

分类号: TP391

单位代码: 10335

学 号: 11021029

浙江大学

博士学位论文



中文论文题目: 基于面网格的物理仿真方法
及其在数字化制造中的应用

英文论文题目: Mesh Based Physical Simulation Methods
and Their Applications in Digital Fabrication

申请人姓名: 张译中

指导教师: 周昆 教授

专业名称: 计算机科学与技术

研究方向: 计算机图形学

所在学院: 计算机科学与技术学院

论文提交日期 2016年4月20日

基于面网格的物理仿真方法
及其在数字化制造中的应用



论文作者签名: _____

指导教师签名: _____

论文评阅人1: _____ 隐名评阅

评阅人2: _____ 隐名评阅

评阅人3: _____ 隐名评阅

评阅人4: _____ 隐名评阅

评阅人5: _____ 隐名评阅

答辩委员会主席: _____ 彭群生\教授\浙江大学数学科学学院

委员1: 刘利刚\教授\中国科学技术大学数学科学学院

委员2: 陈彦云\研究员\中国科学院软件研究所

委员3: 刘新国\教授\浙江大学计算机科学与技术学院

委员4: 周昆\教授\浙江大学计算机科学与技术学院

答辩日期: _____ 2016年6月8日

Mesh Based Physical Simulation Methods and Their Applications in Digital Fabrication



Author's signature: _____

Supervisor's signature: _____

External Reviewers: _____
Anonymous

Anonymous

Anonymous

Anonymous

Anonymous

Examining Committee Chairperson:

Qunsheng Peng\Professor\ZJU

Examining Committee Members:

Ligang Liu\Professor\USTC

Yanyun Chen\Researcher\CAS

Xinguo Liu\Professor\ZJU

Kun Zhou\Professor\ZJU

Date of oral defence: June 8, 2016

浙江大学研究生学位论文独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得 浙江大学 或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文作者签名：

签字日期： 年 月 日

学位论文版权使用授权书

本学位论文作者完全了解 浙江大学 有权保留并向国家有关部门或机构送交本论文的复印件和磁盘，允许论文被查阅和借阅。本人授权 浙江大学 可以将学位论文的全部或部分内容编入有关数据库进行检索和传播，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

(保密的学位论文在解密后适用本授权书)

学位论文作者签名：

导师签名：

签字日期：

年 月 日

签字日期：

年 月 日

摘要

物理仿真基于真实世界物理规律，在计算机中虚拟地重现流体运动、弹性体形变等各种物理现象的过程。由于物理仿真能够达到很高的精度，而成本比物理实验要低，因此被广泛应用于航空航天、机械制造、生物医药等诸多领域的前期研究。影视制作中许多难以实际拍摄的场景，也是通过对流水、风沙、火焰等现象进行物理仿真，并结合高质量的渲染器，制作出的视觉特效。近年来，随着3D打印技术的发展和普及，数字化制造再次成为前沿的研究领域。将物理仿真应用于3D打印，对模型进行结构力学分析，可以及时发现设计缺陷，保证制造出模型的结构稳定；通过运动学分析，可以制造出具有特殊运动规律的模型。物理仿真的应用极大提高了数字化制造的水平。

面网格在物理仿真中有着非常重要的作用。首先，薄层结构的物体（如纸张、布料、板材、气泡等）通常都是用面网格来直接表示。其次，面网格可以表示模型的表面形状，因此被广泛用于碰撞检测以及场景的实时显示。对于流体等无固定形状的物质，还可以用动态面网格追踪其自由液面。尽管自然界中真实物体都具有一定的体积，但是体网格拓扑结构和计算往往都比较复杂。因此，如果能用面网格对计算模型进行近似，就可以显著减少运算量，提高仿真速度。

本文用面网格对水滴、粘性流体薄膜、塑性薄膜等三种物质进行了建模和仿真。水滴体积很小，其形状主要受表面张力的影响，因此可以忽略水滴内部的流体运动，通过面网格的直接形变进行仿真。水滴之间的融合与分裂，可以通过面网格的布尔运算及网格优化来实现。粘性流体薄膜和塑性薄膜都是薄层结构，因此用面网格可以很好地对其进行仿真。我们将其仿真技术应用于两种传统制造工艺的虚拟化：水转印和热塑成型，提出“可计算水转印刷”和“可计算热塑成型”的方法，使其能够用于三维曲面全彩着色。我们研制了这些新型工艺的原型系统，并通过实验证明了其实用性和可靠性。

本文的主要贡献如下：

- 提出了一种基于面网格的实时水滴仿真方法。通过对水滴表面网格直接形变，对水滴的运动、水滴与亲水表面接触，以及水滴在固体表面滑动的现象进行仿真。该方法把流体仿真从三维体素简化为面网格的形变，从而使计算量大幅减少，不仅能够

实时仿真，还可以让用户与水滴进行交互。

- 通过面网格对粘稠流体薄膜漂浮在水面上的现象进行仿真，模拟物体经过薄膜浸入水中，薄膜被拉伸并贴在物体表面的过程。我们将该现象的仿真应用于曲面着色工艺——水转印的虚拟化流程，提出“可计算水转印刷”的方法，解决了传统水转印工艺中对复杂曲面着色时，图案与模型难以精确对齐的问题。我们搭建了计算水转印的原型系统，实现了虚拟仿真、系统标定、运动控制的集成。针对复杂模型水转印过程中，水转印膜拉伸过大导致颜色失真的缺陷，我们提出了多次转印的方法。每次只在一个方向对模型局部着色，多次转移之后图案相叠加，在模型表面上共同形成目标纹理。
- 通过面网格对塑性薄膜进行建模，模拟软化塑料片在大气压力下产生形变，并贴在物体表面的过程。我们将该现象的仿真应用于塑料成型工艺——真空热塑成型的虚拟化流程，提出“可计算热塑成型”的方法，并结合3D打印，将带纹理的虚拟数字化模型制作成为实物模型。首先，将数字化模型作为模具，对真空成型的过程进行虚拟仿真。根据塑料片形变的仿真结果，计算出预形变图案，将其打印在透明塑料片上。通过对抽真空过程的仿真，找出塑料片与模具之间可能出现的气体空腔，据此在模型上设置排气孔，并通过3D打印制作出带有排气孔的模具。我们对一台小型真空成型机进行了改造，使塑料片上的图案能够与物体精确对齐。经过真空成型，打印的纹理就贴在了模具的表面，同时塑料片也为图案覆盖了一层透明保护壳，最终得到与数字化模型相同的实物模型。

关键词：流体仿真，表面张力，平均曲率流，曲面着色，3D打印，粘性薄膜，水转印，塑性薄膜，热塑成型，纹理映射

Abstract

Physically based simulation is a technique to virtually reproduce natural phenomena of the real world in computers based on physical principles. This technique is widely used in many fields, including aerospace/mechanical engineering, and medical science, due to its high precision as well as low cost. Together with physically based rendering, physically based simulation can be used for making special effects in movies, such as flood, hurricane and fire. With the development of 3D printing in recent years, the research topic of digital fabrication has attracted much attention. Physically based simulation can improve the quality of 3D printed products. By performing stress analysis before printing, we can make the structure of the printed model more stable. By mechanical analysis, we can even print models whose physical property will satisfy certain constraints.

Surface mesh plays an important role in simulation. First, thin shell objects (i.e. paper, cloth, shell, bubbles) are usually represented by surface mesh directly. Second, the shape of most objects are represented by surface meshes, which are widely used for collision detection and real-time rendering. Third, dynamic mesh can be used to track the surface of fluid, which has no static shape. Although all objects in real world possess finite volumes, the topology and computation of volumetric mesh are usually very complex. If the computational model can be approximated with surface mesh, the computational cost can be greatly reduced, thus improving the performance of simulation.

In this thesis, we modeled three kinds of materials: water droplet, viscous sheet and plastic sheet, using surface mesh. The shape of water droplet is mainly influenced by surface tension, so we can simulate the motion of water droplet using a deformable surface model while ignoring the water flow inside. Merging and splitting of water droplets can be simulated through Boolean operations and mesh optimization. Surface mesh is very suitable for the simulation of viscous sheet and plastic sheet, as they are both thin shells. We apply these simulation methods in the virtualization of two traditional fabrication techniques: hydrographics and thermoforming,

proposing "Computational Hydrographic Printing" and "Computational Thermoforming", which can be used for 3D surface coloring. We built prototype systems of these new techniques, and validated their usability and robustness through physical experiments.

The contributions of this thesis are as follows.

- We proposed a mesh based real-time water droplet simulation method. Using a deformable surface model, this method is able to simulate droplet motion, the hydrophilic effect, and droplet flowing on solid surfaces. This method reduced water droplet simulation from 3D volume to surface meshes, thus greatly reduced computational cost, making the whole system real-time and allows user interaction.
- We model the viscous sheet floating on water using surface mesh, and simulate the phenomenon of viscous sheet stretches and wraps around the object as it is immersed into the water. We apply this simulation in the virtualization of hydrographics, which is a traditional surface coloring technique. We proposed "Computational Hydrographic Printing" method, which solved the problem of precise alignment between color pattern and 3D model. We built a prototype system using off-the-shelf hardware, which combined virtual simulation, system calibration and controlled immersion. We further extended this technique to multiple immersions to solve the problem of color deviation due to large stretch. In each immersion, we only color a certain part of the model, and finally get a seamless texture after immersing in all directions.
- We model plastic sheet using surface mesh, and simulate its stretching under air pressure, which finally adhere to the mold surface. We apply this simulation in the virtualization of thermoforming, which is widely used in industry to make thin shell plastic products. We proposed "Computational Thermoforming" method, which is able to make digital 3D models into physical objects, with the help of 3D printing. First, we simulate the thermo-forming process, taking the digital 3D model as mold. We calculate a pre-distorted pattern according to the stretch of the simulated plastic sheet, and print it on the transparent plastic sheet physically. By simulating the venting process, we detect potential air pockets between the plastic sheet and mold, setting vacuum holes accordingly, and print vented mold using 3D printing. We improved a small-scale vacuum forming machine, making it possible to align pattern on plastic sheet with mold. After vacuum forming, the printed

pattern will adhere to the mold surface, forming a physical reproduction of the digital model. The plastic sheet will give the pattern a protective layer.

Keywords: Fluid Simulation, Surface Tension, Mean Curvature Flow, Surface Coloring, 3D Printing, Viscous Sheet, Hydrographics, Plastic Sheet, Thermoforming, Texture Mapping

目 次

摘要	I
Abstract	III
目次	
插图	VIII
第1章 绪论	1
1.1 图形学中的物理仿真	1
1.2 数字化制造概述	7
1.3 研究现状	9
1.4 现有方法优缺点分析	21
1.5 本文工作	23
第2章 基于面网格的实时水滴仿真方法	25
2.1 问题的提出	25
2.2 算法概述	27
2.3 网格形变运算	28
2.4 网格拓扑运算	34
2.5 实验结果	38
2.6 小结	42
第3章 粘性流体薄膜仿真与可计算水转印刷	43
3.1 问题的提出	43
3.2 算法概述	47
3.3 虚拟仿真	48
3.4 物理实现	54
3.5 扩展：多次转印	56
3.6 实验结果	59
3.7 小结	63

第4章 塑性薄膜仿真与可计算热塑成型	65
4.1 问题的提出	65
4.2 算法概述	67
4.3 热塑成型过程仿真	68
4.4 排气孔布局	76
4.5 原型着色系统	80
4.6 实验结果	83
4.7 小结	85
第5章 总结和展望	87
5.1 本文工作总结	87
5.2 进一步工作展望	88
参考文献	91
攻读博士学位期间主要研究成果	103
致谢	104

插 图

1.1	图形学中物理仿真主要内容	2
1.2	物理仿真的离散化方法	5
1.3	物理仿真主要对象的分类	6
1.4	商用彩色3D打印产品	8
1.5	刚体运动仿真	9
1.6	弹性体仿真	10
1.7	塑性体仿真	12
1.8	粘稠流体仿真	13
1.9	液体仿真	14
1.10	气体仿真	16
1.11	表面张力主导的流体仿真	17
1.12	多相耦合仿真	18
1.13	特定外观产品制造的最新进展	19
1.14	弹性体应力分析在3D打印中的应用案例	20
1.15	刚体运动仿真在3D打印中的应用案例	21
2.1	自然界中的水滴	26
2.2	水滴实时仿真算法流程图	27
2.3	平均曲率流	29
2.4	边对角的定义	30
2.5	不同亲水性表面上水滴的稳定形状	31
2.6	接触线示意图	31
2.7	平均曲率流运算过程中的体积损失	32
2.8	局部体积修正与全局体积修正效果对比	34
2.9	网格优化	35

2.10 自适应网格	35
2.11 网格切分	36
2.12 网格融合	38
2.13 水滴相撞仿真结果与物理实验对比	39
2.14 水滴落在亲水材料表面仿真结果与物理实验对比	39
2.15 不同表面张力系数的水滴静止效果	40
2.16 水柱截断效果	40
2.17 数百个水滴落在玻璃板上的效果	41
2.18 水滴落在荷叶上的效果	41
2.19 水滴落在小鸟模型上的效果	42
3.1 打印了图案的聚乙烯醇（PVA）膜	44
3.2 传统水转印操作流程	45
3.3 传统水转印工艺着色的产品	45
3.4 传统水转印与计算水转印适用范围对比	46
3.5 计算水转印系统流程图	47
3.6 水转印中，PVA膜被拉伸并贴在物体表面的示意图	48
3.7 粘性薄膜的边界条件	49
3.8 粘性薄膜的计算区域	50
3.9 颜料对薄膜拉伸程度的影响	52
3.10 计算水转印设备	55
3.11 多次转印PVA膜在各转印方向上的拉伸率	57
3.12 多次转印每个方向的转印图案计算过程	58
3.13 水转印可重复性验证	60
3.14 非线性粘性薄膜模型与斯托克斯流模型结果对比	60
3.15 水转印精度验证	61
3.16 大象杯-陶瓷表面的水转印实验	62
3.17 地球仪-塑料及木材表面的水转印实验	62
3.18 两次转印实验	63
3.19 三次转印实验	63
3.20 失败案例-图案在大拉伸下会失真	64
4.1 真空热塑成型原理	66

4.2 常见的真空成型产品	66
4.3 真空热塑成型3D模型着色系统流程图	67
4.4 热塑性材料应力-应变曲线及简化模型	68
4.5 弯曲应力对预形变图案计算的影响	69
4.6 塑料片受力平衡示意图	74
4.7 凹陷模型的大拉伸实验	75
4.8 排气孔设置对真空成型结果的影响	76
4.9 气体空腔的检测方法	77
4.10 气体空腔森林的构造方法	78
4.11 排气孔结构剖面图	79
4.12 真空成型设备示意图	80
4.13 真空成型的过程	82
4.14 真空成型精度实验	84
4.15 通过真空热塑成型制做的彩色模型	85

第1章 绪论

物理仿真就是用计算机对真实世界中的各种现象进行再现的过程。无论何种复杂的现
象，只要找到其本征的物理规律，进行数学建模，然后再用数值方法进行求解，都可以在
一定精度范围内对其进行仿真。大到模拟航天器运行轨迹、模拟大气运动进行天气预报，
小到设计手机的外壳、分析零件结构强度，物理仿真已经影响着人类生活的方方面面，成
为工程技术发展的重要基础。

数字化制造是随着信息技术发展而诞生的一种先进制造技术。计算机设计的数字化模
型，能够通过减材或增材的方法直接加工成型，将设计和制造紧密结合在一起。尤其是
以3D打印为代表的增材制造技术，近几年发展非常迅速。3D打印通过材料逐层堆积的方
式，对于复杂形状，甚至带有封闭空腔的模型都能用统一的方法来制造，因此将数字化制
造推向新的高度。

将物理仿真引入数字化制造技术，会极大促进数字化制造的发展。通过对产品结构、
使用材料、生产流程等进行物理仿真，可以在实际生产之前就对最终产品进行有效的评
估，从而极大地缩短产品开发周期，降低成本。尤其是随着3D打印的普及，以及面向普通
用户设计理念的推广，将进一步把数字化制造推向“个人制造”的新阶段。

本文用面网格对水滴、粘性流体薄膜、塑性薄膜等三种物质进行了建模和仿真。首先
提出了一种基于面网格的实时水滴运动仿真方法，用面网格直接形变模拟水滴运动，并通
过布尔运算及网格优化实现水滴之间的融合与分裂。然后，本文用面网格对薄层结构的粘
性流体薄膜和塑性薄膜进行了仿真，并将其应用于两种传统制造工艺的虚拟化：水转印和
热塑成型。从硬件和软件两方面对这两种工艺进行改进，使其能够用于三维曲面全彩着
色，符合数字化制造的要求，并通过实验证明了其实用性和可靠性。

1.1 图形学中的物理仿真

在工业中，物理仿真几乎被用于各个领域，也有许多非常先进的工业级仿真软件，比
如用于结构力学分析的ANSYS，用于流体仿真的Fluent，用于电磁场仿真的ADS等，可以进

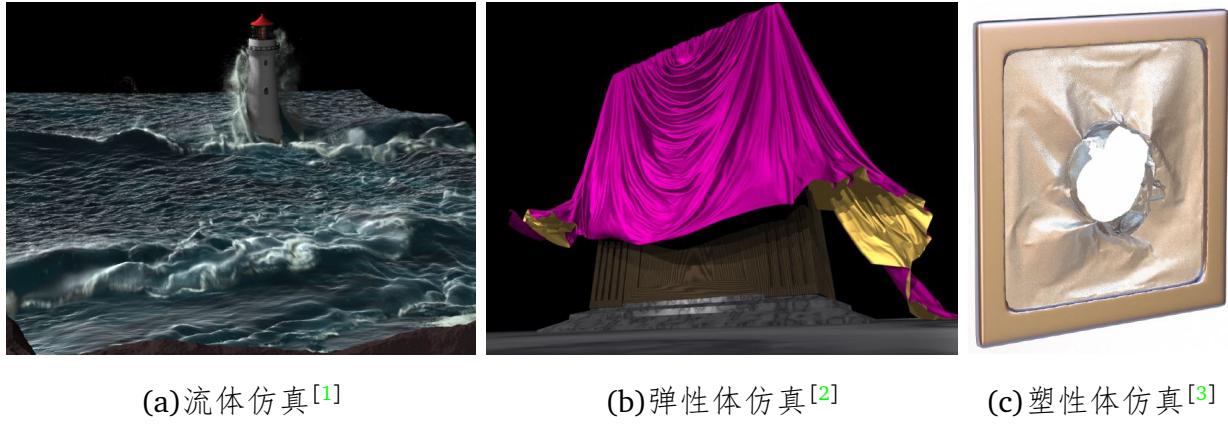


图 1.1 图形学中物理仿真的主要内容

行非常精确的工程模拟。在图形学中，物理仿真则更加偏向于视觉效果的正确，同时是对计算的效率和开销要求更高，在游戏当中甚至要求能够进行实时仿真。

在图形学中，研究较为广泛的方向包括流体、布料、弹性体等，如图1.1所示。除此之外，还有模拟声音传播、磁性物体、刚体碰撞、物体破碎等。为了提高仿真速度，图形学中通常都会在保证视觉效果的前提下对物理模型进行充分的简化。

1.1.1 物理仿真的理论基础

图形学中物理仿真的理论基础是连续介质力学，其仿真对象多是宏观物体。这类宏观物体基本运动规律可以通过微分方程的形式来描述^[4]：

$$\dot{\mathbf{x}} = f(\mathbf{x}, t) \quad (1.1)$$

其中 f 是被仿真对象的状态方程，通常是系统状态 \mathbf{x} 以及时间 t 的函数，而 $\dot{\mathbf{x}}$ 则是 \mathbf{x} 对时间 t 的微分。满足方程1.1的问题称为初值问题，即给定该系统在初始时刻 t_0 的状态 $\mathbf{x}(t_0) = \mathbf{x}_0$ ，就能够通过求解方程1.1，对这个系统在整个时间序列上的状态进行计算。物理仿真就是根据具体问题对方程1.1进行实例化，并求解的过程。

1.1.1.1 前向欧拉法

如果方程1.1能够显式的写出函数表达式，那么就可以用标准的微分方程求解方法对整个系统进行仿真。但是多数复杂系统都采用离散化表达，因此方程1.1也用数值方法来计算。最简单的求解方法为前向欧拉法，即

$$\mathbf{x}(t_0 + h) = \mathbf{x}(t_0) + h\dot{\mathbf{x}}(t_0) \quad (1.2)$$

其中, h 表示迭代计算的时间步长。当系统在 t_0 时刻的状态 $\mathbf{x}(t_0)$ 及状态的变化速度 $\dot{\mathbf{x}}(t_0)$ 已知, 就可以根据公式1.2对下一时刻($t_0 + h$)的状态进行计算。重复此过程, 从而实现整个系统的物理仿真。

前向欧拉法的优点是公式简单, 按照其定义就能够直接求解, 但一个众所周知的缺陷是前向欧拉法数值求解不稳定^[5]。能满足收敛条件的迭代步长 h 与系统离散化的精度有关, 离散化精度越高步长就要越小。对于高精度的物理仿真, 其计算效率就会非常低。因此计算无条件收敛的反向欧拉方法成为了物理仿真的主流方法。

1.1.1.2 反向欧拉法

反向欧拉方法又称隐式方法, 其公式表示为:

$$\mathbf{x}(t_0 + h) = \mathbf{x}(t_0) + h\dot{\mathbf{x}}(t_0 + h) \quad (1.3)$$

与公式1.2的区别在于微分项采用的是($t_0 + h$)时刻的微分。这一改动意味着需要对下一时刻系统的状态进行总体评估, 使得其状态及状态的微分都刚好符合系统的要求。正是由于这一改动, 使得计算无条件收敛, 因此无论离散化精度有多高, 都可以通过大步长来计算, 从而大大提高了仿真速度。不过这一改动使得求解变得复杂, 通常要求 f 是一个线性系统, 然后通过求解线性方程组计算出下一时刻的状态。

参照Baraff^[5]的表述, 用 \mathbf{X}_{old} 表示上一时刻的状态向量, \mathbf{X}_{new} 表示经过时间 h 之后系统待求解的状态向量, 则公式1.3表示为

$$\mathbf{X}_{new} = \mathbf{X}_{old} + hf(\mathbf{X}_{new}) \quad (1.4)$$

定义 $\Delta\mathbf{X} = \mathbf{X}_{new} - \mathbf{X}_{old}$, 于是公式1.4可以写为

$$\mathbf{X}_{old} + \Delta\mathbf{X} = \mathbf{X}_{old} + hf(\mathbf{X}_{old} + \Delta\mathbf{X}) \quad (1.5)$$

即

$$\Delta\mathbf{X} = hf(\mathbf{X}_{old} + \Delta\mathbf{X}) \quad (1.6)$$

对 $f(\mathbf{X}_{old} + \Delta\mathbf{X})$ 通过一阶泰勒展开来近似，

$$f(\mathbf{X}_{old} + \Delta\mathbf{X}) \approx f(\mathbf{X}_{old}) + f'(\mathbf{X}_{old})\Delta\mathbf{X} \quad (1.7)$$

将其带入公式1.6，得

$$\Delta\mathbf{X} = h[f(\mathbf{X}_{old}) + f'(\mathbf{X}_{old})\Delta\mathbf{X}] \quad (1.8)$$

将其合并同类项，并写成矩阵的形式，于是可以算出每一步迭代系统状态的增量：

$$\Delta\mathbf{X} = \left[\frac{1}{h}\mathbf{I} - f'(\mathbf{X}_{old}) \right]^{-1} f(\mathbf{X}_{old}) \quad (1.9)$$

最终整个系统的状态更新为

$$\mathbf{X}_{new} = \mathbf{X}_{old} + \Delta\mathbf{X} \quad (1.10)$$

从整个推导过程可以看出，反向欧拉法物理仿真的核心是公式1.9，即求解每一步迭代系统状态的增量。这一步计算要求解整个线性系统，因此每一步迭代比前向欧拉要复杂得多。但是对于多数问题， f' 是稀疏矩阵，公式1.9的计算可以在线性时间内完成。相反，由于整个线性系统的求解是无条件稳定的，可以采用远大于前向欧拉法的时间步长，因此总的计算效率比前向欧拉法要高。

1.1.1.3 静态平衡问题

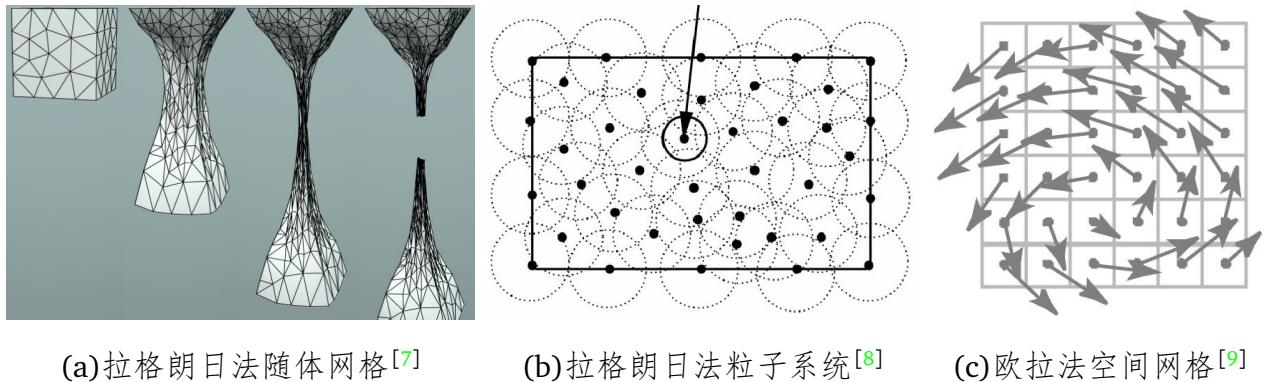
对于一类特殊的问题，如果将 h 进一步增大， $h \leftarrow \infty$ ，于是公式1.9简化为

$$\Delta\mathbf{X} = -[f'(\mathbf{X}_0)]^{-1} f(\mathbf{X}_0) \quad (1.11)$$

其中 \mathbf{X}_0 是整个系统最初初始的状态向量。这一类问题就被称为静态平衡（Quasi-static）问题，它计算的是整个系统经过无限长时间最终达到平衡状态。静态平衡问题在热力学分析当中应用非常广泛，比如计算带有热源系统达到热平衡状态时的最终温度分布。

1.1.2 物理仿真的离散化方法

在1.1.1节中给出了物理仿真的理论基础及求解方法，系统状态 \mathbf{x} 是空间维度， t 是时间维度，物理仿真的实质是对微分方程 $\dot{\mathbf{x}} = f(\mathbf{x}, t)$ （公式1.1）在空间-时间上进行求解。对



(a)拉格朗日法随体网格^[7] (b)拉格朗日法粒子系统^[8] (c)欧拉法空间网格^[9]

图 1.2 物理仿真的离散化方法 根据被仿真对象的物理性质，采用合适的离散化方法，能够显著提高仿真速度和精度。**(a)**拉格朗日法随体网格通常用于弹、塑性体物理仿真，网格顶点可以作为物理参数定义的坐标，通过网格连接关系求解微分方程。模型形变过大会使网格质量变差，就需要对网格进行优化，以保证求解的稳定。**(b)**拉格朗日法粒子系统通常用于流体仿真，物理参数定义在粒子上，由于粒子可以自由运动，求解微分方程需要对搜索临近粒子。**(c)**欧拉法空间网格通常用于流体仿真，物理参数定义在固定的空间坐标上。

于复杂的实际物理系统，微分方程求解需要对其离散化为 $\mathbf{X}_{new} = \mathbf{X}_{old} + h f(\mathbf{X}_{new})$ （公式1.4），其中状态向量 \mathbf{X} 是对空间的离散化，时间步长 h 是对时间进行离散化。所以一个实际的物理系统通常是连续定义，离散求解。

在连续介质力学中，对物体的运动有两种描述方法：拉格朗日法和欧拉法^[6]。在拉格朗日法中，物体的物理参数定义在物体的局部坐标系上，并随着物体一起运动，因此也称随体坐标系。通俗的理解就是将物体切分成许多质点，无论物体如何形变，物理参数一直都直接定义在对应的质点上。欧拉法中，物理参数则是定义在世界坐标系上，表示的是当前空间中固定一点的物理状态。通俗的说，无论物体如何运动，都只是对当前刚好经过采样点的那一点物质物理状态的描述。拉格朗日法和欧拉法理论上是等效的，可以相互转化，但在求解实际物理过程时，采用合适的方法会使问题得到简化。

对应于拉格朗日法和欧拉法，对连续空间的离散化也分为按物体体积离散和按空间坐标系离散，如图1.2所示。直观上讲，拉格朗日法就是把物体体积切成很多小块，并且将每一小块的质量集中在一个质点上，构成一个粒子系统。对于不存在固定结构的物体（如流体），就直接作为粒子系统求解，求解微分方程时需要对每个粒子周边一定范围内进行搜索，如图1.2(b)所示。如果物体存在固定的结构（如弹性体），即粒子之间存在固定连接关系，那么就对整个物体的体积生成网格，作为其离散方法，如图1.2(a)所示。微分方程就通过网格的邻接关系来求解。如果网格变形非常剧烈（如被剧烈拉伸的塑性体），那么还需要对网格进行优化，从而减小计算误差。由于宏观物体都是由微观粒子（分子、原

子) 构成, 从这个意义上说, 拉格朗日法是对物体建模的最本质方法, 因此其适用范围非常广, 从固体到流体都可以仿真。

但是, 用拉格朗日法粒子系统求解微分方程时, 粒子之间的插值误差会非常大, 因此会极大地降低计算精度。针对流体仿真, 欧拉法的计算会更为精确。欧拉法是按空间坐标系离散, 就是对世界坐标系划分网格, 物理参数直接从空间中的固定位置采样, 如图1.2(c)所示。直观上讲, 欧拉法描述的是当前这一时刻, 刚好经过空间中采样点物质微粒的物理参数; 不同时刻采样点位置相同, 但观察到的可能已经不是原来的物质微粒。欧拉法求解微分方程在空间网格中插值, 因此相比用粒子差值精度要高很多。如果物体并不是充满整个欧拉网格, 比如存在自由液面的水, 则需要显式地表达物体所占网格, 并对其施加合适的边界条件。

对固体进行仿真时, 通常用拉格朗日法将固体的体积离散化为四面体、或六面体网格。对于薄壳状物体, 可以将其简化为平面三角网格、或四边形网格, 将三维结构降到二维, 从而大大降低计算复杂度。对于薄膜状流体(如气泡), 一般也用三角网格来表达。在欧拉法流体仿真中, 面网格也可以用来追踪流体的自由液面, 并且通过面网格来对流体施加表面张力。

1.1.3 物理仿真的实例化

由于各种真实感电影特效的大量需求, 计算机图形学中物理仿真的研究范围非常广泛, 几乎涵盖了日常生活的方方面面。从宏观上说, 我们生活中接触到的物体可以分为刚体、弹性体、塑性体、流体四大类。其实自然界中物质形态多种多样, 物理性质也都是连续变化的(如图1.3所示), 只是在进行物理仿真时, 根据物体的基本形态, 选择合适的模型来描述。对于形变时表面形状变化不大的物体, 一般使用静态网格来表达。而对于表面形状变化剧烈的物体(如塑性体), 静态网格已经不足以描述其形状, 此时就需要动态网

仿真对象:	陶瓷	橡胶	黏土	膏状物	水	空气
物质形态:	刚体	弹性体	塑性体	流体		
网格:	静态网格		动态网格		欧拉网格	

图 1.3 物理仿真主要对象的分类^[10] 自然界中的物质形态多种多样, 其物理性质也趋向于连续变化, 因此它们之间并没有明确的界限。我们宏观上将其分为固体、弹性体、塑性体、流体四大类。对于形变时表面形状变化不大的固体、弹性体, 通常用静态网格进行仿真。如果塑性体形状变化比较大, 就需要使用动态网格, 避免因网格质量下降导致数值计算不稳定。由于流体没有固定的形状, 因此多使用欧拉法或拉格朗日粒子系统对其进行仿真。

格。而没有固定形状的流体，则需要用欧拉法或者拉格朗日粒子系统来仿真。根据仿真对象的物理性质，公式1.1被实例化为不同的物理方程，如刚体运动方程、弹性方程、流体Navier-Stokes方程等，对方程进行迭代求解，最终完成物理仿真。

1.2 数字化制造概述

数字化制造是通过CAD软件进行设计，通过计算机控制的减材或增材方法进行加工，将设计和制造紧密结合起来的一种先进制造理念。数控机床采用减材制造，目前技术已经比较成熟，但设备多是大型机器，主要还是工厂化生产。3D打印技术则采用增材制造，近几年发展非常迅速，机器非常容易小型化，甚至已经进入了普通家庭，使得数字化制造进入了“个人制造”时代。

在数控加工技术大规模应用之前，复杂形状模型只能由工匠徒手雕刻，成本很高，每件产品都算是一件艺术品。随着计算机数字控制技术（CNC）的发展，用数控机床完成复杂曲面的雕刻，使得复杂模型制造从艺术变为技术，大大降低了成本。尤其是近年来日趋成熟的3D扫描技术，打通了实物模型与虚拟模型之间的边界。将实物模型通过3D扫描成为数字化模型，然后就可以用数控加工技术大规模复制，于是才有了我们今天产品种类极其丰富的世界。

