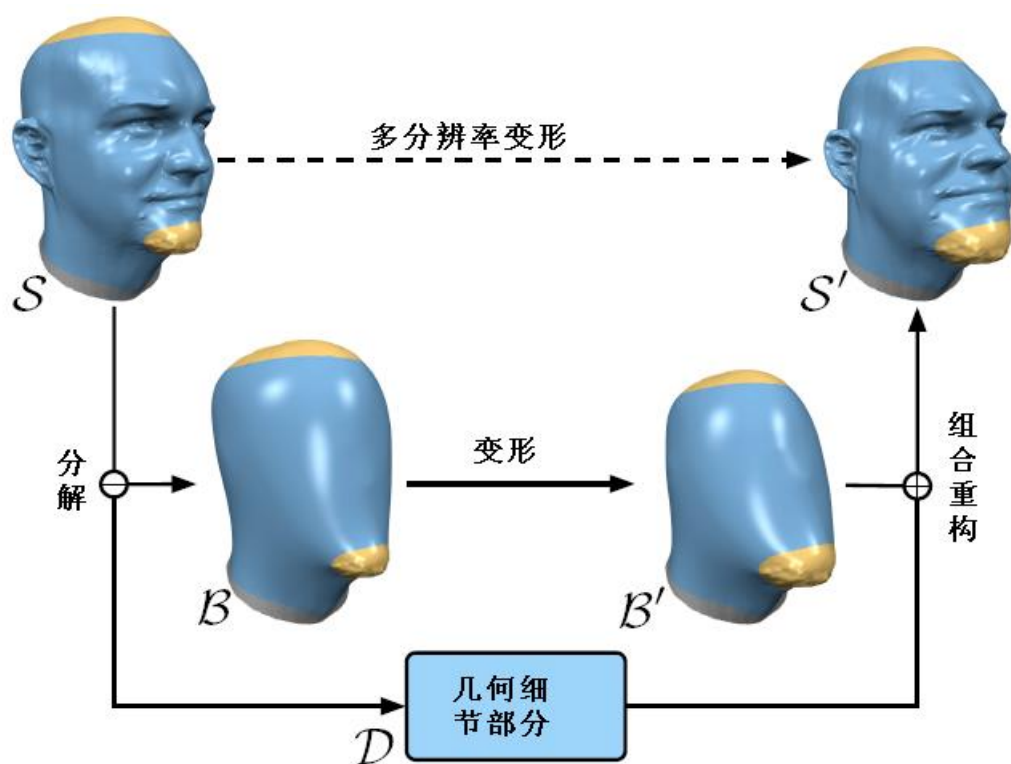


图1-4 基于多分辨率变形方法的基本流程

1.2.3 基于曲面变形能量的网格变形算法

我们对基于曲面变形能量的网格变形算法做了简单的程序实现，并在此阐述他们共同的物理与数学原理。

1.2.3.1 曲面变形能量

对可塑弹性材质的薄壳的力学特性的研究一直是物理与数学中的启发点：比如在变分学中就对在外荷载作用下固定边界内的曲面形态进行了研究并导出了力学中的所谓最小作用原理。由于网格曲面事实上可以看成一张带有边界的很薄的物理弹性薄壳，同时曲面编辑就是为了寻找产生物理上可信的变形效果，因此二者在研究方法上具有相当的相似性。

定义在参数域 Ω 上的一曲面二维流形 S 在受到驱使下形变为曲面 S' ，即有 $S' = S + d$ ， d 是曲面偏移量，Terzopoulos 等在^[25]中指出壳 S' 的内蕴变形能量可写为：

$$E_{shell}(S') = \int_{\Omega} k_s \|\mathbf{I}' - \mathbf{I}\|_F^2 + k_b \|\mathbf{II}' - \mathbf{II}\|_F^2 dudv \quad (1-1)$$

这里的 \mathbf{I} 和 \mathbf{II} 分别代表 S 的第一，二基本型， \mathbf{I}' 和 \mathbf{II}' 分别代表 S' 的第一，二基本型，注意到它们是与曲面的参数化无关的，并且在刚体运动下不变，这符合形变能是由曲面的几何形状决定的本征量的属性；常数 k_s ， k_b 是曲面材质的拉伸和弯曲刚度系数，其值越大，壳反抗拉伸形变或弯曲形变的强度越大，也蕴含更大的形变能。要获得物理可信的变形效果，要求解出使得式1-1取得极小值的曲面，即获得曲面收到驱使力后能达到的能量最小的稳定态。如图1-5所示，图中描述

了正方形边界的一张曲面在我们的系统中由初始态(平直)开始,受到黄色区域代表的“手柄”向上提供的趋势后以最小内蕴形变能为原则发生的变形。

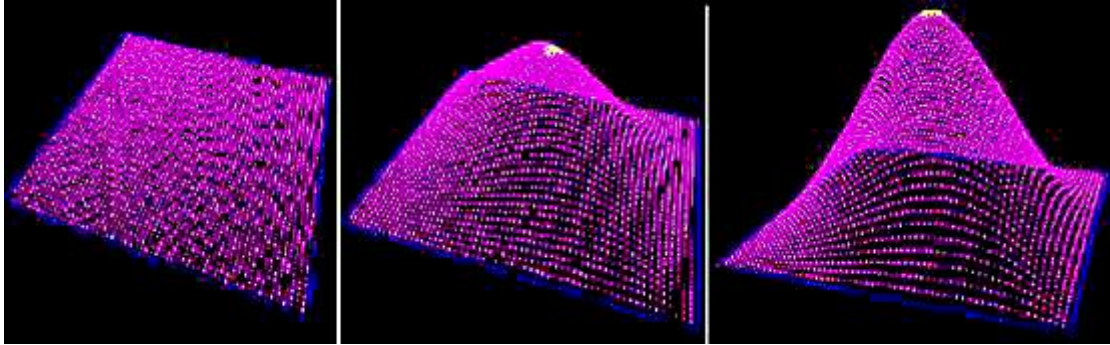


图1-5 形变曲面必须满足能量最小原理,从而决定其形状
(图中设定形变能参数 $k_s = 1$, $k_b = 2$.)

由于式1-1的非线性,直接求其最小是十分耗时的计算。于是Celniker和Gossard在^[26]中,以及Welch和Witkin在^[27]中都指出可以用偏移函数d的一二阶偏导数来代替S'的I, II基本型,且该过程只抛弃了u,v以及 $\sqrt{u^2+v^2}$ 高阶无穷小项,故可得:

$$E'_{shell}(d) = \int_{\Omega} [k_s (\|d_u\|^2 + \|d_v\|^2) + k_b (\|d_{uu}\|^2 + \|d_{uv}\|^2 + \|d_{vv}\|^2)] dudv \quad (1-2)$$

求其变分最小即导出如下的Euler-Lagrange方程:

$$-k_s \Delta d + k_b \Delta^2 d = 0 \quad (1-3)$$

其中的 Δd , $\Delta^2 d$ 表示偏移量d的一阶与二阶拉普拉斯算子,即

$$\Delta d = d_{uu} + d_{vv}, \Delta^2 d = d_{uuuu} + 2d_{uuvv} + d_{vvvv} \quad (1-4)$$

由此可看出薄壳曲面的的拉伸,弯曲能与形变偏移量d的二阶,四阶偏导数之间的对应关系。通过线性的方法求解式1-4即可得到参数域 Ω 上 C^1 连续的被形变曲面。

1.2.3.2 能量最小方程的三角型网格离散求解

式1-1是连续情形下的薄壳形变能量，因此需要讨论其在三角形网格的离散形态才能为我们的变形算法所用。对于式1-3，我们必须解决一二阶拉普拉斯算子在三角形网格上的离散问题。Meyer等对不规则三角形二维流形网格上的偏微分算子和几何量进行了离散定义^[28]，他们不仅提出了梯度，散度，旋度等偏微分算子，还对多维度仿射坐标系内的几何插值坐标进行了定义，如平均值坐标(Mean Value Coordinates, 简称MVC)等；Desbrun等则直接指出二阶偏微分算子，如式1-3所需要的离散拉普拉斯算子可由一个顶点及其一邻域的几何量(如相邻边夹角等)计算得出^[29]。

在方程求解的计算上可以采用有限元方法或是有限差分法，有限元方法可以获得更高的精度但是对于式 1-3 在理论上需要定义一个 Ω 上 C^1 连续的形状函数, Nealen 等在^[30]指出这在三角形网格上通常是很难办到的，因此相比之下有限差分算法则容易实现得多。被离散后的能量最小方程用差分法展开后将会得到一个正定对称的稀疏线性系统，求解该系统可以有多种选择，例如 Golub 提出的著名的共轭梯度法^[31]，这种方法求解精度很高但是这种迭代的方法复杂度达到 $O(n^2)$ ，这对于点数众多的大型模型是不合适的。我们可采用类似 Kobbelt 在^[32]中使用的多重网格等级分解法以 $O(n)$ 复杂度快速的求解。

虽然用二次能量来近似非线性的薄壳变形能量可以求解一个线性系统从而降低每一帧的计算耗时，而且一个全局的能量最小化可以保持曲面变形后的光滑和 C^1 连续；但是我们实验发现线性化会造成几何

细节与法向突出的特征部分受到扭曲，特别是在大角度的旋转变形时，这一点和我们在下面一章将要谈到的基于 laplacian 的网格变形算法的缺点相类似。

第二章 快速初始化的基于 laplacian 的网格变形

在基于曲面微分表示的一类算法中,基于laplacian的算法是非常成功的一种,也是出现的最早的一种微分变形算法^[2]。这种算法的最大优势在于原理易于理解,容易实现而且鲁棒性很强,因此近年来非常流行,在基于laplacian的基本框架内出现了许多分支与改进的新算法,其中使用变换拟合优化的改进是较为成功的一种。

在本章中,我们通过实现基于laplacian的算法,和实现变换拟合优化改进过的基于laplacian的算法,发现了他们由于有形成的线性系统过于庞大而导致初始化十分缓慢甚至不能成功的初始化的问题,这导致用户在实际使用时对一些点数较多的大模型不能很好的变形。

为了解决这个问题,我们设计了大型稀疏矩阵的各种运算算法,有针对性的解决了它们在大模型变形时初始化效率过低的问题,使原来的基于laplacian的算法变得快速,从而让编辑大型模型成为快速简便的工作。

2.1 网格表示与微分 laplacian 坐标

对网格模型M,可视为由两部分组成: $M=(C,V)$, C为顶点间连接关系的集合, $V=\{v_1, v_2, \dots, v_n\}$ 是顶点集合,其网格曲面由 n' 个以 v_i 为顶点的三角形 $\{t_1, t_2, \dots, t_{n'}\}$ 拼合而成。M的空间几何形状由每个顶点的三维坐标 $p_i = p(v_i) \in \mathbf{R}^3$ 给出。

参照上一章1.1.2节所述的变形流程,我们需首先选定ROI区域E

(这里不妨假定是整个网格的所有顶点 $V=\{v_1, v_2, \dots, v_n\}$ 被编辑, 即 $E=M$), “手柄”区域 H , 以及ROI区域之外的固定不动的部分 F ; 然后通过用户交互移动“手柄” H 来控制变形。不失一般性地, 可假定前 $m-1$ 个顶点是编辑区内的自由点, 后面的 $k=n-m+1$ 个顶点是“手柄”部分的控制点: $u_i \in \{u_m, u_{m+1}, \dots, u_n\}$ 。我们这里先只给出一些最基本的记号, 其他的记号将在要用到时逐一出现。

网格模型中的顶点是以三维空间坐标和定点序号来进行标识的, 每个顶点代表一个三维向量 $P(x, y, z)$, 但是这些坐标是对于一个确定的坐标系而言的, 这就是网格模型的模型坐标系(在本研究中所使用的Obj型文件格式里, 模型坐标系的原点一般位于模型的中心对称点, 三个坐标轴则分别与点集的分布主方向对应)。当使用其他的坐标系, 则所有顶点的坐标将全部变化。这就给网格变形带来一个问题: 如果直接使用这些顶点的空间坐标来列出仿射变换的公式, 那么这样的方法将只适用于一个特定的仿射坐标系, 并且在多物体共存在一个场景中的情形下就不再适用了。所以我们必须寻找一种与坐标系尽量无关的量来表示网格, 这样构造出的算法才具有几何不变性。

对网格顶点 v_i 定义微分laplacian坐标 δ_i 及laplacian算子 L 为:

$$\delta_i = L(v_i) = v_i - \frac{1}{d_i} \sum_{j \in N(i)} v_j \quad (2-1)$$

这里 $N(i)$ 是 v_i 的一邻域顶点的集合, 故也写成 $\delta_i = L(v_i) = v_i - \frac{1}{d_i} \sum_{e(i,j) \in E} v_j$ 。

这里 $e(i, j)$ 是连接 v_i 和 v_j 之间的边, E 是模型中所有的边的集合。 d_i 是 v_i 的度即拥有的邻居数, 在规则的网格中, d_i 一般为6。

微分laplacian坐标是最简单的微分坐标之一，也是使用最广泛的。它是对曲面在顶点局部的曲率向量的模拟。本来在微分几何中是没有定义曲面的曲率向量这个概念的，我们在这里可以借用曲线的曲率向量的概念来描述曲面的曲率向量的概念：

对曲线 $P = P(s)$ ， s 是 P 的弧长参数，则有 $kN = \frac{d^2P}{ds^2}$ ，这里 N 是主法矢， k 是曲率，即 $k = \left\| \frac{d^2P}{ds^2} \right\|$ ；在曲面的情形我们可定义曲率向量 K_i ，其方向是曲面在点 v_i 的法向量 N_i ，模长等于 v_i 的平均曲率 $H(v_i)$ ：

$$K_i = H(v_i)N_i = \lim_{len(\gamma) \rightarrow 0} \frac{1}{len(\gamma)} \int_{v \in \gamma} (v_i - v) ds \quad (2-2)$$