然而采用减材成型技术的数控加工仍然有一定的局限性。首先，尽管大规模生产降低了单件产品的成本，但是如果只进行小批量、甚至单件产品生产，其成本依然很高。其次，即便是小型的数控机床价格依然非常昂贵，需要专门的技术人员进行操作，因此小批量生产都是交给专业工厂来完成，生产周期都比较长（通常需要数天甚至数月）。数控加工产品只能是单一材料的硬质物体，曲面形状也会受机器自由度的限制（比如三轴数控铣床只能加工高度场），即使用六轴工业机器人，也只能对产品外表面进行雕刻，无法改变物体的内部结构。因此在产品设计过程中，需要兼顾最终产品的加工方法，甚至为了便于加工而牺牲一部分产品性能。

与之相比，3D打印采用增材制造工艺，将整个模型体素化，并将材料逐层堆积最终形成整个物体。由于可以直接对模型任何一点（甚至包括材料内部的点）进行操作，因此无论多么复杂的模型都可以用统一的方法来制造，甚至可以直接做出中空的内部结构。用多种材料混合打印，还可以制造出材质渐变的物体，或者彩色模型，如图1.4所示。现在小型的塑料熔融堆积（FDM）3D打印机及打印耗材价格已经非常便宜，普通用户也可以接受，在小批量生产中能够有效地降低成本。正是因为这些优点，使得3D打印迅速普及，在许多领域已经得到广泛应用。尤其是在设计阶段，设计师可以更加专注于功能的实现，而不再

(a)热塑性塑料丝3D打印¹(b)光敏树脂3D打印²(c)粉末堆积3D打印³

图 1.4 商用彩色3D打印产品 目前已经有多种商用彩色3D打印机问世。**(a)**塑料熔融堆积(FDM)3D打印机可以装多个喷头，实现非渐变彩色打印；或者打印过程中人工更换塑料丝的颜色，实现颜色层状变化。**(b)**多材料光敏树脂3D打印机将整个模型体素化，通过给每个体素指定一种材料，实现不同材料的混合（包括颜色和硬度）打印，从而产生渐变效果。目前最先进的打印机(Stratasys Connex3)可以实现三种材料混合。**(c)**粉末堆积3D打印机可以在堆积过程中为白色粉末直接CMYK喷墨着色，实现全彩色3D打印。

过分迁就于是否方便制造，原型样机也可以通过3D打印高效制作，从而大大缩短了产品开发周期。在医学领域，义肢、义齿等器官都需要单件定制，3D打印就非常具有成本优势。这些优点使得3D打印成为近几年研究的热点。

3D打印是将数字化模型实体化的过程，将物理仿真应用于模型设计或者制造过程中，可以对模型的视觉、力学或运动学方面的性能进行提升。目前这方面的主要研究方向包括：

- **模型外观制造** 使制造的模型符合特定的纹理颜色、表面BRDF、次表面散射等。
- **模型强度分析** 通过对模型强度进行分析，能够保证其力学性质的稳定。
- **模型运动分析** 模型本身就是一个机械结构，或者使模型能够满足特定的运动条件，如平衡、旋转等。

总体来说，减材制造的数控机床已经是非常成熟的产业；而增材制造的3D打印技术尚处于起步阶段，还存在广阔的研究前景。

¹图片来源：http://3dforged.com/wp-content/uploads/2013/12/Two-color_globe_610x407.jpg

²图片来源：<http://www.engineering.com/Portals/0/BlogFiles/3D%20Printing/0214/color%20prints.jpeg>

³图片来源：<http://3dprinting.com/wp-content/uploads/2012/07/Full-color-sandstone.jpg>

1.3 研究现状

图形学中物理仿真的应用之一是制作真实感的电影特效。通过对物理模型进行简化，高效地计算出爆炸、洪水、火焰、房屋倒塌等符合物理规律的动画，结合真实感渲染器，最终使画面呈现逼真的视觉效果。近几年，数字化制造、尤其是3D打印技术在图形学中也得到了广泛关注，而物理仿真可以用于对3D打印模型的外观、结构进行优化，制作符合给定功能的模型。面网格在物理仿真当中也有很多的应用，包括薄壳结构模型（如布料、纸张、粘稠薄膜等）的建模与物理仿真，模型的碰撞检测，以及最终的渲染。

1.3.1 刚体运动仿真

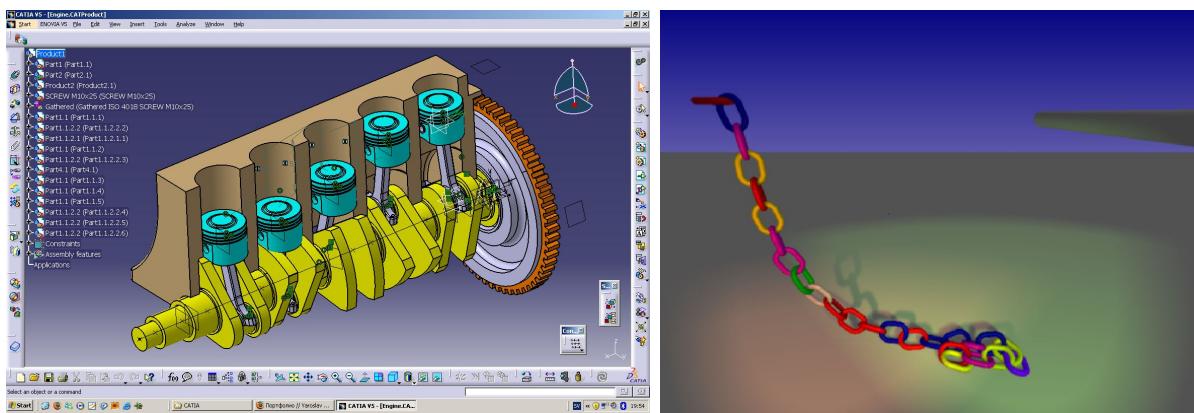
(a)CAD软件CATIA机械运动仿真⁴(b)刚体运动仿真^[11]

图 1.5 刚体运动仿真 刚体运动仿真在CAD领域广泛用于机械结构的运动分析。面网格在刚体运动仿真中的主要作用是碰撞检测和渲染。

刚体运动仿真在工业级CAD软件（如Pro/E, SolidWorks）中应用非常广泛。当各个机械零件设计完成后，将其虚拟的装配在一起，指定主动零件的运动方式（如设定某一根轴匀速转动），从动零件的运动状态就可以根据约束被求解，从而模拟整个机器的运行，甚至连各零件的应力分析也可以同步完成。零件之间的约束既可以是虚拟装配时指定的约束类型（如同轴、共面等），也可以是通过零件之间碰撞检测实时产生的约束。在游戏引擎中，一般也包含简单的刚体运动仿真。如3D作战类游戏的弹道，以及场景中道具跌落的效果；2D游戏（如“愤怒的小鸟”）中各个物体之间的碰撞，都是通过刚体运动仿真来实现的。

⁴图片来源：http://s3images.coroflot.com/user_files/individual_files/original_178545_dO4wcsgBnomSX8LLOGydFf5bg.jpg

面网格在刚体运动仿真中主要表示刚体的形状，用于碰撞检测和渲染。总体来说，刚体的运动较为简单，单个刚体甚至可以写出运动状态的解析表达式。只有在多个刚体发生相互作用时，如碰撞、层叠等，整个系统的运动才需要通过数值方法来进行仿真。因此刚体运动仿真主要研究目标是提升计算性能，尤其是在大规模场景仿真时，提高碰撞检测的速度。

上世纪90年代以前，刚体运动仿真（如Hahn^[12]，Baraff^[13]等人）使用的刚体物理模型与今天并无太大区别，但受限于当时机器的计算性能，只能对简单模型进行仿真。两个包含约200个三角形的模型，每帧计算时间需要8s。到2000年，Mirtich^[14]对400个凸多边形仿真做到了1FPS。2003年，Guendelman等人^[15]对500个凹多面体仿真每帧计算需要3分钟。到2012年，Tonge等人^[11]对5000个高度层叠的凹多面体已经能够达到60FPS的仿真速度。速度的提升一方面归功于硬件运算速度的大幅提高，另一方面也归功于提出的各种加速碰撞检测的算法。

目前，碰撞检测已经成为了独立的研究方向，越来越多的加速结构及算法被提出^[16-21]。通过GPU并行计算，已经能够做到一百万个模型的实时碰撞检测^[22]。这些碰撞检测算法不仅可以用于刚体运动仿真，也可以应用于弹性体、塑性体的仿真当中。

1.3.2 弹性体仿真



图 1.6 弹性体仿真 对一维、二维、三维弹性体采用对应拓扑维度的网格进行仿真。其中，二维弹性体（如布料）使用面网格来建模。

在制造业中，通过有限元方法对机械零件进行强度分析，已经成为CAD软件的标准配置。在电影特效当中，弹性体仿真最重要的应用是人物角色的头发和服装的运动仿真。其中，服装（布料）使用面网格来建模和仿真。

早期的布料仿真用质量弹簧系统实现的^[26]，整块布料是一张四边形网格，质量集

中在各个顶点上。在不同的顶点之间加入三种弹簧分别抵抗三种应力：相邻顶点之间加入弹簧抵抗拉伸应力；对角顶点之间加入弹簧抵抗错切应力；间隔顶点之间加入弹簧抵抗弯曲应力。系统通过前向欧拉方法求解，并对大拉伸弹簧额外施加约束，从而实现布料的仿真。由于采用了前向欧拉求解，为了维持系统稳定只能使用小步长，因此计算比较慢。随后Baraff等人^[27]提出的“大步长布料仿真算法”，通过有限元方法对布料进行建模，并将反向欧拉应用于方程求解，从而极大地提高了仿真速度。从此以后反向欧拉法几乎成为了布料仿真的标准方法。

布料仿真的主要研究方向有：

- 改进布料的物理模型，使其与真实布料物理属性更加接近。
- 提高布料的碰撞检测效率，能高效的处理与其他物体碰撞以及自碰撞。
- 提高仿真计算速度，通过精简模型，提高并行性，或者预计算，实现仿真的加速。

目前已经有许多研究针对这些问题提出了解决方案。在改进布料物理模型方面，Choi等人^[28]通过对布料的弯曲进行建模，实现了逼真的褶皱效果。Goldenthal等人^[24]通过对布料添加额外的约束以及一种快速投影算法，使得在相同的计算速度下大幅提高了布料的拉伸系数，使其同真实的布料更加接近。Kaldor等人^[29;30]提出了一种以纤维为基本单位生成布料，并直接在纤维尺度上进行布料仿真的方法，可以对毛衣形态的布料进行非常精细的仿真。Volino等人^[31]将布料建模为St.Venant-Kirchhoff材料并通过FEM求解，使得仿真结果更加接近真实世界。Wang等人^[32]通过一块按照一定方式拉伸的布料样品，获取FEM中布料的物理参数。

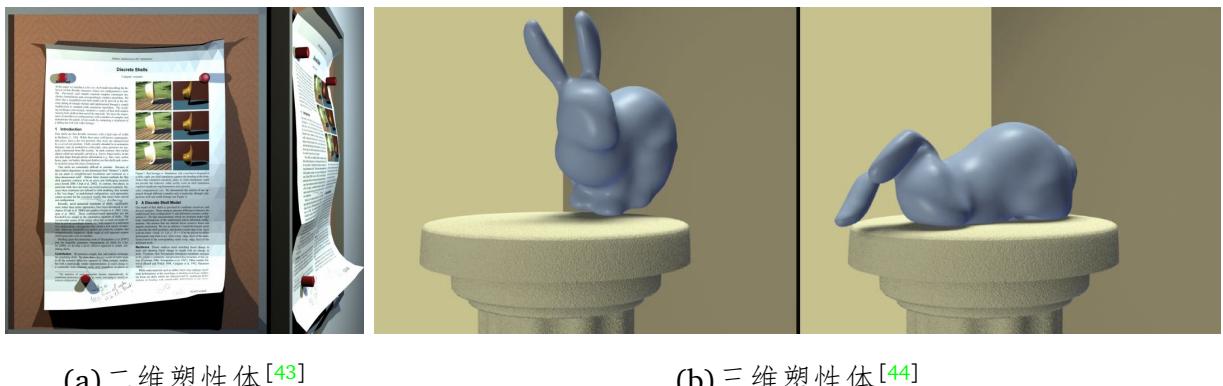
在提高碰撞检测效率方面，Bridson等人^[33]给布料仿真加入了一套有效的碰撞检测和处理机制，使得布料在与其它物体接触或者布料本身叠在一起时，能够因摩擦力的存在呈现正确的形状。Selle等人^[2]在此基础上进一步改进，使得对于极高分辨率的布料模型也能够正确的处理与物体的接触。Baraff等人^[34]提出了一种处理网格自交的办法，通过计算网格的交线，检测出已经穿透的区域，从而对这些区域特殊处理。这种方法在穿有服装的动画角色中特别有用，因为动画角色的网格会出现大量的自交（这是骨骼绑定方法Linear Blend Skinning的固有缺陷），通过特殊处理机保证服装布料仿真的正确性。

在提高仿真速度方面，Müller等人^[35]提出了基于位置的动态仿真方法，通过将布料的受力简化为顶点之间的简单约束，能够获得与仿真相似的视觉效果。这套方法也成为了NVIDIA PhysX布料仿真的基础方法，应用于游戏当中的实时布料仿真。这种方法还可以通过建立层次结构进一步加速^[36]，或者通过两层网格来生成布料的褶皱^[37]。Aguiar等

人^[38]提出了一种介于服装直接骨骼绑定和真实物理仿真之间的生成算法，通过对大量仿真结果的学习构建模型，从而实现了服装动画的实时生成。Feng等人^[39]提出了一种形变映射的方法，将低分辨率布料形状直接映射成高分辨率，实现了实时布料仿真。Wang等人^[40]对紧身服装每个关节处都预计算出高分辨率布料形状，然后在实时仿真时根据每个关节弯曲角度，直接将对应的高分辨率布料形状拼在一起，实现交互级布料仿真。

除了布料仿真外，一维弹性体通常使用线状质量弹簧系统来仿真。Bergou等人^[41]提出了专门针对一维弹性细丝的弯曲形变进行仿真的方法。Selle等人^[23]用质量弹簧系统对头发进行了高精度的仿真。三维弹性常见于工程CAD中对零件强度进行分析，则通常使用四面体或立方体网格，并用有限元方法进行仿真。图形学中通常对三维弹性体进行简化，以满足电影特效的高效计算。Patterson等人^[25]利用立方体网格来近似任意形状弹性体，从而加速仿真。Tan等人^[42]在弹性体中加入伸缩线条，使得用户可以控制弹性体的总体运动。

1.3.3 塑性体仿真



(a)二维塑性体^[43]

(b)三维塑性体^[44]

图 1.7 塑性体仿真 塑性体是弹性体在大拉伸情况下的扩展。将形变分解为弹性和塑性，当弹性形变大于材料的极限，就开始塑性形变，模型无法再回到其初始形态。二维塑性体是通过面网格来仿真的，其中材料的弯曲通过设置二面角的静态角度来实现。

塑性体是弹性体在大拉伸情况下的扩展，当拉伸率超过一定范围，就无法再恢复其原本形状。因此计算方法多是将形变分为弹性和塑性的叠加，当弹性形变超过一定数值，就将形变累加到塑性形变上。

塑性体仿真一个重要应用是模拟物体破裂的过程。O'Brien等人^[45]用四面体网格通过FEM对弹性体进行仿真，根据应力是否已经大于极限应力，判断物体将会产生破裂的位置，从而模拟刚性破裂的过程。他们^[46]还进一步通过将形变分为弹性和塑性两部分，仅依据弹性形变的应力来判断是否发生破碎，从而实现破碎前能够具有一定程度的塑性形

变。Molino等人^[47]提出了一种使破碎产生的位置不受网格走向影响的方法。Pfaff等人^[3]优化了裂缝处的网格，避免了因离散化造成的误差，从而提高了塑性形变及破碎的视觉效果。Chen等人^[48]通过对破碎结果进行后处理，及其呈现不均匀材料破碎的效果。

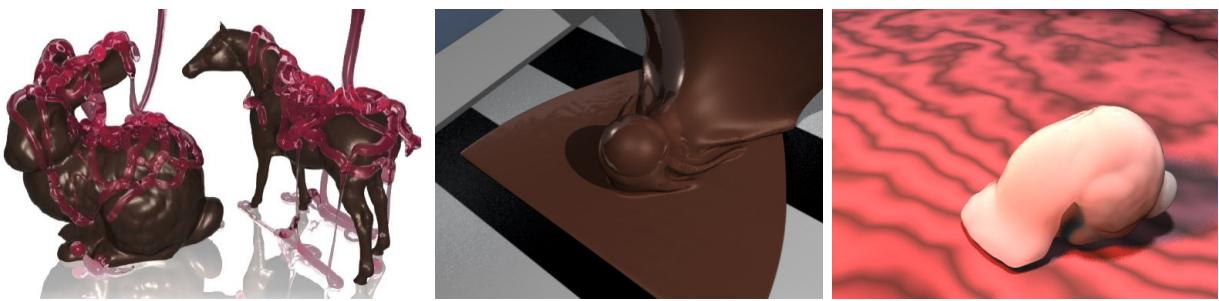
对面网格FEM仿真添加塑性形变，将会产生带折痕的薄壳结构，如折纸或者带褶皱的衣服。Burgoon等人^[49]通过对薄壳模型弯曲形变增加塑性分量，实现了折纸效果的仿真。Rohmer等人^[50]进一步通过沿折痕对网格进行细分，使得折痕与网格的边对齐。Narain等人^[51]通过动态网格实现了网格的边与折痕的对齐，并且能够对纸、金属、布料等多种薄壳结构的折痕进行仿真。Miguel等人^[52]通过对编织成的布料内部摩擦力进行仿真，从而使布料总体上呈现出类似塑性的滞回特性。

1.3.4 流体仿真

流体没有固定的形态，因此流体仿真算法不仅要计算物理参数，还能够处理其形态的剧烈变化，因此是物理仿真当中最为复杂的一类。从计算方法上分类，可以分为拉格朗日法和欧拉法。从流体表现形态上分类，可以大致分为粘稠流体、液体、气体，以及其他特殊形态的流体。每种形态的流体物理性质差别比较大，因此在仿真时需要针对各自特点，选择合适的模型进行仿真。面网格在二维流体仿真、三维薄膜状粘性流体仿真、流体自由液面跟踪等领域都有应用。

1.3.4.1 粘稠流体

粘稠流体是指粘滞阻力非常大，但依然呈现流体特征，符合流体运动方程的物质，比如蜂蜜、沥青。图形学中对粘稠流体的仿真，则更加广泛地指生活中常见的粘稠物，甚



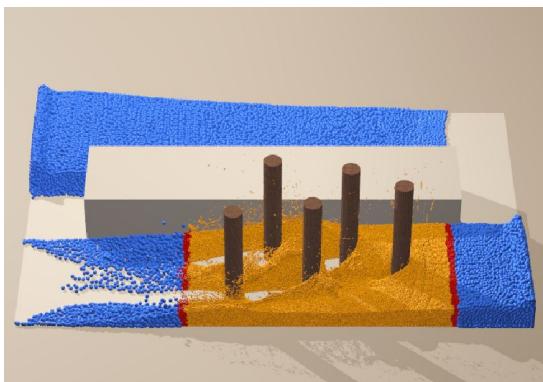
(a)一维粘稠液柱^[53] (b)二维粘稠薄膜^[54] (c)三维粘稠物体^[55]

图 1.8 粘稠流体仿真 图形学中的粘稠流体，广泛地指生活中常见的粘稠物。其中二维粘稠薄膜仿真当中，薄膜采用面网格来表示。

至将形变非常剧烈的塑性体也归入其中。因此，对粘稠流体的仿真，既有拉格朗日SPH方法，也有欧拉法，还有通过塑性体仿真与动态网格结合的方法。

Carlson等人^[55]用大的粘性系数来表示粘稠流体，并用粒子来跟踪流体表面运动，从而模拟出蜡质的融化。Goktekin等人^[56]将弹性公式直接加入Navier-Stokes方程，从而使流体具有一定的弹性，模拟胶状物质的运动。Bargteil等人^[44]通过四面体网格的塑性形变对粘性流体进行模拟，当网格形变过大时，对四面体网格进行重采样。Batty等人^[57]指出粘性流体边界条件的设置对于计算准确性影响很大。Wojtan等人^[10]直接用塑性形变有限元方法，用动态四面体网格对粘性流体进行仿真。Bergou等人^[53]通过简化模型对粘性细丝进行了仿真。Batty等人^[54]通过粘性薄膜模型，对粘性流体膜进行了仿真。Stomakhin等人^[58]对雪进行了物理仿真，通过笛卡尔网格将拉格朗日和欧拉两种方法结合使用，达到了逼真的效果。Paiva等人^[59]通过改进SPH中粘性的表达，实现了对粘稠流体的仿真。

1.3.4.2 液体



(a) SPH流体仿真^[60]



(b) 欧拉法流体仿真，面网格跟踪自由液面^[61]

图 1.9 液体仿真 液体的显著特征是存在自由液面。**SPH**是对流体体积的直接离散化，因此天然就能够表示流体存在的空间区域。欧拉法将整个空间统一划分网格，因此还需要记录流体的自由液面。面网格可以被用于显式的跟踪液面运动，并且通过网格实现表面张力的效果。

液体的显著特征是存在自由液面，用拉格朗日法（SPH）和欧拉法都有各自的优势。SPH将流体按体积划分并抽象为粒子，因此这些粒子天然就能够表示流体存在的空间区域。而且SPH应用范围很广，计算易于并行，可以达到实时仿真。欧拉法将整个空间统一划分网格，因此需要记录流体的边界，从而对流体内部进行仿真，并在表面上设置边界条件。表面网格就是一种能够显式跟踪流体自由液面的方法，同时还能够通过网格计算表面张力，并进行最终的高质量渲染。

Müller等人^[62]将SPH引入图形学流体仿真时，已经可以做到交互级别的流体仿真。随后大家在此基础上对SPH进行了各种改进，使其能够应用于更多的场景，或者达到更高的计算速率。Solenthaler等人^[63]提出了Predictive-Corrective Incompressible SPH（PCISPH），通过对粒子的密度进行修正，使SPH仿真中流体被压缩的现象得到了解决。Solenthaler等人^[60]提出了一种两尺度SPH仿真方法，大尺度粒子用于计算流体的总体运动效果，而在工况复杂的局部增加进行小尺度粒子，使流体表面具有更多的细节，只需增加少量计算，就能使结果得到较大的提升。Schechter等人^[64]还提出了通过在固体表面添加一层“幽灵粒子”来模拟固体表面对流体的吸附作用，这样流体就可以附着在物体表面而不是直接散开，从而实现更逼真的仿真效果。Mercier等人^[65]对低精度SPH仿真的结果进行后处理，在其表面直接合成高精度细节，从而实现与高精度流体仿真类似的结果。

SPH流体仿真的主要缺点是精度低。由于每次计算都需要从粒子的周围进行差值，插值误差会不断累积。因此SPH主要应用于电影特效、游戏等对精度要求不高的场合。而欧拉法使用固定网格，避免了插值误差，可以实现更高精度的仿真，因此是工业级CFD软件，以及高精度流体仿真的主流方法。

图形学发展的早期，对自由液面的流体多是当成高度场来计算。通过这种方法可以模拟大片水域的波纹^[66]，并且计算流体在平面内的运动情况^[67]。如果与粒子系统相结合，还可以模仿简单的水花溅开的效果^[68]。Foster等人^[69]则通过在流体中均匀散布粒子，让粒子随着流体的流场运动，将有粒子分布和无粒子分布的网格交界处作为自由液面，并添加对应的边界条件。他们还将此方法扩展到三维仿真，只是受限于运算能力，使用的网格分辨率很低，只能看出流体的大概运动方向。同时他们采用前向欧拉法求解，因此迭代步长会受到限制，计算效率很低。Jos Stam^[70]提出了一种在任何条件下都能够保证Navier-Stokes方程求解稳定的半拉格朗日法：stable fluid。由于稳定性能够保证，因此在计算中可以使用更大的步长，从而显著提高了运算效率，甚至已经达到实时计算。但是stable fluid模拟的是充满整个容器的流体，不包含自由液面，而且由于其求解方法耗散很大，相当于粘性流体，因此在工程计算中误差较大，只是在图形学中用于产生视觉效果的流体仿真。从此以后，如何对自由液面进行表达，成为流体仿真研究的重点。

Foster等人^[71]采用粒子与level set相结合的办法表达自由液面，以stable fluid作为流体求解方法，在进行被固体搅动过程的仿真时，既通过level set表示了流体的表面，又通过粒子产生喷溅的效果。Enright等人^[72]进一步提升了自由液面的追踪精度，与渲染器结合起来，可以做出非常高质量的动画。Losasso等人^[73]通过八叉树来对求解空间进行划分，从而能够更好地保持流体表面的细节。Selle等人^[74]也是利用粒子与level set混合的方法，将计算过程中由于耗散而损失的细节重新加到流体表面，从而实现更加丰富的视觉效果。

最近几年，由于对流体仿真表面结果的要求不断提高，出现了直接用面网格表达流体表面的方法，其优点是能够直接保持流体表面的细节，避免level set等方法在生成曲面时的细节丢失。Wojtan等人^[75]使用了动态三角网格，利用体素结构检测网格形变过程中的融合与截断，并且能够通过多种物理方法进行仿真。还改进了曲面融合与截断的方法^[76]，使得细小的结构也能通过曲面来表达。Thürey等人^[61]在这套方法之上，给网格表面也加入了表面张力，从而使得仿真结果在宏观和微观上都能够符合物理规律。Edward等人^[77]通过使用一种自适应三角网格方法，实现了只用低分辨率的流体仿真网格就能保留丰富的流体表面细节。

1.3.4.3 气体



(a) 火焰仿真^[60] (b) 烟雾仿真，网格跟踪烟雾表面^[78]

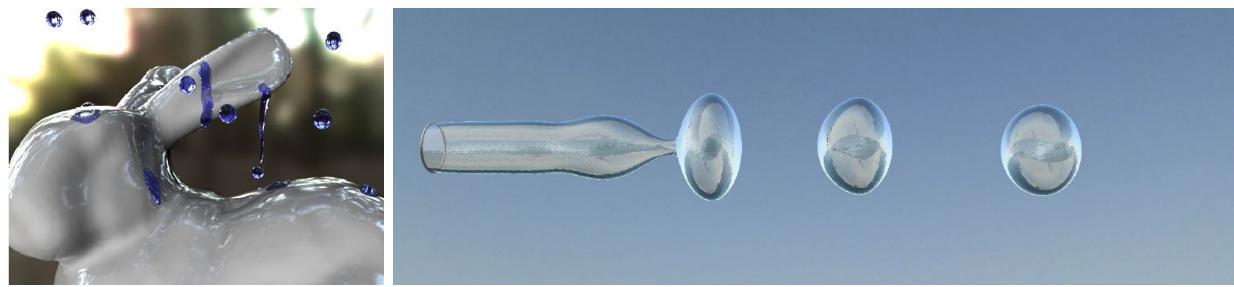
图 1.10 气体仿真 气体能够充满任意空间，因此气体中的火焰或烟雾等现象需要用粒子或者网格来表示。

与液体仿真不同，气体能够膨胀并充满任何空间，因此不存在自由液面。但气体在固定空间的运动过程中，密度变化可忽略不计，因此可以作为不可压缩流体适用Navier-Stokes方程。气体仿真的对象（烟雾、火焰等）通常没有明显的分界面，因此除网格外，还可以采用密度场或粒子表示。

Fedkiw等人^[79]通过粗网格进行流体仿真，并显式地对涡旋进行加强，从而恢复出在网格中被平滑掉的细节。Nguyen等人^[80]通过模拟燃料燃烧导致的气体膨胀，对空气中的火焰进行了仿真。Feldman等人^[81]模拟了空气中悬浮颗粒爆炸的效果，还则使用不规则体网格来表达流体的求解区域，从而更好地处理流体与固体的边界^[82]。Kim等人^[83]在烟雾的表面直接进行小波合成，从低分辨率的烟雾以可编辑的方式生成高分辨率烟雾。Horvath等人^[84]将3D流体空间切片成为一系列的2D流体仿真，以较低的计算开销实现了电

影级别火焰渲染。Mullen等人^[85]提出了一种能量守能的积分方法，解决了隐式积分求解过程中能量耗散问题。Pfaff等人^[78]直接用三角网格跟踪流场的运动，并且在网格上直接添加细节，从而实现高精度的烟雾效果。

1.3.4.4 其他特殊形态流体



(a) 水滴仿真^[86]

(b) 气泡仿真^[87]

图 1.11 表面张力主导的流体仿真 当流体表面积与体积之比很大时，其表面张力对流体运动将起到显著的影响，通常通过平均曲率流对表面张力进行仿真。平均曲率多是通过表示物体形状的面网格来计算的。

Navier-Stokes方程表述的是流体的宏观运动，而对于流体各种各样的微观现象（如表面张力、气泡等）则需要通过专门的方法对其进行描述。

Wang等人^[86]通过在流体与固体接触位置构造虚拟平面，使水滴形成接触角，从而模拟出液滴在固体表面吸附并滚动的效果。Kim等人^[88]通过体积修正的方法，使得level set表示的流体泡沫仿真过程中体积损失得到修正。而Cleary^[89]，Thürey^[90]，Hong^[91]，以及Busaryev^[92]等人都提出了对水中存在的大量小气泡建模的方法。Lenaerts等人^[93]将多孔材料的吸水现象应用于SPH当中，通过将材料当中密集的孔洞抽象成为宏观的物理参数，从而实现对这种现象的快速模拟。

1.3.5 多相耦合仿真

多相耦合仿真是模拟将多种自然现象的相互作用。比如浮在水面的小船，刚体和流体之间的作用是相互的，因此需要对它们一起进行仿真。多相耦合的核心是各相之间的边界条件需要保持连续，而各相内部则根据自身的物理方程去计算。

在包含SPH流体的系统中，可以将所有物体都生成粒子，从而采用统一的框架进行仿真。Akinci等人^[97]通过在固体表面生成粒子，将固体与流体粒子放入同一的框架中计

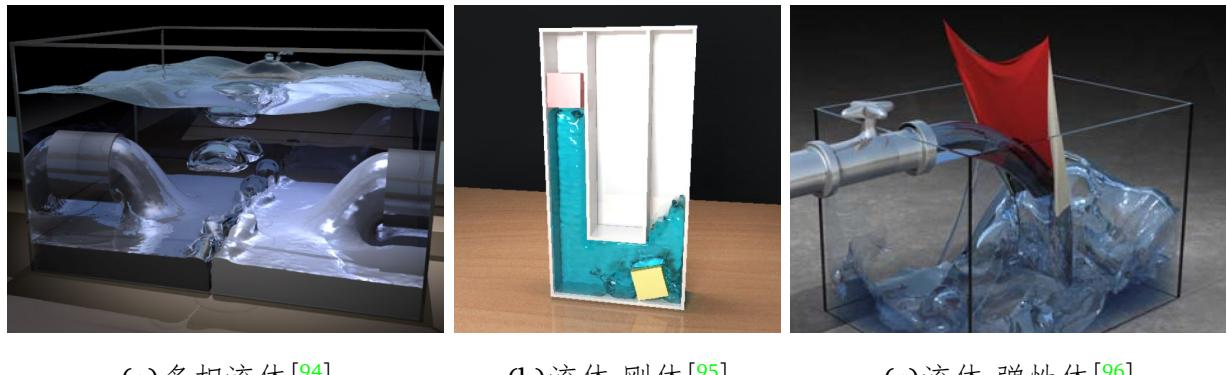
(a) 多相流体^[94](b) 流体-刚体^[95](c) 流体-弹性体^[96]

图 1.12 多相耦合仿真 多相耦合仿真模拟将多种自然现象的相互作用，每相之间边界条件连续，每相内部运动符合自己的物理方程。

算，实现了SPH流体仿真过程中的流体固体耦合计算。Ren等人^[98]用SPH对多相流进行了仿真，可以模拟出不同颜色的流体混合时颜色的变化。欧拉法也可以用于多相耦合仿真。Génevaux等人^[95]通过计算固体与流体之间的相互作用力，模拟了流体与刚体的相互作用。Guendelman等人^[96]对流体和薄膜弹性体的相互作用进行了仿真，其中薄膜弹性体是用面网格来仿真的。Losasso等人^[94]通过扩展的level set实现了多相流仿真。

1.3.6 物理仿真在数字化制造中的应用

图形学中物理仿真原本只是用于电影、游戏等虚拟场景的动画制作。随着数字化制造制造的发展，尤其是3D打印技术的进步，物理仿真在制造领域也在发挥重要作用。对模型进行强度分析，可以在实际制造前检测脆弱的部位。对模型进行运动分析，找出可能发生干涉的结构。用于制造的物理仿真已经不仅限于视觉效果，因此对计算的精度就有了更高的要求。3D打印最新的研究方向包括特定外观模型制造、通过弹性体仿真进行模型强度分析、通过刚体运动仿真进行模型运动分析等。

1.3.6.1 模型外观制造

模型的外观主要取决于几何形状和纹理颜色，可以通过商用全彩色3D打印机来制造。但模型表面BRDF，次表面散射等也会影响视觉效果。目前商用全彩色3D打印机价格昂贵，而且只能使用固定材料。因此针对目前3D打印在模型外观制造方面存在的缺陷，科学界提出了多种改进方案。Brunton等人^[99]将平面半色调技术推广到体素当中，通过把支撑材料也作为一种原材料，用原本只有三种材料的Stratasys Objet500 Connex3打印机实现了全彩色打印。Lan等人^[101]提出了一种制作特定BRDF表面的方法。先通过3D打印制作带有微



(a)纹理打印^[99] (b)BSSRDF打印^[100] (c)BRDF打印^[101]