这里的 $len(\gamma)$ 表示包含 v_i 在内的一圈围线的长度， $H(v_i)$ 反映了曲面的弯曲程度， N_i 则反映了弯曲的朝向^[33]。比较式2-1与式2-2可看出两者定义的相似性：laplacian坐标即是连续曲面的曲率向量在三角形网格中的离散，见图2-1。

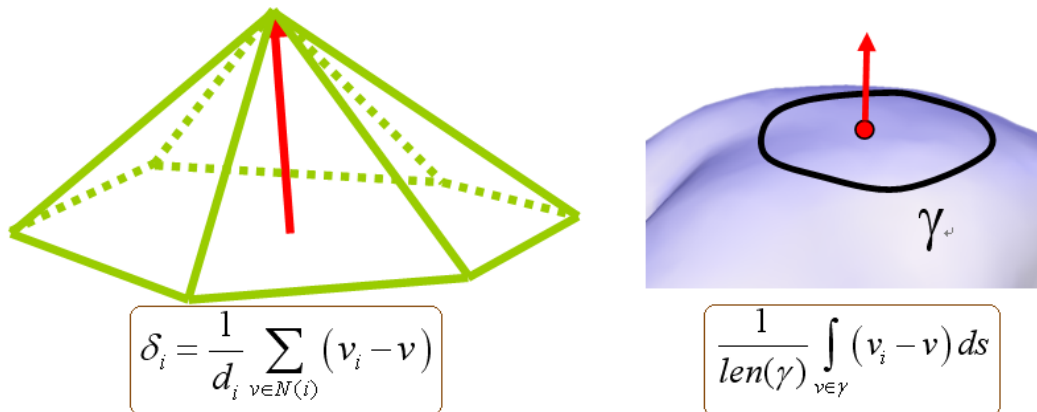


图2-1 laplacian坐标是曲率向量的离散情形

2.2 基于 Laplacian 的网格形变算法及原理

2.2.1 误差能量函数

因为Laplacian坐标刻画了网格局部的几何细节,所以我们可以利用这个特点来重建网格。对原始网格 \mathbf{M} 和形变后的网格 \mathbf{M}' ,二者在几何细节上应尽量保持一致,我们用 \mathbf{M} 和 \mathbf{M}' 的Laplacian坐标之差的模长平方和来衡量这种一致性,即定义细节误差函数:

$$V' = \arg \min_{V'} E(V') = \arg \min \left(\sum_{i=1}^n \|L(v_i') - \delta_i\|^2 + \sum_{i=m}^n \|v_i' - u_i\|^2 \right) \quad (2-3)$$

同时“手柄”H部分的点是在用户交互下受迫移动的,因此在 \mathbf{M}' 中的H'部分的顶点 $\{v_m', v_{m+1}', \dots, v_n'\}$ 应移动至用户交互指定的控制位置 $\{u_m, u_{m+1}, \dots, u_n\}$,即应有 $v_i = u_i, i \in \{m, \dots, n\}$ 。

现在可按两种方式来定义最终的误差能量函数 $E(V')$:以细节误差函数作为 $E(V')$,即 $E(V') = D(V') = \sum_{i=1}^n \|L(v_i') - \delta_i\|^2$,这样“手柄”H部分的顶点坐标固定在 u_i ,这样的交互称为硬约束。但是由于“手柄”H部分是包含于编辑E部分的,所以两者应在同一算法下求出新坐标,而不应区别对待,同时我们的实验发现由硬约束方式进行交互极容易造成H部分在变形后与E区域的其他部分格格不入的“撕裂”现象。鉴于以上考虑,目前流行的算法^{[2][7][34]}一般采用软约束交互形式来定义误差能量函数 $E(V')$:

$$E(V') = \sum_{i=1}^n \|L(v_i') - \delta_i\|^2 + \sum_{i=m}^n \|v_i' - u_i\|^2 \quad (2-4)$$

变形后的网格 \mathbf{M}' 应按照视觉要求使得 $E(V')$ 最小:

$$V' = \arg \min_{V'} E(V') = \arg \min \left(\sum_{i=1}^n \|L(v_i') - \delta_i\|^2 + \sum_{i=m}^n \|v_i' - u_i\|^2 \right) \quad (2-5)$$

软约束的合理性在于其对编辑区域的全局考虑与全局优化。

2.2.2 最小二乘线性优化

式2-4是未知网格 \mathbf{M}' 的顶点 V' 坐标 (x_i', y_i', z_i') 的二次函数, 其取最小值的必要条件是偏导数 $\frac{\partial E(V')}{\partial x_i'} = \frac{\partial E(V')}{\partial y_i'} = \frac{\partial E(V')}{\partial z_i'} = 0$; 由于在 $E(V')$ 中坐标的对称性, 可以单独讨论 $\frac{\partial E(V')}{\partial x_i'} = 0, (i=1, 2, \dots, n)$ 的求解。简单推导可发现, x 坐标向量 $X' = (x_1', x_2', \dots, x_n')$ 应满足:

$$\arg \min_{X'} E(X') \left\| \begin{pmatrix} \mathbf{L} \\ \mathbf{0} | I_{k \times k} \end{pmatrix} X' - \begin{pmatrix} \delta x_n \\ u x_k \end{pmatrix} \right\|^2 = \arg \min_{X'} \| \mathbf{A} X - \mathbf{B} \|^2 \quad (2-6)$$

\mathbf{L} 是网格 \mathbf{M} 的 n 阶 laplacian 矩阵, 它完全由连接关系决定, 即

$$\mathbf{L}_{ij} = \begin{cases} 1 & i = j \\ -\frac{1}{d_i} & j \in \{j : e(j, i) \in E\} \\ 0 & \text{其余} \end{cases} \quad (2-7)$$

$I_{k \times k}$ 是 k 阶单位矩阵, 表示了后面 k 个控制点受到的位置约束; n 维 laplacian 坐标 $\delta x_n = (\delta x_1, \delta x_2, \dots, \delta x_n)$ 和 k 维约束向量 $(\mathbf{A}^T \mathbf{A}) X = \mathbf{A}^T \mathbf{B}$ 提供了系统求解的边界条件。

式2-6是 $n+k$ 行 n 列的超定线性组, 按线性优化得到其法方程^[1]:

$$(\mathbf{A}^T \mathbf{A}) X = \mathbf{A}^T \mathbf{B} \Rightarrow X = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{B} \quad (2-8)$$

这里的 $\mathbf{A}^T \mathbf{A}$ 可被证明是满秩的 n 阶实对称方阵^[35]。这一点对我们是非

常重要的，它使我们能够快速的分解这样的稀疏矩阵从而连续地快速的解出未知量。

2.2.3 线性系统求解与交互前初始化

为了交互时达到实时速度，基于laplacian的变形采取了一次分解，多次代入求解的技巧。实际上整个算法就是将未知量全体用已知的顶点坐标以及预先求出的laplacian坐标线性表出的过程，它保证了最后线性求解的可能。观察式2-8的组成元素，可发现在交互过程开始之前，除了方程式右端向量 \mathbf{B} 因包含了H部分的控制点位置约束值而必须在运行时刻生成并实时更新外，其余的 \mathbf{A} ， \mathbf{A}^T ， $(\mathbf{A}^T \mathbf{A})^{-1}$ ， $(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T$ 都可以在交互前初始化时生成；而且直到下一次选择新的编辑区域和控制点，这些数据都不用更改。

交互前初始化主要的计算量在于 $\mathbf{A}^T \mathbf{A}$ 的矩阵相乘和 $(\mathbf{A}^T \mathbf{A})^{-1}$ 的求逆，特别对于点数较大的模型，我们的实验发现其计算量是普通计算机的内存和运算速度难以顺利完成的。这个问题在接下来的2.3节中介绍的用变换拟合优化的laplacian算法实现时将更加突出和棘手。

初始化生成的矩阵都是大型的稀疏矩阵。稀疏的原因在于对每一个顶点 v_i 而言，其未知坐标量的方程中只含有 d_i 个邻居顶点坐标的未知数参与构成方程，所以从某种意义上我们可以将 \mathbf{A} 看成带宽为 d_i 的条带矩阵，由于 $d_i \ll n$ ，故其稀疏度是比较大的；同时由矩阵 \mathbf{A} 计算出的中间矩阵 \mathbf{A} ， \mathbf{A}^T ， $(\mathbf{A}^T \mathbf{A})$ 等也都是大型稀疏阵。

对于式2-8表示的线性方程组 $(\mathbf{A}^T \mathbf{A})\mathbf{X} = \mathbf{A}^T \mathbf{B}$ ，我们不能直接进行求

逆 $(\mathbf{A}^T \mathbf{A})^{-1}$ 来求解，因为 $(\mathbf{A}^T \mathbf{A})$ 是 n 阶方阵，当 n 较大时，不但十分耗时，而且求逆带来的精度损失是十分大的：在我们的实验中发现若直接求逆 $(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}$ ，在使用double型数组计算时精度损失就非常大，单精度float型的结果则更加病态。而我们编辑的网格模型其点数 n 常常是以万计数，所以这个精度问题更加需要解决。

我们采用的方法是对 $(\mathbf{A}^T \mathbf{A})$ 进行分解，将其分解为上三角和下三角矩阵之后，在交互时只需在计算完相对较快的向量与矩阵间的乘法： $\mathbf{A}^T \mathbf{B}$ 后，即可快速的回代求出所有的未知数。分析矩阵 \mathbf{A} 可发现，由于本顶点未知数的系数绝对值是邻居顶点未知数系数绝对值的 d_i 倍，故 \mathbf{A} 是一个主对角占优的稀疏矩阵，因此可证明乘积矩阵 $(\mathbf{A}^T \mathbf{A})$ ^[35]是一个对称正定/半正定矩阵，所以我们可以对 $(\mathbf{A}^T \mathbf{A})$ 进行LU分解，而更快的办法是使用专为对称正定矩阵而设计的Cholesky分解或者Bunch-Kaufman分解法来做，我们通过大量的实验证明Cholesky在应用中要略快于另外两种方法。

在完成 $(\mathbf{A}^T \mathbf{A})$ 一次分解之后，方程组式2-8就只需要等待右端向量的更新来进行回代求解，其回代速度是十分快的。这就是本节开始提到的laplacian框架的最大优点：一次分解，多次代入求解的技巧。通过在大量不同规模模型上的试验我们证实了这一点，见试验结果图表2-1的“交互帧速”栏目，可知对10000点之内的中型模型基本可在交互中达到实时。

2.3 用变换拟合优化的 Laplacian 算法及原理

2.3.1 Laplacian 坐标在仿射变换下的可变性

由于Laplacian坐标是定义在一个模型坐标系中,即对于模型而言的一个全局坐标系,因此当在一顶点的局部发生相对于全局坐标系的旋转变换时,Laplacian坐标无法感知这样的变化,所以Laplacian坐标不是旋转不变的。因此在大角度旋转的交互下,普通的Laplacian算法不能很好的处理几何细节朝向的问题。若单独处理法向量,则将遇到法向量是非线性的难题。

只要稍加推导就可看出:Laplacian算子 L 是平移变化不变的,但 L 不是旋转和缩放不变的,在各向同比缩放下可保持方向不变,在旋转和各向不同比的缩放变换下方向和模长都将变化,如图2-2所示。

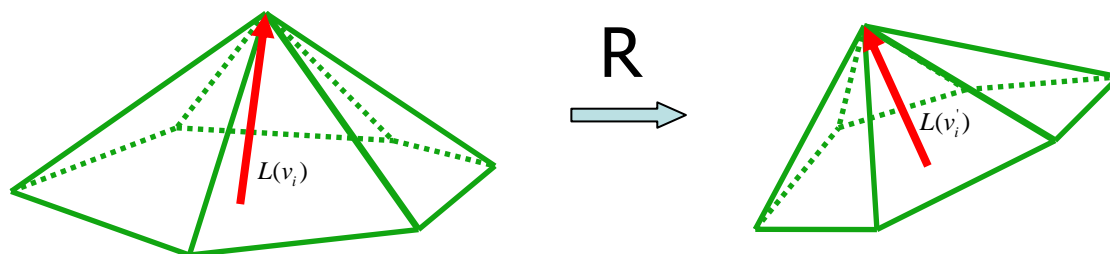


图2-2 旋转变换改变了Laplacian坐标

2.3.2 Laplacian 算法不能很好的处理旋转

理论上Laplacian的缺点直接导致了在实验上的有缺陷的结果。在图2-3可以看到,对兔子头的一只耳朵仅仅进行拖拽的平移操作,也应在产生某些部分的微小的旋转效应(如图中绿色线框内的耳朵与头相连的部位)以达到视觉上与物理上的自然。然而Laplacian方法却不能

得到较好的形变效果。其原因就是上面提到的laplacian坐标不是旋转不变的。如果仅从由平移变换带来的位置约束中得到这样的局部旋转是一个非线性的问题，因此不能使用线性化的方法来完美的求解。当然我们可以考虑使用非线性的迭代方法，如采用金字塔坐标取代laplacian坐标^[12]；更进一步，如果跳出网格微分表示方法的框架，我们还可以使用基于多分辨率或是多尺度等级的某些方法。但是为了保持线性的框架，同时能达到较为符合直观的变形效果，我们必须对原始的laplacian算法框架进行某些改造。Sorkine等在^[2]中就提出了用变换拟合优化的laplacian算法，部分的解决了以上问题。

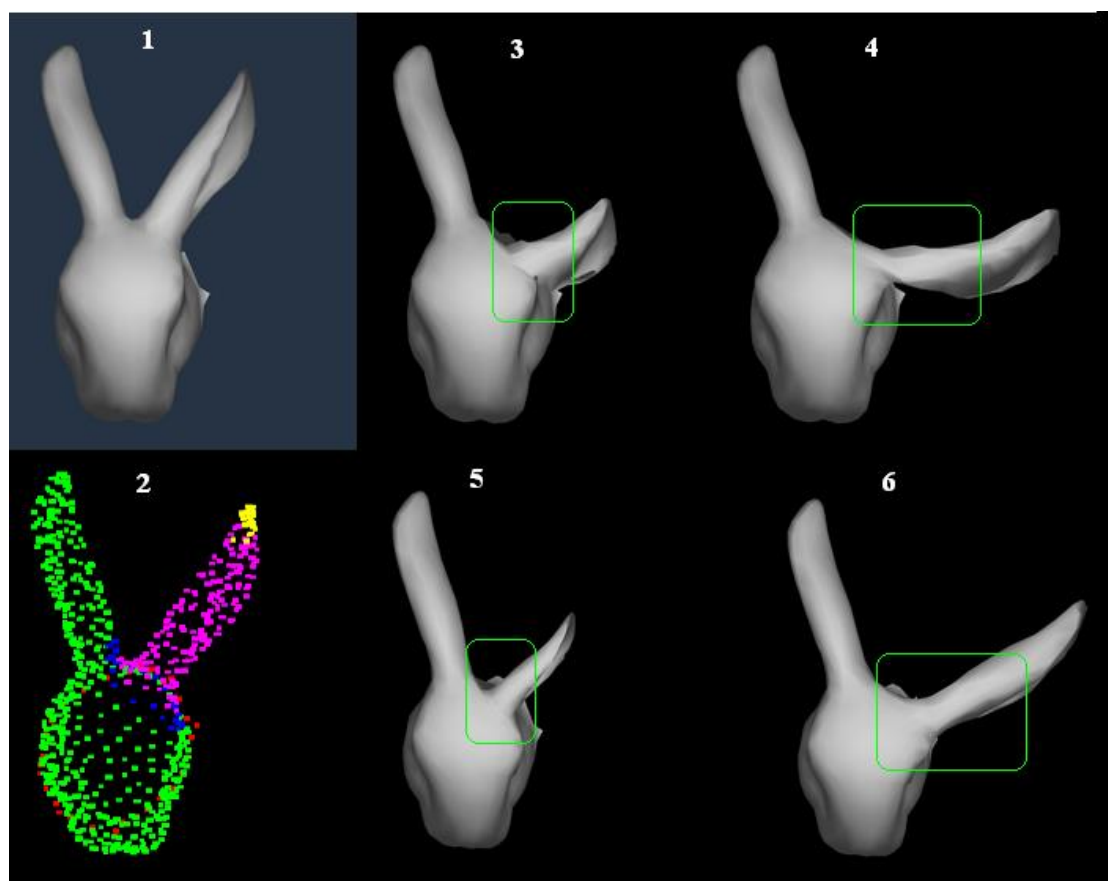


图2-3 拖拽兔子耳朵变形效果图

(图中分图(1)原始兔头，(2)交互式所选的编辑ROI区域(红色的右侧耳朵)和“手柄”H部分的控制点(耳尖的黄色部分)，(3)(4)原始laplacian

算法的缺陷效果，(5)(6)变换拟合优化的laplacian算法在效果上有明显的提升。)

2.3.3 变换估计

为了解决旋转等一些线性变换不能在 laplacian 坐标中被反应的缺点，一个解决办法是对其拟合一个局部的变换^[2]。其主要思想是为每一个编辑区域内的顶点 v_i 估算一个合适的仿射矩阵 T_i ，将原网格 M 的 laplacian 坐标变换之后的来与 M' 的进行比较，这样能绕过 L 算子对除平移以外的其他变换不守恒的缺点。将对 v_i 直接求 laplacian 坐标改成对仿射变换后的坐标顶点 Tv_i 来求，在逻辑上是更加合理的算法。

利用 L 算子和仿射变换 T 的交换律，即

$$L(Tv_i) = T(Lv_i) \quad (2-9)$$

我们得到新的全局误差函数是：

$$\tilde{E}(V') = \left(\sum_{i=1}^n \|L(v_i') - T_i \delta_i\|^2 + \sum_{i \in H} \|v_i' - u_i\|^2 \right) \quad (2-10)$$

现在最关键的问题在于得到 T 。为了保持式2-10的线性，必须寻找 $T_i = T_i(v_i')$ 是未知的位置坐标 v_i' 的线性函数，考虑到最小二乘可导出线性函数，以及在一顶点与其一邻域应保持同一个仿射变换 T ，所以可利用如下的二次极小导出 T ：

$$T_i = \arg \min_{T_i} \left(\|v_i' - T_i v_i\|^2 + \sum_{j=1}^k \|v_{i_j}' - T_i v_{i_j}\|^2 \right) \quad (2-11)$$

其中的 v_{i_j} 表示顶点 v_i 的第 j 个邻居。

一般的变换 T 应包含平移，旋转和缩放等元素，但是由于各向异

性缩放会带来高度的非线性^[2]，同时各向异性缩放在用户进行自然变形时是较少用到的操作，故舍去不考虑；又因为2.3.1节提到的laplacian算子的平移不变性，则在对一个邻域局部的整体平移变换下也是保持不变的，所以又可将平移分量舍去。因而得到只包含旋转，同比缩放的变换矩阵 T 形如：

$$T_i = \begin{pmatrix} s & -h_3 & h_2 & 0 \\ h_3 & s & -h_1 & 0 \\ -h_2 & h_1 & s & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2-12)$$

这样 T_i 被表成四个未知因子 (s, h_1, h_2, h_3) 的反对称矩阵，对式2-11取极小即对四个未知因子 (s, h_1, h_2, h_3) 求偏导可解出：

$$(s_i, h_{i1}, h_{i2}, h_{i3})^T = (A_i^T A_i)^{-1} A_i^T \mathbf{b}_i \quad (2-13)$$

其中的

$$A_i = \begin{pmatrix} v_{jx} & 0 & v_{jz} & -v_{jy} \\ v_{jy} & -v_{jz} & 0 & v_{jx} \\ v_{jz} & v_{jy} & -v_{jx} & 0 \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}, j \in \{i\} \cup N_i, \text{ 以及 } \mathbf{b}_i = \begin{pmatrix} v_{jx} \\ v_{jy} \\ v_{jz} \\ \vdots \end{pmatrix}, j \in \{i\} \cup N_i.$$

这样我们将局部变换阵 T_i 表成了未知顶点 v_i 的线性组合，达成了线性地嵌入laplacian算法框架的目的。

2.3.4 线性系统求解

比照式2-10与前面给出的原始laplacian的线性系统式2-4，经简单计算可知，式2-10不具有式2-4的xyz坐标独立性，即对 $E(V)$ 而言所有的未知点的x坐标可以独立于y，z坐标之外在一个线性系统中单独求解，对y，z亦然，而且这三者的线性系统的左端矩阵 $(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T$ 一摸

一样，只有右端向量 \mathbf{B} 在交互时有区别（他们分别获得控制顶点的交互位置的xyz坐标值），这意味着原始的laplacian算法的 $(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T$ 是 $n*n$ 阶方阵；而对于 $E(V)$ 而言，由于拟合的局部变换阵 T_i 的引入，使得变换拟合优化的laplacian算法的 $(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T$ 矩阵是一个x,y,z三者相互耦合的 $3n*3n$ 的大矩阵，其存储空间是原始的laplacian算法的九倍。

新算法生成的矩阵也都是大型的稀疏矩阵：对于新算法的 \mathbf{A} 矩阵的稀疏性可以理解为，它是由原始算法的 \mathbf{A} 矩阵将每个非零元扩展为一个 $3*3$ 的小矩阵而得来，如下图2-4所示。不难发现，新算法的 \mathbf{A} 矩阵的非零元占总体的比率是和原始laplacian算法一样的，故还是一个大型的稀疏矩阵。同理，可证明 \mathbf{A} 的主对角占优性与待分解矩阵 $\mathbf{A}^T \mathbf{A}$ 的对称正定/半正定性，故也可采用Cholesky分解法。

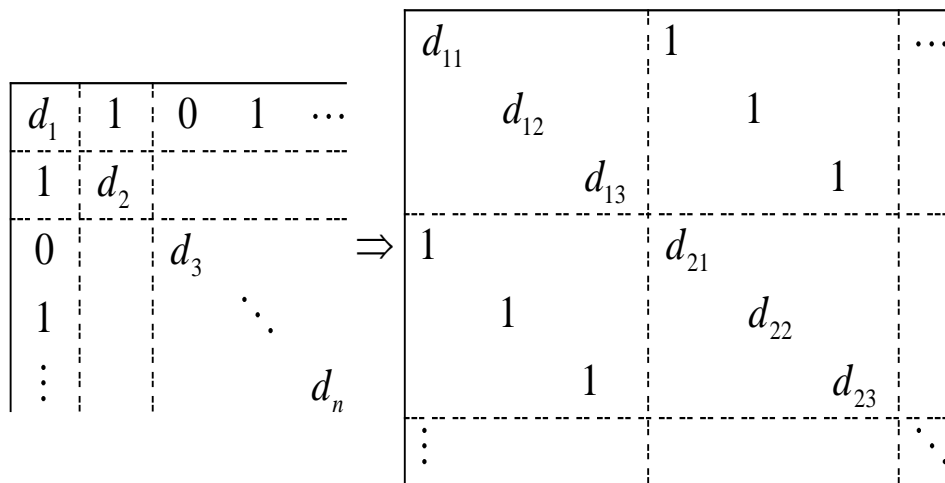


图2-4 原始laplacian算法的 \mathbf{A} 矩阵从形式上扩展到新算法 \mathbf{A} 矩阵的过程示意图

2.4 快速初始化的基于 laplacian 的网格曲面变形

2.4.1 初始化过程分析

通过前面的论述和实际实现我们统计出在初始化过程中需要完成

的任务有：生成 \mathbf{A} (尺寸为 $3(n+k)*3n$)，计算矩阵 \mathbf{A}^T (尺寸为 $3n*3(n+k)$)，计算矩阵 $\mathbf{A}^T\mathbf{A}$ (尺寸为 $3n*3n$)，计算矩阵 $(\mathbf{A}^T\mathbf{A})^{-1}$ (尺寸为 $(3n*3n)$)，计算矩阵 $(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$ (尺寸为 $3n*3(n+k)$)。由上节知，由于三坐标的相互耦合性，相比于原始的laplacian算法，用拟合变换优化的laplacian算法所创建与使用的矩阵都变大为了原来的9倍。

在矩阵计算方面，我们使用了intel公司开发的MKL: Math Kernel library的计算库，这是一个高性能的商用数学计算软件库，其功能涵盖广泛：支持一般的矩阵三角化，分解，求奇异值等，还支持许多常用的计算方法：如FFT快速离散傅立叶变换，DST离散余弦变换等，而且还支持大型稀疏线性系统的求解等。在程序实现时我们发现，MKL的乘法调用时必须在内存中同时保留矩阵 \mathbf{A} 和 \mathbf{A}^T ，接下来的分解与回代调用时又必须在内存中同时保留矩阵 $\mathbf{A}^T\mathbf{A}$ 的分解结果阵和 \mathbf{A}^T ；考虑到新算法矩阵的尺寸是原来的9倍，其占用的内存量是十分庞大的，操作系统在MKL调用时为这些大矩阵分配内存需要花费相当长的时间：我们的实验发现，相比起来，真正用于乘法，分解等等运算的时间占整个初始化过程的时间却并不多，大部分的时间是花在了硬盘虚拟内存的开辟和内存间数据移动上，见实验结果图表[表2]。此外对于一些大型的模型，如10万个点的模型，若“手柄”部分控制点为100个，则其单精度float型 \mathbf{A} 矩阵所占内存空间为：
 $3*100000*3*(100000+100)*4 = 36599062500$ 字节，约合34903GB字节，这是不可能满足的内存需求。所以从实现角度来说，新laplacian算法对这样的大型模型根本是无能为力的。

2.4.2 改进线性系统生成效率与试验结果

为此我们对整个计算过程作了优化工作：首先我们利用 \mathbf{A} 矩阵高度稀疏的特点，在生成 \mathbf{A} 时采用稀疏存储。我们根据全部顶点度数之和来计算 \mathbf{A} 矩阵中非零元素的数目，再根据一顶点 v_i 与其邻居顶点在顶点序列编号上的位置关系来逐次计算出这些非零元，并按照顶点行主序来进行存放。这样就避免了花费大量的内存空间存储零元素。实验发现对于那些正则度较高的模型，即平均的度数 d_i 约为6，稀疏度就是 $6/n$ ，回顾上面提到的10万点的模型，这样的稀疏存储即可使 \mathbf{A} 矩阵存储空间降为 $34903 \times 6 / 100000 = 2.0914\text{GB}$ 字节，这对于内存比较充裕的电脑已是可以接受的了。

接下来在矩阵相乘和分解方面，我们充分利用了 \mathbf{A} 矩阵的对称性和顶点相邻关系，推测 $(\mathbf{A}^T \mathbf{A})$ 中非零元的位置，这样我们就可以只使用 \mathbf{A}^T 就可以单独完成这次乘法与分解，无需再花内存空间存储 \mathbf{A} ，这样又能节省不少的存储空间；另外在速度方面，我们的优化只是在推测顶点相邻关系的传递性上需要做一些判断与遍历，所以在整个计算上复杂度并未太大的增加。计入节省下的开辟硬盘虚拟内存与大批量数据移动的时间，我们的优化使得初始化过程的时间大大缩短，同时也能在家用电脑配置下处理大型或超大型模型成为可能。对各种模型的初始化过程所耗内存和运算时间优化前后的对比结果见表2-1。