图 1.13 特定外观产品制造的最新进展 通过对打印算法进行改进，不仅可以打印出高质量的全彩色模型，甚至可以形成给定的次表面散射，或表面BRDF。**(a)**通过体素半色调算法，用多材料光敏树脂3D打印机实现高质量全彩色打印。**(b)**打印按层分布的多种材料并优化其组合，形成特定次表面散射效果的模型。**(c)**3D打印与平板喷墨打印相结合，使模型表面具有特定BRDF。两幅图为相同视角观察的同一个模型，光源位置导致不同的视觉效果。

结构的高度场，然后再用UV平板打印机在微结构上喷墨，使几何+颜色的总效果形成给定的BRDF。Chen等人^[102]以及Vidimče等人^[103]通过多材料SLA-3D打印机，对给定外观模型进行打印的方法，并进一步制造了一台多材料3D打印机^[104]，实现更高级的外观效果。

除模型表面纹理颜色外，由于有些打印材料具有一定的透光性，因此还可以通过计算使模型具有特定的次表面散射效果。Dong等人^[105]通过不均厚度的材料分层及组合，打印出了具有特定双向表面散射反射率分布函数（BSSRDF）的模型。Miloš等人^[100]也是通过对多种基础材料建立BSSRDF数据库，并对其进行组合，使其组合之后的最终结果与目标模型的BSSRDF一致，从而实现对特定次表面散射效果的制造。

1.3.6.2 模型强度分析

将3D打印模型作为弹性体对其进行应力分析，可以在打印前对模型的坚固程度做评估，避免打印之后模型因强度过低而破坏。Stava等人^[106]针对模型结构强度弱到甚至不能支撑自身重量的问题，提出一种通过应力分析，找出结构较弱的部分，从而对其进行加强的方法。首先对模型生成四面体网格，然后对整个模型在重力场中以及局部受到外力情况进行FEM应力分析，通过内部设置空洞减轻重量、模型局部加厚，以及增加支撑结构的办法消除应力集中的区域，最终保证模型打印出来的结构有较高的强度。Zhou等人^[109]提出了一种更加鲁棒的计算模型结构最弱部位的方法，进而最大限度的避免3D打印的实物模型在使用过程中因应力集中而造成破坏。Lu等人^[110]为减少3D打印过程中材料的消耗，提出了一种在模型内部生成蜂窝状空洞的方法。这种方法通过对模型进行力学分析，优化蜂窝



图 1.14 弹性体应力分析在3D打印中的应用案例 (a)将模型当成弹性体进行应力分析, 找出结构上薄弱之处, 并通过加粗、中空减重、增加支撑等方法进行改进, 使最终打印的模型具有较高的强度。 (b)对弹性体进行预形变, 使打印的软体模型在重力作用下形变之后, 与设计模型形状一致。 (c)通过微结构实现宏观上不同的弹性强度, 使得单一材料打印的模型宏观上成为不均匀弹性的物体。

的形状, 使模型达到强度与重量的均衡。

材料强度分析在制作软体模型时可以发挥重要的作用。 Skouras 等人^[111]提出一种通过多种弹性材料混合打印, 制造特殊力学性能模型的方法。 Chen 等人^[107]针对软体材料在重力下会发生形变的问题, 提出一种对初始打印形状进行优化的方法, 使得模型在重力下形变之后的最终形态, 与设计的一致。 Pérez 等人^[112]对六面体网状模型的结构进行优化, 使得其真实模型在给定约束条件下呈现的形状与设计的一致。

尽管材料的种类有限, 但通过打印出微结构, 对整体力学性质也会起到重要的作用。 Bickel 等人^[113]提出了一种制造符合给定物理性质的弹性体模型的方法。通过构建一个包含各种材料、各种微结构的材料库, 将各种片状材料层叠在一起, 通过计算使得这些材料及结构组合出的弹性体与目标弹性体物理性质最接近, 并通过3D打印将模型制作出来。 Panetta 等人^[114]用六种微结构组合为整体模型, 使模型各部分能够具有不同的力学性质, 甚至能够使总体具有负的泊松比(材料两侧被压缩时, 总体积反而增大)。 Schumacher 等人^[108]也用类似的方法做到了总体呈现非均匀弹性的模型。

受力分析在设计模型形状时也会有一些实际的用途。对于体积大于3D打印机工作范围的模型, 可以将其拆分为许多小块分别打印, 最后再组装为一个整体(胶水粘合或打印时预留销钉和销孔)。 Luo^[115]以及 Yao 等人^[116]将应力分析作为模型拆分算法优化目标之一, 避免在大应力处切分, 使得组装的模型更加结实。 Wang 等人^[117]提出了一种打印模型时减少耗材使用的方法。将模型分为外壳与四棱柱结构支撑, 对支撑进行受力分析并优化结构, 使得在不减弱强度的前提下减少材料消耗。最后再加入支撑结构, 使得整个模型不需

要额外的支撑材料，即可直接用FDM打印机打印。Dumas等人^[118]通过用户给定纹理图例生成3D打印模型的几何形状，使几何与图案有类似的风格。而强度分析可以保证细小的结构不会受到破坏。

1.3.6.3 模型运动分析

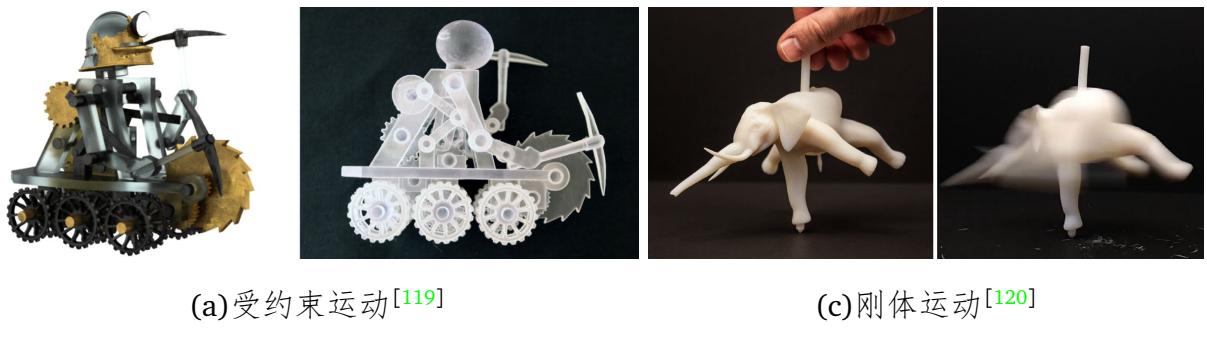


图 1.15 刚体运动仿真在3D打印中的应用案例 (a)对刚体受约束运动轨迹进行仿真，使生成的机械结构符合特定的运动规律。(b)对刚体运动、尤其是旋转进行仿真，并对模型进行优化，使其能够像陀螺一样旋转时保持平衡。

对3D打印模型进行刚体运动仿真，可以制作质量分布满足特定要求的模型。Prévost等人^[121]提出一种让3D打印模型保持平衡的方法，通过改变体素化模型内部结构或者对模型进行形变来调整重心位置，使其能够按照设计站立在平面上。Bächer等人^[120]提出一种通过调整体素化模型重心，使其能够像陀螺一样旋转的方法。Musalski等人^[122]用位移曲面实现类似的效果，使模型具有更大的优化空间。

对于3D打印的机械连接结构，对其进行受约束的运动仿真，可以使其符合给定的运动规律。Zhu等人^[123]提出一种自动生成可运动机械玩具的方法。用户指定模型的运动方式之后，算法自动生成控制模型运动的机械结构，使模型按照设定轨迹运行。Coros等人^[119]提出的连杆及齿轮优化算法，能够让非专业用户设计机械玩具。Ceylan等人^[124]通过优化齿轮的配合，使得机械模型能够以给定骨骼动画类似的方式运动。Megaro等人^[125]设计了一套能够让普通用户制作多足机器人的系统。系统能够自动生成可打印部件，用简单零件和舵机就可以完成组装。Bächer等人^[126]提出了一种能够让用户方便编辑连杆系统的方法。用户只需指定少量信息，系统就可以生成可打印，并且能正确运行的连杆模型。

1.4 现有方法优缺点分析

物理仿真涉及领域众多、对各种自然现象有不同的方法对其进行仿真，而网格的使用

也非常广泛。本文涉及其中三种现象的仿真：水滴、粘性流体薄膜、塑性薄膜，因此重点同相关的现有方法进行比较。粘性流体薄膜和塑性薄膜都是用面网格来表示的。

1. 水滴

Wang等人^[86]专门针对水滴在物体表面流动现象的仿真方法，通过虚拟平面来构造流体与固体的接触角。他们采用欧拉法进行流体仿真，通过粒子level set表示水滴表面，因此能够直接实现水滴的融合与分裂。由于水滴尺寸很小，欧拉法需要对空间进行很精密的细分，因此必须采用稀疏结构存储物理参数。这种方法的好处是精度很高，仿真结果与真实物理现象一致性很好；而主要缺点是运算量太大导致效率极低，几秒钟的动画需要数天的时间来仿真。Wang等人^[127]还提出了通过浅水方程，用高度场表示水滴的方法。这种方法使运算量大幅减少，但仅限于对水滴在固体表面滑动的现象进行仿真。Thürey等人^[61]用动态网格跟踪流体自由液面的方法也能够产生水滴，但这种欧拉-拉格朗日结合的方法主要针对通用流体仿真，没有针对水滴进行优化，因此仿真速度比较慢。

2. 粘性流体薄膜

Batty等人^[54]对粘性薄膜进行了仿真。薄膜采用了动态网格，应对大拉伸时的剧烈形变，甚至破裂的情况，其内力包含薄膜拉伸粘滞力、弯曲力、表面张力。由于采用了拉格朗日网格，因此这个方法只适用于二维粘性薄膜，无法直接扩展到一维和三维的应用。对于一些特定的应用（如浮在水面上的粘性薄膜），内力的模型还可以继续简化，从而提高仿真速度。

3. 塑性薄膜

Pfaff等人^[3]对薄膜材料的塑性形变及破碎效果进行了仿真。薄膜用高精度三角网格表达，并且会根据应力动态重三角化，使得裂縫能够沿着正确的应力分布产生。材料形变为弹性形变和塑性形变的叠加，塑性形变的程度和时间有关，因此整个系统需要通过动态仿真来模拟。对于准静态系统，如果需要用静态平衡方式求解，就要改变塑性形变的模型方程。

目前物理仿真的相关工作多是应用于电影特效，而在数字化制造领域，应用主要集中在弹性体仿真和刚体运动仿真。这些仿真都是针对3D打印模型本身进行的，其实许多工艺流程也可以通过仿真而得到提高和改善。对传统工艺进行物理仿真，有可能会极大地拓展传统工艺的适用范围，使其在数字化制造中能够发挥新的作用。本文所进行的物理仿真主要用于三维曲面着色，具体包含水转印和真空热塑成型两种工艺。

对于水转印工艺的仿真只有Panozzo等人^[128]与我们几乎同时进行的工作，方法与我们相似，但他们的成果发布要迟于我们。他们通过静态平衡的方式对物理方程进行求解。由于去掉了时间的参数，计算速度可以加快，但是同实际物理过程相比误差会增加。而且他们的方法不支持多次转印，因此无法处理需要对整个曲面进行完整着色的复杂模型。

工业中对真空成型工艺的仿真已经有许多工作^[129-131]。但是这些工作不涉及曲面着色，只对真空成型的薄膜运动进行仿真，计算薄膜形变后的厚度分布，从而优化模具设计及成型过程中的辅助工艺。一些工业级的仿真软件^[132]尽管有针对带图案的定位真空成型进行图案预形变功能，但没有考虑拉伸导致的色彩变化并对其进行预校正，也不涉及模具排气孔的设置与优化。这些软件都是面向专业用户，软件价格很高，而且还需要专门的设备对使用材料的物理参数进行测试，并不适用于个人用户给3D打印模型着色。Schüller等人^[133]提出的方法与这些软件实现的功能相同，同时还提出了基于视觉的塑料薄膜参数估计方法。Mellor等人^[134]和相关专利^[135]提到了用带有标定网格的塑料板进行真空成型，并用视觉系统测量塑料板形变，从而对图案进行预形变的方法。这种方法的优点是能避免测试复杂的物理参数，因此适用范围很广。缺点是需要专门的扫描装置，对每一个模型都需要进行一次标定，因此适合规模化生产，而对单件定制产品操作就显得过于复杂。

1.5 本文工作

本文用面网格对水滴、粘性流体薄膜、塑性薄膜等三种物质进行了建模和仿真。水滴体积很小，其形状主要受表面张力的影响，因此可以忽略水滴内部的流体运动，通过面网格的直接形变进行仿真。水滴之间的融合与分裂，可以通过面网格的布尔运算及网格优化来实现。粘性流体薄膜和塑性薄膜都是薄层结构，因此用面网格可以很好的对其进行仿真。本文将这两种仿真技术应用于两种传统制造工艺的虚拟化：水转印和热塑成型，提出“可计算水转印刷”和“可计算热塑成型”的方法，使其能够用于三维曲面全彩着色。我们制造了这些新型工艺的原型系统，并通过实验证明了其实用性和可靠性。

第二章我们提出了一种微小流体基于表面网格直接形变的实时仿真方法，能够对液滴的运动以及与物体接触并在其表面滑动的现象进行模拟。基于小尺度假设，可以将流体内部的流动简化成为其面网格的形变。在每一个计算步长中，该模型先用隐式平均曲率流运算产生表面张力的效果，然后显式的构造与固体表面的接触角，最后再用一系列的网格运算处理拓扑结构变化并提高网格质量。通过实验证明了该模型的尽管简单，但是能够产生丰富的流体效果，能够达到实时仿真的运算速度。

第三章我们对浮在水面上的粘性薄膜在物体浸入水中时产生拉伸的效果进行了仿真，

并将其用于模拟水转印技术，提出了“计算水转印刷”的方法。通过对整个水转印物理过程进行仿真，可以根据水转印膜的形变，得到预变形图案，使得印有这个图案的水转印膜在形变之后，能够在模型表面刚好形成想要的纹理。这个方法兼具传统水转印工艺印刷质量高、适用范围广的优点，同时解决了对复杂曲面印刷时图案与模型难以对齐的问题。

第四章我们对软化塑料片在大气压力下拉伸并贴在模具表面的现象进行了仿真，并将其用于模拟真空热塑成型。通过对整个真空成型过程进行模拟，根据塑料板拉伸情况，计算出预形变图案，将其印在透明塑料片上。根据塑料板贴在模型表面的过程，得到3D模型需要设置排气孔的位置，并通过3D打印得到实物模型，作为模具。最终用真空热塑成型将带有图案的透明塑料片贴到模型的表面。于是图案就贴在了模型的表面，同时形变的塑料片也为图案覆盖了一层透明保护壳。

最后对全文了进行总结，并提出今后可能的研究方向。

第2章 基于面网格的实时水滴仿真方法

在流体仿真中，体积会影响流体的运动效果。大量流体在运动时，水面形状主要受内部流场影响。流体体积越小，表面张力的作用就会越明显。直到体积小到水滴，表面张力将起主导作用。基于这一观察，本章提出了一种基于面网格的实时水滴仿真方法。通过网格直接形变，对水滴的运动、水滴与亲水表面接触，以及水滴在固体表面滑动的现象进行仿真。基于小体积假设，将流体内部的三维体素流动简化成为其面网格的形变，从而使计算量大幅减少。在每一个计算步长中，先用隐式平均曲率流运算产生表面张力的效果，然后显式构造流体与固体表面的接触角，最后再通过一系列的网格运算处理拓扑结构变化，并提高网格质量。

我们通过一系列实时仿真实验，验证了模型的准确性。该模型尽管非常简单，却能够产生丰富的流体效果。所设计的实验包括模拟水滴碰撞时出现在表面的波纹，水柱因表面张力而截断的效果，以及水滴在固体表面滑动等。所提出的仿真方法能够达到同真实物理实验较为一致的结果，并且比现有仿真算法的计算速度快至少一个数量级。

2.1 问题的提出

水滴在日常生活中随处可见，比如浴室的玻璃板，或者雨天的车窗。同大尺度的流体仿真相比，水滴的仿真则具有独特的挑战性^[86]。水滴体积很小，因此表面张力作用效果非常明显，使得水滴具有许多独特的物理性质，比如同固体表面接触时，会因表面亲水性的差异而产生不同的接触角（图2.1）。如果用三维体素来表示水滴，就需要非常细的网格划分。同时在这种大粘度和表面张力作用下，需要用非常小的计算步长，才能保证计算的精度和稳定性，因此空间和时间开销都非常大^[86]。总之，大多数通用的流体仿真算法都不适合水滴运动的仿真。

如果用面网格来描述水滴，则可以比三维体素更好地表达粘性和表面张力的效果。大粘度使水滴内部的流场变得平滑，其效果就像是忽略水滴内部的运动，而只考虑水滴表面的状态。表面张力可以用表面的平均曲率来计算，其效果是使得曲面变得更加平滑，同时



(a)亲水表面，水滴会在表面上摊开¹ (b)疏水表面，水滴会缩成一团²

图 2.1 自然界中的水滴 表面的亲水性会影响水滴的形态，形成不同的接触角

可以让曲面拓扑结构的运算更加鲁棒。

基于上述观察，本文提出了一种基于面网格的实时水滴仿真方法。该方法属于拉格朗日法，对表面网格直接形变模拟水滴运动，还能模拟固体表面对水滴的亲水作用。通过这种方法，将流体仿真的自由度从三维体素降低到面网格的顶点上，大大减少了运算量。这个方法主要包含两部分：

- **网格形变运算** 提出了一组网格形变运算方法，使水滴的运动符合物理规律。比如，隐式曲率流运算可以直接产生表面张力效果，而接触角运算能够模拟水滴在固体表面的亲水效果。
- **网格拓扑运算** 使用了一组网格拓扑运算，用于提高网格质量，以及处理拓扑变化，用于模拟水滴自由的融合与分裂。

整套系统是所有这些运算的有机结合，相互配合才能实现水滴的模拟，每一个都针对水滴仿真进行了优化。比如，尽管网格优化已经被研究了很多年，但正如Wojtan等人指出的那样^[61;75;76]，在真正应用于流体仿真这种拓扑变换极为频繁的应用时，仍然需要专门进行优化才能保证鲁棒性。除此之外，尽管Müller^[136]和Brochu^[137]都曾经研究过表面曲率流和保持体积方法，但他们的方法在水滴频繁融合与分裂的复杂场景中无法直接应用。通过将这些运算融合进一个系统中，并通过对真实物理现象进行仿真，来验证该方法对水滴仿真的高效性。这些物理现象包括水滴的碰撞，水柱的截断，以及水滴在多种物体表面的滑动。本文的典型实验场景包含20000个三角形，其运算速度可以达到30fps。我们还同真实的物理实验视频做了比较，以验证该方法的准确性。

¹图片来源：<http://www.designyourway.net/blog/resources/big-collection-of-water-and-ice-textures/>

²图片来源：<http://www.seriocomic.com/photos/surface-tension>

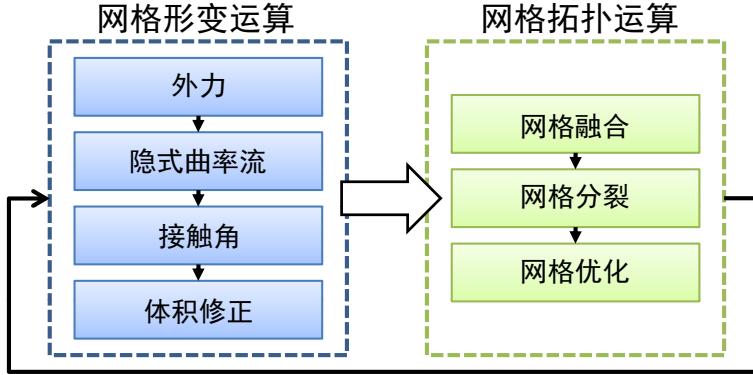


图 2.2 水滴实时仿真算法流程图 网格形变运算用于单个水滴的物理仿真，网格拓扑运算则用于处理水滴的融合与分裂，并对网格进行优化。通过这两组运算迭代求解，实现了包含许多水滴的大场景的实时仿真。

2.2 算法概述

本章提出的表面网格直接形变方法，专门用于水滴尺度的流体仿真。在这个尺度下，水滴的粘性和表面张力效果非常明显，因而运动主要由其表面形状所决定。这个方法可以分为两个主要部分。第一部分负责单个水滴的物理仿真，包含一系列的网格形变运算。这些运算将外力、摩擦力、表面张力，以及其他一些微小流体所具有的现象都考虑进去，从而使水滴运动符合物理规律。第二部分负责水滴间的相互作用，包含一系列的网格拓扑运算，包括网格求交，网格切分，以及能提高网格质量的优化。这些拓扑运算能够使水滴产生融合与分裂的效果，并且保证了网格的质量，从而提高了形变运算的鲁棒性。整个方法中，所有运算都是对面网格直接操作，因此处理速度很快，保证了系统的实时性。

整个系统的流程如图2.2所示。网格形变运算用于单个水滴的物理仿真，包含了四个形变运算。首先将外力作用在网格上，包括重力和固体表面的接触力。然后通过隐式积分计算平均曲率流，使网格产生表面张力的效果。为了让水滴与固体表面接触时产生亲水或疏水的效果，通过接触角运算，根据液体与固体之间的亲水性，显式的构造接触线。最后，用体积修正运算来弥补数值计算过程中产生的体积损失。当水滴面网格形变完成后，我们用网格拓扑运算来处理水滴之间的融合与分裂，同时不断地对网格进行优化，提高网格质量以保证形变运算的鲁棒性。

在整个系统中，运算之间相互独立，总体串行。每个运算都可以视为对网格顶点速度和位移的一次更新。前一个运算中顶点速度和位移的输出，就作为下一个运算的输入。为了简化表达，我们用 v_i^{old} 和 x_i^{old} 表示顶点*i*在运算之前的速度和位移，用 v_i^{new} 和 x_i^{new} 表示顶点*i*在运算之后的速度和位移。

由于各个运算串行且相互独立，各运算之间的约束条件可能会产生冲突。比如，体积

修正运算会改变网格的形状。在极端情况下，更新后的网格可能会产生自交，违背网格拓扑结构的约束。约束之间的冲突可以通过类似Gauss - Seidel迭代的方法逐一化解，直到所有的约束都得到满足。但在实验中，每一步迭代都不会使网格产生剧烈变化，因此这个问题基本可以忽略。唯一需要特殊对待的是，当水滴与固体表面、尤其是复杂曲面接触时，流体网格上所受外力情况非常复杂，如果处理不当就可能导致体积损失，或者使水滴的网格穿透固体。针对这种情况，我们在施加外力运算时，将网格上所有与固体表面接触的顶点都标记出来，在其它运算中添加额外的位置约束，防止其穿透固体表面。

2.3 网格形变运算

本文通过网格形变运算，对单个水滴进行物理仿真，实现了与求解Navier-Stokes方程类似的效果。与基于体素的流体仿真不同，我们只考虑水滴的表面形状，因此将问题极大地简化，提高了计算效率。

2.3.1 外力

在该的模型中，外力包括重力、与固体表面接触力、摩擦力以及粘滞阻力。我们用 v_i^{old} 表示顶点*i*的当前速度，那么重力的作用可以直接表示为 $v_i^{\text{new}} = v_i^{\text{old}} + g\Delta t$ ，其中 v_i^{new} 是更新后的速度， g 是重力加速度， Δt 是时间步长。于是，顶点*i*的位移 x_i 可以通过前向欧拉法来计算： $x_i^{\text{new}} = x_i^{\text{old}} + v_i^{\text{new}}\Delta t$ 。如果仿真过程中流体会与固体表面接触，那么在更新顶点位移之后，需要判断每个顶点是否穿透了固体表面。如果已经穿透，就把该顶点的位移 x_i^{new} 更新为在固体表面上到 x_i^{new} 的最近点，并将速度以完全非弹性的方式更新为：

$$v_i^{\text{new}} = v_i^{\text{old}} - ((v_i^{\text{old}} - v_s) \cdot n_i) n_i \quad (2.1)$$

其中 v_s 是固体在 x_i^{new} 的速度， n_i 是在 x_i^{new} 的法向。直观的解释就是通过公式2.1，使得水滴与固体表面在法向上相对静止，但切向上相对速度不变。同时还根据流体与固体之间的粘滞性，对这些顶点施加摩擦力，其方法为：

$$v_i^{\text{new}} = \begin{cases} 0 & |v_i^{\text{old}}| < \varepsilon \\ v_i^{\text{old}} - \varepsilon v_i^{\text{old}} |v_i^{\text{old}}|^{-1} & \text{otherwise} \end{cases} \quad (2.2)$$

其中 ε 表示摩擦系数的大小。而顶点位移可以根据 v_i^{new} 和 v_i^{old} 之差直接修正： $x_i^{\text{new}} = x_i^{\text{old}} + (v_i^{\text{new}} - v_i^{\text{old}})\Delta t$ 。

在欧拉法流体仿真中，粘滞阻力不会被作为外力来计算。但在拉格朗日法中，需要在流体表面的顶点额外施加阻力，这样就能够实现类似于粘滞阻力的效果。同质量弹簧模型中的阻力类似，该模型的阻力由两项来构成：

$$\mathbf{v}_i^{\text{new}} = (1 - \mu \Delta t) \mathbf{v}_i^{\text{old}} + \eta \Delta t \Delta \mathbf{v}_i^{\text{old}} \quad (2.3)$$

其中第一项是固定的粘滞阻力，每一步迭代都按照固定的比率 μ 使速度减小，类似于空气阻力；第二项是通过Laplacian-Beltrami算子来计算，类似于流体内部相对流动产生的阻力，其阻力系数为 η 。其中第二项与Navier-Stokes方程中的粘滞阻力类似，只是该粘滞阻力是定义在面网格上，而Navier-Stokes方程是定义在体网格中。 μ 的值通常在0.3到0.5之间， η 的值通常在0到0.1之间。通过实验发现，增大 η 的值能够使网格变化更加平滑，从而能够避免因网格质量下降导致的计算不稳定性，因此在计算过程中有可能会额外的增大 η 的值(2.4节)。

2.3.2 平均曲率流

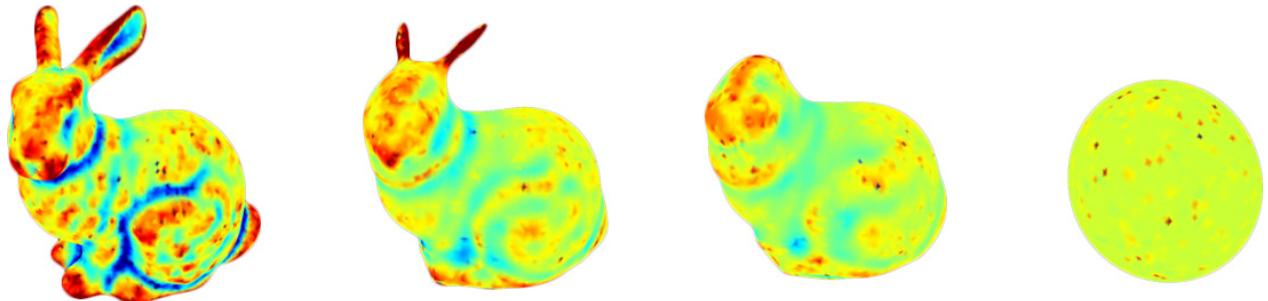


图 2.3 平均曲率流 平均曲率描述了曲面上一点的弯曲程度，根据弯曲方向不同而有正有负。平均曲率流使曲面趋向平滑，以此来模拟表面张力的效果。

同Sussman等人^[138]的方法类似，我们用平均曲率流来模拟表面张力的效果。设 \mathbf{x} 为水滴表面 S 上的任意一点， γ 是曲率流系数， ∇_S^2 是 S 的Laplace-Beltrami算子。平均曲率流在连续曲面上的定义为

$$\frac{\partial \mathbf{x}}{\partial t} = -2\gamma \kappa \mathbf{n} = \gamma \nabla_S^2 \mathbf{x} \quad (2.4)$$

其中 $\kappa \mathbf{n}$ 是在法向 \mathbf{n} 上定义的平均曲率。根据Desbrun等人^[139]提出的半隐式方法，平均曲率流在离散三角网格上计算公式为：

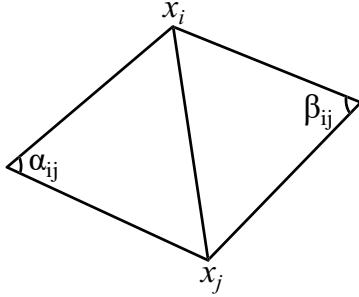


图 2.4 边对角的定义 α_{ij} 和 β_{ij} 称为边 $x_i - x_j$ 的对角。其余弦和 $\cot \alpha_{ij} + \cot \beta_{ij}$ 是构造网格拉普拉斯矩阵的参数，同时作为网格优化运算中边翻转的判别条件。

$$(\mathbf{I} + \gamma \Delta t \mathbf{M}^{-1} \mathbf{L}) \mathbf{X}^{\text{new}} = \mathbf{X}^{\text{old}} \quad (2.5)$$

其中 \mathbf{X} 是顶点位移构成的列向量， Δt 是迭代步长， \mathbf{M} 是质量矩阵， \mathbf{L} 是通过 Laplace-Beltrami 算子构造的拉普拉斯矩阵（对称阵），其构造方法为：

定义 $N(i)$ 为顶点 i 的一环邻域，则矩阵 \mathbf{L} 的非零元为

$$l_{ij} = -\frac{1}{2}(\cot \alpha_{ij} + \cot \beta_{ij}) \quad \text{for } j \in N(i) \quad (2.6)$$

$$l_{ii} = -\sum_{j \in N(i)} l_{ij} \quad (2.7)$$

其中 α_{ij} 和 β_{ij} 是边 $i - j$ 的对角，如图 2.4 所示。由于 \mathbf{I} 和 \mathbf{L} 都是对称阵，公式 2.5 可以通过两边都乘以 \mathbf{M} 变成正定对称的线性系统。用带有非完全 Cholesky 预条件器的共轭梯度法来求解这个线性系统，解出新的顶点位移向量。当算出新的顶点位移之后，每一个顶点的速度就可以再次被更新： $v_i^{\text{new}} = v_i^{\text{old}} + (x_i^{\text{new}} - x_i^{\text{old}})/\Delta t$ 。

2.3.3 接触角

水滴与固体接触时，固体表面的亲水性会使水滴摊开，而水滴的表面张力又会使水滴收缩，两者达到平衡状态时，便形成了稳定的接触角。固体表面亲水性的大小会影响水滴摊开的程度，通常用稳定接触角作为对亲水性的定量描述。稳定接触角定义为：当水滴在固体表面达到平衡状态时，固体、液体、空气三相交界处，液体/空气交界平面与固体平面所成的夹角。

本文提出了一种直接接触角运算方法，通过计算稳定接触角与当前接触角之差来直接驱动接触线，以此来实现亲水的效果，并让水滴网格以稳定接触角静止在固体表面。通过



图 2.5 不同亲水性表面上水滴的稳定形状 通过设置不同的稳定接触角，可以模拟水滴在不同亲水性表面上的静止时的形状，图中的水滴从左到右稳定接触角依次为 60° , 90° , 120° 。

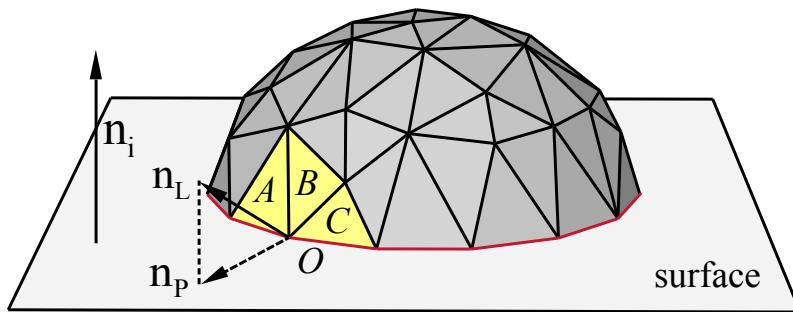


图 2.6 接触线示意图 红色曲线为水滴与固体表面的接触线

这种方法，可以模拟出水滴在各种亲水性表面上的形状，如图2.5所示。如果固体表面非水平放置，那么通过接触角运算还可以驱动网格运动，产生图2.17中小水滴在倾斜玻璃板上滑动的效果。

接触角运算的第一步是找出水滴网格与固体表面的接触线。在2.3.1节计算外力时，已经把所有与固体表面贴合的点标记了出来（记为贴合点），于是我们把所有与非贴合点相邻的贴合点定义为接触点。接触点就存在于三相（气体、流体、固体）交界的位置上。接触线就是将接触点连接成闭合曲线的边的集合，如图2.6所示。

当接触线构造好之后，接下来就要前后移动接触线，使其形成稳定接触角。与Wang等人^[86]针对体素level set提出的虚拟表面方法（通过在接触线上添加虚拟表面，通过平均曲率流隐式驱动接触线的移动）不同，我们的平均曲率定义在网格上，并且受网格质量影响很大。而接触线附近往往是网格质量最差的地方，存在许多的凸起和凹陷，因此接触线附近计算出来的曲率误差可能会很大。

因此本文采用显式方法构造接触角。先计算当前的接触角，然后根据其数值大小移动

接触线上的顶点。如图2.6所示，选取一个接触点，记为 O ，并将其三个邻接三角形记为 A ， B ， C ，于是顶点 O 处流体/空气交界面的法向 n_L 由 A ， B ， C 三个面的面积加权平均法向来表示。固体表面法向 n_i 和 n_L 的夹角也是以其为法向的两个切平面的夹角，因此可以作为顶点 O 处接触角的估计。