表 2-1 不同级规模的模型在优化前后初始化时间空间消耗对比

模型名称	模型点数	耗内存量(MB)		初始化用时(sec)		交互帧速(fps)
		改进前	改进后	改进前	改进后	
hand	1254	135.3	12.2	9	0.5	250
man	10139	1205.8	111.4	279	16.5	4.4
armadillo	20031	2405.8	218.8	2008	191	2.4
horse	42568	#over	737	#over	559	1.1
Planck	100086	#over	2146	#over	1245	0.35

我们的测试平台是AMD64 3000+的CPU, 1G内存和普通显示卡的个人电脑, 表2-1中的“#over”表示因为我们的系统无法为没被我们优化的laplacian变形程序开辟所需的巨大的内存以及磁盘空间(即已超过windows的4G虚拟内存上限)而无法成功初始化, 所以不能获取到内存(包括虚拟内存)和耗时的数据。如图2-5显示了一个大型模型(点数为48969)的形变, 说明我们的工作使得用户更加方便。



图2-5 初始化优化使大模型 (armadillo) 形变编辑快速而方便

第三章 基于线性旋转不变微分坐标的网格变形算法研究

第二章研究的基于laplacian的算法的优点是能保持网格细节, 并有一个线性的框架; 而它最大的缺点在于不能很好的处理法向量与旋转变换。虽然上一章的2.3节所引出的变换拟合优化的laplacian算法从一定程度上解决了小角度旋转与同比缩放的变换问题, 但是它是从一个最小二乘优化的角度隐式地拟合出一个较为合适的变换矩阵, 并没有从根本上解决如何处理法向量的问题, 因而还是不能应付大角度多周期的旋转变形。实际上, 这个缺点是由laplacian坐标本身不是旋转不变的内在缺陷所导致的。所以为了更好的处理旋转特别是大角度多周期的旋转, 我们必须跳出laplacian框架, 建立另外一种能够在旋转变换下保持不变的微分坐标。

3.1 曲面的微分结构与离散基本型

为了更深入的剖析三角形网格曲面的几何构造, 我们可先来研究具有一定连续阶(无论是参数连续还是几何连续)的正则曲面的微分属性。微分几何的研究发现: 对给定的一曲面, 无论采用哪种参数化方式, 我们总可以通过分析的方法得到一些与参数选取和坐标系都无关的几何量, 这些量反映了曲面形状所决定的本征性质。换句话说, 它们是曲面的刚体运动不变量, 而刚体运动就包含了我们进行曲面变形编辑所需要的平移和旋转等变换。第二章研究了laplacian坐标所代表的曲面在一点 v_i 的曲率向量, 但曲率向量在曲面旋转后也随之变

化,所以曲率向量和laplacian坐标不属于这样的刚体运动不变本征量。

3.1.1 切面与 Guass 标架

对正则 C^2 连续参数曲面 $P(u_1, u_2)$, 第一个碰到的与参数选取无关的是切平面: 对其上一点 $p_0 = P(u_1^0, u_2^0)$, 此处单位法向量 n 为

$$n = \left(\frac{\partial P}{\partial u_1} \times \frac{\partial P}{\partial u_2} \middle/ \left\| \frac{\partial P}{\partial u_1} \times \frac{\partial P}{\partial u_2} \right\| \right)_{(u_1^0, u_2^0)} \quad (3-1)$$

它的切平面 $T_{p_0}S$ 由 n 唯一决定, 为

$$T_{p_0}S = \{T \in \mathbb{R}^3 \mid \langle T - p_0, n \rangle = 0\} \quad (3-2)$$

易知 $T_{p_0}S$ 是两个参数偏导矢 $\frac{\partial P}{\partial u_1}, \frac{\partial P}{\partial u_2}$ 的线性组合, 即有

$$T_{p_0}S = P(u_1^0, u_2^0) + s \frac{\partial P}{\partial u_1}(p_0) + t \frac{\partial P}{\partial u_2}(p_0) \quad (3-3)$$

将 $P(u_1, u_2)$ 在 p_0 点附近按参数 Taylor 展开, 有

$$P = p_0 + \frac{\partial P}{\partial u_1}(p_0)\Delta u_1 + \frac{\partial P}{\partial u_2}(p_0)\Delta u_2 + \frac{1}{2} \left[\frac{\partial^2 P}{\partial u_1^2}(p_0)\Delta u_1^2 + 2 \frac{\partial^2 P}{\partial u_1 \partial u_2}(p_0)\Delta u_1 \Delta u_2 + \frac{\partial^2 P}{\partial u_2^2}(p_0)\Delta u_2^2 \right] + o(\Delta u_1^2 + \Delta u_2^2) \quad (3-4)$$

上式最末包含有参数二次增量的高阶无穷小, 可抛弃。

从分析角度看, 比对式3-3和式3-4, 我们可用 $T_{p_0}S$ 上的点近似的代替曲面 S 上 p_0 附近的点, 于是我们称 $T_{p_0}S$ 为 S 的一次切平面。

再者, 若更精确的用式3-4的前三项定义一二次曲面 Q :

$$Q = p_0 + \frac{\partial P}{\partial u_1}(p_0)s + \frac{\partial P}{\partial u_2}(p_0)t + \frac{1}{2} \left[\frac{\partial^2 P}{\partial u_1^2}(p_0)s^2 + 2st \frac{\partial^2 P}{\partial u_1 \partial u_2}(p_0) + \frac{\partial^2 P}{\partial u_2^2}(p_0)t^2 \right] \quad (3-5)$$

该二次曲面可比 $T_{p_0}S$ 更好的近似代替曲面 S 上 p_0 附近的点, 但是它的二次项与曲面的曲纹参数 (u_1, u_2) 有关^[33], 因此不能算是几何的本征性质, 更好的办法是用二次项在 S 的法方向的投影来代替之, 于是我们得到另一二次曲面:

$$m_{p_0}(s, t) = p_0 + \frac{\partial P}{\partial u_1}(p_0)s + \frac{\partial P}{\partial u_2}(p_0)t + \frac{1}{2} \left\langle \left(\frac{\partial^2 P}{\partial u_1^2}(p_0)s^2 + 2st \frac{\partial^2 P}{\partial u_1 \partial u_2}(p_0) + \frac{\partial^2 P}{\partial u_2^2}(p_0)t^2 \right), n(p_0) \right\rangle n(p_0) \quad (3-6)$$

$$\left(s, t, \frac{1}{2} \left\langle \left(\frac{\partial^2 P}{\partial u_1^2}(p_0)s^2 + 2st \frac{\partial^2 P}{\partial u_1 \partial u_2}(p_0) + \frac{\partial^2 P}{\partial u_2^2}(p_0)t^2 \right), n(p_0) \right\rangle \right)$$

可以证明 $m_{p_0}(s, t)$ 在参数变换下保持不变^[33], 于是我们称 $m_{p_0}(s, t)$ 为 S 的二次切面。

观察一, 二次切面的形式与三个线性无关的向量 $\frac{\partial P}{\partial u_1}, \frac{\partial P}{\partial u_2}$ 和 n 的联系, 可得出结论: 对曲面 S 上 p_0 的微分区域, 存在局部标架(又名 Gauss 标架): $\left\{ \frac{\partial P}{\partial u_1}, \frac{\partial P}{\partial u_2}, n \right\}$, 可将该区域的点在该标架中分布。若以 p_0 为坐

标原点, 则在该局部坐标系中, $m_{p_0}(s, t)$ 上的点有仿射坐标为

$$\left(s, t, \frac{1}{2} \left\langle \left(\frac{\partial^2 P}{\partial u_1^2}(p_0)s^2 + 2st \frac{\partial^2 P}{\partial u_1 \partial u_2}(p_0) + \frac{\partial^2 P}{\partial u_2^2}(p_0)t^2 \right), n(p_0) \right\rangle \right) \quad (3-7)$$

进一步选取参数使得 $\frac{\partial P}{\partial u_1}, \frac{\partial P}{\partial u_2}$ 正交, 得到正交的 Gauss 标架, $m_{p_0}(s, t)$ 成为该坐标系中的抛物面, 所以也称为 S 在 p_0 处的密切抛物面。

3.1.2 基本型与离散基本型

首先看曲面 S 在在切平面上的度量: 对于增量 $u + du = (u_1 + du_1, u_2 + du_2)$

在 S 上我们可定义全微分: $dP(u) = \frac{\partial P}{\partial u_1} du_1 + \frac{\partial P}{\partial u_2} du_2$, 由式3-3知

$dP(u) \in T_{P(u)}S$, 所以 $P(u)+dP(u)$ 是 $T_{P(u)}S$ 上的点, 则有二次微分式

$$I(dP(u), dP(u)) = \langle dP(u), dP(u) \rangle \quad (3-8)$$

称为曲面 S 的第一基本形式^[33], 以矩阵式展开式3-8易得:

$$I(dP(u), dP(u)) = (du_1, du_2) \begin{bmatrix} E & F \\ F & G \end{bmatrix} \begin{pmatrix} du_1 \\ du_2 \end{pmatrix} \quad (3-9)$$

其中 $E = \left\langle \frac{\partial P}{\partial u_1}, \frac{\partial P}{\partial u_1} \right\rangle$, $F = \left\langle \frac{\partial P}{\partial u_1}, \frac{\partial P}{\partial u_2} \right\rangle$ 以及 $G = \left\langle \frac{\partial P}{\partial u_2}, \frac{\partial P}{\partial u_2} \right\rangle$ 是曲面 S 的第一基本

量, 可证明第一基本形式 I 与参数选取无关。更一般地我们可利用 I 来衡量 $T_{P(u)}S$ 上两个矢量的夹角: 对 $X, Y \in T_{P(u)}S$ 我们定义 $I(X, Y) = \langle X, Y \rangle$,

则有
$$\cos \angle(X, Y) = \frac{I(X, Y)}{\sqrt{I(X, Y)I(X, Y)}} \quad (3-10)$$

可见 I 可以刻划切平面上两个方向间的夹角。

再来到密切抛物面 $m_{p_0}(s, t)$ 上, 类似的定义第二基本型^[33]:

$$II(dP, dP) = (du_1, du_2) \begin{bmatrix} L & M \\ M & N \end{bmatrix} \begin{pmatrix} du_1 \\ du_2 \end{pmatrix} \quad (3-11)$$

其中 $L = \left\langle \frac{\partial^2 P}{\partial u_1^2}, n \right\rangle$, $M = \left\langle \frac{\partial^2 P}{\partial u_1 \partial u_2}, n \right\rangle$ 以及 $N = \left\langle \frac{\partial^2 P}{\partial u_2^2}, n \right\rangle$ 是曲面 S 的第二基本

量。第二基本型刻画了 S 上的点离开切平面 $T_{P(u)}S$ 的高度信息。通过

I, II 基本型我们可以将 p_0 周围微小区域的点的空间位置关系刻画出来,

这就是我们希望得到的曲面微分结构。

Lipman在^[11]中成功的将以上的微分几何理论运用到三角形网格的离散情形, 定义了线性旋转不变的微分坐标, 并建立了与之对应的网格变形编辑框架。

对嵌入二维流形的三角形网格 M ，其上一顶点 v_i ，其模型空间坐标为 \hat{x}^i ，记 $\hat{x}_1^i, \hat{x}_2^i, \dots, \hat{x}_{d_i}^i$ 为从 v_i 指向其各个一邻域邻居顶点的边向量，如图3-1所示； T_iM 为 v_i 处的切平面，则将 $\hat{x}_1^i, \hat{x}_2^i, \dots, \hat{x}_{d_i}^i$ 投影于 T_iM ，可得在切平面上的邻居方位投影向量 $\hat{x}_1^{i'}, \hat{x}_2^{i'}, \dots, \hat{x}_{d_i}^{i'}$ ，对 T_iM 上一向量 $u = u_1 \hat{x}_k^{i'} + u_2 \hat{x}_{k+1}^{i'} \in T_iM$ 我们可模仿式3-8定义第一离散基本型为：

$$\hat{I}^i(u) = \langle u, u \rangle = (u_1, u_2) \begin{bmatrix} g_{k,k}^i & g_{k,k+1}^i \\ g_{k,k+1}^i & g_{k+1,k+1}^i \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \quad (3-12)$$

其中的 $g_{k,k}^i = \langle \hat{x}_k^{i'}, \hat{x}_k^{i'} \rangle$ 和 $g_{k,k+1}^i = \langle \hat{x}_k^{i'}, \hat{x}_{k+1}^{i'} \rangle$ ，注意整个 $\hat{I}^i(u)$ 的定义是在以 v_i 及 v_i 的两个相邻的邻居顶点 v_k^i, v_{k+1}^i 的三角形 $\Delta v_k^i v_i v_{k+1}^i$ 中的。我们同时须定义 $O_k^i = \text{sign}(\det(\hat{x}_k^i, \hat{x}_{k+1}^i, N^i))$ 以反映三元组标架的组合定向，若三者形成右手系则 O_k^i 为正(但是我们的实验发现，在大角度旋转时，标架定向改变的现象也很少出现)。

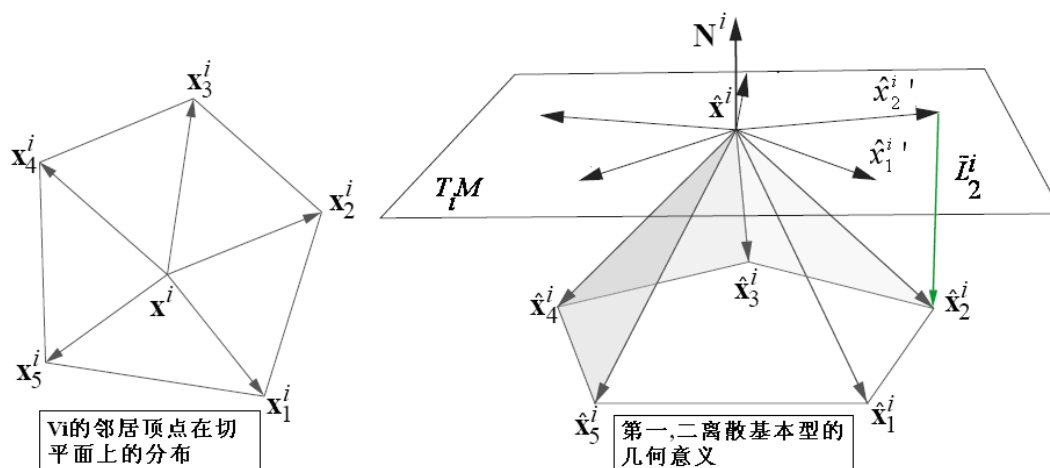


图3-1 第一，二离散基本型的几何意义

第一离散基本型 $\hat{I}^i(u)$ 与第一基本型一样都具有与曲面参数化无关的本征性，实际上 $\hat{I}^i(u)$ 是以 v_i 发出邻边的投影长度信息加上相邻两条投影边之间的夹角来描述 v_i 及其一邻域在切平面 T_iM 上的结构的。

由于网格 M 的各三角片是以 C^0 连续方式进行拼接的，所以我们可以

以分片线性的将第二离散基本型定义为：

$$\hat{\Pi}^i(u) = u_1 \langle \hat{x}_k^i, N^i \rangle + u_2 \langle \hat{x}_{k+1}^i, N^i \rangle = u_1 \hat{L}_k^i + u_2 \hat{L}_{k+1}^i \quad (3-13)$$

其几何意义见图3-1；不难看出第二离散基本型 $\hat{\Pi}^i(u)$ 很好的反映了 $\hat{I}^i(u)$ 所缺乏的法向信息。式中的 $\hat{L}_k^i = \langle \hat{x}_k^i, N^i \rangle = \|\hat{x}_k^i\| \cos \varphi$ 实际上是 v_i 的一圈邻居离开切平面 $T_i M$ 的法向高度值，见图3-1中的 \hat{x}_2^i 。

有了第一,二离散基本型及其系数，实际上就有了在一个顶点局部标架内所有邻居顶点的仿射空间坐标；有了一个顶点 v_i 和其上的局部标架，以及第一,二离散基本型及其系数，就可以把这个邻域的其他顶点坐标恢复出来，所以我们又把他们称为网格顶点的旋转不变微分坐标。

Lipman还证明了离散基本型可以收敛到连续情形下的基本型^[11]。值得提到的是，离散基本型是平移及旋转不变的，但却不是缩放不变：在各向同性缩放下，法向的第二基本型系数 \hat{L}_k^i 会变化；而在剪切变换中，切向坐标也会变化。

3.2 离散曲面方程

3.2.1 建立局部标架

由上节可知，为了重建网格，必须同时具有坐标和标架，二者缺一不可。而且在计算坐标之前我们必须先定义好标架。

由第二离散型的定义知我们首先可以确定顶点 v_i 局部标架纵坐标Z轴应是法向量 N^i ；至于在切平面，由第一离散型的定义知我们必须选定一个平行与 $T_i M$ 的基准X方向才能按照相邻边夹角的方式逐步

确定各个邻居顶点。Lipman 在^[11]中认为, 可以任意选取一个邻居顶点的投影边方向来作为 X 轴, 故不妨选择第一个邻居顶点的 \hat{x}_1 的单位方向 b_1^i 来作为 X 轴; 然后记 $b_2^i = b_1^i \times N^i$ 为切方向的 Y 轴; 这样我们就得到了一个原点置于 v_i 的右手标准正交标架 (b_1^i, b_2^i, N^i) 。注意这样定义的标架的 X 轴向实际上是随意放置的, Lipman 在^[11]中认为无论哪种方式选择 b_1^i 都会有同样的结果。但是我们的研究发现, 可以用更好的方式来选择标架。

3.2.2 法向量的计算

目前出现了较好的计算离散三角形曲面顶点的法向量的方法, 可参见^{[36] [37] [38]}。我们这里简单的使用面积加权法来计算, 即:

$$N^i = \frac{\sum_{k=1}^{d_i-1} S\Delta v^i v_k^i v_{k+1}^i (\hat{x}_k^i \times \hat{x}_{k+1}^i) / \|\hat{x}_k^i \times \hat{x}_{k+1}^i\|}{\sum_{k=1}^{d_i-1} S\Delta v^i v_k^i v_{k+1}^i} \quad (3-14)$$

其中的 $S\Delta v^i v_k^i v_{k+1}^i$ 是以 v^i 为顶点的三角片面积。

3.2.3 利用标架差异定义离散曲面方程

考察一对相邻顶点 $e(i, j) \in E$, 我们定义矢量差分算子 δ 为:

$$\begin{cases} \delta_j(b_1^i) = b_1^j - b_1^i \\ \delta_j(b_2^i) = b_2^j - b_2^i \\ \delta_j(N^i) = N^j - N^i \end{cases} \quad (3-15)$$

Lipman在中利用 δ 定义了所谓离散曲面方程^[11]:

$$\begin{cases} \delta_j(b_1^i) = \Gamma_{j,1}^{i,1}b_1^i + \Gamma_{j,1}^{i,2}b_2^i + A_{j,1}^iN^i \\ \delta_j(b_2^i) = \Gamma_{j,2}^{i,1}b_1^i + \Gamma_{j,2}^{i,2}b_2^i + A_{j,2}^iN^i \\ \delta_j(N^i) = \Gamma_{j,3}^{i,1}b_1^i + \Gamma_{j,3}^{i,2}b_2^i + A_{j,3}^iN^i \end{cases} \quad (3-16)$$

上式将相邻的局部标架的差异矢量放到 v_i 的右手正交标架 (b_1^i, b_2^i, N^i) 进行表示。可以证明差异矢量在标准正交基 (b_1^i, b_2^i, N^i) 中的坐标 Γ 和 A 是可以由第一二离散型计算出。实际上，式3-15代入上式即可得：

$$\begin{cases} b_1^j = (\Gamma_{j,1}^{i,1} + 1)b_1^i + \Gamma_{j,1}^{i,2}b_2^i + A_{j,1}^iN^i \\ b_2^j = \Gamma_{j,2}^{i,1}b_1^i + (\Gamma_{j,2}^{i,2} + 1)b_2^i + A_{j,2}^iN^i \\ N^j = \Gamma_{j,3}^{i,1}b_1^i + \Gamma_{j,3}^{i,2}b_2^i + (A_{j,3}^i + 1)N^i \end{cases} \quad (3-17)$$

于是寻找坐标 Γ 和 A 就等价于寻找 (b_1^j, b_2^j, N^j) 在相邻局部标架 (b_1^i, b_2^i, N^i) 中的坐标，这样的表示当然是存在且唯一的，如图3-2所示。

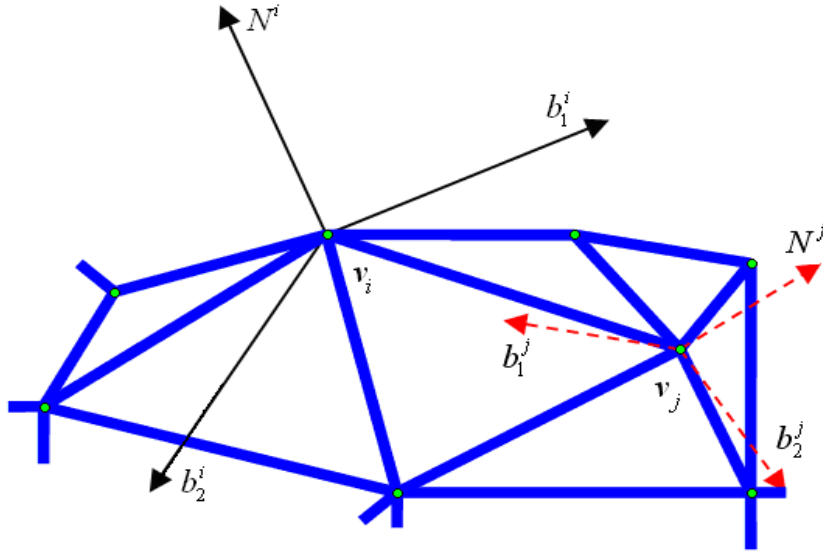


图 3-2 邻居顶点 v_j 标架可在本顶点 v_i 标架局部坐标系内被唯一表出

3.3 利用离散曲面方程与离散基本系数进行网格曲面变形

利用离散曲面方程式3-16可形成以标架 (b_1^i, b_2^i, N^i) 为未知数的线性

方程组，该方程组的系数矩阵是 $(3n * 3(\sum_{i=1}^n d_i + k))$ 的大型稀疏矩阵， k 为H部分的控制点数。这是一个超定方程，故可模仿上一章的式2-8取其最小二乘意义的法方程，得到 n 阶方阵的系统后，也可采用一次分解，多次代入的模式实时求解出新的标架场。下面两个命题进一步说明了本框架可以从交互提供的约束中重建变形曲面的能力。

3.3.1 网格曲面标架场的可唯一恢复

Lipman在^[11]中指出：给定初始网格 M 在每顶点定义的标架三元组向量场 $\Theta := \{v_i(b_1^i, b_2^i, N^i) | v_i \in M\}$ ，以及第一二离散基本系数，就可在被手柄部分控制顶点的约束下，从式3-16形成的离散曲面方程组中解出变形后网格 M' 的新标架三组向量场 $\Theta' := \{v_i'(b_1^i, b_2^i, N^i) | v_i' \in M'\}$ ，其中ROI区域边界固定部分的标架近似等于原顶点被锁定标架，其余的被编辑点的新标架可旋转一定角度与原向量场 $\Theta := \{v_i(b_1^i, b_2^i, N^i) | v_i \in M\}$ 向匹配。

实际上，解出的变形后网格 M' 的新标架三组向量场 Θ' 在最小二乘意义上与初始标架场 Θ 满足离散曲面方程，这是由所定义的旋转不变的局部标架在刚体运动变换下保持相对差异结构不变所决定的。

3.3.2 网格曲面顶点模型空间坐标的可唯一恢复性

从3.3.1节的论述可知，求解出的新标架场在每一个顶点 v_i' 处生成了一个新的局部标架 (b_1^i, b_2^i, N^i) 。我们将原来求出的从 v_i 指向其各个一邻域邻居顶点的边向量 $\hat{x}_1^i, \hat{x}_2^i, \dots, \hat{x}_{d_i}^i$ 的第一二离散基本型看成在新局

部标架中的坐标，即可从其中ROI区域边界固定部分的顶点坐标逐步外推的得到所有被变形点的新模型空间坐标，且这种恢复过程显然是唯一的。我们可将这个从新标价场恢复曲面顶点坐标的过程写成：

$$\hat{x}^j - \hat{x}^i = \langle \hat{x}_k^i, b_1^i \rangle b_1^i + \langle \hat{x}_k^i, b_2^i \rangle b_2^i + \hat{L}_k^i \cdot N^i \quad (3-18)$$

其中的 \hat{x}^j 表示顶点 v_i 的第 k 个邻居点的未知模型坐标。实际上这就是一个简单的从局部坐标恢复全局坐标的过程。类似的，式 3-18 形成的线性方程组也是 $n * (\sum_{i=1}^n d_i + k)$ 的大型稀疏超定线性系统，也需要在最小二乘意义优化下求解。我们实验发现，其法方程的系数方阵也是对称正定的。

3.4 基于线性旋转不变微分坐标的变形流程

3.4.1 初始化与交互流程

基于线性旋转不变微分坐标的网格变形框架与基于laplacian的网格变形框架在流程上有明显的不同。虽然二者都是线性的方法，都能在交互中获得实时的速度，但是基于线性旋转不变微分坐标的变形需要连续求解两个线性系统，第一个是求解由离散曲面方程式3-16组成的线性系统，得到的是受到用户交互约束变形后的新的标价场；第二个是求解由式3-18组成的线性系统，得到的是由新的标价场重建出的编辑区域网格顶点的新位置，从而使得 \mathbf{M} 变形为 \mathbf{M}' 。整个变形算法的流程图如图3-3所示。