为了使接触角的值等于稳定接触角，我们将接触点沿固体表面前后移动。通过给接触点施加边界力的方法动态的实现接触角调整。实际物理过程中，还存在滞回现象^[86]。而边界力对于滞回现象也能够很好的模拟。边界力公式为：

$$f_{\text{bnd}} = \begin{cases} 0 & \theta_c^r < \theta < \theta_c^a \\ \alpha(\theta - \theta_c^r)n_p|n_p|^{-1} & \theta < \theta_c^r \\ \alpha(\theta - \theta_c^a)n_p|n_p|^{-1} & \theta > \theta_c^a \end{cases} \quad (2.8)$$

其中 θ_c^r 和 θ_c^a 分别是后退和前进接触角， α 是边界力大小的系数， n_p 是水面法向 n_L 在固体表面的投影方向（图2.6）。公式2.8的直观解释是接触点总是朝着能够使边界力减小的方向移动，而且由于后退和前进接触角数值不同，存在一定的滞回区域，在整个区域内都能够保持稳定。通过这个公式，不仅可以实现图2.17中那样水滴在倾斜玻璃板上静止的效果，甚至在边界力与表面张力的共同作用下，还能够实现图2.19中水滴被悬挂在物体底部的效果。

2.3.4 体积修正

网格形变运算和拓扑运算并不能保证水滴体积恒定。尤其是平均曲率流运算，会使网格的体积不断缩小，如图2.7。为此本文提出了两种体积修正运算，局部体积修正和全局体积修正，避免体积误差积累导致最终产生明显的体积变化。

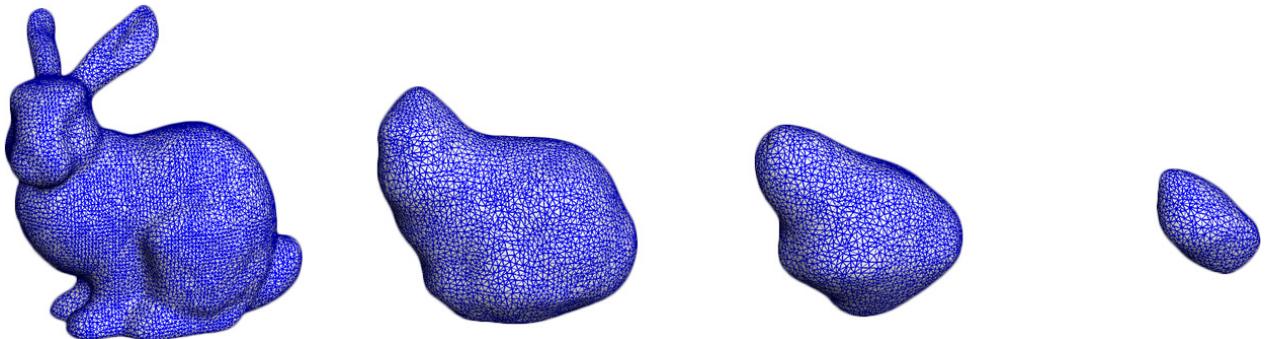


图 2.7 平均曲率流运算过程中的体积损失 由于闭合曲面的总曲率为正（向外突出），因此平均曲率流趋向于使网格缩小。如果不进行体积修正，网格就会在平均曲率流的作用下逐渐收缩，最终消失。

局部体积修正通过修正形变的速度来隐式的修正体积变化。Müller^[136]和Eckstein^[140]等人在平均曲率流计算中引入了体积修正，而本文提出的体积修正方法能够修正所有运算中水滴的体积误差（包括平均曲率流运算中的误差）。首先计算每一个水滴的刚性速度（在仿真中将每个水滴都当作刚体计算平移速度和旋转速度，并显式的记录），并将其从网格的速度场中去除，这样剩余的局部速度 \mathbf{u}_i 才使得水滴产生形变，并导致体积变化。设 a_i 为单位表面积上的体积变化率，计算方法为：

$$a_i = \mathbf{u}_i \cdot \mathbf{n}_i \quad (2.9)$$

其中 \mathbf{n}_i 是流体表面在顶点*i*处的法向。同时计算出顶点*i*邻域 $N(i)$ 内 a_i 的均值 \bar{a}_i ：

$$\bar{a}_i = \frac{\sum_{j \in N(i)} A_j a_j}{\sum_{j \in N(i)} A_j} \quad (2.10)$$

其中 A_i 是顶点*i*的混合面积（定义参见^[141]）。将 \bar{a}_i 从 \mathbf{u}_i 中除去：

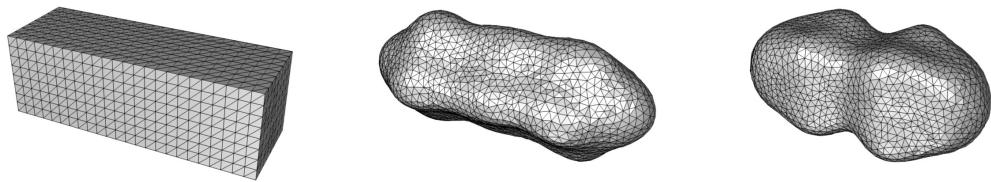
$$\mathbf{u}_i = \mathbf{u}_i - \bar{a}_i \mathbf{n}_i \quad (2.11)$$

此方法同Eckstein等人^[140]提出的局部体积修正方法类似，只是他们是对平均曲率流直接修正，而我们是对导致形变的速度 \mathbf{u}_i 来修正。这样可以使局部体积修正的覆盖面更广，对所有引起体积变化的形变都有效。

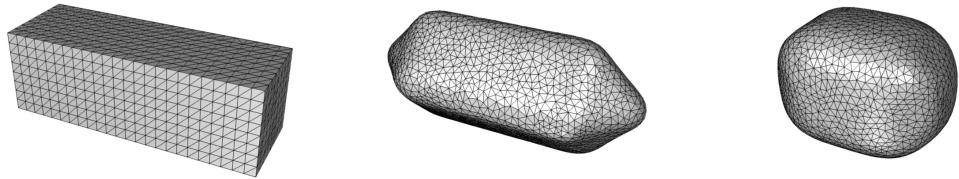
全局体积修正则是显式的直接修正体积变化。首先计算当前体积与水滴的初始体积之差，然后计算为消除体积差每个顶点需要移动的距离 $d = \Delta V / A$ ，其中 ΔV 是体积差， A 是总的表面积。最后将顶点沿其法向移动 $\mathbf{u}_i = \mathbf{u}_i + d \mathbf{n}_i$ 。由于体积差总是通过显式记录的初始体积来计算，因此不存在累积误差；而且每次迭代 ΔV 都很小，因此全局体积修正可以达到很高的精度。

通过实验，我们发现局部体积修正能够比全局体积修正保留更多的表面波纹（如图2.8所示），然而却不能像全局体积修正那样没有累积误差。这是因为公式2.10中， \bar{a}_i 只是顶点*i*处体积变化速度的估算，没有初始体积作为约束，因此会存在累积误差。我们需要根据应用来选择合适的体积修正方法。对大量的水滴进行模拟时，几乎观察不到单个水滴表面的波纹，因此只用全局体积修正。当动画需要突出这些细节波动时，首先进行局部体积修正，然后再进行全局体积修正。

当体积修正完成后，再把水滴的刚性速度加回 \mathbf{u}_i 中。此时网格表面的速度场便是能够保持体积守恒的速度场。



(a)局部体积修正，网格波动效果明显。



(b)全局体积修正，网格波动效果很弱。

图 2.8 局部体积修正与全局体积修正效果对比 一个初始状态为长条形的水滴在表面张力作用下产生形变。**(a)**局部体积修正中，每个顶点修正量都取决于其邻域，因此液面会像波浪一样传递运动。**(b)**全局体积修正中，顶点的修正量取决于整个面网格，因此不再具有波动效果，只有因顶点惯性导致的震动。

2.4 网格拓扑运算

前文中介绍了对单个水滴进行物理仿真的网格形变运算，本节我们提出模拟水滴之间融合与分裂的网格拓扑运算。网格拓扑运算的作用是处理网格拓扑变化，包括网格融合、网格分裂，调整网格采样密度，以及提高网格质量。与Wojtan等人^[75;76]依赖于体素的动态网格算法不同，我们的方法只依赖于面网格本身。几种网格拓扑运算之间并不是完全独立的，比如网格切分本质上是对网格优化中“短边移除”的特殊处理，因此本节依照概念提出的次序，依次介绍网格优化、网格切分和网格融合。实际运算中，则是先进行网格融合运算，然后在网格优化的同时完成网格切分，这样在网格融合运算中产生的许多非常短的边，都能够被直接消除，从而保证网格的质量。

2.4.1 网格优化

网格优化的目标是提高网格质量，保证网格形变运算的稳定性，尤其是平均曲率计算的正确性。在每个时间步长里，都会进行网格的局部拓扑优化，主要包括Hoppe等人^[142]提出的三个拓扑操作：短边移除、长边切割以及边翻转，如图2.9所示。对于长度超

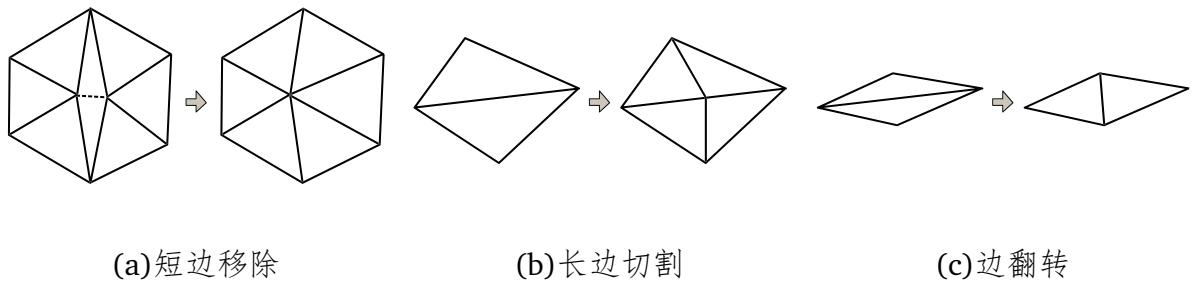


图 2.9 网格优化 通过三种方法对网格进行优化，短边移除、长边切割，以及边翻转，使得网格所有的边长都在一定范围内，避免狭长三角形的出现。通过网格优化，水滴网格的质量得到提高，从而提升了网格形变运算的稳定性。

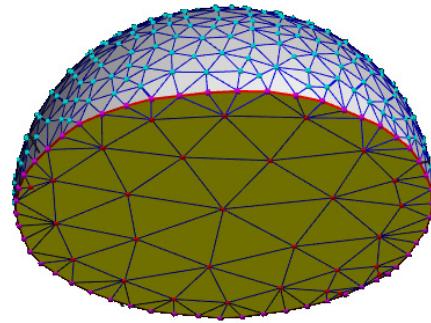


图 2.10 自适应网格 由于水滴与固体表面接触的部分对水滴的形状没有影响，因此为了提高计算速度，我们通过自适应采样，降低接触面上顶点的密度。

过阈值的边，将其从中间切割；对于长度小于阈值的边，将其移除。我们将边长度下限设为0.04cm，长度上限是其三倍，这样可以保证最佳的网格质量^[143]。为了提高速度，我们动态地维护两个优先队列（一个最大堆，一个最小堆），使得能够快速的按边长顺序对边进行访问。首先进行短边移除。每次循环都通过最小堆找到当前最短的边将其移除，而移除过程中会导致与其相邻的边长度发生变化，因此需要将这些边在两个堆中的位置都进行更新。重复这个过程，直到最小的边长度大于阈值。长边切割的方法与之类似。经过这一过程，网格的质量将得到极大的提升。然后再将对角余弦和（参见图2.4）为负数（意味着违背了Delaunay条件）的边进行翻转，进一步对网格进行优化。网格的质量对网格形变运算的计算速度与计算稳定性有直接影响，因此在整个实时系统中有着非常重要的作用。

在这套网格优化框架下，自适应网格可以通过对每条边长度乘以一个权重而方便的实现，如图2.10所示。比如，当一条边处于流体与固体交界的平面上时，就给这条边的长度乘以一个小的权重，相当于在做短边移除和长边切割的时候，长度阈值被放大，因此该区域顶点的密度就会显著降低。在不需要顶点密集采样的区域（如水滴与固体的接触面上），

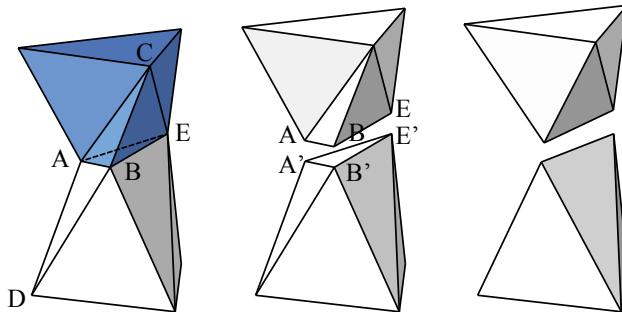
就可以在不增加额外计算量的前提下降低网格顶点密度。

水滴与固体接触线（图2.6）上的边在网格拓扑运算中需要特殊处理。这条线的形状是显式构建的，不能轻易改变，否则就会使水滴的形状产生明显变化，甚至会影响到计算稳定性。因此这条线上的边不能进行翻转操作，因为边翻转会切断接触线；但可以进行短边移除和长边切割，因为这两个操作不会改变接触线的连续性，而且还能提高接触线上的网格质量。

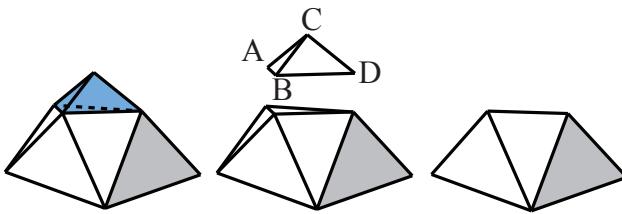
2.4.2 网格切分

网格切分模拟的是大水滴分裂成小水滴。网格切分并非独立的运算，而是在短边移除的过程中，对特殊拓扑结构的边进行特殊处理而实现的。具体来说，如果通过移除一个短边，发现网格就无法保证流形性质（连接两点的边不止一条——出现细管结构），那么此时就应对网格进行切分。

流形网格中，两相邻顶点应该被唯一的一条边相连。在图2.11(a)(左)中，边A-B的长度已经小于边长最小阈值，而且是网格中长度最短的边，因此在短边移除操作中将被首先移除。所有的流形网格中，顶点A和B的公共邻接顶点C和D都会存在，其特征是C和D分别包含在边A-B的邻接面中。但是在图示的情形中，顶点E也是其公共邻接顶点，却不包含在边A-B的邻接面中。此时如果将A-B直接移除，那么就将形成非流形的细管结构：点A（将



(a) 如果移除短边A-B将形成细管状结构，则在应当沿A-B-E切分网格



(b) 如果网格切分后一部分形成四面体A-B-C-D，则将其直接删除

图 2.11 网格切分 网格切分模拟大水滴分裂成为小水滴，通过在短边移除操作中，对符合特殊拓扑结构的边进行特殊处理而实现。

点B并入点A) 和点E将有两条边相连(原A-E边和原B-E边)。网格拓扑运算的每一步操作都必须保持网格的流形特性，所以在这种情况下，我们的操作是将整个网格沿着A-B-E剪开，然后添加两个面ABE和A'B'E'把网格补为封闭曲面。注意此时图2.11(a)(中)上下两半分离表示网格已经切分成两块，实际上网格剪开之后并不移动点的位置，A和A'在空间中是重合的，网格只是拓扑结构上被分离。(同理B和B'，E和E')。新生成的边A-B，A'-B'长度与原边A-B相同，因此仍然是网格中最短的边，在继续进行的短边移除操作中将被依次移除，从而形成图2.11(a)(右)所示的形状，也就完成了一次网格切分。

在短边移除操作中可能出现如下的情形：包含待移除边的连通网格只是一个四面体，如图2.11(b)示。这种情形出现的原因是在网格切分操作时，我们切下的其实是网格的一个尖角，如图2.11(b)(中)所示。这个尖角体积很小，而且只包含几个顶点，继续移除短边时就只剩下了一个四面体。如果四面体的边还要被继续移除，那么就将整个四面体一起删除，如图2.11(b)(右)所示。

网格优化过程中，如果对短边移除操作时进行了网格切分，就通过网格连通性检测，将所有被分开的子网格都作为一个独立的水滴。在此过程中的体积损失平均分摊到各个子网格上，并通过全局体积修正运算来修正。

2.4.3 网格融合

为了模拟两个小水滴相撞，并融合成一个大水滴的情形，需要对网格进行融合运算。计算过程中，每个水滴都动态维护一个包围盒，如果两个水滴的包围盒相交，就进一步通过边-面相交测试判断网格是否相交。为了加速相交检测，我们为参与计算的网格构建AABB树。如果经测试两个网格的确发生相交，即两个水滴应当融合，就对这两个网格进行布尔求并操作，方法如图2.12所示。

首先用Baraff等人^[34]提出的方法算出两个网格的交线。两个无自交的流形网格相交，其交线一定是一条(或几条)闭合曲线。曲线由两个网格所有的边-面交点依次连接而成。然后沿交线将网格切开，对被切开的面重新三角化，使得两个网格在切线上有一致的拓扑结构。接着以切线为分界线，对网格进行分割标记^[34]，判断每一个三角形位于另一个网格的内部还是外部。每个网格上被分割开的每一块，一定完全位于另一个网格的内部或外部。可以通过选取网格上一点向某一方向发射线与另一个网格求交，数交点个数的奇偶性来判断。最后我们将所有位于另一个网格内部的面删除，将余下的面拼接起来，于是形成一个封闭的网格。因三角化而产生的大量短边，将在网格优化过程中被消除。在此过程中损失的体积，通过全局体积修正运算(2.3.4节)来补偿。

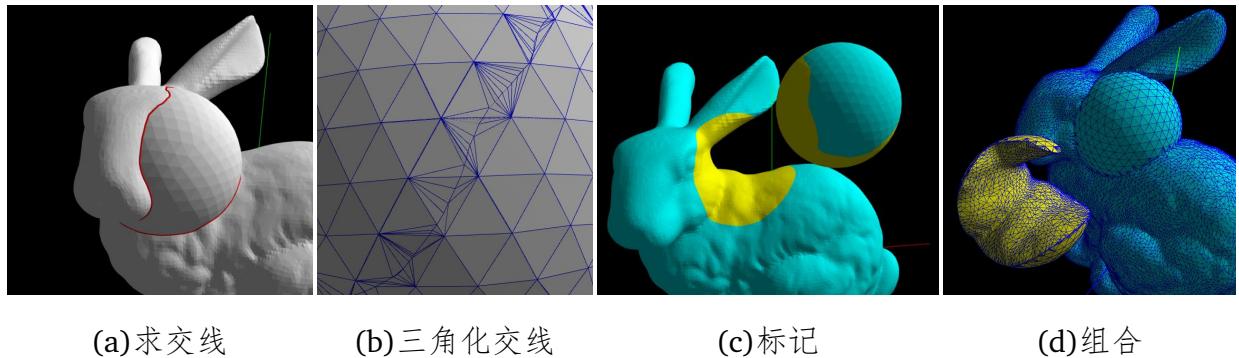


图 2.12 网格融合 网格融合模拟两个小水滴融合为一个大水滴，其实质是网格求并布尔运算。

(a)首先计算两个网格的交线。两个无自交的流形网格相交，交线是一条（或几条）封闭曲线。曲线由两个网格所有的边-面交点依次连接而成。(b)然后沿交线将两个网格切分，并重三角化。(c)在每个网格上标记出所有位于另一个网格外部的面，(d)用这些面组成新的网格，结果即为两个初始网格的布尔“并”。于是两个小水滴便融合成为一个大水滴。

网格的自交的检测和处理与此类似。不过通过实验发现在实时仿真实验中，自交检测并无必要。这是因为水滴表面张力的效果是使其形状变得圆滑，因此即便网格暂时存在自交，也会很快消失，在大量水滴存在时不影响视觉效果。相反，自交检测非常耗时，通常运算时间是两个网格相交检测的十倍以上。这是因为相交检测时通过AABB树来剔除大量距离很远的线和面，而自交检测时所有的线和面距离都很近，AABB树就无法充分发挥效果。但在网格存在自交时，用上述方法进行网格求并可能会出现错误。比如交线可能不再是闭合曲线。因此，如果网格求并运算中出现任何错误，可能就是网格自交导致的。我们的应对方法是暂时不进行网格融合，直到网格形变使错误消失后再进行融合。

2.5 实验结果

本文用所提出的表面网格直接形变的方法制作了一系列的流体动画。我们用一台8核Xeon 2.4GHz处理器，16G RAM的工作站对系统进行测试。实验中，场景包含1万到5万个三角形，计算速度通常能达到10FPS到50FPS。平均58%的时间用于求解隐式曲率流并完成网格形变，30%的时间用于碰撞检测，其余时间用于对网格进行拓扑操作，包括网格融合、分裂，以及网格质量优化。通过一系列的验证性实验，证明了仿真结果与真实物理现象具有较高的吻合程度。并进行了实时水滴仿真测试，来证明系统的高效性与可靠性。

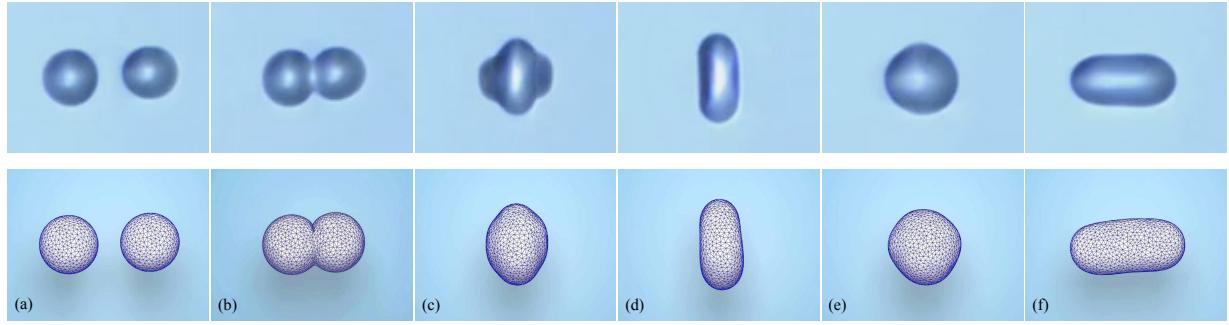


图 2.13 水滴相撞仿真结果与物理实验对比 上方为真实物理实验², 下方为仿真结果。

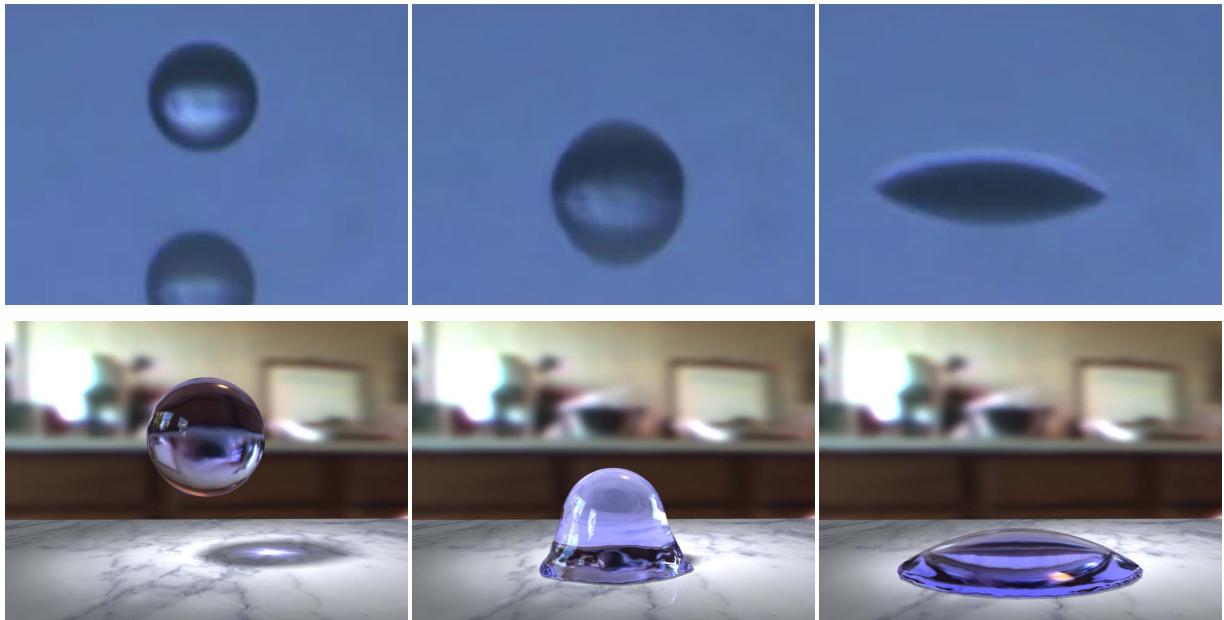


图 2.14 水滴落在亲水材料表面仿真结果与物理实验对比 上方为实验结果⁴, 下方为仿真结果。

2.5.1 验证性实验

- 水滴相撞

这个测试对两个水滴正面相撞进行了仿真，并同真实物理实验的录像进行了对比，结果如图2.13所示。可以看出，尽管本文的方法非常简单，但还是能够准确的模拟出两个水滴碰撞过程中产生的纹波。

- 水滴向固体表面滴落

为了验证接触角运算在极限条件下的可靠性，我们对水滴落在并附着在亲水固体表面的过程进行仿真，其过程如图2.14所示。水滴在接触到固体表面之后，由于表面的

³图片来源：http://www.youtube.com/watch?v=uVQS2W0_r7U

⁴图片来源：<http://www.youtube.com/watch?v=m1KKbJo4nYk&feature=related>

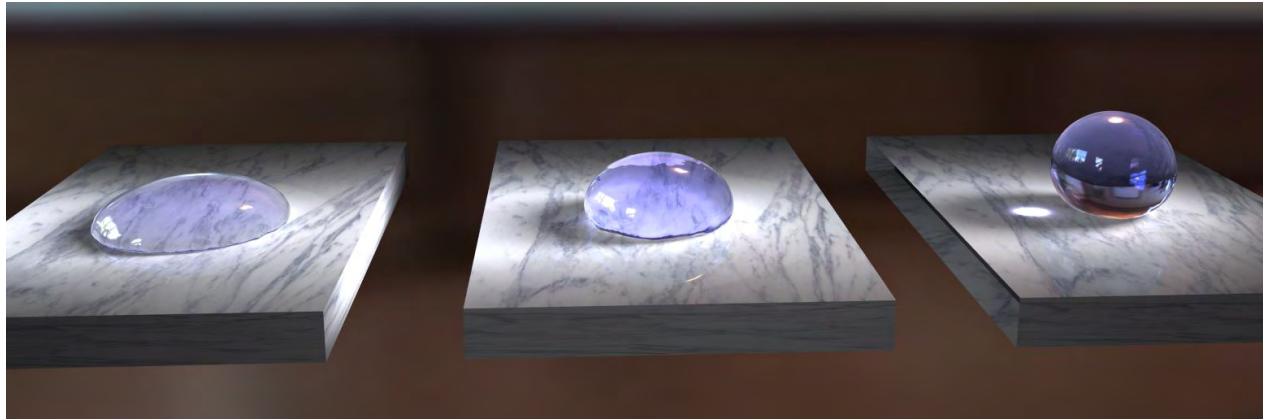


图 2.15 不同表面张力系数的水滴静止效果 固体为完全疏水的表面，从左到右水滴体积相同，表面张力系数分别为0.05, 0.2, 1.0。



图 2.16 水柱截断效果 水柱因Rayleigh-Plateau不稳定性会被截断，不同表面张力系数效果会有差别。该实验仿真速度为50FPS，从左往右表面张力系数为0.01, 0.2, 1.0。

亲水性快速推开，并达到稳定。同时与实际的物理实验做了对比，与仿真结果具有较高的一致性。

- 平面上的静止水滴

图2.15展示了不同表面张力系数下，静止在疏水固体表面的水滴呈现的不同形态。图2.5是相同表面张力系数水滴在不同亲水性固体表面（不同稳定接触角）呈现的形态。尽管本文的方法是物理模型的高度简化，但这些参数能够让我们获得丰富的近乎真实的水滴效果。

2.5.2 实时流体动画

- 水龙头出水

图2.16展示的是水龙头出水的仿真效果。水柱由于表面张力不均匀而导致的Rayleigh-Plateau不稳定性^[61]，会将水柱截断为小水滴。表面张力系数越大，水柱就越容易被



图 2.17 数百个水滴落在玻璃板上的效果 该实验对水滴在固体表面滑动的现象进行了仿真。小水滴会融合成大水滴，滑到玻璃板边缘会滴落。该实验仿真速度能达到30FPS，还允许用户对水滴进行交互。



图 2.18 水滴落在荷叶上的效果 该场景中水滴包含约3万个三角形，仿真速度为35FPS。

截断。该实验运行速度为50FPS。

- 雨中的玻璃板

图2.17展示的是数百个水滴落在玻璃板上，融合、滑动的效果。由于玻璃板的亲水性，水滴被吸附在玻璃表面。当水滴体积继续增加，其重力也越重，因此开始在玻璃板上滑动，最终达到恒定速度。这个实时系统还可以实现用户交互。用户通过鼠标，控制一个擦子在玻璃板上运动，擦子运动会将其轨迹上的水滴推开。整个系统的运行速度会随着水滴的增多而下降，但当场景包含5万个三角形的时候，仿真速度仍然能够达到30FPS。整个场景通过OpenGL实时绘制。

- 雨中的荷叶

图2.18展示的是水滴掉落在荷叶上的效果。荷叶是完全疏水的表面，因此接触角为180°。在仿真过程中，总共约180个水滴，共包含3万个三角形。仿真速度约为35FPS。

- 水滴落在复杂曲面上



图 2.19 水滴落在小鸟模型上的效果 该场景中水滴包含约1万个三角形，仿真速度为10FPS。

图2.19展示的是水滴落在在复杂曲面上的效果。为了加速碰撞检测，我们给模型也建了AABB树。物体表面是光滑的，曲面上的接触角与平面上计算方法基本相同。最小接触角设为 40° 。当亲水性强到足以吸附住水滴时，水滴就会悬挂在模型的底部，并因重力拉扯而断裂成小水滴。整个场景包含约1万个三角形，仿真速度约10FPS。

2.6 小结

本文提出了一种基于面网格的实时水滴仿真方法。该方法通过对面网格直接形变，能够对水滴的运动，及其与物体表面接触并滑动的现象进行仿真。该方法可以分为两部分：网格形变运算，对单个水滴进行物理仿真；网格拓扑运算，处理水滴的融合与分裂，并对网格进行优化。该方法可以对水滴的多种现象进行仿真，也可以将其集成到现有的流体仿真系统中，为通用流体仿真增强小尺度仿真的细节。

目前，我们的方法还存在一定的缺陷。首先，尽管面网格直接形变的方法给仿真带来了速度的提升，但由于模型高度简化，完全不考虑水滴内部的运动，因此同基于体素的欧拉法流体仿真相比，准确性要差一些。其次，当水滴不断汇集、变成大体积流体之后，粘性和表面张力的作用就不那么显著，该方法也就不再适用了。再次，尽管用反向欧拉求解平均曲率流，但由于会受到网格质量的影响很大，并不能保证一定收敛。步长越大，求解就越可能出问题。最后，在模拟水滴在曲面上滑动时，我们要求曲面是光滑的。对于更加复杂的曲面形状，比如带有尖角的曲面，接触角操作可能会产生不稳定。

接下来的工作，我们希望将这套系统完全在GPU上实现，以进一步提升速度。同时还希望能够找到更好的自适应网格构造方法，进一步减小计算开销。还有如何能够进一步提升系统的稳定性，比如处理网格自交的情况等。

第3章 粘性流体薄膜仿真与可计算水转印刷

在本章中，我们用面网格对浮在水面上的粘稠流体薄膜进行仿真，模拟固体物经过薄膜浸入水中，薄膜被拉伸并包裹住物体表面的过程。该物理过程是曲面着色工艺——水转印的虚拟化流程。水转印是把图案印刷在一种薄膜上，并将其平铺在水面上。喷洒活化剂后，这种薄膜就变为像胶水一样的粘稠流体。将待着色物体慢慢通过这层膜浸入水中，薄膜会贴在物体的表面，并逐渐将物体包裹起来，从而把颜料转印到物体表面。因此通过对该物理现象进行仿真，就能够将水转印虚拟化。

水转印可以对各种不规则曲面、多种材料的模型进行高质量着色，因此已经在工业中广泛应用于曲面纹理的印刷。然而，在水转印过程中，水转印膜因拉伸会发生形变，因此很难在曲面上印刷与模型对齐的图案。而且传统的水转印工艺通常依赖于手工操作，很难被普通用户使用。

本章提出了“计算水转印刷”的方法。该方法兼具传统水转印工艺印刷质量高、适用范围广的优点，同时解决了对复杂曲面印刷时图案与模型难以对齐的问题。我们提出了水转印过程的计算模型。通过对整个水转印物理过程进行仿真，可以根据水转印膜的形变，得到预变形图案，使得印有这个图案的水转印膜在形变之后，能够在模型表面刚好形成想要的纹理。为了真实重现与虚拟仿真相同的物理过程，我们采用常见的机械零件设计并搭建了一套定位水转印设备，用于实现模型定位，以及水转印中模型的运动控制。为了解决复杂曲面着色过程中水转印膜拉伸过大导致颜色失真、甚至膜破裂的问题，进一步将单次水转印扩展为多次水转印，即每次转印只对模型的局部进行着色，经过多次水转印颜料在物体表面叠加形成的图案，与设计的模型纹理刚好一致。通过实验，验证了所提出计算模型的精度，并证明了整套系统能够对多种曲面进行高质量的着色。