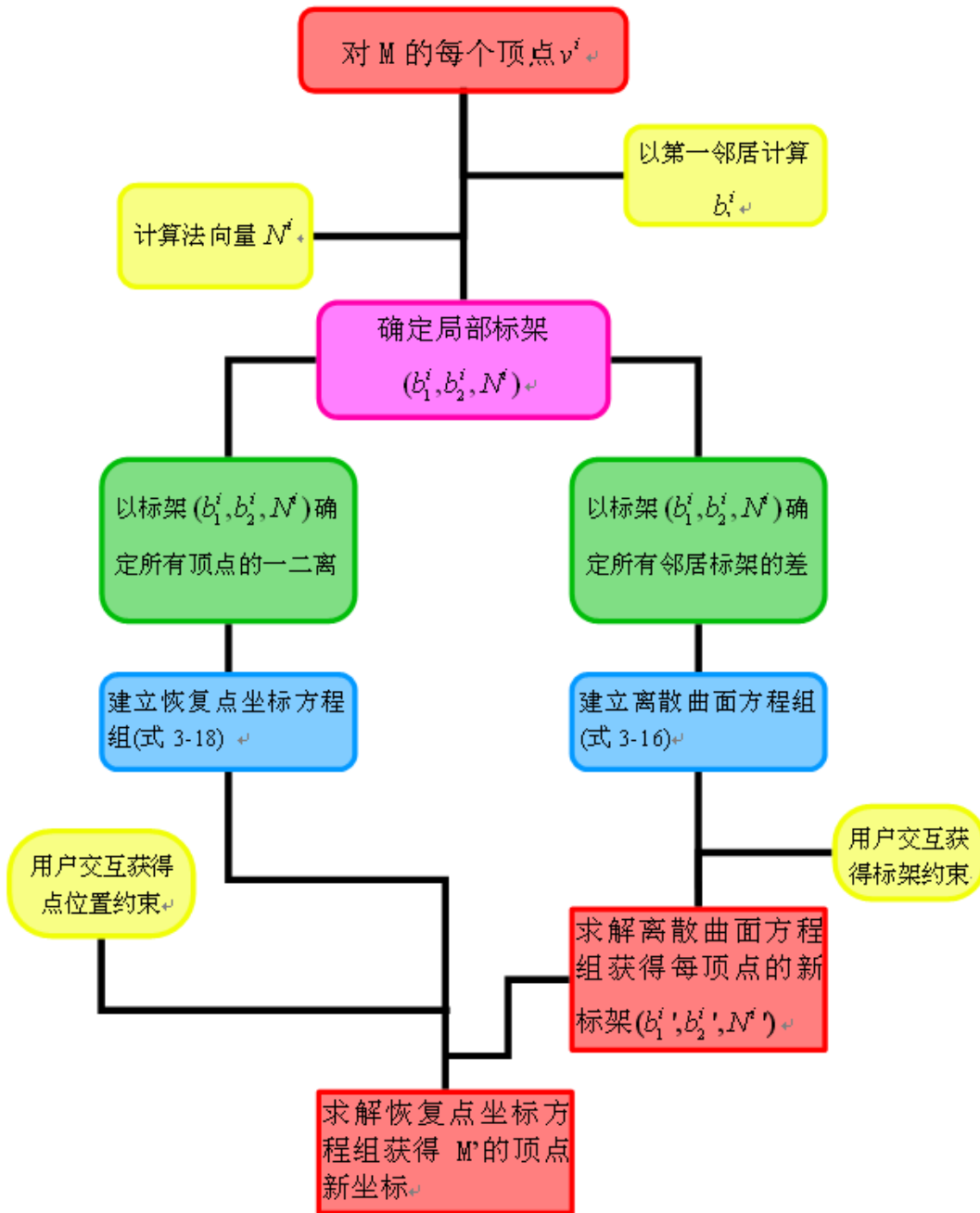


图3-3 基于线性旋转不变网格变形算法流程图

由上图可以看出，两个线性系统在算法运行时是交替求解运行的。所以在初始化的过程中，我们必须为这两个方程组的稀疏矩阵开辟足够内存空间，并且在交互时必须同时保存两者的数据，这导致本算法比第二章介绍的用变换拟合优化的laplacian算法要消耗更多的内存空间，也需要较长的初始化时间。在用户选定一片ROI区域之后，系统

开始生成系数矩阵并进行分解，在之前3.3节提到了这两个系统的系数矩阵都是对称正定阵，故都能采用快速的Cholesky分解。

3.4.2 双重交互方式

在交互提供的约束上，本算法和基于laplacian的算法一样都是使用较合理的软约束方式，实际上这也是所有基于微分网格表达的线性变形算法的共通之处^[39]。但是由于本算法需要求解标架和坐标两个线性系统，所以用户在交互时必须为两个系统都提供所需的约束值，以生成系统所需的右端向量进行回代求解。

对离散曲面方程3-16，交互需改变“手柄”H部分顶点的局部标架，即提供标架的约束；对接下来的恢复点方程3-18，交互需移动H部分的顶点，即提供位置的约束。这里遇到一个问题是：两个交互实际上对用户而言是同时发生的。以往的方法中用户根本看不到标架的旋转，而只关心手柄位置的移动，但在本算法中，标架的旋转必须和顶点的移动同时进行，而且必须紧密而协调的配合起来，才能产生较自然合理的变形效果。我们的实验发现，若标架的控制与点位置的控制不能达到协调，则会产生很不好的效果，如图3-4所示。

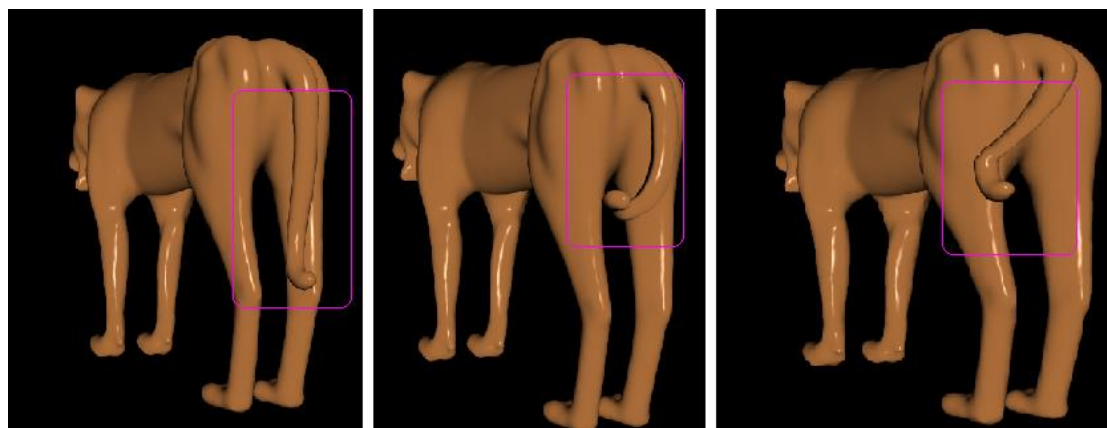


图3-4 协调的交互操控的重要性

(左幅)对同一个部位模型的操作,(中幅)一个好的交互是标架与控制点的约束相协调的,(右幅)不协调的交互控制将造成不自然的形变效果)

值得提到的是,对网格曲面上的标架向量场与顶点位置进行协调的操控是一个较难解决的问题,有许多研究者对这个问题进行了研究与创新,如Zayer在^[40]中提出了一种利用生成调和向量场来指导标架旋转以达到协调操控的方法。

第四章 选取主方向的线性旋转不变的曲面变形方法

在本章中，我们在实现并研究了上一章的线性旋转不变网格变形方法的基础上，提出了一种改进的方案。我们通过在每个顶点处选取离散主曲率方向的方法，建立了主方向局部标架。通过实验证明我们的改进使得算法在处理大角度多周期旋转时能达到更视觉可信的效果。

4.1 线性旋转不变方法保持几何细节的数学原理

基于线性旋转不变微分坐标的方法是能够在连续的曲面变形编辑中保持住曲面在局部的几何细节的，我们在大量的实验中验证了这一点，见图4-1。经分析发现，其数学原因在于离散曲面方程有保持曲面上曲线的曲率和挠率的作用。

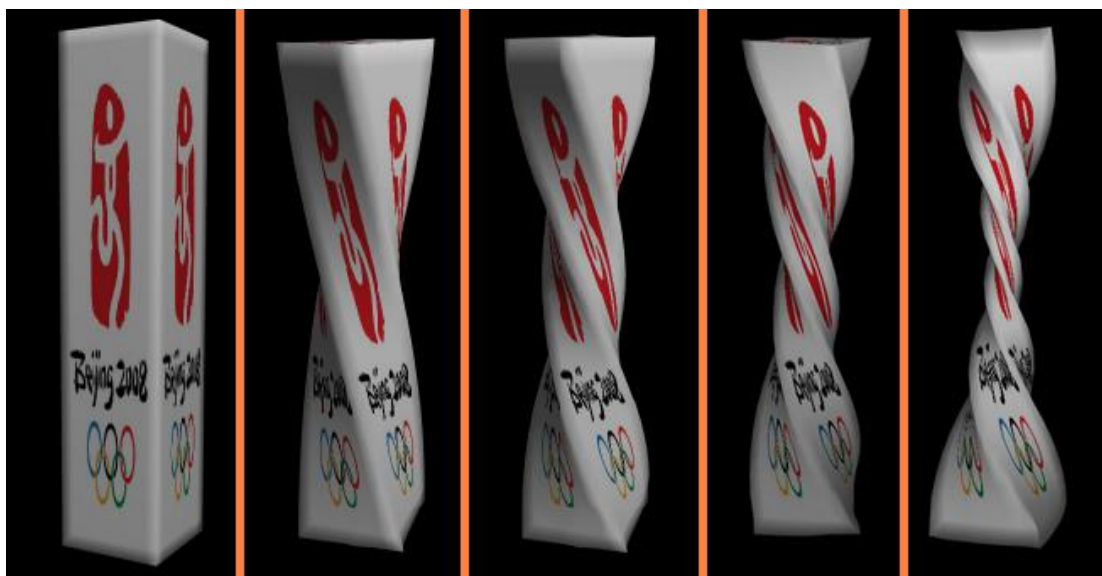


图4-1 多周期旋转也能维持视觉可信度

观察上一章的离散曲面方程式3-16，并联想到曲线的微分几何的重要的Frenet公式，即对弧长参数曲线 $C(s)$ 有：

$$\frac{d}{ds} \begin{pmatrix} T(s) \\ N(s) \\ B(s) \end{pmatrix} = \begin{pmatrix} 0 & \kappa(s) & 0 \\ -\kappa(s) & 0 & \tau(s) \\ 0 & -\tau(s) & 0 \end{pmatrix} \quad (4-1)$$

式4-1中的 $(T(s), N(s), B(s))$ 是曲线的正交Frenet标架，而 $\kappa(s), \tau(s)$ 分别是参数 s 处的曲率与挠率。

若将式3-16中的标架差异算子 δ 视作连续情形下的对弧长参数的导矢算子 $\frac{d}{ds}$ ，那么我们可以认为有下式成立，即：

$$\begin{pmatrix} \delta_j(b_1^i) \\ \delta_j(b_2^i) \\ \delta_j(N^i) \end{pmatrix} \cong \frac{d}{ds} \begin{pmatrix} T(s) \\ N(s) \\ B(s) \end{pmatrix} = \begin{pmatrix} 0 & \kappa(s) & 0 \\ -\kappa(s) & 0 & \tau(s) \\ 0 & -\tau(s) & 0 \end{pmatrix}, e(i, j) \in E \quad (4-2)$$

这样我们可以得出结论，以离散曲面方程导出的优化系统求解出的形变后网格 M' 的标架场所蕴含的曲率与挠率信息在最小二乘意义上与原网格 M 保持尽可能的一致，而曲率与挠率就是局部几何细节的具体表现。因此线性旋转不变的算法框架总是在保持网格上所有曲线段 $e(i, j) \in E$ 各个顶点的曲率和挠率。

4.2 网格曲面的主曲率方向

本节我们沿用上章3.1节的记号。在曲面 S 的切平面 $T_{P_0}S$ 上有向量 $X = x_1 \frac{\partial P}{\partial u_1} + x_2 \frac{\partial P}{\partial u_2} \in T_{P_0}S$ ， n 为法方向， α 是 X 方向与 $T_{P_0}S$ 上 du_1 方向的夹角。称 $\kappa_n(X) = II(X, X) / I(X, X)$ 是在 P_0 点沿 X 方向的法曲率。可见 $\kappa_n(X)$ 是方向 X 的函数，见图4-2。

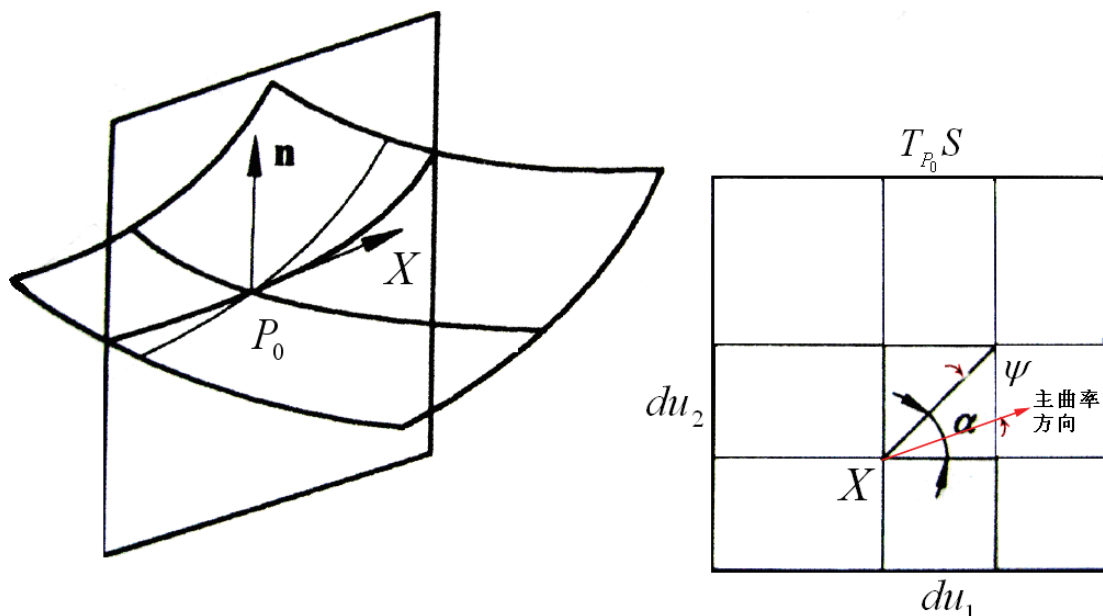


图4-2 切平面上不同的方位有不同的法曲率

更进一步，在微分几何中关于 $\kappa_n(X)$ 有如下的Euler定理：

$$\begin{aligned} \kappa_n(X) &= k_1 \cos^2 \psi + k_2 \sin^2 \psi \\ \kappa_n(X)_{\max} &= k_1, \kappa_n(X)_{\min} = k_2, \end{aligned} \quad (4-3)$$