3.1 问题的提出

3D打印作为一种新型的快速原型制造技术，无论是在工业中还是在科研领域都被越来越多的关注。在计算机图形学领域，近几年对3D打印的研究主要集中在几何造型^[107]、物

理仿真^[106;111;113]以及表面反射特性^[101;144]等方面。除了3D打印物体的几何形状外，打印使用的材料以及模型表面的纹理图案，对于最终成型的产品外观同样非常重要。然而这些还是当前研究领域的薄弱环节。

普通家用级3D打印机每次只能使用一种材料，进行单一颜色打印。目前一些高端3D打印机实现了彩色3D打印，但仅限于少数几种材料（如塑料、石膏），而且这种打印机的价格很高，无法面向普通用户。工业中有一些比较成熟的三维曲面着色技术，包括电镀^[145]、贴花^[146]、上釉^[147]，甚至是手绘等。但是这些技术往往需要专门的设备，熟练的操作人员才能实现，因此这些方法主要用于大规模生产，而且不能保证可以用于复杂形状曲面。对于个性化定制的产品，这些着色方法并不一定适合。

我们提出了“计算水转印刷”的方法，首次将计算引入传统水转印工艺，实现了对个性化订制彩色三维模型的物理着色。该方法不仅能够适应多种复杂曲面，还能够用于多种材质的物体，如塑料、木材、陶瓷等。整套系统造价低廉，耗材费用也很低（单次转印的成本不超过人民币3元），具有成为家用级产品的潜力。

3.1.1 传统水转印工艺

水转印是工业中用于复杂曲面着色的方法^[148]。首先把图案打印在聚乙烯醇（PVA）膜上（图3.1(a)），把膜平整的置于水面上使其漂浮，然后将活化剂均匀喷洒在PVA膜上。PVA膜一旦被活化，就会变得非常柔软，像一层浮在水面的胶水一样（图3.1(b)），可以非常容易被拉伸。最后将待着色物体慢慢通过活化的PVA膜，浸入水中。PVA膜一旦接触物体，就会贴在物体的表面，并随着物体浸入不断拉伸，将物体包裹起来。在此过程中，打



图 3.1 打印了图案的聚乙烯醇（PVA）膜 (a) 出厂时的PVA膜贴在一张光面A4纸上，因此使用普通喷墨打印机就可以在上面打印图案。(b) PVA膜从光面纸上揭下，四周用胶带固定并放置在水面，被活化之后就可以对物体转印图案了。



(a)喷活化剂

(b)浸入物体

(c)转印完成

图 3.2 传统水转印操作流程 将水转印膜（PVA膜）平铺在水面之后，(a) 在其表面均匀喷洒活化剂，使其与水转印膜的材料发生化学反应，变成一层漂浮在水面的粘稠液体。(b) 然后将待着色物体缓慢浸入水中，水转印膜就开始被拉伸并包裹在物体表面。(c) 当物体完全浸入水中之后，在水中轻轻晃动物体，清除水面上漂浮的膜，并将物体捞出。用水冲洗掉表面的胶状物，并烘干物体之后，即完成着色。

(a)仿碳纤维汽车装饰¹(b)仿大理石建材²

图 3.3 传统水转印工艺着色的产品 水转印能够为多种材料进行着色，但由于模型与图案无法精确对齐，因此只能转印重复的图案。

印在膜上的颜料也会随着PVA膜一起运动，并最终印在物体的表面。水转印对物体的材质没有特殊要求，因此能够在多种材质的曲面上印刷高质量的图案。

然而，如果用水转印技术在曲面上印刷需要与其形状对齐的图案，传统水转印工艺就会存在问题。首先，传统水转印中模型入水多是人工操作（每个模型都有最佳的入水角度，甚至入水的过程中还要旋转模型），因此很难保证曲面与图案完全对齐。其次在模型入水过程中，PVA膜被拉伸会使图案被扭曲，甚至膜因拉伸过大而破裂。因此传统水转印工艺印刷的图案多是些重复的不需要对齐的纹理，如木纹、大理石纹等。主要产品有汽车木纹内饰、仿大理石建材、仿碳纤维装饰等，如图3.3所示。

¹图片来源: produto.mercadolivre.com.br²图片来源: www.dg114.com



(a)待着色模型

(b)传统工艺 (重复纹理)

(c)计算水转印刷

图 3.4 传统水转印与计算水转印适用范围对比 同传统水转印工艺相比，通过对水转印过程进行仿真，计算出预形变的图案，通过定位水转印贴在物体表面之后能够刚好与目标纹理一致，因此着色不再限制于重复纹理，而是能够按照用户设计的纹理为物体着色。

3.1.2 对传统水转印工艺的改进

本文通过计算方法解决了传统水转印工艺中曲面与图案无法精确对齐的问题。我们对水转印过程进行了虚拟仿真，活化的PVA膜建模为非线性粘性薄膜模型，模型的边界条件由物体入水的过程动态更新。通过模拟物体入水的全过程，模拟出PVA膜在水转印过程中的拉伸与形变，并且计算出膜上的点与其贴到物体表面上位置的映射关系。通过该映射可以反算出需要在PVA膜上打印的预变形图案，用这个图案来做水转印就能刚好给曲面上想要的纹理。

为了给真实物体印上图案，还需要真实的重现虚拟仿真中的物理过程。因此我们设计并制造了一套定位水转印设备，包含一套基于3D视觉的定位系统，用于给物体精确定位；以及物体夹持与运动装置，可以控制物体以给定速度匀速运动。整套设备是用常见的零件来搭建，因此成本很低，普通用户都可以很容易实现。这套设备与计算模型结合起来，能够通过水转印将纹理图案精确的转印到物体表面。这样就能够做出传统水转印工艺无法实现的需要曲面与图案对齐的产品，如图3.4所示。

为解决复杂曲面着色时，PVA膜因拉伸过大而产生严重失真甚至破裂的问题，我们进一步将单次水转印扩展为多次水转印，每次选取一个着色方向，计算出该方向上打印的图案，并通过水转印对物体的局部进行着色。物体表面多次着色叠加在一起的效果，与给定的模型纹理相同。

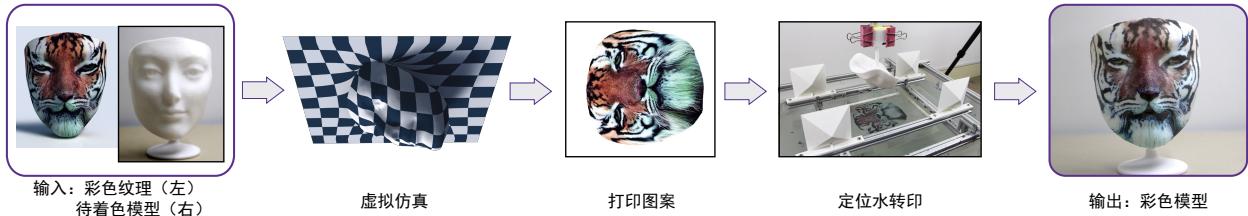


图 3.5 计算水转印刷系统流程图 系统的输入是一个待着色的实物模型，同时还需要有其贴了纹理的数字化模型。将待着色模型固定在转印设备上之后，通过视觉系统测量其位置和朝向，并用这些参数进行虚拟仿真。计算出PVA膜在水转印过程中的形变之后，根据形变计算预形变图案，并将其打印在PVA膜上。最后将膜放置于水面并活化，启动装置控制物体穿过活化的PVA膜浸入水中，最终得到转印上图案的彩色模型。

3.1.3 创新点

本工作的主要创新是首次提出水转印的计算模型，并建立起定位水转印系统，实现了三维实物按照设计纹理的精确着色。具体包括：

1. 首次提出了水转印物理过程的计算模型，可以对整个物理过程进行仿真，计算出PVA膜在此过程中的拉伸与形变。
2. 制造了能真实重现虚拟仿真的定位水转印装置，包括系统标定和运动控制。
3. 提出了对物体多次水转印的方法，有效的解决了复杂曲面着色时图案失真的问题。

3.2 算法概述

整套计算水转印刷系统的流程如图3.5所示。实现水转印刷的步骤如下：

1. **系统标定** 为了对水转印过程进行物理仿真，首先要对模型以及PVA膜的相对位置进行标定。将模型固定在定位水转印装置上，用一套基于深度相机的视觉系统，测量出模型相对于PVA膜的位置及朝向。
2. **虚拟仿真** 得到模型与PVA膜的相对位置之后，对整个水转印的物理过程进行模拟，计算出PVA膜在整个水转印过程中是如何被拉伸的。
3. **计算预形变图案** 在虚拟仿真过程中，得到了PVA膜上的每一点附着到模型表面上的坐标，从而建立起PVA膜与三维模型表面的映射关系。根据这个映射以及数字化模型的纹理，就可以计算出预形变的图案。

4. 物理实现 最后将预形变图案打印在PVA膜上，将PVA膜放置在水面上并固定其边界，然后开始水转印操作，PVA膜上的颜料就将被转印到物体的表面。对于复杂曲面模型，可以在不同方向上重复这个过程，通过多次转印实现整体的曲面着色。

整套系统的核心是虚拟仿真(3.3节)和物理实现(3.4节)，即计算预变形的图案，并通过设备控制实现水转印的物理过程。

3.3 虚拟仿真

计算水转印刷的核心是计算打印在PVA膜上的预形变图案，通过预形变使得图案被拉伸之后，能够在模型的表面呈现出设计的纹理。虚拟仿真的输入是三维模型的几何形状、表面的纹理、以及模型相对PVA膜的朝向和位置（计算方法将在3.4节描述）。目前我们搭建的定位水转印设备中，用于控制模型的直线电机是竖直安装的，因此虚拟仿真中的模型也只在竖直方向上运动。

虚拟仿真目的是要为PVA膜上的点与模型表面建立起一个映射关系。通过这个映射关系，可以查找到膜上的一点在水转印过程之后，贴在模型表面上的位置。因此这个映射关系反映的是水转印的物理过程，故只能通过虚拟仿真来构建。

3.3.1 粘性薄膜

在水转印开始之前，打印好图案的PVA膜被放置在水面上，随后在膜的表面均匀的喷洒一层活化剂。活化剂能够与PVA膜发生化学反应，使得PVA膜迅速变软并具有粘附性，从而在水面形成一层类似于胶水的粘性薄膜。随后待着色模型缓慢匀速下降，接触到薄膜并浸入水中。在此过程中，颜料连同PVA膜一起将模型包裹起来，并吸附在模型的表面。通

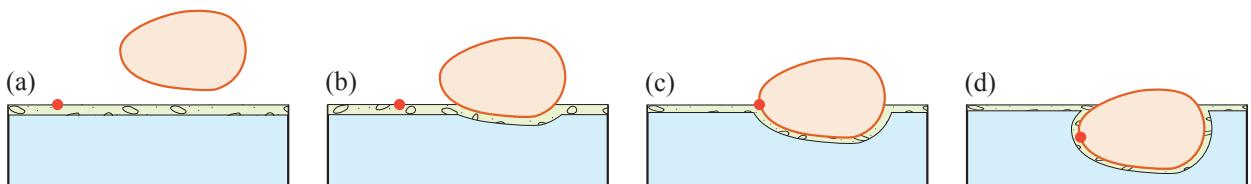


图 3.6 水转印中，PVA膜被拉伸并贴在物体表面的示意图 (a)一个物体开始竖直向下运动；红点表示浮在水面的PVA膜上的一点。(b)物体一部分开始入水。一部分PVA膜贴在物体的表面，浮在水面上的部分开始被拉伸，因此红点开始移动。(c)物体继续浸入水中。红点运动到物体表面并贴在物体上。(d)红点贴在物体上之后就牢牢地固定在物体表面并随着物体一起运动。

常，PVA膜的外边界是被胶带固定的，并且不可移动。已经附着在模型表面的PVA膜随模型一起下沉，而其余部分仍旧浮在水面，但获得了速度并开始拉伸。整个过程如图3.6所示。

基于上述观察，可以采用粘性薄膜模型来描述PVA膜的物理性质。在我们的系统中，由于物体入水速度恒定且缓慢（5mm/s），因此可以作如下假设：

1. 承载PVA膜的水处于静止稳态。
2. 粘性薄膜没有竖直方向的速度。
3. 粘性薄膜在水面上只有水平方向的速度，一旦接触物体就将贴在物体表面，并随物体一起竖直运动。

基于上述假设，可以将粘性薄膜简化到二维平面上来计算。在这个二维区域内薄膜向着物体与水面的交界线方向被拉伸，直到其运动到交界线之后贴在物体的表面。由于薄膜与水相对运动的摩擦力远小于薄膜拉伸的内力，因此可以将摩擦忽略。

将粘性薄膜简化到二维平面之后，需要计算其边界条件。PVA膜的外边界被胶带固定，因此其速度为0。PVA膜与模型接触部分的边界上（内边界），其速度可以根据几何关系来计算。对于内边界上一点 x ，其速度方向与模型在 x 点法向在水面的投影方向相同；由于薄膜持续被拉伸并附着在模型的表面，因此速度的大小与 x 点在物体表面的速度大小相同。我们用 $u(x), x \in \mathbb{R}^2$ 表示粘性薄膜的二维速度场。在我们的系统中模型在竖直方向匀速下降，因此根据几何关系可以得出边界上速度 $u(x)$ 的公式

$$u(x) = \begin{cases} v_d n(1 - \cos \theta) / \sin \theta, & x \in \Gamma_s, \\ 0, & x \in \Gamma_v, \end{cases} \quad (3.1)$$

其中， Γ_s 和 Γ_v 分别表示薄膜接触物体的边界和被固定的外边界； v_d 是模型下降的速度， θ 是模型在与 x 接触点处切平面与水平面之间的夹角， n 是薄膜边界 x 点朝向边界外部的法向。在整个水转印过程中，边界 Γ_v 始终不变，而 Γ_s 则根据模型的入水深度不停在更新。

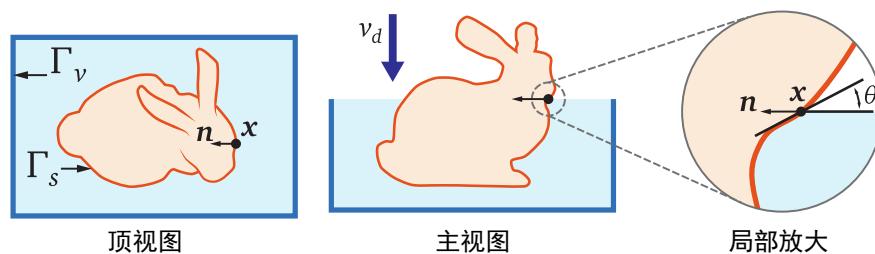


图 3.7 粘性薄膜的边界条件

当边界条件定下来之后，下一步就是要计算在给定边界条件下薄膜内部的形变。我们提出了两种计算方法：斯托克斯流模型以及非线性粘性薄膜模型。这两种计算方法在计算速度与计算精度上各有优势，因此在迭代计算时（3.3.5节）会将其结合使用。

3.3.2 斯托克斯流模型

由于薄膜在喷洒活化剂后变得具有黏性，而且模型浸入速度也很慢，这种粘性薄膜具有很小的雷诺数^{3[149]}，因此可以用斯托克斯流（Stokes Flow）来描述这种粘性薄膜的运动^[150]。在二维运动中，斯托克斯流被简单描述为一个拉普拉斯方程

$$\nabla^2 u(x) = 0, \quad x \in \Omega \quad (3.2)$$

其中 Ω 为粘性薄膜浮在水面上的区域，也就是其计算区域，如图3.8，其形状随着物体浸入水中不断在变化。公式3.2的边界条件为公式3.1。在数值计算中（3.3.5节），对时域离散化，然后在每个时间步长中求解公式3.2。通常流体仿真需要让散度为零，从而保证流体不可压缩的特性。但是薄膜由于拉伸而产生面积变化，同时会使薄膜的厚度也发生变化，从而保证体积守恒，因此在公式3.2中并不包含散度项。

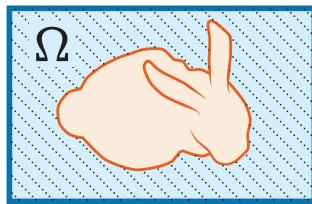


图 3.8 粘性薄膜的计算区域

公式3.2的计算可以非常快，因此在薄膜形变较小的时候（比如模型刚入水的阶段），可以非常快速的计算出结果。但是随着模型浸入水面越来越深，膜的形变也越来越大，形变的非线性也越来越强，线性的斯托克斯流模型计算精度就会下降。因此这时就需要采用非线性粘性薄膜模型来计算薄膜的形变。

3.3.3 非线性粘性薄膜模型

由于PVA膜在喷活化剂之后将表现出类似牛顿粘性流体的性质^[151]，因此采用Batty等人的论文^[54]中描述的方法作为非线性粘性薄膜模型。通过离散斯托克斯-瑞利类比，在每

³雷诺数，定义为 $Re = UL/\mu$ ，在粘性流体Navier-Stokes方程中是一个无量纲数。其中 U 为速度特征尺度， L 为长度特征尺度， μ 为流体的运动学黏性系数。

一个迭代周期中首先计算二维弹性形变，然后再计算粘性。具体来说，在这个模型中弹性力 $\mathbf{f}_{int}(x)$ 的大小与这一点材料弹性能量的变化速率有关。这些力将会使得粘性薄膜产生运动，其运动满足方程：

$$\rho \dot{\mathbf{u}} + \mathbf{f}_{int}(x) = 0 \quad (3.3)$$

由于本文中非粘性薄膜模型专门用于水转印过程的计算，因此对Batty论文^[54]中的公式进行了一定的简化。首先，在计算弹性形变能量的时候，忽略薄膜弯曲能量（bending energy），只考虑薄膜（拉伸）能量(membrane energy)。这是因为薄膜只在平面中运动，因此弯曲能量为零。薄膜能量是在水面上整张薄膜的薄膜能量密度的积分， $W_m(x), x \in \Omega$ 。如Gingold在论文中所述^[152]，薄膜能量密度通过应变张量来定义：

$$W_m(x) = \frac{Yh(x)}{2(1-\nu^2)}((1-\nu)\text{tr}(\epsilon^2) + \nu\text{tr}(\epsilon)^2) \quad (3.4)$$

其中 $h(x)$ 是薄膜在 x 点的厚度， tr 是弹性张量的迹， Y 是PVA膜的杨氏模量， ν 是PVA膜的泊松比。在计算不可压缩粘性流体时， $\nu = 0.5$ ；杨氏模量则通过实验估算（3.6节）。公式3.4中，薄膜应变 ϵ 采用格林应变张量，用于描述薄膜的平面内几何形变。通过定义薄膜能量，非线性薄膜的内力可以表示为整个薄膜区域 Ω 上的积分：

$$\mathbf{f}_{int}(x) = \nabla_x \left(\int_{\Omega} W_m(x) dS \right) \quad (3.5)$$

3.3.4 非均匀厚度薄膜

薄膜的厚度 $h(x)$ 需要特殊对待，可以简单假设在仿真的每一步过程中，薄膜在每一点处厚度恒定，一步计算完成后再假设体积恒定来更新薄膜厚度。然而，由于在PVA膜上打印上了颜料，颜料会增加薄膜的厚度。而且打印的图案不同，薄膜增加的厚度也不同。PVA膜本身就很薄（约0.5mm），由颜料带来的厚度增加，足以使薄膜局部的可拉伸性降低，并导致最终结果与仿真结果有明显的区别。在图3.9中，左侧面具的图案在计算过程中考虑了颜料厚度的影响，因此着色非常准确；而右侧面具未考虑颜料厚度影响，因此计算出的图案偏小。

然而在进行虚拟仿真前，薄膜上的图案却是未知的，也就无法根据颜色来设置薄膜在每一点处的厚度。为了解决这个问题，我们分成两步来求解：首先假设整张薄膜的厚度都相同，用均匀的薄膜进行虚拟仿真，通过这次仿真先算出来一幅图案。有了这幅图案之



图 3.9 颜料对薄膜拉伸程度的影响 水转印物理过程中，颜料的厚度会对PVA膜拉伸性产生影响：颜料越厚就越难拉伸，因此如果在仿真计算时就考虑颜料厚度，就可以提高计算的精度。左侧模型是仿真时考虑了颜料厚度的着色结果，模型与图案精确对齐。右侧模型是将PVA膜当成均一厚度仿真的着色结果，图案计算偏小，因而没有完全与模型对齐。

后，第二步就可以根据颜色来设置薄膜上每一点的厚度，用这个非均匀厚度薄膜作为另一个虚拟仿真的输入。这个过程可以迭代进行，直到算出的图案不再发生变化。不过根据实验结果，通常只需要两步迭代就能够达到收敛。因此在实际计算中，第一步采用斯托克斯流模型迅速计算并估算出薄膜每一点的厚度，接下来就采用非线性粘性薄膜模型用于进一步提高计算精度。

颜料使薄膜增加的厚度，可以通过打印机喷在薄膜上的墨水量来估算。将打印在薄膜上图片当做位图来处理，每一个像素的颜色都可以转换到CMYK颜色空间中。CMYK颜色值是打印机四原色（青、品红、黄、黑）每种颜料喷墨量占其最大喷墨量的百分比表达，每个通道的值为[0,1]，因此将这四通道的值加起来（最大4.0）可以作为这种颜色总喷墨量的估计。将这个值记为 t ，那么，喷墨厚度正比于 t 。通过实验，发现用 $\Delta h = 0.05t\text{mm}$ 作为颜料使薄膜增加的厚度可以获得不错的结果。

在计算过程中，我们不停地更新薄膜的厚度，使得薄膜的总体积保持恒定。薄膜采用三角网格的离散化表达，薄膜在三角形内厚度一致，因此在每一步计算的结尾，都将薄膜厚度更新为 $h(x) = h_0(x)A_0(x)/A_r(x)$ ，其中 $h_0(x)$ 是包含点 x 三角形的薄膜初始厚度， $A_0(x)$ 和 $A_r(x)$ 分别为包含点 x 三角形的初始面积和当前面积。

3.3.5 数值计算

斯托克斯流和非线性粘性薄膜都采用有限元方法（Finite Element Method）来计算。薄膜在水面上的部分用三角网格来表达。速度场 $u(x)$ 的值离散的存储在三角形的每个顶点上，在三角形内部用一阶线性插值来逼近。

3.3.5.1 动态边界的网格更新

在模型入水的过程中，薄膜的内边界就是模型与水面交界线。由于模型始终匀速下降，每一帧都能够计算模型入水的深度，并且更新与水面的交界线（薄膜内边界）。边界更新之后，先用几何算法库（如Triangle^[153]）重新对薄膜进行三角化，然后再进行物理仿真。由于交界线（薄膜内边界）平滑的变化，对网格变化也平滑处理（将部分上一帧的顶点作为新生成三角网格的采样点），以防止频繁拓扑变化导致插值误差。在实验中， Γ_s 的边长取1mm，三角形内角角度下限取30°，三角形面积上限取1cm²。在用斯托克斯流模型（3.3.2节）进行计算时，在更新之后的网格上直接构造并求解拉普拉斯方程。在用非线性粘性薄膜模型（3.3.3节）进行计算时，还需要用上一步计算的网格物理参数，通过重心差值为更新之后的网格物理参数赋值，然后再求解公式3.3。算法1是用非线性粘性薄膜模型一个时间步长的算法，与Batty等人的方法^[54]基本一致。

Algorithm 1 非线性粘性薄膜一个时间步长计算流程

- 1: 将模型匀速下降给定时间步长
 - 2: 更新边界线 Γ_s （见图3.7）
 - 3: 三角化 Ω 区域，其边界为 Γ_v 和 Γ_s
 - 4: 将上一帧的网格的速度场插值到新网格上
 - 5: 更新边界条件（公式3.1）
 - 6: 隐式欧拉法迭代一步
 - 7: 更新薄膜厚度： $h \leftarrow V_r/A_r$
 - 8: 插值出像素粒子的速度（见3.3.6节）
 - 9: 更新像素粒子位置
-

3.3.5.2 计算模型选择

我们提出了斯托克斯流和非线性粘性薄膜两个模型，都可以用于粘性流体的计算。斯托克斯流计算速度会快很多，但被限制只能模拟小形变。例如，对图3.15中的球进行虚拟仿真，斯托克斯流计算只需要7秒，而非线性粘性薄膜则需要5分钟。在实验中，我们发现如果水转印过程中最大局部拉伸率不超过0.3，那么斯托克斯流已经足够算出比较精确的结果。但超过这个拉伸率，误差会越来越大，就只能采用非线性粘性薄膜来进行仿真了。所以斯托克斯流被用于正式仿真前计算估计图案，用于估计薄膜的厚度。这样只用很小的开销，就能够很好的提升非线性粘性薄膜的计算精度。除此之外，在多次转印中（3.5节），斯托克斯流也用来快速计算各个方向薄膜的形变，从而对曲面进行分割。

3.3.6 纹理跟踪

虚拟仿真的目的就是为了建立一个PVA膜上的点与物体表面的映射关系，对于图像的每个像素可以通过该映射找到水转印后贴在物体表面的位置，读取这一点的颜色，从而生成预形变图案。由于打印机的分辨率有限，为了生成尽可能清晰地图案，图案尺寸根据打印机最高分辨率来生成。为了获取每一个像素的颜色，在每个像素中点生成一颗粒子（像素粒子），并且在虚拟仿真中，让其根据速度场 $u(x, t)$ 去移动，直到粒子贴在模型表面。从模型的纹理图案中读取贴合点的颜色，并将其赋予对应的像素。

在虚拟仿真的每一步计算中，对每一个像素粒子找到包含它的三角形，这一点的速度就从三角形顶点上插值得到。对于彩色薄膜上原本位于 x 点的像素粒子，我们跟踪其运动，直到贴在物体表面 \tilde{x} 点，于是就有了映射 $\phi(x) : x \mapsto \tilde{x}$ 。等模型完全浸入水中，得到所有的映射，并得到逆映射 $\phi^{-1}(\tilde{x}) : \tilde{x} \mapsto x$ ，这样就可以将 \tilde{x} 处模型的纹理颜色赋值给像素 $\phi^{-1}(\tilde{x})$ 。把所有这些映射合起来，就获得了需要打印的图案。

3.4 物理实现

当虚拟仿真完成，计算出预形变图案后，需要一套能够实现与仿真相同物理过程的系统，将虚拟仿真变为现实。该系统包含了一套基于深度相机的三维视觉系统，用于模型的定位，以及执行水转印操作的机械装置。

3.4.1 系统搭建

我们的系统如图3.10所示。整套系统由常用零件搭建而成，能够精确控制物体的朝向和下降位置。模型通过两个夹子被固定在直线电机的下端，通过电机控制能够在竖直方向匀速运动。电机的速度通过控制器，可以在0mm/s到10mm/s范围内连续可调。在实验中，物体的浸入速度都设为5mm/s。

装置的下方是一个恒温水池，温度通过加热棒恒定在30°C。打印好图案的PVA膜就放置在水面上，四周用胶带固定，位置用固定杆限定。水池上装有固定杆的限位块，因此可以保证每次PVA膜都被放置在相同的位置。我们使用A4尺寸的PVA膜，因此用普通喷墨打印机就可以在上面打印图案。

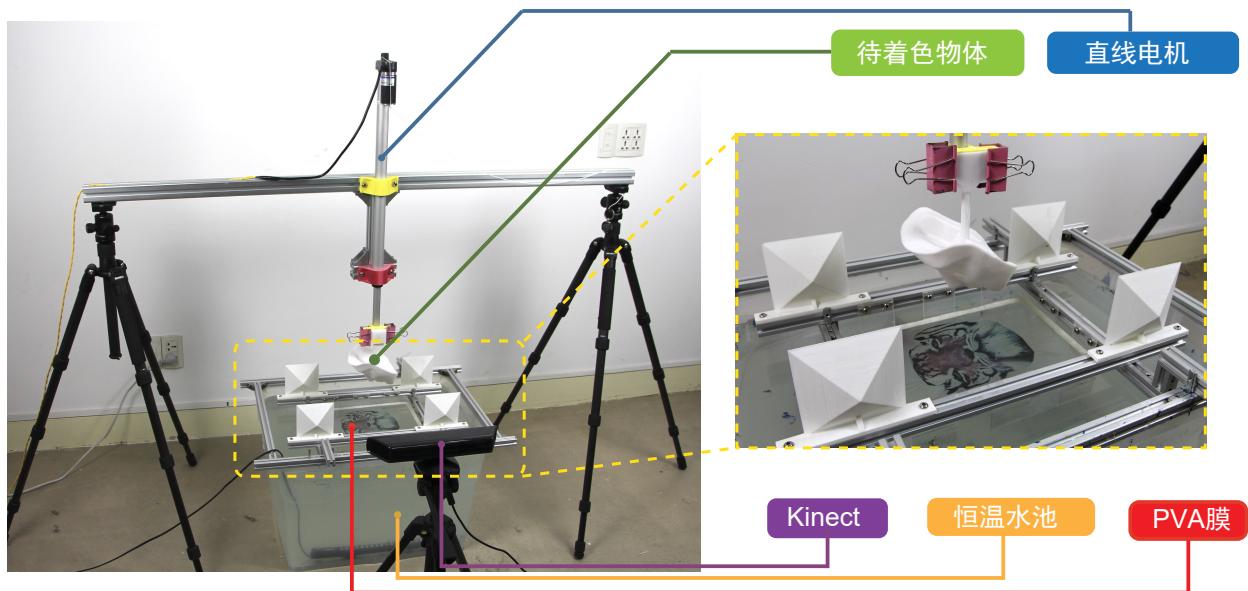


图 3.10 计算水转印刷设备 我们的设备主要包括一个盛满水的水池来承载PVA膜，一个能够加持并控制待着色物体竖直方向上下运动的直线电机，以及一个深度相机（Kinect）用于获取待着色物体的位置和朝向。

3.4.2 模型三维定位

由于目前的机械装置只能让模型上下运动，因此很难实现将模型自动放置在指定位置，然而模型的位置和朝向却是虚拟仿真所需要的。为了解决这个问题，首先把模型固定在装置上，然后通过三维扫描系统，计算出模型相对于PVA膜的位置和朝向。用这个位置和朝向进行虚拟仿真，得到预形变图案之后，再执行水转印操作。

为了获取模型的位置和朝向，我们用一台深度相机Kinect，通过KinectFusion算法^[154]扫描得到整个场景的3D点云。为了指示PVA膜的位置，在膜的固定杆上安装了四个金字塔形的3D模型，因此PVA膜相对于金字塔模型的位置也是已知的。由于金字塔模型和待着色模型的形状都是已知的，可以用ICP算法^[155]将其与扫描到的点云对齐，于是可以计算出待着色模型与金字塔模型的相对位置，从而计算出与PVA膜的相对位置。在实验中，这种定位方法的误差不超过1.5mm。

3.4.3 系统流水线总结

我们总结整套水转印系统流水线（图3.5）的工作原理。用户提供一个数字3D模型，模型的纹理，以及与数字模型完全一致的3D实物模型。把实物模型用夹子固定在装置上，用三维定位系统计算出实物模型相对于PVA膜的位置和朝向，然后通过虚拟仿真计算出预

形变图案，并用喷墨打印机打印在PVA膜上。最后将PVA膜固定在水面上，并执行水转印操作。

3.5 扩展：多次转印

我们的计算水转印方法能够为多种曲面进行高质量的着色，但是随着物体入水深度不断增加，薄膜拉伸率也不断增加，着色质量会因此下降。这是由于薄膜拉伸的实际物理过程，其实是高度非线性的，而用简单的数学模型来逼近，拉伸率越大，计算结果与物理实验结果的偏差也就越大。当拉伸率过大的时候，薄膜甚至还会被拉破。因此对于表面形状复杂的物体，单次水转印要让薄膜覆盖整个曲面几乎是不可能的。

单次转印可能会遇到困难的另一个原因是，当物体表面法线朝上，并且几乎与水面平行的时候（如圆球入水实验时，最后入水的那一点），公式3.1中的 θ 值接近 180° 。此时根据公式3.1计算出来的边界速度会非常大，在这种病态的边界条件下计算出来的结果误差也就会很大。

因此本文提出了水转印的一个扩展方法：对同一个物体在不同方向上多次转印。这个方法可行是因为打印图案上的某像素如果是白色，那么打印机这一点上不会喷墨水，因此PVA膜在此处是透明的，相当于这一点没有被转印。模型上的一点如果转印了多次，那么这一点上的颜色呈现的将是这几次着色中颜料混合的结果。

3.5.1 纹理分割

首先用户为物体指定多个转印的方向，据此将模型的纹理分割成多个区域。在物理操作时，每次转印只对物体的一部分进行着色，多次转印共同拼接出完整的纹理。因此需要计算在各个方向上转印的图案。对物体表面上一点，先假定这一点只在一次转印中被着色，着色方向的选取原则是选择使膜在这一点拉伸率最小的方向。用户指定转印方向的方法也是通过把物体固定在装置上，3D扫描计算出模型的位置和朝向（3.4.2节），再输入到虚拟仿真中的。

具体算法如下：把转印的方向记为 i ，然后用斯托克斯流方法（3.3.2节）对其进行虚拟仿真。通过纹理跟踪（3.3.6节），可以获得从二维薄膜到三维模型表面的映射图 $\phi_i(x)$ 。 $\phi_i(x)$ 的Jacobian就是膜上 x 点的形变，其最大奇异值就是 x 点处膜的局部拉伸率。对于一个三角形的三个顶点 $\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3 \in \mathbb{R}^3$ ，其在二维平面上的坐标为 $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3 \in \mathbb{R}^2$ 。因此在这个三角形中存在唯一的映射 $S(\mathbf{p}) : \mathbf{p} \mapsto \mathbf{q}$ ，计算公式为^[156]：