其中的 k_1, k_2 分别是法曲率 $\kappa_n(X)$ 能取到的极大值和极小值，他们叫做曲面 S 在 P_0 点的两个主曲率，使 $\kappa_n(X)$ 去极值的方向称为主曲率方向，亦称主方向，式中的 ψ 是 X 和极大值主方向的夹角(图4-1)。当 $\psi = 0$ 时， $\kappa_n(X)$ 取得极大值 k_1 ；当 $\psi = \pi/2$ 时， $\kappa_n(X)$ 取得极小值 k_2 。

Euler定理说明，曲面上一点处朝不同方向的曲面曲线的法曲率可由主曲率和 ψ 角唯一确定。主曲率 k_1, k_2 很好的描述了曲面在一点所有方向曲面曲线的弯曲情况，当然也反映了曲面本身的弯曲状态。

4.3 选取主曲率方向建立局部标架

结合前两节的讨论，使我们认为选用每一顶点处的主方向来建立局部标架，而不是原算法^[11]采用的随意使用第一个邻居顶点来生成局部标架能更好的保持曲面上曲线的曲率与挠率。其原因在于，原方法

所保持的对象是曲面上随意走向的曲线，而我们采用极大或极小主方向后，则是保持了曲面上主曲率线的曲率与挠率，主曲率线上每一点的方向都是主方向，且每一点的曲率都是主曲率，因此能比随意方向的曲线更好的反应曲面的弯曲情况。

在具体实现方面，我们简单的采用了比较邻居点间高度角度的方法来选取主方向顶点。我们首先计算了 v_i 的所有邻居顶点离开 T_iM 的高度角 θ ：

$$\theta_j^i = \arccos \left(\frac{\langle \hat{x}_j^i, N^i \rangle}{\|\hat{x}_j^i\|} \right) - \frac{\pi}{2} \quad (4-4)$$

即如图4-3所示。

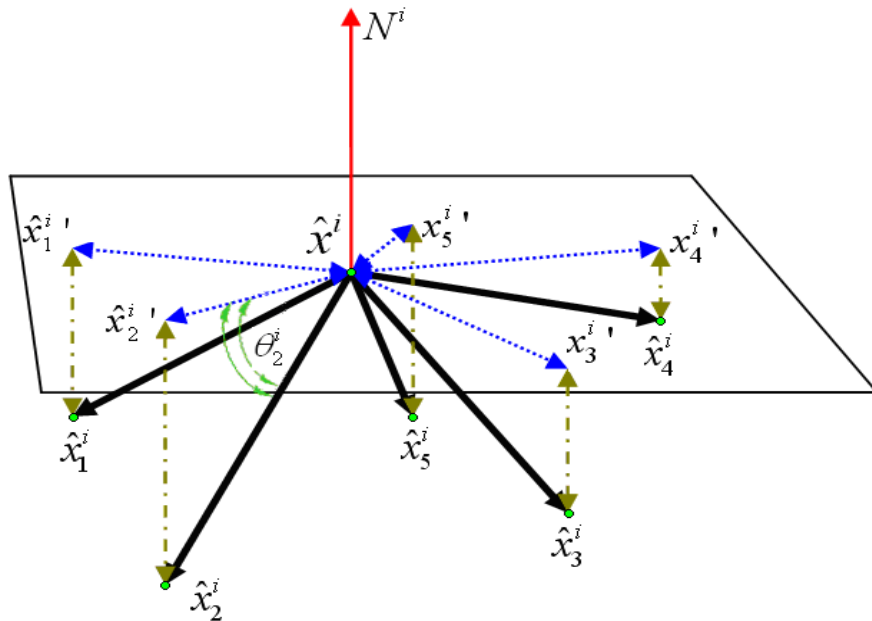


图4-3 邻居顶点离开 T_iM 的高度角 θ

然后我们选取其中最大的一个顶点作为主方向顶点，并以指向它的投影边为标架的 b_1^j ，最终建立正交标准局部标架。

4.4 实验结果与分析

通过实验对比，我们的改进使得算法在大角度如180度的弯折变形中可以生成较为合理自然的效果，如图4-4所示：

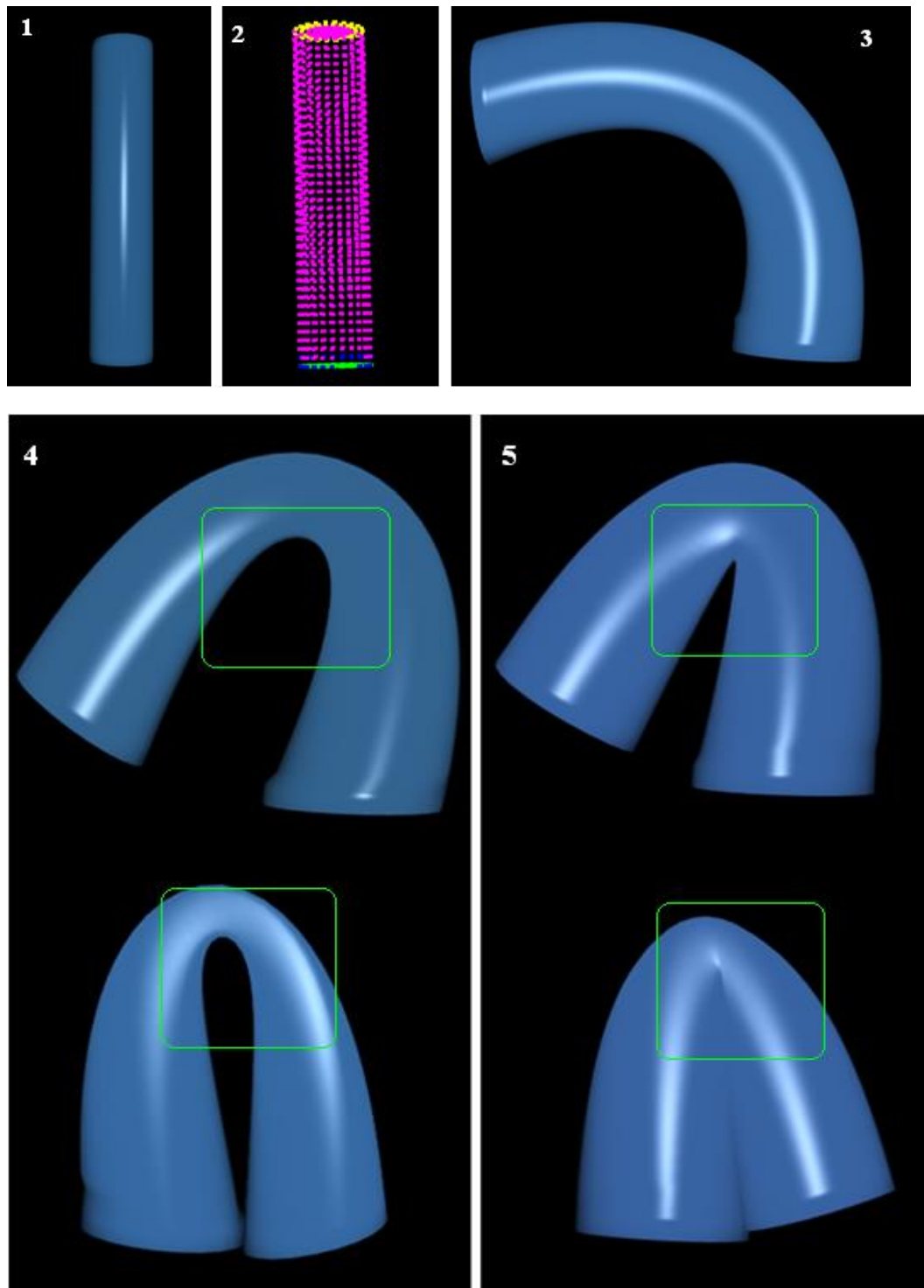


图4-4 圆柱弯折变形实验

(编辑区域是红色部分，顶上的黄色部分是手柄区域(2)； 90度弯折效果(3)，改进后算法变形效果(4)，改进前算法变形效果(5)。注意中间的拱形弯折区域，改进后的效果更为合理。)

另外，我们的改进使得算法在多周期旋转变形中也可以生成比原算法更正确自然的效果。我们以最能体现多周期旋转的长方体扭曲为例，如图4-4所示。我们还展示了更多的结果，如图4-5所示。

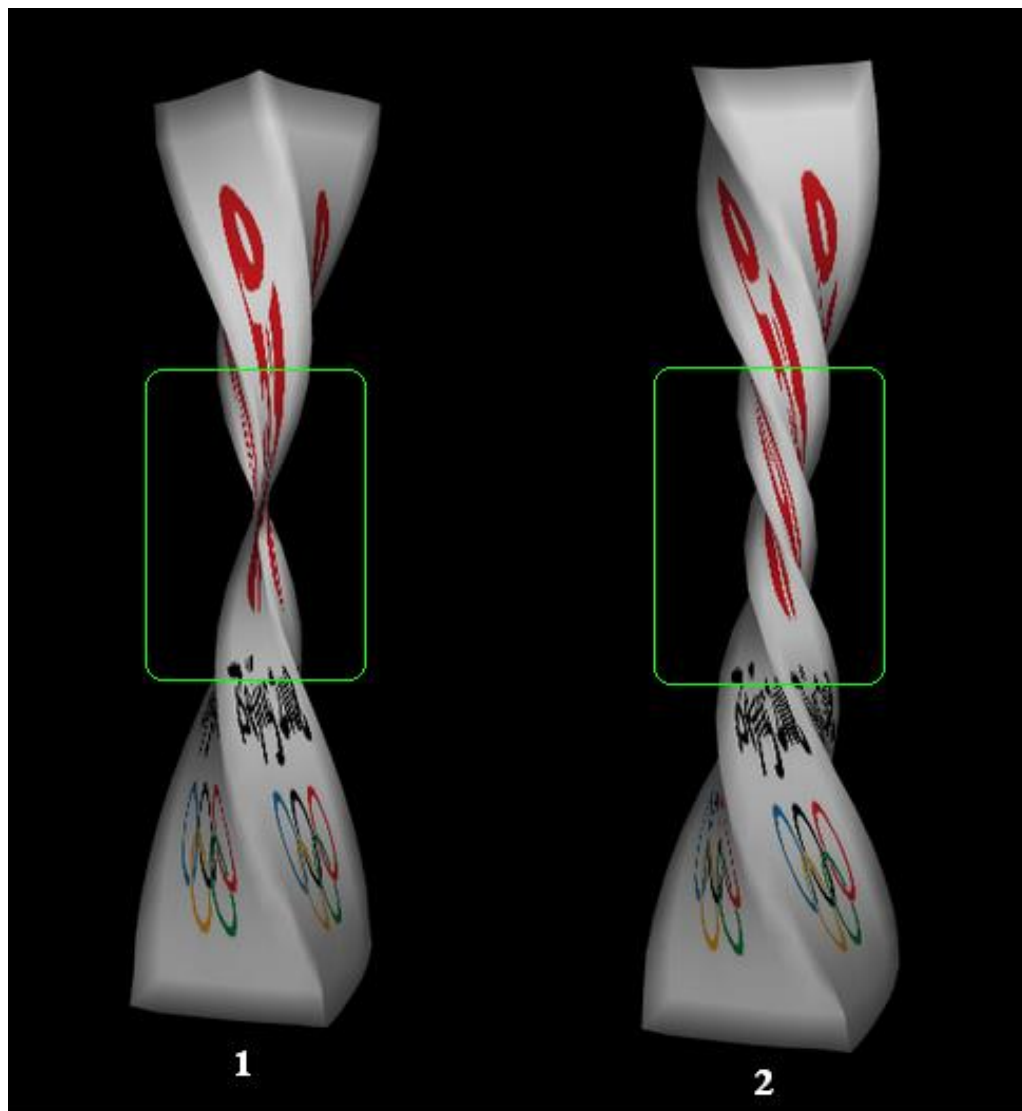


图 4-4 多周期旋转变形实验
(改进前效果(1)，改进后的效果(2)，注意物体中间部分的变形情况。)

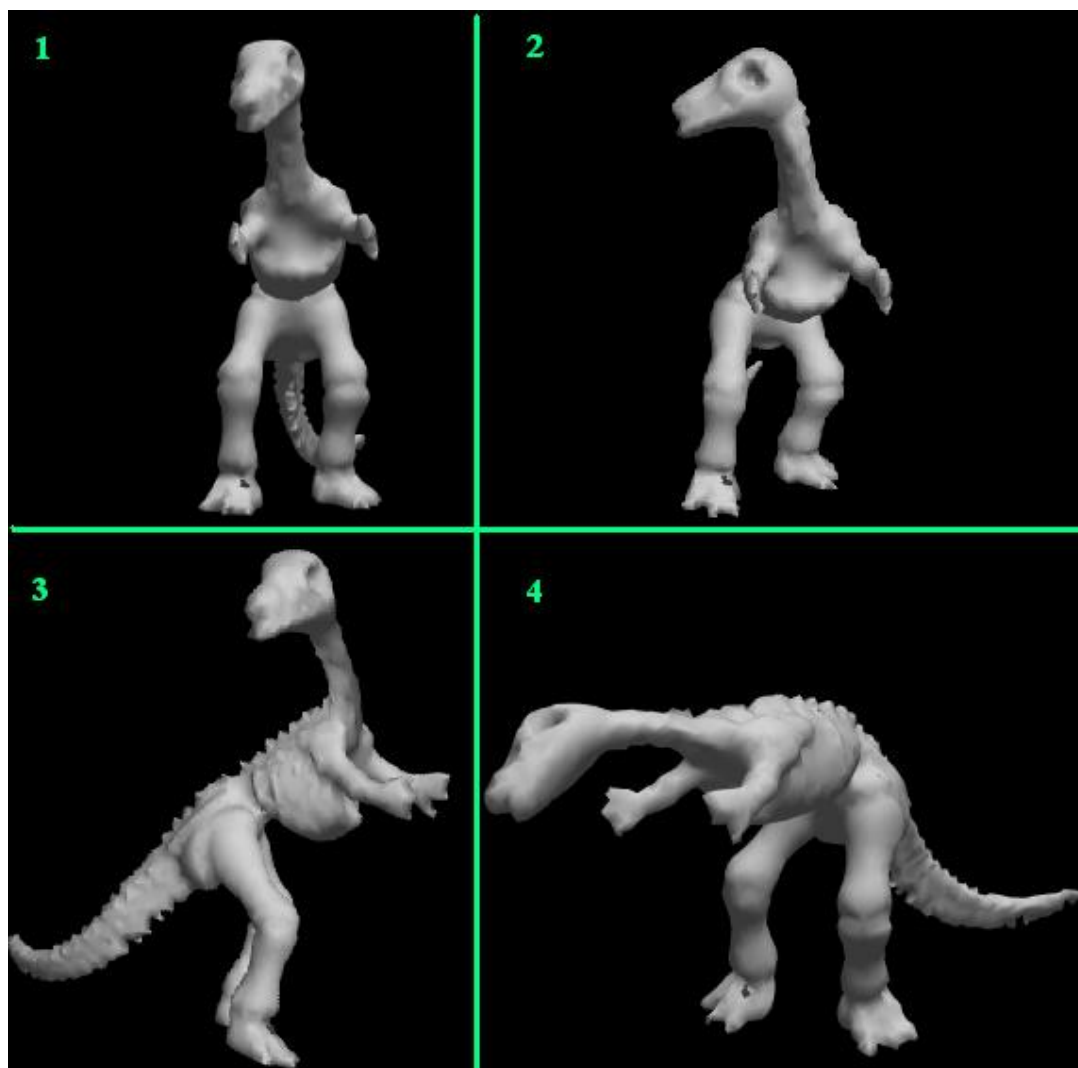


图 4-5 使用我们的算法进行模型变形编辑
(原始恐龙模型(1), 右扭头 90 度(2)(3), 右偏头 90 度(4)。)

结 语

本文着重研究并实现了两种目前较为流行的基于网格曲面微分表示变形算法。

第一个是基于 laplacian 的网格曲面变形算法，我们除了实现了原始的算法，还实现了它的一种优化算法即用变换拟合优化的 laplacian 算法。我们通过实验发现它们存在初始化时间过长甚至不能进行完毕的缺陷，通过设计优化的矩阵储存与运算方式，我们成功的解决了这个问题，提高了算法的可用度。

第二个是基于线性旋转不变微分坐标的变形，本文在认真分析算法的微分几何原理的基础上，提出了一种选取主方向的改进算法，通过实验证明，我们的工作使得算法能够更好的处理大角度多周期的旋转变换，生成更加自然的形变效果

本文在理论上，利用浙江大学刘利刚教授提供的网格文件读取与观察平台，在 Visual C++6.0 MFC 的开发环境实现了以上各算法，进行了大量的实验。证明了本文观点的正确性和可行性。

参考文献

- [1] O. Sorkine, D. Cohen-Or. Least-squares meshes. In Proceedings of Shape Modeling International. IEEE Computer Society Press, 2004. 191-199.
- [2] O. Sorkine, Y. Lipman, D. Cohen-Or, M. Alexa, C. Ross, H. Seidel. Laplacian surface editing. In Proceedings of the Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, ACM Press, 2004. 179 - 188.
- [3] M. Alexa. Local control for mesh morphing. In Proceedings of Shape Modeling International. IEEE Computer Society Press, 2001. 209 - 215
- [4] M. Alexa. Differential coordinates for local mesh morphing and deformation. The Visual Computer, 2003, 9(2): 105 - 112.
- [5] Y. Lipman, O. Sorkine, D. Cohen-Or, D. Levin, C. Ross, H. Seidel. Differential coordinates for interactive mesh editing. In Proceedings of Shape Modeling International. IEEE Computer Society Press, 2004. 181 - 190.
- [6] Y. Lipman, O. Sorkine, M. Alexa, D. Cohen-Or, D. Levin, C. Ross, H. Seidel. Laplacian framework for interactive mesh editing. International Journal of Shape Modeling, 2005, 11(1): 43 - 62.
- [7] P. Perez, M. Gangnet, A. Blake. Poisson image editing. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH), 2005, 22(3): 313 - 318.
- [8] M. Desbrun, M. Meyer, P. Schroder, A. H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In Proceedings of ACM SIGGRAPH, ACM Press/Addison-Wesley Publishing Co., 1999, 317 - 324.
- [9] Y. Yu, K. Zhou, D. Xu, X. Shi, H. Bao, B. Guo, H-Y. Shum. Mesh editing with Poisson-based gradient field manipulation. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH), 2004, 23(3): 644 - 651.
- [10] Dong Xu, Hongxin Zhang, Qing Wang, Hujun Bao. Poisson shape interpolation. In Proceedings of the ACM Symposium on Solid and Physical Modeling. ACM Press, 2005. 267 - 274.
- [11] Y. Lipman, O. Sorkine, D. Levin, D. Cohen-Or. Linear rotation-invariant coordinates for meshes. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH), 2005, 24(3): 479-487.
- [12] A. Sheffer, V. Kraevoy. Pyramid coordinates for morphing and deformation. In Proceedings of the Second International Symposium on 3DPVT (3D Data Processing, Visualization, and Transmission). IEEE Computer Society Press, 2004. 68 - 75.
- [13] Mario Botsch, Mark Pauly, Markus Gross, and Leif Kobbelt. PriMo: Coupled prisms for intuitive surface modeling. In Eurographics/ACM SIGGRAPH Symposium on Geometry Processing. ACM press, 2006. 11 - 20.
- [14] Kun Zhou, Jin Huang, J. Snyder, Xinguo Liu, Hujun Bao, Baining Guo, and Heung-Yeung Shum. Large mesh deformation using the volumetric graph Laplacian. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH), 2005, 24(3): 496 - 503.
- [15] Jin Huang, Xiaohan Shi, Xinguo Liu, Kun Zhou, Li-Yi Wei, Shanghua Teng, Hujun

- Bao, Baining Guo, and Heung-Yeung Shum. Subspace gradient domain mesh deformation. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH)*, 2006, 25(3): 1126 - 1134.
- [16] Lin Shi, Yizhou Yu, Nathan Bell, and Wei-Wen Feng. A fast multigrid algorithm for mesh deformation. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)*, 2006, 25(3): 1108 - 1117.
- [17] I. Guskov, W. Sweldens, P. Schroder. Multi-resolution signal processing for meshes. In *Proceedings of ACM SIGGRAPH*, ACM Press, 1999. 325 - 334.
- [18] Gabriel Taubin. A signal processing approach to fair surface design. In *Proceedings of ACM SIGGRAPH*, ACM Press, 1995. 351 - 358.
- [19] L. Kobbelt, S. Campagna, J. Vorsatz, H. Seidel. Interactive multiresolution modeling on arbitrary meshes. In *Proceedings of ACM SIGGRAPH*, ACM Press, 1998. 105 - 114.
- [20] D. Zorin, P. Schroder, W. Sweldens. Interactive multiresolution mesh editing. In *Proceedings of ACM SIGGRAPH*, ACM Press/Addison-Wesley Publishing Co., 1997. 259 - 268.
- [21] M. Botsch, R. Sumner, M. Pauly, M. Gross. Deformation transfer for detail-preserving surface editing. In *Proceedings of Vision, Modeling and Visualization*, ACM Press, 2006. 357 - 364.
- [22] M. Botsch, L. Kobbelt. Multiresolution surface representation based on displacement volumes. *Computer Graphics Forum (Proceedings of Eurographics)*, 2003, 22(3): 483 - 491, .
- [23] M. Marinov, L. Kobbelt. Automatic generation of structure preserving multiresolution models. *Computer Graphics Forum (Proceedings of Eurographics)*, 2005, 24(3): 479 - 486.
- [24] M. Botsch, L. Kobbelt. A remeshing approach to multiresolution modeling. In *Proceeding of the Euro-graphics/ACM SIGGRAPH Symposium on Geometry Processing*, ACM Press, 2004. 189 - 196.
- [25] D. Terzopoulos, J. Platt, A. Barr, K. Fleischer. Elastically deformable models. In *Proceedings of ACM SIGGRAPH*, ACM Press, 1987. 205 - 214.
- [26] G. Celniker, D. Gossard. Deformable curve and surface finite-elements for free-form shape design. In *Proceedings of ACM SIGGRAPH*, ACM Press, 1991. 257 - 266.
- [27] W. Welch, A. Witkin. Variational surface modeling. In *Proceedings of ACM SIGGRAPH*, ACM press, 1992. 157 - 166.
- [28] M. Meyer, M. Desbrun, P. Schroder, A. Barr. Discrete differential geometry operators for triangulated 2-manifolds. In Hans-Christian Hege and Konrad Polthier, editors, *Visualization and Mathematics III*, Springer-Verlag, 2003. 35 - 57.
- [29] Eitan Grinspun, Yotam Gingold, Jason Reisman, and Denis Zorin. Computing discrete shape operators on general meshes. *Computer Graphics Forum (Proceedings of Eurographics)*, 2006, 25(3): 547 - 556.
- [30] A. Nealen, M. Muller, R. Keiser, E. Boxerman, M. Carlson. Physically based deformable models in computer graphics. *Computer Graphics Forum*, 2006, 25(4):

- 809 - 836.
- [31] G. H. Golub, C. F. Van Loan. *Matrix Computations*. 3rd edition. Baltimore: Johns Hopkins University Press, 1996. 29-78.
 - [32] L. Kobbelt, S. Campagna, J. Vorsatz, H. Seidel. Interactive multi-resolution modeling on arbitrary meshes. In *Proceedings of ACM SIGGRAPH*, ACM Press, 1998. 105 - 114.
 - [33] 孟道骥, 梁科. *微分几何*. 第二版. 北京: 科学出版社, 1999. 10-150.
 - [34] Oscar Kin-Chung Au, Chiew-Lan Tai, Ligang Liu, Hongbo Fu. Dual Laplacian editing for meshes. *IEEE Transactions on Visualization and Computer Graphics*, 2006, 12(3): 386 - 395.
 - [35] M. Botsch, D. Bommes, L. Kobbelt. *Efficient linear system solvers for mesh processing*. Lecture Notes in Computer Science. Springer Berlin/Heidelberg, 2005. 62 - 83.
 - [36] D. S. Meek, D. J. WALTON. On surface normal and Gaussian curvature approximations given data sampled from a smooth surface. *Computer Aided Geometric Design*, 2000, 17(6): 521 - 543.
 - [37] F. Cazals, M. Pouget. Estimating differential quantities using polynomial fitting of osculating jets. In *Proceedings of the Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*. Eurographics Association, 2003. 177 - 187.
 - [38] D. Cohen-Steiner, J. M. Morvan. Restricted delaunay triangulations and normal cycle. In *Proceedings of the 19th annual symposium on computational geometry*, ACM Press, 2003. 312-321.
 - [39] Y. Lipman, D. Cohen-Or, R. Gal, D. Levin. Volume and shape preservation via moving frame manipulation. *ACM Transactions on Graphics*, 2007, 26(1): 105-112.
 - [40] R. Zayer, C. Rossel, Z. Karni, H. Seidel. Harmonic guidance for surface deformation. *Computer Graphics Forum (Proceedings of Eurographics)*, 2005, 24(3): 601 - 609.

附 录

读硕士期间发表的论文：

[1]周世哲,满家巨. 基于多重网格法的实时流体模拟.计算机辅助设计与图形学学, 2007, 07: 121-126.

后 记

三年的时间对于整个人生而言，也许是短暂而微不足道的。但即将过去的这个三年对我而言，却是人生一个重要的里程碑，众多的良师益友给予我生活上的关怀和照顾，科研上的指导和帮助以及思想上的鞭策和鼓励，都将使我终身难忘。

首先，值此论文完成之际，衷心感谢我的导师满家巨副教授这段时间来对我关键的指导和耐心的帮助。论文撰写期间，他一直通过当面指导、发送 E-mail 等方式对我的论文从最初的选题、分析研究、课题的顺利进行到最后的写作都进行了悉心的指导。浙江大学的刘利刚教授以其高瞻远瞩的目光将我带到这个极富挑战性的学术方向上，他不仅让我学到了很多知识，更重要的是使我也学会了许多进行科学研究的方法。两位导师们严谨的治学态度、渊博的知识、深厚的理论功底、敏捷的思路以及踏实的科研和工作作风都给我留下了深刻的印象，也为我树立了终身学习的榜样。在此，谨向我敬爱的两位老师致以最诚挚的谢意！

向湖南师范大学数学与计算机科学学院所有关心和帮助过我的领导、老师表示由衷的谢意！

我还要感谢和我一起学习生活的同学，他们的关心和帮助让我克服了学习生活中一个又一个的困难，希望这份友情能长久保留下去。

最后，衷心地感谢在百忙中抽出时间参加本次论文评阅和答辩的各位专家、教授！