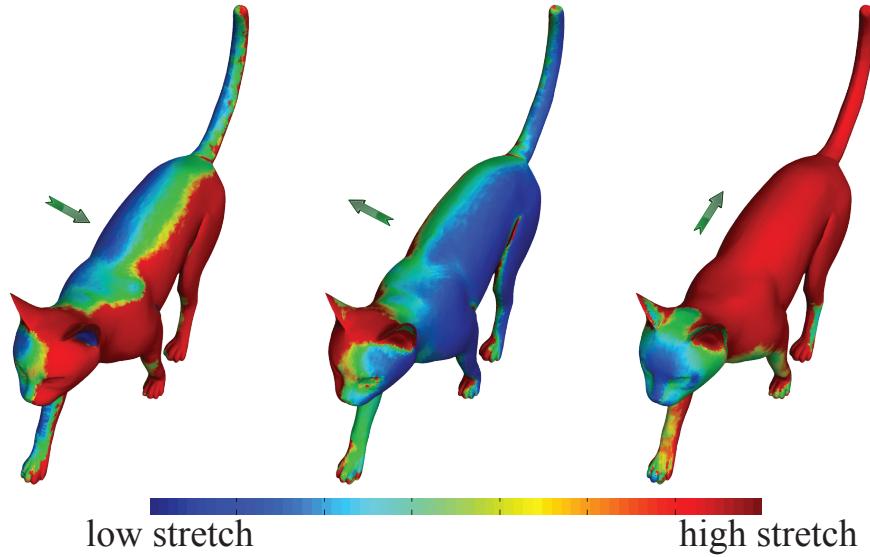


图 3.11 多次转印PVA膜在各转印方向上的拉伸率 将PVA膜贴在物体表面时的拉伸率进行可视化，绿色箭头表示物体三次转印中着色的方向。

$$S(\mathbf{p}) = \frac{1}{\langle \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3 \rangle} (\langle \mathbf{p}, \mathbf{p}_2, \mathbf{p}_3 \rangle \mathbf{q}_1 + \langle \mathbf{p}_3, \mathbf{p}_1, \mathbf{p} \rangle \mathbf{q}_2 + \langle \mathbf{p}_2, \mathbf{p}, \mathbf{p}_1 \rangle \mathbf{q}_3) \quad (3.6)$$

其中 $\langle \cdot, \cdot, \cdot \rangle$ 表示这三个顶点构成三角形的面积。这个仿射变换的Jacobian是一个 3×2 的矩阵，其最大奇异值就是这个三角形的最大拉伸率^[156]。计算出所有三角形的最大拉伸率之后，在 i 方向上每一个顶点的拉伸率记为与该顶点相连的所有三角形最大拉伸率的均值。对每个方向 i 都进行该计算，就得到了在各个方向上模型上每一点的拉伸率，如图3.11所示。

最后，对模型上每一点选取使其拉伸率最小的着色方向，这样就对模型表面进行了分割，分割的每一块都对应于一个着色方向。

3.5.2 计算单方向的转印图案

在上一步中我们对模型表面在各个转印方向的分割，计算拉伸率时采用斯托克斯流（3.3.2节）进行快速计算，算出各个方向拉伸率，同时还算出各个方向着色的估算图案。完成了整个模型表面的分割值后，用斯托克斯流中算出的图案估计薄膜厚度，再用非线性粘性薄膜模型（3.3.3节）进行精确的计算。

这一步中，用户按照计算水转印的标准流程，以与上一步中相同的方向将物体固定在装置上。在完成了模型定位之后，用非线性粘性薄膜模型（3.3.3节）进行虚拟仿真，同时将颜色带来的薄膜厚度变化代入计算，得到更加精确的着色图案，如图3.12左侧一列所



图 3.12 多次转印每个方向的转印图案计算过程 每行代表一个方向上转印图案的计算过程。(左) 每个方向上计算出来的完整转印图案。(中) 边界平滑过渡后, 对应方向上每个像素的权重。(右) 最终打印在PVA膜上的图案。

示。各个方向的图案乘以各个方向上着色权重(图3.12中间一列), 得到各个方向上分割后的打印图案(图3.12右侧一列)。然后进行转印操作, 对物体在该方向对应的部分进行着色。等物体完全干燥后, 更换着色方向重复这个过程, 直到所有方向转印都完成, 整个模型的纹理就通过多次转印被绘制在了物体上。

3.5.3 边界平滑过渡

多次转印可以避免PVA膜在着色过程中被剧烈拉伸, 从而提升了着色的效果, 但是需要保证每次着色区域能够刚好拼接在一起, 否则就会产生明显的接缝。然而实际物理过程不可避免的会产生误差, 因此通过边界平滑过渡来减小因拼接产生的图案失真。如在图3.12中, 最终计算出的打印图案边缘是光滑过渡的。对于转印方向 i , 其在模型上着色区域为 Θ_i 。在此区域中的每个顶点 j 都被赋予着色权值 $w_{i,j} = 1$ 。而在区域 Θ_i 外的顶点, 其着色权值按照到区域边界的测地距离均匀递减, 公式为:

$$w_{i,j} = 1 - \frac{d_{i,j}}{H}, \text{ for } d_{i,j} \leq H, \text{ and } w_{i,j} = 0, \text{ otherwise} \quad (3.7)$$

其中 $d_{i,j}$ 是顶点 j 到 Θ_i 最近边界的测地距离, H 是控制过渡范围宽度的常数, 在实验中取5mm。为了保证通过多次着色, 物体每一点转印颜料的总量与只进行单次着色相同, 需要对每个顶点多次着色的权值进行均一化。对顶点 j , 其在*i*方向的着色权值为 $w_{i,j} = w_{i,j} / \sum_i w_{i,j}$ 。设顶点 j 处在CMYK空间中的颜色为 c_j , 则顶点 j 在*i*方向着色时, 打印图案对应像素的颜色为 c_j 与白色通过 $w_{i,j}$ 的线性插值, 即 $w_{i,j}c_j + (1 - w_{i,j})W$, 其中 $W = (0, 0, 0, 0)$ 为CMYK颜色空间中的白色,。

3.6 实验结果

我们首先设计了验证性实验 (3.6.1节) 来验证物理模型的准确性, 然后用定位水转印对多种三维模型进行了着色, 包括单次转印着色 (3.6.2节) 以及多次转印着色 (3.6.3节)。仿真运行在Intel i5 4430 CPU上, 像素粒子跟踪运行在Nvidia GTX 780 GPU上。所有的实验都能够再5分钟之内完成单次浸入着色的仿真。

粘性PVA膜有3个主要物理参数: 密度、厚度、杨氏模量。密度 $\rho = 1300\text{kg/m}^3$ 和厚度 $h = 0.5\text{mm}$ 参数值由PVA膜供应商提供。杨氏模量($Y = 1.82\text{MPa}$)通过物理实验测量得到。通过水转印将一个棋盘格图案转印到一个球上, 同时用不同的杨氏模量对这个过程进行虚拟仿真, 选取与物理实验结果最接近的虚拟仿真结果, 得到其杨氏模量值。这个值就作为实际的物理参数用于所有的实验。

3.6.1 验证性实验

我们通过验证性实验回答了以下三个问题:

1. 单次转印用水转印系统是否可重复?
2. 非线性粘性薄膜模型是否比斯托克斯模型的结果更加准确?
3. 如果水转印系统可以重复, 虚拟仿真能够达到多高的精度?

3.6.1.1 可重复性验证

由于依赖手工操作, 传统水转印工艺一个很大的缺点就是不可重复。而计算水转印通过机械装置对模型进行控制, 可重复性得到了极大的提高。我们用相同的棋盘格图案往同一个模型上多次着色, 看图案是否能够重合来验证可重复性。我们严格按照水转印的标准流程操作: 在A4纸大小的水转印膜上喷洒1ml活化剂, 等待20s然后浸入模型。如果图案能



图 3.13 水转印可重复性验证 用相同的棋盘格图案，完全相同的转印过程，对面具模型进行着色。从左到右，模型分别转印了一次、两次、三次。每次转印图案都能够对齐，因此看起来颜色在逐渐加深，从而证明水转印过程是可重复的。



图 3.14 非线性粘性薄膜模型与斯托克斯流模型结果对比 对倒置的U型物体通过水转印贴棋盘格纹理，同时对同一物理过程用非线性粘性薄膜与斯托克斯流两种模型进行仿真，比较仿真与物理实验的结果，证明非线性粘性薄膜模型计算准确性更高。

够重合，则证明水转印过程是可以重复进行的。该实验结果如图3.13所示，图中的模型从左到右分别被着色了1次、2次、3次，每次着色都使用完全相同的操作。通过对比可以看出，每次图案都能够几乎完全覆盖上一次着色的图案，因此颜色随着墨量累加而不断加深。

3.6.1.2 两种模型对比

图3.14是斯托克斯流模型与非线性粘性薄膜模型实验结果对比。采用了一个倒置的U形模型来做测试，模型的两只脚先入水，等两只脚完全浸入水中之后，上部的平板才入水。随着模型入水深度的增加，两平板之间薄膜的拉伸率迅速增大，等到上部的平板入水时薄膜的拉伸率已经非常大，因此能够非常清晰地观察出薄膜的形变。比较图3.14(b)和(c)的仿真结果可以看出，非线性粘性薄膜模型比斯托克斯模型的计算精度要

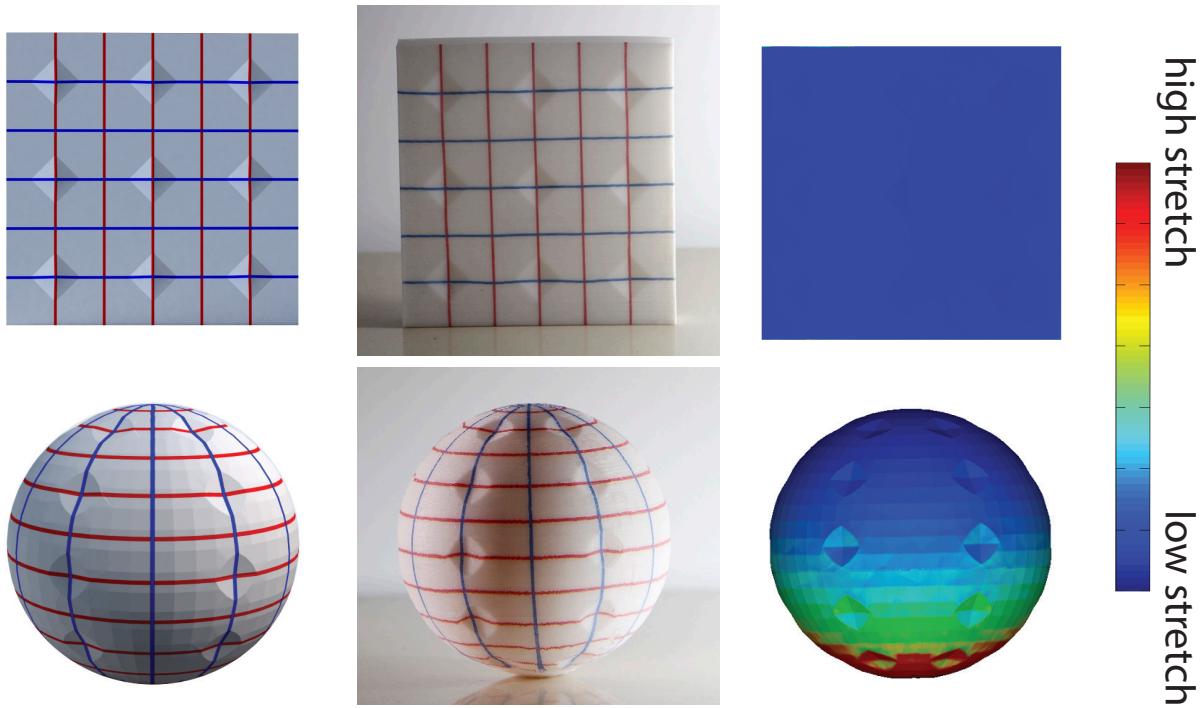


图 3.15 水转印精度验证 通过向带有几何特征（表面凹陷）的模型上转印具有图案特征（经纬线交叉点）的纹理，测量图案特征与几何特征对齐的误差来验证精度。平板模型的拉伸率较小，误差不超过1.5mm。圆球模型北半球误差不超过2mm，但过半球后拉伸急剧增大，因此误差也变大（约5mm）。(左) 测试模型及其纹理。(中) 转印了纹理的实物。(右) PVA膜的拉伸率。

高。

3.6.1.3 精度验证

我们用两个模型来验证系统的精度：一个尺寸为 $12\text{cm} \times 12\text{cm} \times 1\text{cm}$ 的平板和一个直径8cm的圆球，如图3.15所示。在模型上设置了一些规则排列的凹陷，这些凹陷用于指示模型上的位置。这些凹陷都非常浅，因此不足以影响PVA膜的形变。这些模型的纹理设计为经纬线，并且其交叉点与模型上凹陷的底部刚好重合。通过制作它们的实物模型，定位水转印系统的误差就可以通过纹理特征（经纬线交点）与几何特征（凹陷底部）的偏差来进行定量测量。

平板模型以 30° 入水，最终结果误差不超过1.5mm。圆球模型北极竖直向下入水，北半球的误差不超过2mm，但南半球的误差显著增大（约5mm）。这是因为当超过一半的圆球模型浸入水中之后，PVA的拉伸程度将会迅速增加，靠近南极附近水面与模型表面的夹角更是接近 180° ，在虚拟仿真时会形成病态的边界条件，因此计算误差也会明显增大。



图 3.16 大象杯-陶瓷表面的水转印实验 白色的陶瓷大象杯为直接购买的成品，因此先通过3D扫描获取其几何形状，由艺术家在数字化模型上设计纹理，最终通过水转印系统为其着色。



图 3.17 地球仪-塑料及木材表面的水转印实验 水转印膜能够拉伸并包裹整个圆球，因此地球仪的制作只需单次着色。图中左二为通过3D打印制作的塑料球，左四为木球，旁边为着色后的效果。

3.6.2 单次转印实验

对于许多简单物体，单次水转印就能够制作出效果不错的结果。图3.4为单次转印的结果，同传统水转印工艺相比，我们的方法能够将纹理图案与模型表面精确对齐。

水转印工艺能够对多种材料的表面进行着色，我们的方法同样能够扩展到多种材料。图3.16所示的是用计算水转印为陶瓷杯贴上的大象纹理。陶瓷杯是购买的制成品，因此首先用三维扫描仪获取其几何形状，然后由艺术家为模型设计了纹理图案。用扫描的模型及彩色纹理进行虚拟仿真，计算得到了打印在PVA膜上的预形变图案，并通过定位水转印将其印刷在陶瓷杯的表面。

图3.17展示的是为塑料球和木球贴上的地球仪纹理，我们的方法能够对几乎整个球面进行着色。不过尽管模型的底色并不会影响到水转印过程，但却会影响最终的着色效果。打印机默认是在白纸上打印，更换底色便会导致色偏。不过对于颜色不太深的物体，通过预先对打印机进行校色能在一定程度上纠正色偏。



图 3.18 两次转印实验 每个结果大图为着色之后的模型，左上小图为数字化模型的渲染图，左下小图为未着色的3D模型。



图 3.19 三次转印实验 每个结果大图为着色之后的模型，左上小图为数字化模型的渲染图，左下小图为未着色的3D模型。

3.6.3 多次转印实验

图3.18和图3.19分别是两次转印和三次转印制作的结果。通过多次转印，对复杂形状的模型也能够进行着色。

3.7 小结

本工作用面网格对浮在水面上的粘稠流体薄膜进行仿真，模拟固体物经过薄膜浸入水中，薄膜被拉伸并包裹住物体表面的过程。并将该仿真方法应用于曲面着色工艺——水转印的虚拟化，首次提出了“计算水转印刷”的方法，实现了三维实物按照用户设计纹理的准确着色。我们首次提出了水转印物理过程的计算模型，对整个物理过程进行仿真，从而计算出膜在水转印过程中的拉伸与形变。进一步将单次水转印扩展为多次水转印，解决了复杂曲面着色时图案失真的问题。我们用常用零件制造出了定位水转印装置，能够重现虚

拟仿真的结果，按照用户设计的纹理将图案印在物体的表面。通过一系列的实验，证明了这套系统能够进行高质量的曲面着色。

目前这套方法仍然存在一定的局限性。首先，如果模型有很深的凹陷（如图3.20所示），或者模型的形状存在自遮挡，那么即便用多次转移也很难对曲面进行完整的着色，这也是水转印工艺本身存在的问题。第二，多次转印中使用的边界平滑过渡方法，有可能会导致过渡区域的颜色失真。这是由于物体的外观并不是颜料的简单相加，即便用已知量的墨水一次转印，与把墨水平均分成两次转印叠加在一起，最终的效果看起来都会有区别。影响外观的可能还有喷墨打印机喷墨方式、颜色校准方式等等。因此如果要实现更加精确的颜色混合，需要专门为此建立一套模型，并进行标定。第三，PVA膜在水转印过程中的拉伸也会影响最终的视觉效果。膜单位面积变大会使得颜料密度下降，从而使颜色变淡。我们可以通过增加喷墨量的方法来矫正因拉伸导致的色偏，但增加喷墨量会使膜变厚，反过来又会影响膜形变的计算，从而影响图案的计算。因此在接下来的工作中，可以尝试迭代的方法来计算图案，但会牺牲运算速度。第四，目前估算墨水层厚度的方法只是一种估计，很难定量的描述。因此今后的工作中，可以尝试CMYKW色彩空间的喷墨打印机，使得整幅图像上墨水层的厚度都一致，从而彻底避免膜厚不均匀导致的计算问题。最后，现在的多次转印方法中，转印方向是用户指定的。也许今后可以将最优的方向也自动算出，甚至连模型的运动轨迹也一起优化，用机器人实现整个定位水转印过程的自动化。



(a)数字模型

(b)水转印实物结果

图 3.20 失败案例-图案在大拉伸下会失真 对于这种包含大凹陷区域的模型，无论从什么方向着色，PVA膜贴在凹陷内壁时都会被剧烈拉伸（甚至被拉破）。因此这种模型的凹陷区域不适合用水转印来进行着色。

第4章 塑性薄膜仿真与可计算热塑成型

在本章中，我们用面网格对塑性薄膜进行建模，模拟软化塑料片在大气压力下发生形变，并贴在固体表面的运动过程。该物理过程是塑料成型工艺——真空热塑成型的虚拟化流程。首先塑料片被加热软化，然后在模具一侧抽真空，塑料片就在另一侧的大气的压力下被拉伸，并贴紧模具的表面。如果塑料片上印有图案，那么通过真空成型就做成了彩色的塑料薄壳产品。因此对该物理现象进行仿真，就能够将真空热塑成型的工艺流程虚拟化。

热塑成型是将塑料加热软化，通过模具使其形成特定形状，并冷却成形的一类塑料加工方法。真空热塑成型简称真空成型，通过大气压力将塑料片压入模具成型，广泛用于薄壳塑料产品的制造。塑料片的恒定工况下，拉伸形变具有很好的可重复性。本章提出了“可计算热塑成型”的方法，并结合3D打印，将带纹理的虚拟数字化模型制作成为实物模型。首先将数字化模型作为模具，对真空成型的过程进行虚拟仿真。根据仿真中塑料片的形变，计算出预形变图案，将其印在透明塑料片上。通过对抽真空过程的仿真，找出塑料片与模具之间可能出现的气体空腔，并据此在模型上设置排气孔，通过3D打印制作出带有排气孔的模具。最后将塑料片和模具固定在真空成型机上，将其位置对齐并进行真空成型操作，于是纹理就贴在了模具的表面，同时塑料片也为图案覆盖了一层透明保护壳。通过实验，验证了计算模型以及真空成型着色系统的高效性与精确性。

4.1 问题的提出

在第3章，我们提出的可计算水转印刷技术，为曲面着色提供了一种新的途径。但计算水转印刷仍然存在一些缺陷，尤其是对物体凹陷较深部分着色效果较差。除此之外，水转印设备体积较大，操作复杂，在水中操作精度较低，通过简易设备已经很难再进一步提升着色质量。

真空热塑成型的原理如图4.1所示。将塑料片加热变软，通过大气压力将塑料片压在模具表面，冷却后裁剪掉多余部分，即成为最终产品（图4.2）。真空成型在工业中已经有

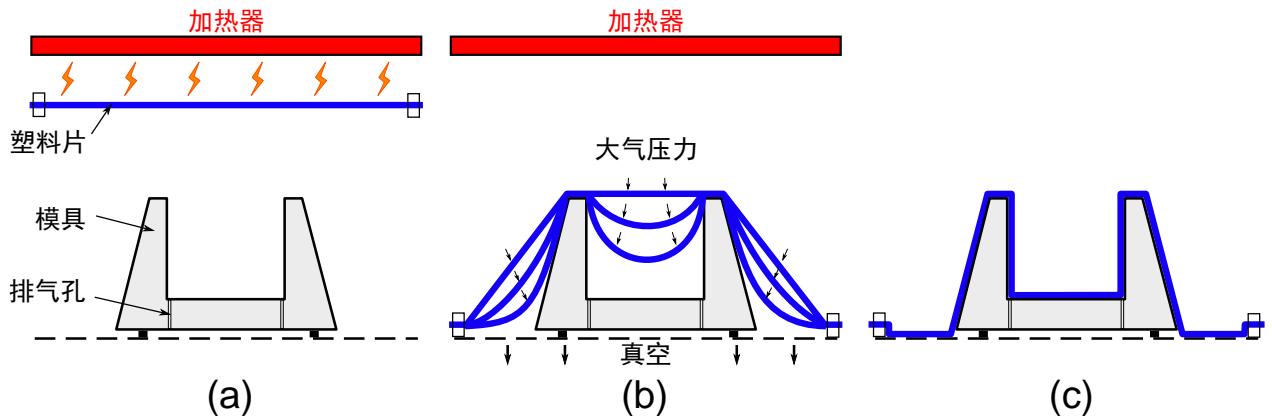


图 4.1 真空热塑成型原理 (a)首先塑料片被加热软化。(b)然后将其移动到工作位置,使塑料片下部形成密闭空间。开始抽真空后,空气通过排气孔被排出,于是塑料片在大气压力作用下被拉伸,并贴紧模具的表面。(c)待塑料片冷却后,形状就固定下来。将其从模具上取下,裁掉边角料,就制成了薄壳塑料制品。

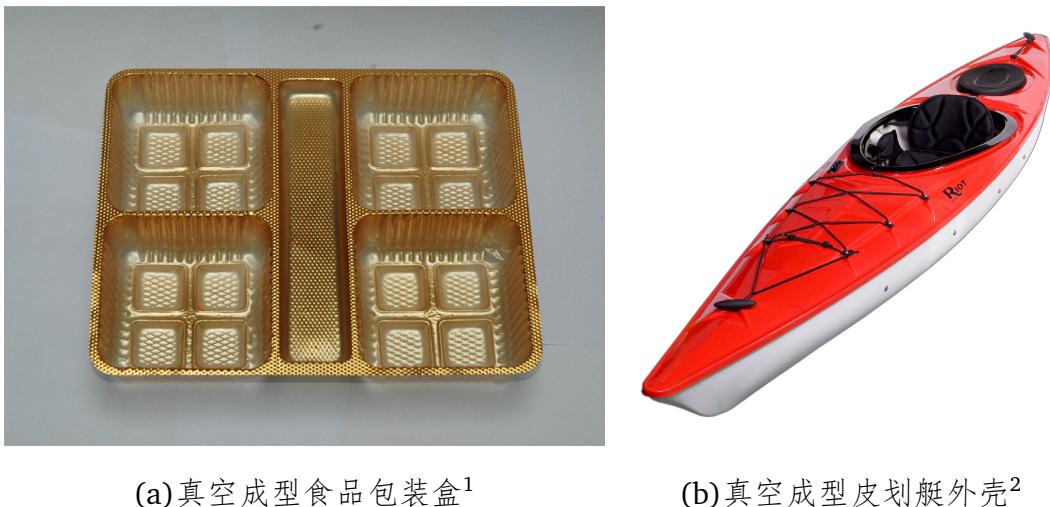


图 4.2 常见真空成型产品 真空成型主要用于制造薄壳塑料产品,生产效率高,应用广泛。

超过70年的历史,其生产效率高,应用广泛,小到食品包装,大到游艇外壳都能够生产。

基于真空热塑成型原理,我们提出了一种设备更加简单,操作更加方便,结果更加稳定的新型着色方法——可计算热塑成型。同水转印原理类似,如果在平整的塑料片上预先印好图案,再进行真空成型,就可以制造出带图案的薄壳塑料产品。如果按照合适的方法对图案进行预形变,还能使纹理与模型几何结构对齐。

本文提出的“可计算热塑成型”,基于虚拟仿真对图案进行预形变。将塑料片建模为用面网格表示的塑性薄膜,并虚拟地再现真空成型中塑料片从加载到贴紧模具的过程。根

¹图片来源: <http://www.eastsoo.com/pifa/traffic-10035591.html/>

²图片来源: <http://www.lago.com.cn/goods.php?id=38>

据仿真中塑料片的拉伸对图案进行预形变，并将图案打印在塑料片上。与此同时，还对抽真空过程中可能出现的气体空腔进行检测，使打印的模具带有排气孔，保证塑料片能够贴紧模具表面。我们用一台简易的真空成型机将塑料片与模具对齐，并进行真空成型，制造出与带纹理的数字模型一致的实物模型。

塑料片的拉伸均匀性远好于水转印膜，因此该方法能够在大拉伸区域保证图案的质量。整套系统的设备及耗材成本都非常低廉，能够进一步降低全彩色3D模型的制造成本。

本文的主要贡献包括：

- 基于面网格的塑性薄膜仿真方法，能够对真空成型过程进行精确的仿真。
- 模具排气孔位置的计算方法，避免在真空成型过程中产生气体空腔。
- 一套可计算真空成型系统，通过虚拟仿真计算出预形变图案，并将图案印在塑料片上；打印带有排气孔的模型，最终通过真空成型，制造出与带纹理的数字模型一致的实物模型。

4.2 算法概述

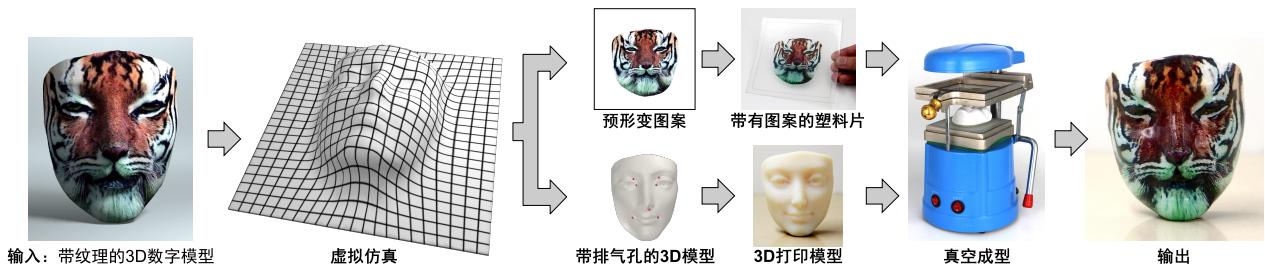


图 4.3 真空热塑成型3D模型着色系统流程图 通过对真空成型的整个过程进行仿真，计算出塑料片在成型过程中的形变，并在塑料片表面印刷预形变图案；同时分析出需要设置排气孔的位置，通过3D打印制作带排气孔的模型，最后通过真空成型将塑料片贴在模型表面，形成与带纹理的数字模型一致的实物模型。

我们的方法主要是通过对真空成型进行仿真，计算塑料片的形变，在塑料片表面印上的预形变图案；通过3D打印制作带排气孔的模型，并通过真空成型将塑料片贴在模型表面，最终制作出与带纹理的数字模型一致的实物模型。系统流程如图4.3所示，主要包含如下步骤：

1. **虚拟仿真** 首先对真空成型整个过程进行仿真，包括塑料片软化之后加载至工作位置，以及抽真空时塑料片贴紧模具表面的整个过程。

2. **计算预形变图案** 通过虚拟仿真，获得了3D模型表面与塑料片之间的映射关系，再通过3D模型的纹理映射，可以得到经过预形变的图案。将这个图案印在塑料片上，真空成型之后就能够得到与数字模型纹理一致的实物。
3. **模具排气孔设置** 通过对抽真空过程的仿真，找出塑料片与模具之间可能出现的气体空腔，并据此在模型上设置排气孔，通过3D打印制作出带有排气孔的模具。
4. **真空热塑成型** 最终，把印刷了预形变图案的塑料片，以及带排气孔的模具装到真空成型机上，进行真空成型操作，从而把印刷有图案的塑料片贴在3D模型上。

4.3 热塑成型过程仿真

对热塑成型进行虚拟仿真，可以在实际操作前，对塑料片在大气压力下拉伸形变并贴在模型表面的过程进行模拟。通过虚拟仿真获得塑料片图案上每个像素与其贴在模型上位置的对应关系，进而通过模型纹理读取每个像素的颜色，算出预形变图案，使得形变之后的效果与目标一致的。

4.3.1 热塑性塑料片

真空成型采用的塑料片为热塑性材料，即当材料温度达到其软化温度后变得可拉伸，而冷却之后又能重新变得坚硬。软化后的拉伸形变属于弹性-塑性形变，其应力-应变曲线

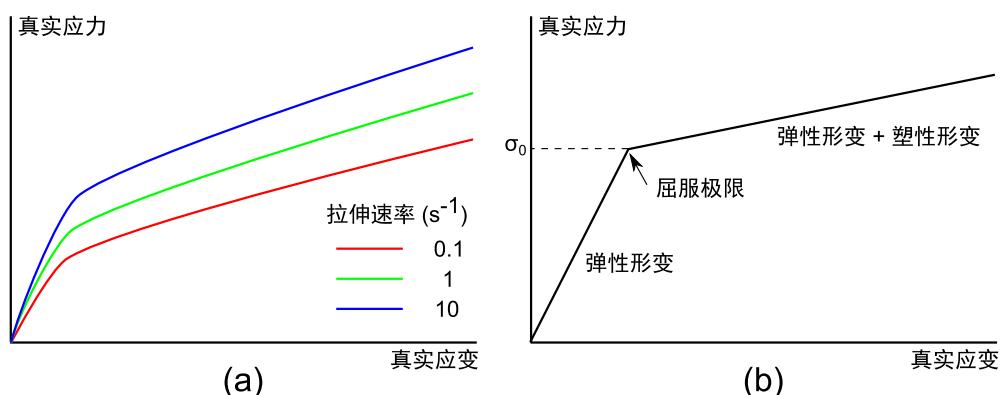


图 4.4 热塑性材料应力-应变曲线及简化模型 (a) 热塑性材料典型真实应力-真实应变曲线。在小形变区域内，材料的形变是弹性形变。应力超过屈服极限，材料开始塑性形变，并伴随强化。材料可拉伸性也会受到拉伸速度的影响。但是拉伸速度变化10倍，应力变化不超过20%，因此可以认为整个仿真过程中拉伸率恒定，并不会造成显著的误差。(b) 简化的应力-应变曲线，以屈服点为界分成弹性、弹性+塑性两段。

如图4.4(a)所示。与真实应力-真实应变曲线对应的是工程应力-工程应变曲线，即用标准材料试样在拉伸试验机上直接测出的曲线。热塑性材料在拉伸的时候，材料本身会变细，而工程应力仍然用材料变细之前的截面积来计算，因此比真实应力偏小。曲线上的拐点叫做屈服点，应力达到屈服点之前是弹性形变，超过屈服点就开始塑性形变，此时即便去掉外力也不能再回到初始的状态。随着塑性形变的增加，材料会被强化——卸掉拉力后再次拉伸时，屈服极限会增加。材料被拉伸的速度也会影响材料的可拉伸性，但这个影响相对较小^[157]，因此在我们的系统中将其忽略并不会造成显著的误差。

我们使用的弹性-塑性模型如图4.4(b)所示。 σ_0 表示屈服极限。当材料的von Mises应力小于屈服限时，就是一个弹性薄膜应变模型；而超过 σ_0 之后就开始塑性形变，同时伴随着材料的强化，再次拉伸时的屈服极限也在增加，因此在这一段的形变同时具有弹性形变和塑性形变的性质。由于材料拉伸的速度对结果影响不大，整个模型都与时间无关，因此通过静态平衡的方式求解。

4.3.1.1 弹性形变

真空成型中使用的热聚合物塑料片属于薄片结构，材料的弯曲应力同拉伸应力相比小很多，因此可以忽略弯曲应力，只考虑材料的拉伸^[131]。经测试，加入弯曲应力的模型计算结果同只考虑拉伸的模型几乎没有区别（如图4.5所示），但运算速度会显著降低，因此在仿真计算时忽略弯曲应力。材料的拉伸应力通过Pfaff等人^[3]提出的方法来计算。由于真空成型过程中，网格初始为平面，因此天然具有二维的参数化，形变梯度 \mathbf{F} 可以用 3×2 矩

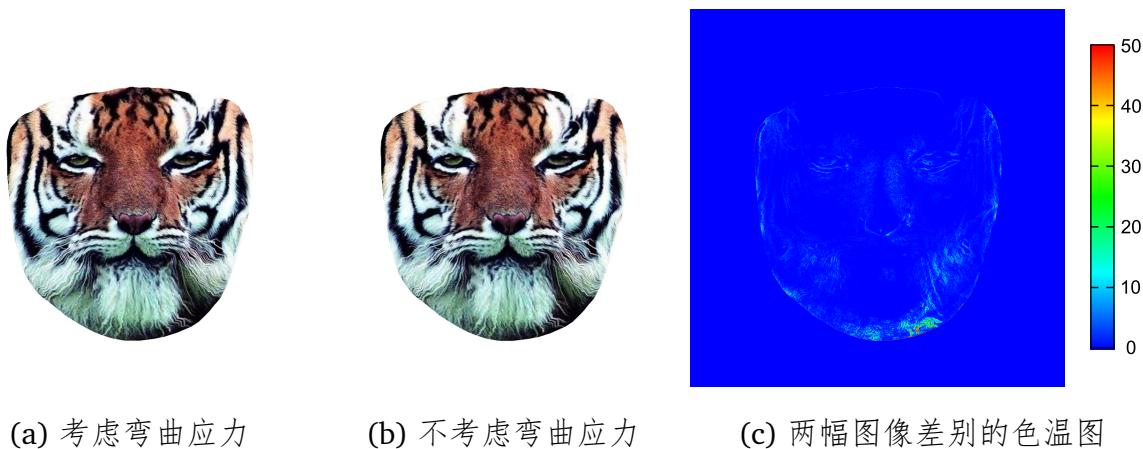


图 4.5 弯曲应力对预形变图案计算的影响 我们分别计算考虑弯曲应力、不考虑弯曲应力的预形变图案，并计算两幅图像之差的色温图（每个像素RGB三通道数值差的绝对值均值）。可以看出，是否考虑弯曲应力对图案计算的影响很小，计算出的图案只有像素级别的误差。因此在仿真过程中，可以将弯曲应力忽略，从而提高仿真速度。

阵表示。描述平面内形变程度的格林应变为：

$$\mathbf{G} = \frac{1}{2}(\mathbf{F}^\top \mathbf{F} - \mathbf{I}) \quad (4.1)$$

其中 \mathbf{I} 是 2×2 的单位阵，于是膜内应力为

$$\boldsymbol{\sigma} = k_s [(1 - \nu) \mathbf{G} + \nu \text{tr}(\mathbf{G}) \mathbf{I}] \quad (4.2)$$

其中 $k_s = Yh/(1 - \nu^2)$ 描述材料的可拉伸性， Y 是杨氏模量， h 是塑料片的厚度， ν 是材料的泊松比（塑性形变阶段为 $0.5^{[158]}$ ）。

4.3.1.2 塑性形变

塑性形变模型是基于 Pfaff^[3] 和 Bargteil^[44] 等人的方法，将其改造为可通过静态平衡方式求解的模型。薄膜形变模型中，形变梯度 \mathbf{F} 可以分解为弹性 \mathbf{F}_e 和塑性 \mathbf{F}_p 两部分：

$$\mathbf{F} = \mathbf{F}_e \mathbf{F}_p \quad (4.3)$$

这个表达与 Bargteil 等人^[44] 的三维模型类似，只是我们的薄膜应力存在于平面内，因此 \mathbf{F}_e 是 3×2 矩阵， \mathbf{F}_p 是 2×2 矩阵。模型通过调整局部厚度 h 实现体积守恒，因此去掉了 Bargteil 等人^[44] $\det(\mathbf{F}_p) = 1$ 的限制。仿真开始前，还没有任何塑性形变时 \mathbf{F}_p 为单位矩阵。一旦开始产生塑性形变， \mathbf{F}_p 就开始变化，于是 \mathbf{F}_e 与总形变 \mathbf{F} 就不再一致了。换句话说， \mathbf{F}_p 改变了模型的静态形状，因此在计算公式 4.1（格林应变）和公式 4.2（膜内应力）时，用 \mathbf{F}_e 取代 \mathbf{F} 。

在仿真的每一步中，根据 von Mises 判别条件，判断材料的应力是否已经超过了屈服极限。膜内 von Mises 应力公式为

$$\sigma_v = \sqrt{\sigma_1^2 + \sigma_2^2 - \sigma_1 \sigma_2} = \sqrt{(\text{tr} \boldsymbol{\sigma})^2 - 3 \det \boldsymbol{\sigma}} \quad (4.4)$$

其中 σ_1 和 σ_2 是公式 4.2 中 $\boldsymbol{\sigma}$ 的两个特征值。一旦 σ_v 超过了材料的初始屈服极限 σ_0 ，材料就开始塑性形变。

塑性形变会使材料发生强化，使得材料的屈服极限 σ_y 增加^[159]。如果假设材料的强化实线性的^[160]，那么当前屈服极限 σ_y 可以表示为 \mathbf{F}_p 的函数，写成类似于 Ludwik's 方程的形式：

$$\sigma_y = \sigma_0 + K \|\mathbf{F}_p^\top \mathbf{F}_p - \mathbf{I}\| \quad (4.5)$$

其中 σ_0 是材料的初始屈服极限， K 是材料的强化系数，用于控制图4.4(b)中弹性+塑性部分的斜率。

屈服极限 σ_y 的增加，会使得已经开始塑性形变的模型重新开始弹性形变，直到 $\sigma_v > \sigma_y$ 。而实际的物理过程应当是弹性形变与塑性形变同步增加，并且处处保持静态平衡，即完全塑性^[161]。因此在静态平衡求解的每一步中，如果发现 von Mises 应力超过了屈服极限，那么就对材料进行强化，直到刚好达到平衡为止，即 $\sigma_v = \sigma_y$ 。每一步中总形变梯度 \mathbf{F} 固定，通过增加塑性形变梯度 \mathbf{F}_p 来减小弹性形变部分 $\mathbf{F}_e = \mathbf{F} \mathbf{F}_p^{-1}$ ，于是 σ_v 减小， σ_y 增加，从而达到平衡。迭代方法与 Pfaff 和 Bargteil 等人^[3;44]类似。

更新塑性形变梯度 \mathbf{F}_p 的本质是要减小弹性应变，也就是 \mathbf{F}_e 的主应变的值，因此首先要计算出 \mathbf{F}_e 的主应变。通过奇异值分解 (Singular Value Decomposition)，得到 $\mathbf{F}_e = \mathbf{U} \Sigma \mathbf{V}^\top$ ，其中 \mathbf{U} 和 \mathbf{V} 分别是三维和二维旋转矩阵， Σ 是 3×2 伪对角矩阵。移除 Σ 中元素都为零的最后一行，变为 2×2 对角阵 $\mathbf{S} = \text{diag}(\lambda_1, \lambda_2)$ 。于是塑性形变梯度 \mathbf{F}_p 更新为：

$$\mathbf{F}_p \leftarrow \mathbf{V} (\det(\mathbf{S})^{-\frac{1}{2}} \mathbf{S})^\gamma \mathbf{V}^\top \mathbf{F}_p \quad (4.6)$$

其中指数 γ 的大小表示塑性形变 \mathbf{F}_p 的变化程度。为了证实增大 γ 确实能够减小弹性形变量，对弹性形变梯度 \mathbf{F}_e 也进行更新：

$$\mathbf{F}_e \leftarrow \mathbf{F} \mathbf{F}_p^{-1} = \det(\mathbf{S})^{\frac{\gamma}{2}} \mathbf{U} \text{diag}(\lambda_1^{1-\gamma}, \lambda_2^{1-\gamma}) \mathbf{V}^\top. \quad (4.7)$$

系数 $\det(\mathbf{S})^{\frac{\gamma}{2}}$ 保证塑性形变不改变薄膜的单位面积，但由于材料是不可压缩的，厚度的改变远小于长度的拉伸，因此这个值接近1。因此， γ 在 $[0, 1]$ 之间取值能够将格林应变 (公式 4.1) 减小到 $\sigma_v = \sigma_y$ 。Pfaff^[3] 和 Bargteil^[44] 等人用动态仿真求解， γ 是时间或者迭代步数的函数。而在静态平衡仿真中，只要尚未达到静态平衡，塑性形变就要进行下去，直到材料强化到足以对抗弹性形变。因此每一步都要计算能够使 $\sigma_v = \sigma_y$ 的 γ 值。

具体方法为：一旦通过弹性迭代计算出了新的总形变梯度 \mathbf{F} ，便用二分查找法对 $\sigma_v - \sigma_y$ 寻找一个公式 4.4 中的 γ 值。注意，对于一个固定的总形变梯度 \mathbf{F} ， σ_v 和 σ_y 都是 γ 的函数。由于 γ 能够使 σ_v 减小到 0，而 $\sigma_y > \sigma_0$ ，因此一定能够在 $[0, 1]$ 区间内找到一个解^[3]。

4.3.1.3 外力

弹性和塑性作用都属于材料的内力，在实际物理过程中，还存在外力的作用，包括重力、模具的支持力、摩擦力，以及大气压力。

1. 重力

材料会受到重力的作用，不过由于材料本身很薄很轻，与其内力相比要小很多。材料的初始内应力是由于其制造工艺：挤出成型^[162]产生的。大块材料在被软化之后，被硬挤成薄板，然后立即冷却成型，因此具有很大的内部应力，这些内应力将在加热过程中被释放。在开始加热前，塑料片的边缘被固定在真空成型机的塑料片固定夹上，边界在整个仿真过程中都将作为硬约束来计算。由于边界被固定，在加热的初始阶段（约1-3分钟），塑料片由于内应力的释放会产生一定的卷曲。待内应力的释放变得均匀之后，塑料片会重新变得平整（约3-5分钟）。一旦达到工作温度之后，就可以开始成型操作了。整个加热阶段，塑料片都没有观察到明显的下垂，说明相比于材料的内应力，重力小到可以忽略。当抽真空开始之后，空气压力和模具支持力就成为了主导的外力。因此整个仿真过程中都忽略重力的作用。对于更大的塑料片（常见于工业级别的真空成型加工），重力通过塑料片的单位质量乘以重力加速度，可以非常容易的加入计算模型中。

2. 模具的支持力

待塑料片软化之后，将热塑料片加载至工作位置，使塑料片下部形成封闭空间。在此过程中，塑料片接触到模型表面会受到其支持力，并且会在其表面上产生滑动。由于模型表面在整个仿真过程中都静止，因此对3D模型预计算一个用 $2\text{mm} \times 2\text{mm} \times 2\text{mm}$ 规则体素构建的距离场^[163]（signed distance field）用于快速的进行碰撞检测。同Bridson等人^[33]的方法类似，通过添加与距离成正比的排斥力来防止塑料片穿透模型表面。对塑料片网格的每一个顶点，在距离场中用三线性插值算出其到模型表面的距离。如果某顶点距离 $d < h_0/2$ ，其中 h_0 是塑料片的初始厚度（实验中用1mm厚的塑料板），其受到的支持力计算方法为

$$\hat{\mathbf{f}}_r = \begin{cases} k_r \left(\frac{h_0}{2} - d \right) M \hat{\mathbf{n}}, & d < h_0/2 \\ \hat{\mathbf{0}}, & \text{otherwise} \end{cases}, \quad (4.8)$$

其中 M 是该顶点一环邻域面积的三分之一（即网格分摊到该顶点上的面积）， $\hat{\mathbf{n}}$ 是距离场在该点的单位梯度向量， k_r 是控制支持力大小的常数。这个简单的线性模型对

于我们的实验已经足够精确了。

3. 模具表面摩擦力

摩擦力 $\hat{\mathbf{f}}_f$ 是阻碍塑料片与模具表面相对滑动的力，方向与模具表面相切，通常用库伦摩擦力来描述。对于塑料片上一点，摩擦力的最大值为 $\mu \|\hat{\mathbf{f}}_r\|$ ，其中 μ 是塑料片与模型之间的摩擦系数。摩擦力 $\hat{\mathbf{f}}_f$ 的求解公式为：

$$\hat{\mathbf{f}}_f = \begin{cases} -\mathbf{f}_t, & \|\mathbf{f}_t\| < \mu \|\hat{\mathbf{f}}_r\| \\ -\mu \frac{\|\hat{\mathbf{f}}_r\|}{\|\mathbf{f}_t\|} \mathbf{f}_t, & \text{otherwise} \end{cases}, \quad (4.9)$$

其中 \mathbf{f}_t 是该点所受除摩擦力外的合力 \mathbf{f} 在切平面上的投影。由于我们采用静态平衡方法求解，通过虚拟仿真计算出的薄膜应力会自动与其余的外力相平衡。

4. 大气压力

当塑料片同模型表面相对静止之后，开始真空成型。随着大气压强逐渐增加，塑料片逐渐被拉伸，直到完全贴紧模型的表面。在仿真过程中，首先计算每个三角形上的大气压力，然后将其均匀地分摊到三个顶点上。每个三角形上的力为

$$\hat{\mathbf{f}}_p = -p A \hat{\mathbf{n}}, \quad (4.10)$$

其中 p 是当前塑料片两侧的大气压强差， A 是该三角形的面积， $\hat{\mathbf{n}}$ 是该三角形的单位法向量。塑料片一旦接触模型，就会在大气压力作用下贴紧，不再产生滑动。因此在此阶段不再考虑摩擦力，而是检测顶点到模型的距离。如果 $d < h_0/2$ ，就对该顶点添加硬约束，其位置在当前坐标和上一时刻坐标之间，距离模型表面距离为 $d = h_0/2$ 的点处。由于薄膜的厚度变化相对模型的尺寸变化不大，因此在计算中都用薄膜的初始厚度 h_0 。

4.3.2 数值计算方法

我们通过有限元来对塑料片的运动进行仿真。用三角网格对整张塑料片进行离散化，其边界是与塑料片夹尺寸一致的矩形，边界内通过Triangle^[153]生成三角网格。每个三角形内角下限设定为 30° ，面积上限为 $2mm^2$ 。塑料片初始位置位于模具的上部（图4.1(a)）。虚拟仿真分为两个阶段，第一个阶段塑料片从初始位置加载到工作位置。在此过程中，塑料片的边缘匀速下降，塑料片接触到模具表面会被拉伸，并有可能在模具表面滑动。当到达

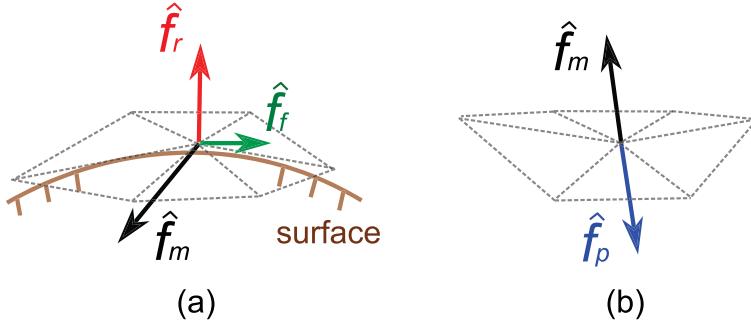


图 4.6 塑料片受力平衡示意图 仿真采用静态平衡方式求解，在仿真的每一步每个顶点都会达到受力平衡状态。图中展示的是热塑成型的两个阶段，塑料片的力平衡状态。**(a)** 在加载塑料片阶段，塑料片接触模具，会在模具表面产生滑动。在此阶段，塑料片的弹性应力 \hat{f}_m 会与模具表面的支持力 \hat{f}_r ，以及摩擦力 \hat{f}_f 达到平衡。**(b)**在抽真空阶段，大气压力会将塑料片贴紧模具表面，阻止相对滑动。因此当顶点接触模具之后，其坐标就被约束在当前位置。而尚未接触模具表面的顶点，塑料片的弹性应力 \hat{f}_m 会与模具表面的支持力 \hat{f}_p 达到平衡状态。

工作位置之后，开始第二个阶段：抽真空。在此阶段，塑料片在大气压力作用下贴紧模具表面（图4.1(b))。由于整个过程时间很短，因此可以忽略该过程中的温度变化，认为塑料片的物理性质是恒定的。

仿真通过反向欧拉迭代来实现，每一步迭代都是对顶点位移进行松弛，使其满足顶点的受力条件。内部应力通过弹性形变梯度 $\mathbf{F}\mathbf{F}_p^{-1}$ 来计算，其中 \mathbf{F} 是顶点位移的函数， \mathbf{F}_p^{-1} 是上一步迭代中的塑性形变梯度。其余的外力，包括大气压力、模具的支持力、模具表面摩擦力都是顶点位置的函数。这些力的导数都可以方便计算，因此通过牛顿迭代来求解非线性方程，每个仿真步长迭代50次，而每次迭代都通过共轭梯度法（conjugate gradient）来求解。

当顶点位置更新之后，就进行塑性形变的更新。首先检测出那些符合von Mises判别条件 $\sigma_v > \sigma_y$ 的三角形。对于这些三角形，用二分法来更新 \mathbf{F}_p 使得 $\sigma_v = \sigma_y$ 。更新后的塑性形变梯度将用于下一个仿真步长的计算。

4.3.3 打印图案生成

当塑料片完全贴在3D模型表面之后，也就构建出了塑料片与模型表面之间的映射关系。这个映射决定了塑料片上印刷的图案与3D模型表面最终视觉效果之间的关系。打印图案的计算方法与计算水转印中基本相同。

计算塑料片上的打印图案可以看成是给塑料片网格贴上纹理。对于图案的每个像素，根据平面坐标 \mathbf{x} 找到其在塑料片网格中落入的三角形，并计算其重心坐标。由于网格的拓

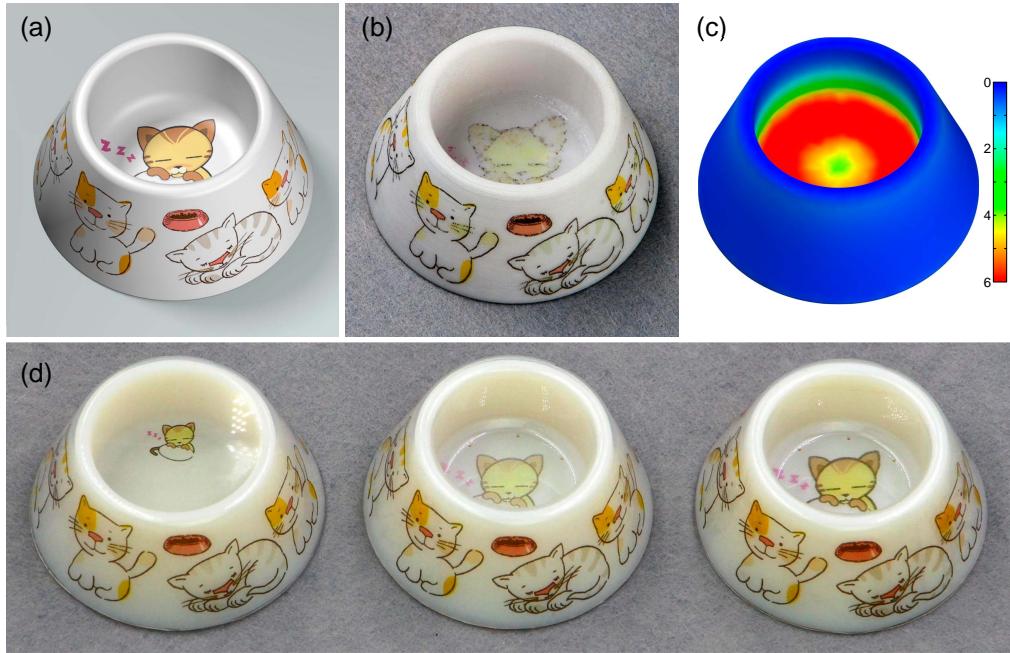


图 4.7 凹陷模型的大拉伸实验 通过真空成型对深度凹陷的猫盆模型内外表面同时进行着色。
(a) 目标模型及纹理。 (b) 计算水转印制作的模型，模型底部由于拉伸量过大导致严重失真。 (c) 贴在模型表面的塑料片面积拉伸率。 (d) 真空成型着色的结果，我们把三个模型放在一起对比颜色变化。(左) 没有设置排气孔时，空气被困在了凹陷区域，导致塑料片无法贴紧模型内壁。中间的猫的图案反映了在塑料片被拉伸之前，打印出来图案的大小与颜色深度。(中) 模型打了排气孔，但图案在打印时没有进行颜色校正，因此中间图案由于拉伸量过大而变淡。(右) 打印图案时进行了颜色校正，颜色失真现象有所减弱。

扑结构在计算过程中不改变，因此这一像素在网格变形后的位置 \mathbf{x}' 可以直接在同一个三角形中，用同样的重心坐标计算出来。 \mathbf{x}' 是塑料片网格上的点，要找到模型上的一点，就用 \mathbf{x}' 向模型上投影，找到其最近点 $\tilde{\mathbf{x}}$ 。而 $\tilde{\mathbf{x}}$ 就位于模型上，其颜色可以从3D模型的纹理中获得，并赋给 \mathbf{x} 处图案对应的像素。对图案上所有的像素都进行该操作，就得到了完整的预形变图案。

为进一步提高着色质量，我们将塑料片的拉伸也考虑进图案的计算中。塑料片上各处拉伸并不均匀，导致颜料密度会产生不均匀的变化（如图4.7(d)所示），因此需要对颜色进行修正，方法与Panozzo等人^[128]类似。对于 \mathbf{x} 点的一个像素，在网格中找到包含这点的三角形，计算其仿真前后的面积 A_x 和 A'_x 。颜料的密度变化与面积变化成反比。记 $\tilde{\mathbf{x}}$ 点在CMYK颜色空间中的颜色为 $c_{\tilde{\mathbf{x}}}$ ，于是 \mathbf{x} 点处像素颜色修正为 $(A'_x/A_x)c_{\tilde{\mathbf{x}}}$ 。对于拉伸量非常大的区域，颜色变化已经不足以用这种方法进行调整了，就可以通过多次打印将颜料累加起来，从而使颜料密度增加。我们用一台经过校色的CMYK激光打印机，完成所有的实验。颜色矫正的效果如图4.7(d)所示。在真空成型过程中，塑料片的杨氏模量远比水转印膜要

大，颜料对塑料片的拉伸没有任何影响，因此一步计算就可以到位，不必像水转印那样更新图案后就要重新仿真。

4.4 排气孔布局

在虚拟仿真的抽真空阶段，我们假设所有尚未贴在模型表面的点，其两侧气压差都一样。但是，这个假设只有当塑料片和模型之间的所有空间都与真空泵连通时才成立。实际上，如果3D模型上有凹陷区域，那么很有可能出现凹陷的四周都已经被塑料片覆盖，于是空气被困在凹陷中无法排出的情况，如图4.8所示。为了防止这种情况，需要在虚拟仿真阶段就把所有可能出现的气体空腔给检测出来，并且在这些区域设置排气孔。这样一来，空气就能够通过这些排气孔被顺利排出，使得塑料片能够贴紧模型的表面。

4.4.1 气体空腔的检测

在虚拟仿真中，气体空腔形成于塑料片的网格与模型表面之间，可以在仿真的过程中直接检测出来。在虚拟仿真抽真空阶段的每一步中，如果网格上的任何一个顶点贴在了模型上，那么这一点就永远附着在模型表面，称为附着顶点。如果一条边的两个顶点都成为了附着顶点，那么这条边也就附着在了模型表面，称为附着边，并且将会阻止空气从这条边下方流过。假设模型本身没有洞，当贴在模型上的边形成一个环，将尚未贴在模型上的

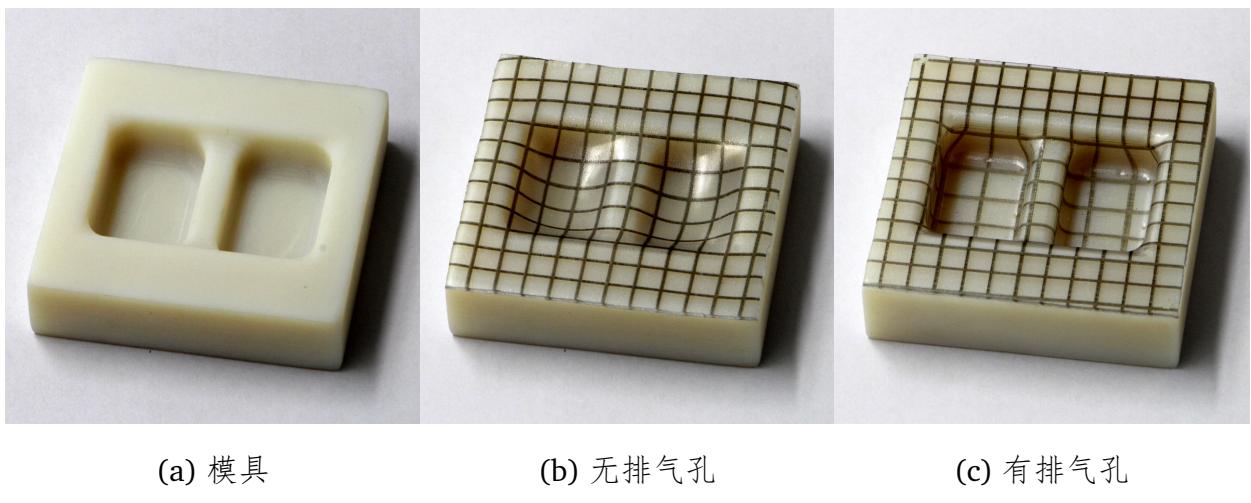


图 4.8 排气孔设置对真空成型结果的影响 通过一个带有凹陷的模具，测试排气孔的有无对真空成型结果的影响：(a) 用于测试的模具。(b) 如果没有设置排气孔，空气会被困在凹陷部分，使得塑料片无法与模型贴合。(c) 当设置了排气孔之后，凹陷部分的空气就可以通过排气孔被排除，于是塑料片就能够与模型顺利贴合。

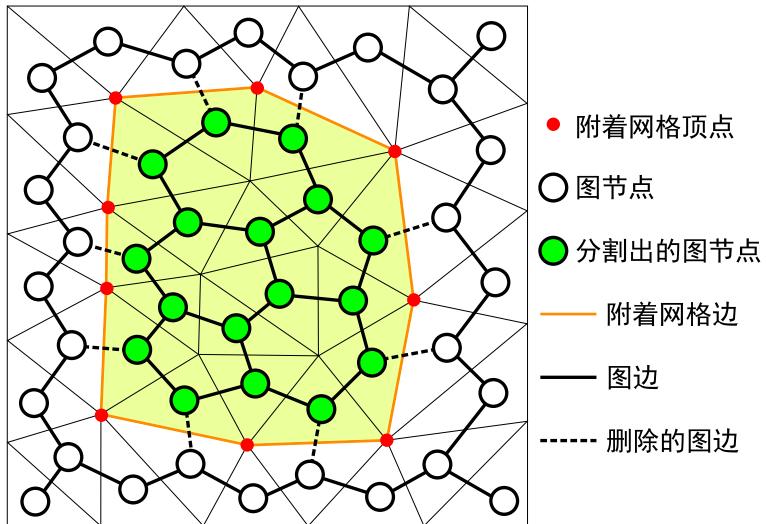


图 4.9 气体空腔的检测方法 为了检测气体空腔，对塑料片的三角网格生成弱对偶图。在仿真过程中，网格顶点将逐渐贴在模型表面，形成附着顶点。如果网格中一条边的两个顶点都是附着顶点，那么这条边成为附着边，在图中对应的边将被删除（图中虚线）。如果检测出一个连通区域，并且这个区域是被附着边所包围，那么这片网格区域（图中的浅黄色区域）就将形成一个气体空腔，其边界由附着边构成（图中的黄实线）。

区域围起来，那么这个区域下方的空气已经没有排出的通道，也就形成了一个气体空腔。这个空腔的边界就是附着边形成的环。因此通过检测这种环，就可以检测气体空腔。

检测气体空腔的方法可以通过图论来精确定义。图4.9所示的是网格的弱对偶图 $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ 。图的每个节点 $V_i \in \mathbf{V}$ 表示网格上的一个三角形，图的每条边 $E_i \in \mathbf{E}$ 连接相邻三角形。如果网格中一条边成为了附着边，那么图中对应的边 E_i 将从 \mathbf{E} 中被移除。在仿真的每一步，整个图 \mathbf{G} 都被更新，然后从图上检测连通分量 $C \subset \mathbf{G}$ 。如果检测出一个新的连通分量 C_i ，并且其对应的三角网格区域 R_i 包含尚未附着的点，而区域的边界都是附着边，那么这个区域下方的空间就将形成一个新的气体空腔。当然如果模型本身有洞，那么这个空间此时还是有可能直接与真空泵相连的。这种情况可以通过检测模型的曲面识别出来。但由于我们假设模型的亏格为零，这种情形不会发生。

4.4.2 气体空腔森林

在虚拟仿真的过程中每一步都对气体空腔进行了检测，跟踪气体空腔的变化情况，并据此计算排气孔的位置。为此我们引入“气体空腔森林”的概念，森林中每棵树的根节点代表从塑料片网格上直接检测出来的气体空腔；每一个叶节点表示最后完全贴在模型表面的气体空腔；而每个中间节点则是由一个气体空腔分裂成两个而形成的，如图4.10所示。笼统地说，只要在每个叶节点处设置排气孔，那么所有的气体空腔都能够被消除。然而，

如果单纯用这套方法检测气体空腔，有可能叶节点数量会非常多。比如由于模型表面的起伏，或者离散误差而产生的体积过小的气体空腔。这些很小的气体空腔只是在数值计算中存在，实验中会通过模型表面的微小凹凸被消除，或者体积小到根本看不出来。因此要把这些气体空腔从气体空腔森林中给去掉。

如果把气体空腔森林的构造过程按照时间反过来观察，就成为了将节点和边不断加入网格对偶图的过程，也就是将小的气体空腔不断合并在一起的过程。这个过程直接对应了计算拓扑学中持续同调的概念，因此可以直接用相关的数学工具来消除持续时间很短的气体空腔。不过为了更好地体现真空成型中形成气体空腔的实际物理过程，我们用空腔的体积大小作为是否需要对其进行处理的标准。

虚拟仿真过程中，假设在 t_1 时刻检测出了连通区域 C_i ，形成气体空腔 P_i ，并且在 t_2 时刻气体空腔对应的网格区域 R_i 内所有顶点都附着在了模型表面。于是在 t_1 、 t_2 两时刻，

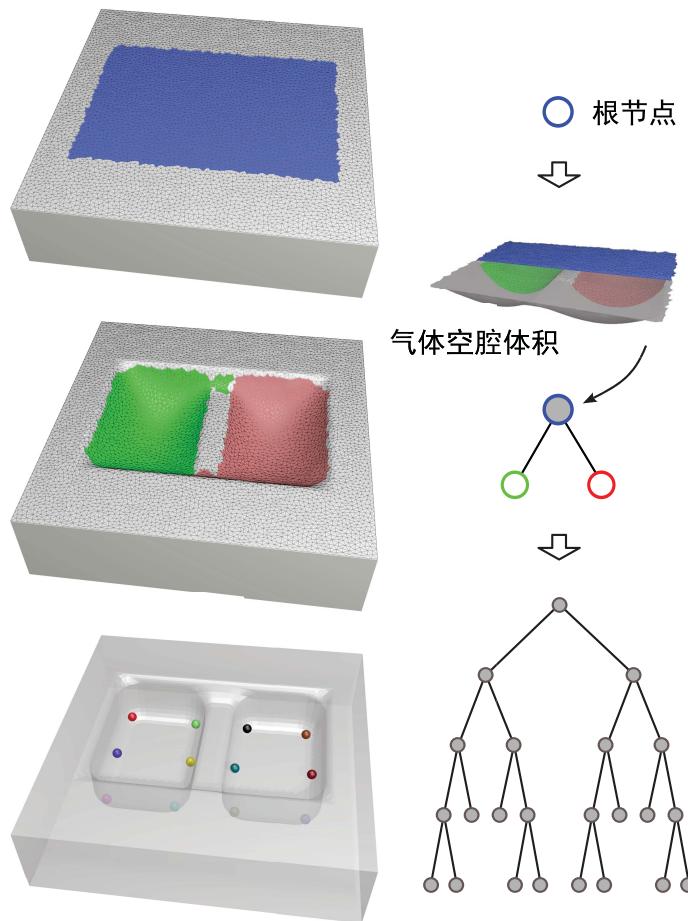


图 4.10 气体空腔森林的构造方法 当一个气体空腔被直接从塑料片网格上检测出来，这个气体空腔就形成一棵树的根节点。如果一个气体空腔分裂成两个气体空腔，对应的节点也分裂为两个子节点。当一个气体空腔完全贴在模型表面之后，这个节点就成为了叶节点，排气孔位置就将从叶节点中产生。

R_i 的形状都被记录下来，记为 $R_i^{t_1}$ 和 $R_i^{t_2}$ ，它们之间所形成的空间即为气体空腔 P_i 。因此 P_i 的体积的就通过 $R_i^{t_1}$ 和 $R_i^{t_2}$ 所围成的封闭三角网格来计算。

当一个大的连通区域 C_i 被分割成几个小的连通区域时，对应的气体空腔 P_i 也就从这一步开始被分裂为几个小的气体空腔，于是 P_i 就形成气体空腔树中的一个内部节点，每一个分裂出的小空腔都是其子树。这种内部节点也就意味着其任何一个子节点排出的空气，都会将其父节点空腔的空气先排出。因此，对于体积小于一定阈值的气体空腔，递归地对其进行剪枝，并将其体积加入父节点，直到整个森林中所有的节点体积都大于了设定的阈值。如果直到根节点体积仍然小于阈值，那么这棵树就将被删除。在我们的实验中，阈值设为 1mm^3 。

4.4.3 排气孔布置

当气体空腔树剪枝处理完成后，就开始计算模型上布置排气孔的位置。图4.11所示的是排气孔的典型形状。排气孔的开口需要足够小，从而避免在塑料片上留下痕迹，而下方加粗，从而增大排气量。排气孔开孔的最佳直径根据塑料片的材料和厚度来设置^[162]。在我们的模型中，表面开孔直径设为 1mm ，超过表面下方 3mm 后，孔的直径增加到 3mm 。在我们的真空成型机上，这个尺寸的孔排气量可以达到 $100\text{cm}^3/\text{s}$ 。

对于气体空腔森林中的每个叶节点，记录其最后一个贴到模型上的顶点。在这个顶点贴在模型上的位置设置排气孔，能够排出该气体空腔，以及其全部祖先节点对应气体空腔中的全部空气。由于每个排气孔排气速度都有限，因此当模型和塑料片的尺寸非常大的时候（比如大型真空成型产品制造），排气孔的数量也需要进行计算。不过在实验中，模型尺寸都很小，因此每个叶节点设置一个排气孔已经足够了。由于整个气体空腔森林中每个节点的体积都记录了下来，因此给每个孔的排气量增加限制，并据此计算需要排气孔的数



图 4.11 排气孔结构剖面图 排气孔上端直径 1mm ，防止孔在塑料片上留下痕迹；下端直径 3mm ，增大排气量。在数字模型文件上设置好排气孔后，直接3D打印带有排气孔的模型。

量也是很容易实现的。

对设置的每个排气孔，细孔部分的方向与模型表面垂直，粗孔部分方向竖直向下。对于复杂模型来说，为了防止开孔会与模型表面相交，从而破坏表面形状，我们允许用户手动调整排气孔的位置。在以后的工作中，也许这一步也可以自动来完成。

4.5 原型着色系统

我们采用一台小型真空成型机来完成真空成型操作。这种真空成型机是口腔医学中为制作牙套设计的，我们对其进行了一定的改造。设备如图4.12所示，其工作范围是 $11\text{cm} \times 11\text{cm}$ 。这台机器使用的是尺寸为 $13\text{cm} \times 13\text{cm}$ ，厚度 $1\sim 2\text{mm}$ 的塑料片。机器上有固定塑料片的夹子，通过一个把手可以调整其高度。在实验中，我们使用真空成型机厂家提供的 1mm 厚透明聚乙烯（PE）塑料片。在工业中，还有很多类型的塑料都可以用于真空成型，比如ABS塑料、聚丙烯（PP），聚苯乙烯（PS）等。这些材料在真空成型过程中性质都类似，通过调整参数都适用于仿真计算。

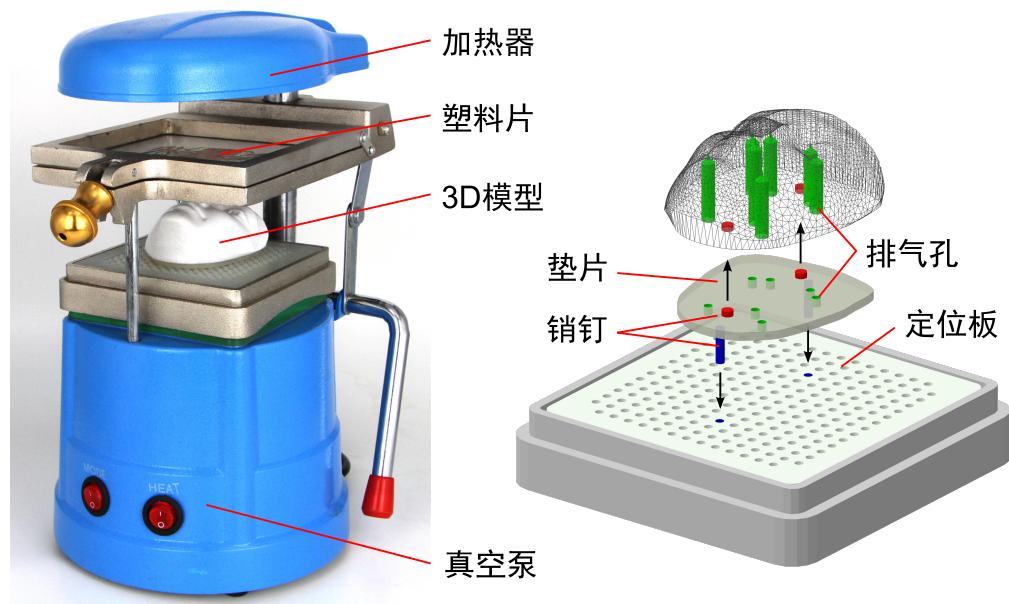


图 4.12 真空成型设备示意图 (左)我们使用一台小型真空成型机。打印了图案的塑料片和模型都已经加载到了设备上，待加热完成后，转动手柄可以将软化的塑料片向下移动到工作位置，打开真空泵开关即可完成真空成型操作。(右)模型固定在设备上的方法。一块与模型底面形状相同的垫片上，向上下方向都设有销钉，打印模型时留有销孔，从而将模型同定位板连为一个整体，装在真空成型机的模具槽内。

为了给3D模型添加设计的纹理，系统需要能够将塑料片上的图案，与3D打印的模型按照仿真计算时的位置关系对齐。否则，实际着色就无法像虚拟仿真中那样精确了。因此我们设计了一些简单的机制，使得图案可以和模型对齐。

4.5.1 塑料片上印刷图案

在虚拟计算中，使用的塑料片网格尺寸与机器的工作区域一致。在生成图像时，把塑料片网格的矩形边界一起绘制出来，在经校准的打印机打印出来之后，这个矩形的大小与机器上塑料片夹的内边缘刚好能够重合。因此在夹装塑料片时，只要将线条与边框重合便可以保证塑料片安装在了正确的位置。

我们使用贴花工艺将图案印在塑料片上。图案首先用HP CP3525dn彩色激光打印机打印在贴花膜上。贴花膜是一层很薄的透明塑料膜，因此打印的时候是附着在一层硬纸上的，使用时再将其撕下。然后将贴花膜裁剪成塑料片的尺寸，将其贴在塑料片的表面，于是印有预形变图案的塑料片就制作好了。

由于我们使用的是透明塑料片，因此将贴花膜贴在正面或反面均可。在实验中，我们将贴花膜贴在塑料片和模型中间，这样塑料片就可以作为彩色图案的保护膜，同时还为模型表面增加了高光的效果。当塑料片上附着了贴花膜之后，其拉伸性会变弱，因此虚拟仿真中塑料片物理参数应当是附着了贴花膜之后物理参数。

如果用UV平板打印机在塑料片上直接打印图案，会更加方便和精确。但是由于在真空成型过程中塑料片要经受高温和大比例拉伸，普通UV墨水在此过程中会变色，甚至被拉出裂缝。因此如果用UV平板打印机，需要使用真空成型专用的墨水^[164]。相比之下，激光打印机通过半色调（halftoning）来构成图案，而非一层完整的墨水，因此通过贴花工艺印刷的图案没有遇到这个问题。

4.5.2 模型3D打印

3D打印模型在真空成型机上的位置同样需要固定。我们用定位销将模型与机器固定在一起，如图4.12所示。在模型下方，打印了一块能牢牢卡在机器模具槽中的定位板。这块定位板上开了许多规则排列的孔，既作为通风孔，又作为定位销孔，实验中所有模型都通过这些孔固定在定位板上。模型和定位板之间还夹了一块垫片，是按照模型底部的形状生成的，用于将模型抬高，能够在剪掉塑料片多余部分时更加方便。模型的排气孔同样延伸到垫片上，除此之外还设有四枚销钉，朝下两枚将垫片固定在定位板上，朝上两枚固定模型。通过这种方法就把模型在机器上也固定了下来。

我们用Stratasys Eden260v 3D打印机，使用VeroWhitePlus光敏树脂打印实验中用到的模型。该打印机能够打印高精度的模型，并且模型表面在打磨之后能够非常光滑。如果模型的工作表面是一个高度场，那么成型之后的塑料片能够非常容易的从模型上取下，因此这样打印的模型可以在大批量彩色塑料曲面生产中作为模具使用。当然也可以采用低成本的材料熔融堆积（FDM）3D打印机，但这种模型表面比较粗糙，一旦与热塑料片粘在一起，就不容易取下来了。因此对于这种模型，成型后将塑料片多余部分剪掉后，就直接作为最终的产品了。

4.5.3 真空成型

当把塑料片和模型都固定在机器上之后，就可以开始真空成型操作了。首先移除模具槽，把塑料片降到最低，并开始加热，这样可以远离加热管使其受热更加均匀。塑料片的温度会影响其物理性质，因此每次操作需要相同的加热时间，并且与测定塑料片参数时加热的时间相同（实验中用5分钟）。

加热结束后，立即将塑料片升起，把模具槽装入机器，并把塑料片压入工作位置。塑料片一接触模型表面就开始被拉伸，如图4.13(a)所示。当塑料片被放置于工作位置，其下方将形成一个密闭空间。开启真空泵，塑料片将在大气压力作用下迅速被拉伸，并贴紧模型表面。于是图案也就随塑料片一起贴在了模型上，如图4.13(b)所示。等温度降至室温，关闭真空泵，沿模型底部将塑料片多余部分剪掉，一个贴了彩色纹理的模型就做好了（图4.13(c)）。

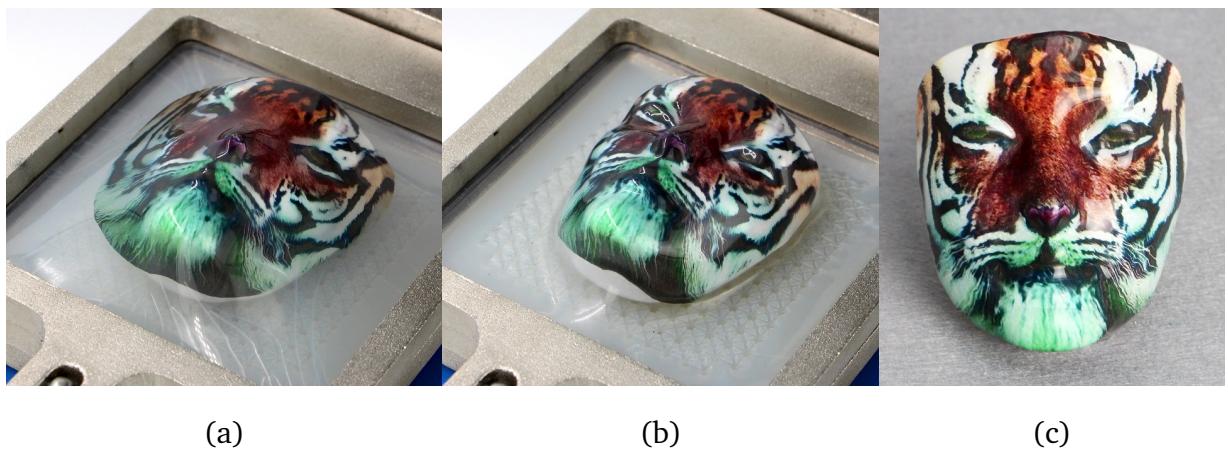


图 4.13 真空成型的过程 (a) 塑料片加热到工作温度后，放置到工作位置。热塑料片一接触到模型表面，就开始产生形变。(b) 打开真空泵，大气压力将塑料片压紧在模型的表面上。(c) 当温度降至室温，塑料片重新硬化。把塑料片多余部分剪掉，彩色模型就做好了。

4.6 实验结果

我们用一台装了Intel i7-4930K 3.4GHz处理器，64G内存的机器进行虚拟仿真。仿真使用的塑料片网格包含5148个顶点，10058个三角形。更高精度的离散化已经不会使着色精度进一步提升了。目前我们的程序并没有经过优化，而且只用了一个CPU线程，不过所有实验都能够一分钟内计算完成（包括仿真和实时渲染）。我们相信通过优化程序，多线程甚至GPU计算，仿真速度还能够大幅提高。

在塑料片物理模型中，材料的物理性质主要有4个参数：杨氏模量、泊松比、屈服极限以及强化系数。材料在工作温度的杨氏模量($Y = 1\text{ MPa}$)由材料生产商给出。高分子材料在拉伸超过屈服极限之后，泊松比接近0.5^[158]，这意味着塑性形变阶段材料体积不可压缩。由于材料的屈服极限比较低，塑料片在开始拉伸不久就开始塑性形变，因此简单地在整个仿真过程都把泊松比($\nu = 0.5$)当做常数。初始屈服极限($\sigma_0 = 220\text{ Pa}$)以及强化系数($K = 1000\text{ Pa}$)的值通过实验来确定。

在实验中确定初始屈服极限 σ_0 和强化系数 K 值的方法为，先用画有网格线的塑料片，以图4.14中的平顶圆锥作为模具进行热塑成型。然后对 σ_0 和 K 的值各取20个采样点，构成 20×20 组测试数据，用同样的网格线作为塑料片的纹理进行虚拟仿真，得到400个仿真结果，并从中选取与真实物理实验结果最接近的仿真结果。在一台12核CPU的机器上用多进程计算，400个仿真时间耗时约1小时。然后可以根据需要，在该结果对应参数附近再进行更加密集的采样，进一步优化参数的估计。由于在物理参数的连续变化过程中，仿真结果也是连续变化的，因此可以很容易地选出与实验结果最接近的仿真结果。也可以将实验结果扫面下来，用自动的算法来计算仿真结果与实际物理试验的误差，从而选出最优的参数估计^[133]。

控制模型对塑料片支持力的系数 k_r 人为设为 $k_r = 5 \times 10^8 \text{ N/m}^3$ 。该数值足以保证正确的碰撞。摩擦系数 μ 与塑料片和3D打印的材料都有关。对于用Stratasys Eden260v 3D打印机VeroWhitePlus材料来说， $\mu = 0.8$ 能够计算出不错的结果。对于非常粗糙的模型，比如用FDM打印机打印的模型，几乎看不出塑料片在模型上的滑动。这种情况下，可以进一步将仿真模型简化：网格顶点一旦接触到模型表面就对其施加位置约束，从而使计算量进一步降低。

4.6.1 精度

我们用一个平顶圆锥模型来验证系统的精度，如图4.14所示。这个模型高3cm，顶面半径1cm，底面半径2cm。7条蓝色圆圈在模型侧面均匀分布，其中最小和最大的圆分别与

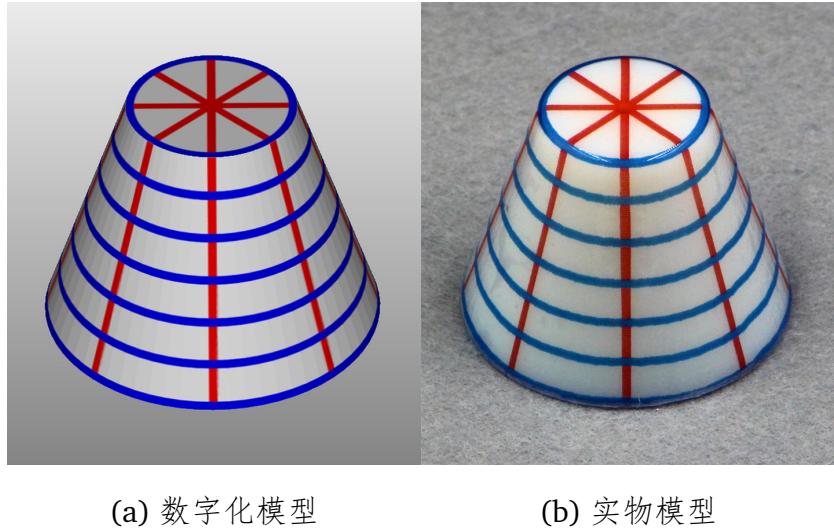


图 4.14 真空成型精度实验 通过带有经纬线纹理的平顶圆锥来测试真空成型的精度。我们方法的着色误差不超过1mm。

模型的顶面和底面圆重合。线条的宽度为1mm。

通过我们的方法制作了带有相同纹理的实物模型，结果如图4.14(b)所示。可以看出实物模型上蓝色圆圈出现的位置与数字化模型几乎完全一致。由于模型的形状非常规则，可以方便地用游标卡尺对模型上的图案进行测量。经检验，热塑成型的误差不超过1mm。误差也可以采用Schüller等人^[133]提出的方法，将制作出的实物模型扫描成数字化模型之后，再计算其误差。

4.6.2 与水转印的对比

我们将真空热塑成型的结果与计算水转印进行了对比，如图4.7所示。塑料片贴到猫盆模型底部的时候，被拉伸到了超过原面积的6倍。在水转印中，如此大的拉伸足以将PVA膜给撕破，使得转印的图案变得非常扭曲，如图4.7(b)所示。与之相比，塑料在拉伸率很大的时候依然能够保持均匀拉伸，因此形成的图案被很好地保存了下来。如4.3.3节所述，塑料片拉伸导致颜料密度降低，因此颜色矫正能够对这种颜色失真进行很好地矫正（图4.7(d)）。

水转印方法中，PVA膜可以在水中完全溶解，因此模型经过清洗后只会在其表面留下颜料，而热塑成型则会将作为载体的塑料片一起贴在模型的表面。因此水转印和热塑成型有各自所使用的场合。



图 4.15 通过真空热塑成型制做的彩色模型 每个结果大图为着色之后的模型，左上小图为数字化模型的渲染图，左下小图为未着色的3D模型。

4.6.3 更多的结果

图4.15中展示了更多用真空热塑成型方法制作的彩色模型。女性头像的浮雕是用Chai等人^[165]的方法从单张图片中生成的，高度场与图片的每个像素是对齐的。3D打印制作出头像模型，并用我们的方法使得3D模型上的点也与像素对齐。海龟模型的腿、尾巴、头部都向外凸出，这种模型通过真空热塑成型也可以制作。跑车模型并不是一个高度场，通过计算预形变图案，可以使得塑料片在真空热塑成型中包裹整个模型，因此除了底面外，其余各面都能够被着色。

除了海龟模型不需要设置排气孔外，其余模型都必须设置排气孔，才能使塑料片贴紧模型表面。具体来说，老虎面具模型（图4.13）上设置了7个排气孔，浮雕模型上有12个，跑车模型上有6个，猫盆模型上有8个。

4.7 小结

本工作用面网格对塑性薄膜进行建模，模拟软化塑料片在大气压力下发生形变，并贴在固体表面的运动过程。并将该仿真方法应用于塑料成型工艺——真空热塑成型的虚拟化。我们提出“可计算热塑成型”的方法，并结合3D打印，将带纹理的虚拟数字化模型制作成为实物模型。通过虚拟仿真计算出预形变图案，并通过贴花工艺印在塑料片上。通过在仿真过程中跟踪气体空腔的出现和消失，计算出排气孔的布局，从而使最终的模型不会有明显的气体空腔。最后我们提供了用简单部件搭建硬件系统的方法，最终完成真空热塑成型。

目前这项技术还存在一定的局限性。比如，现在还只能处理亏格为0的模型，而且对于类似高度场形状的模型着色效果最好。目前的仿真没有对塑料片的自碰撞进行检测，因此对于带有类似把手结构的模型就无法计算出塑料片贴在一起的效果。如果3D模型的内部结构被限定不能改变，那么也就不能用这种办法设置排气孔了。对于接下来的工作，我

们希望将此方法应用于更高亏格的模型，类似于水转印的多次着色，对模型放置位置也进行优化从而使得更大的面积，甚至底部都能被着色，用动态仿真取代静态平衡的计算方法来获得更高的精度，以及尝试更多的材料来获得包括漫反射、各向异性、甚至用户指定BRDF效果的曲面。

第5章 总结和展望

物理仿真是图形学重要的研究领域，其目标是在计算机中再现真实物理世界的各种现象，不仅能够用于制作真实感的电影特效，为用户带来丰富的视觉体验，还能够用于提升传统制造工艺，将工艺流程虚拟化并通过仿真对其进行改进，拓宽应用范围，使其满足数字化制造的要求。

本文用面网格对水滴、粘性流体薄膜、塑性薄膜等三种物质进行了建模和仿真。首先提出了一种基于面网格的实时水滴运动仿真方法，用面网格直接形变模拟水滴运动，并通过布尔运算及网格优化实现水滴之间的融合与分裂。然后，本文用面网格对薄层结构的粘性流体薄膜和塑性薄膜进行了仿真，并将这些仿真技术应用于两种传统制造工艺的虚拟化，提出“可计算水转印刷”和“可计算热塑成型”的方法，使其能够用于三维曲面全彩着色。我们制造了这些新型工艺的原型系统，并通过实验证明了其实用性和可靠性。

5.1 本文工作总结

本文的第一章，对物理仿真及其在数字化制造中的应用进行了综述。对图形学中物理仿真，尤其是通过面网格进行物理仿真的相关工作进行了总结，并介绍了将物理仿真应用于3D打印领域的相关研究成果。

本文的第二章，我们提出了一种基于面网格的实时水滴仿真方法。该方法通过对面网格直接形变，能够对水滴的运动，及其与物体表面接触并滑动的现象进行仿真。该方法可以分为两部分：网格形变运算，对单个水滴进行物理仿真；网格拓扑运算，处理水滴的融合与分裂，并对网格进行优化。该方法可以对水滴的多种现象进行仿真，也可以将其集成到现有的流体仿真系统中，为通用流体仿真增强小尺度仿真的细节。

本文的第三章，我们用面网格对浮在水面上的粘稠流体薄膜进行仿真，模拟固体物经过薄膜浸入水中，薄膜被拉伸并包裹住物体表面的过程。并将该仿真方法应用于曲面着色工艺——水转印的虚拟化，首次提出了“计算水转印刷”的方法，实现了三维实物按照用户设计纹理的准确着色。首次提出了水转印物理过程的计算模型，对整个物理过程进行仿

真，从而计算出膜在水转印过程中的拉伸与形变。进一步将单次水转印扩展为多次水转印，解决了复杂曲面着色时图案失真的问题。我们用常用零件制造出了定位水转印装置，能够重现虚拟仿真的结果，按照用户设计的纹理将图案印在物体的表面。通过一系列的实验，证明了这套系统能够进行高质量的曲面着色。

本文的第四章，我们用面网格对塑性薄膜进行建模，模拟软化塑料片在大气压力下发生形变，并贴在固体表面的运动过程。并将该仿真方法应用于塑料成型工艺——真空热塑成型的虚拟化。提出“可计算热塑成型”的方法，并结合3D打印，将带纹理的虚拟数字化模型制作成为实物模型。通过虚拟仿真计算出预形变图案，并通过贴花工艺印在塑料片上。通过在仿真过程中跟踪气体空腔的出现和消失，计算出排气孔的布局，从而使最终的模型不会有明显的气体空腔。最后提供了用简单部件搭建硬件系统的方法，最终完成真空热塑成型。

5.2 进一步工作展望

物理仿真是一个很广阔的研究领域，还有更多的物理现象等待我们去发掘。接下来的研究可以从下述方向着手：

- **更精确的物理模型** 真实物理世界往往都是复杂的非线性系统，而物理模型是对真实现象的简化与近似，通常都希望用线性模型来简化计算。但随着对精度要求的不断提高，就需要更加准确的非线性物理模型。一个可能的方案，是采用数据驱动（data driven）的方式进行仿真。通过大量的真实物理实验采集数据，拟合出非线性模型，并用这种模型进行仿真。
- **更快的仿真速度** 随着物理仿真精度要求的提高，计算量也迅速增加，因此如何提高仿真速度一直是研究的内容。物理仿真时有许多计算都是用来进行碰撞检测，因此提高碰撞检测的效率会是研究的重点。在物理方程求解时，可以通过多线程、GPU并行等方法提高运算速度。
- **更多现象的仿真** 物理仿真能够对真实世界进行一定程度的还原和预测。自然现象、工程技术、人类行为都能够作为仿真的对象。在某些特殊领域，物理仿真甚至取代了部分物理实验，如核武器爆炸实验。因此对更多现象进行仿真，能够让我们加深对物理世界的理解，发现更多的规律。在图形学中，也需要对更多的自然现象提出简化物理模型，以满足日益增加的电影特效制作的需求。

- **更多制造工艺的虚拟化** 工业4.0的目标是将整个工厂的生产都进行虚拟化，因此会有更多的制造工艺通过物理仿真得到提升。这些工艺可能涉及多种物理、化学变化，在长期的生产实践中工艺已经非常成熟。通过物理仿真，可以迅速的对最终结果进行预测，从而为工艺的改进提供更多的途径。

我们提出的两种通过物理仿真实现的曲面着色技术，还有许多进一步提升的空间，从而实现更加精准、丰富的着色效果。

我们已经用简单零件搭建了一套计算水转印原型系统，并成功将其应用于曲面着色。不过目前这套系统还存在一些缺点。首先，每个模型都需要先用深度相机定位之后才能进行仿真，而且每次更换模型都需要重新定位并重新仿真，然后才能进行一次水转印，操作非常麻烦，不适合流水线生产。其次，深度相机测量的是模型相对于PVA膜定位框的位置，而PVA膜与定位框之间依然存在微小的间隙，喷洒活化剂之后会使图案产生微小的形变，这些因素使得模型与图案之间的误差偏大。最后，这套简单的系统只有一个直线电机控制模型运动，因此只能以固定角度沿竖直方向运动。为了克服这些缺陷，进一步提升计算水转印的性能及自动化程度，我们打算使用一台六自由度工业机器人，实现水转印的自动化。通过计算机控制，机器人可以使模型沿任意轨迹精确运动。将其应用于水转印中，能够显著提升水转印的工艺水平。首先，对同一模型可以先进行仿真并批量打印图案，水转印操作时用机器人将模型移动到目标位置，并执行水转印操作，这样只需一次仿真就能够实现批量模型着色。其次，通过视觉系统直接测量喷洒活化剂之后图案的位置，并用机器人将模型移动到与图案匹配的位置上，可以使模型直接与图案对齐，避免了机械间隙与PVA膜形变导致的定位误差。最后，可以计算出水转印更优化的轨迹，并通过机器人来控制模型运动，这样可以使膜的平均拉伸程度更小，甚至原本需要多次转印的模型现在只要单次转印就能实现。

在可计算热塑成型中，我们使用了一台经过改造的简易小型真空成型机。如果使用更专业的设备，则可以对工艺进行更多的优化。比如通过辅助的薄膜拉伸装置，对薄膜局部进行预拉伸，使得抽真空时薄膜形变更加均匀。可以仿照水转印中多次转印的原理，将一个模型分割为两半，用两个模具同时真空成型，并在塑料片冷却硬化之前将其拼成一个整体，从而制作出封闭的完整彩色模型。在塑料片的选择上，我们目前只用了厂家提供的透明塑料片，使得制作的模型都呈现高光的效果。在未来，可以尝试更多类型的塑料片，包括纯白底色、亚光表面、甚至成型之前先在塑料片上打印特定的BRDF，使得制作出的模型具有特殊的外观效果。

传统工艺是一个巨大的宝库，有许多成熟的技术在经过物理仿真的提升之后，能够发挥更大的威力。下面列举几项有进一步开发的潜力，并用于曲面着色的技术。在接下来的工作中，可以对其进行尝试。1，参照传统感光相纸的原理，将感光物质喷洒在三维曲面上，并用投影仪曝光，经显影、定影，在模型表面呈现彩色纹理。2，利用移印的原理，先将预形变图案印刷在平面上，用移印头蘸取尚未干燥的油墨，并转印在模型表面。由于移印头用柔软的硅胶制作而成，压在模型表面上可以与其表面相贴合，从而将油墨保留在模型表面。3，可以对喷墨打印机进行改造，将送纸机构更换成为六自由度运动平台，使得安装在平台上的模型能够与喷头的运动相互配合，从而实现曲面上直接彩色喷墨。

参考文献

- [1] Frank Losasso, Jerry Talton, Nipun Kwatra, Ronald Fedkiw. Two-way coupled sph and particle level set fluid simulation[J]. IEEE Transactions on Visualization and Computer Graphics, July 2008. 14(4):797–804. ISSN 1077-2626.
- [2] A. Selle, J. Su, G. Irving, R. Fedkiw. Robust high-resolution cloth using parallelism, history-based collisions, and accurate friction[J]. IEEE Transactions on Visualization and Computer Graphics, March 2009. 15(2):339–350. ISSN 1077-2626.
- [3] Tobias Pfaff, Rahul Narain, Juan Miguel de Joya, James F. O'Brien. Adaptive tearing and cracking of thin sheets[J]. ACM Trans. Graph., July 2014. 33(4):110:1–110:9. ISSN 0730-0301.
- [4] Andrew Witkin, David Baraff. Physically based modeling differential equation basics[J]. online]. California, Pixar. Available from: <http://www.pixar.com/companyinfo/research/pbm2001/notesc.pdf> [Accessed 5 July 2005], 2001.
- [5] David Baraff. Physically based modeling implicit methods for differential equations[J]. Course Notes, SIGGRAPH, 2001.
- [6] Ellis H Dill. Continuum mechanics: elasticity, plasticity, viscoelasticity[M]. Boca Raton, London, New York:CRC Press, 2006.
- [7] Martin Wicke, Daniel Ritchie, Bryan M. Klingner, Sebastian Burke, Jonathan R. Shewchuk, James F. O'Brien. Dynamic local remeshing for elastoplastic simulation[J]. ACM Transactions on Graphics (SIGGRAPH 2010), July 2010. 29(4):49:1–49:11. ISSN 0730-0301.
- [8] Shaofan Li, Wing Kam Liu. Meshfree and particle methods and their applications[J]. Applied Mechanics Reviews, 2002. 55(1):1–34.
- [9] Jos Stam. Real-time fluid dynamics for games[C]//Proceedings of the game developer conference. volume 18. 2003:25.
- [10] Chris Wojtan, Greg Turk. Fast viscoelastic behavior with thin features[J]. ACM Trans. Graph., August 2008. 27(3):47:1–47:8.
- [11] Richard Tonge, Feodor Benevolenski, Andrey Voroshilov. Mass splitting for jitter-free parallel rigid body simulation[J]. ACM Transactions on Graphics (TOG), 2012. 31(4):105.
- [12] James K Hahn. Realistic animation of rigid bodies[J]. ACM SIGGRAPH Computer Graphics, 1988.

- 22(4):299–308.
- [13] David Baraff. Analytical methods for dynamic simulation of non-penetrating rigid bodies[J]. ACM SIGGRAPH Computer Graphics, 1989. 23(3):223–232.
- [14] Brian Mirtich. Timewarp rigid body simulation[C]//Proceedings of the 27th annual conference on Computer graphics and interactive techniques. ACM Press/Addison-Wesley Publishing Co., 2000:193–200.
- [15] Eran Guendelman, Robert Bridson, Ronald Fedkiw. Nonconvex rigid bodies with stacking[J]. ACM Transactions on Graphics (TOG), 2003. 22(3):871–878.
- [16] Bin Wang, François Faure, Dinesh K Pai. Adaptive image-based intersection volume[J]. ACM Transactions on Graphics (TOG), 2012. 31(4):97.
- [17] Samantha Ainsley, Etienne Vouga, Eitan Grinspun, Rasmus Tamstorf. Speculative parallel asynchronous contact mechanics[J]. ACM Transactions on Graphics (TOG), 2012. 31(6):151.
- [18] Sara C Schvartzman, Álvaro G Pérez, Miguel A Otaduy. Star-contours for efficient hierarchical self-collision detection[J]. ACM Transactions on Graphics (TOG), 2010. 29(4):80.
- [19] Shu-Wei Hsu, John Keyser. Piles of objects[J]. ACM Transactions on Graphics (TOG), 2010. 29(6):155.
- [20] Jernej Barbič, Doug L James. Subspace self-collision culling[J]. ACM Transactions on Graphics (TOG), 2010. 29(4):81.
- [21] Jérémie Allard, François Faure, Hadrien Courtecuisse, Florent Falipou, Christian Duriez, Paul G Kry. Volume contact constraints at arbitrary resolution[J]. ACM Transactions on Graphics (TOG), 2010. 29(4):82.
- [22] Fuchang Liu, Takahiro Harada, Youngeun Lee, Young J Kim. Real-time collision culling of a million bodies on graphics processing units[J]. ACM Transactions on Graphics (TOG), 2010. 29(6):154.
- [23] Andrew Selle, Michael Lentine, Ronald Fedkiw. A mass spring model for hair simulation[J]. ACM Transactions on Graphics (TOG), 2008. 27(3):64.
- [24] Rony Goldenthal, David Harmon, Raanan Fattal, Michel Bercovier, Eitan Grinspun. Efficient simulation of inextensible cloth[J]. ACM Transactions on Graphics (TOG), 2007. 26(3):49.
- [25] Taylor Patterson, Nathan Mitchell, Eftychios Sifakis. Simulation of complex nonlinear elastic bodies using lattice deformers[J]. ACM Transactions on Graphics (TOG), 2012. 31(6):197.
- [26] Xavier Provot. Deformation constraints in a mass-spring model to describe rigid cloth behaviour[C]//Graphics interface. Canadian Information Processing Society, 1995:147–147.
- [27] David Baraff, Andrew Witkin. Large steps in cloth simulation[C]//Proceedings of the 25th annual conference on Computer graphics and interactive techniques. ACM, 1998:43–54.
- [28] Kwang-Jin Choi, Hyeong-Seok Ko. Stable but responsive cloth[C]//ACM SIGGRAPH 2005 Courses. ACM, 2005:1.

- [29] Jonathan M Kaldor, Doug L James, Steve Marschner. Simulating knitted cloth at the yarn level[J]. ACM Transactions on Graphics (TOG), 2008. 27(3):65.
- [30] Jonathan M Kaldor, Doug L James, Steve Marschner. Efficient yarn-based cloth with adaptive contact linearization[J]. ACM Transactions on Graphics (TOG), 2010. 29(4):105.
- [31] Pascal Volino, Nadia Magnenat-Thalmann, Francois Faure. A simple approach to nonlinear tensile stiffness for accurate cloth simulation[J]. ACM Transactions on Graphics, 2009. 28(4):Article–No.
- [32] Huamin Wang, James F O'Brien, Ravi Ramamoorthi. Data-driven elastic models for cloth: modeling and measurement[J]. ACM Transactions on Graphics (TOG), 2011. 30(4):71.
- [33] Robert Bridson, Ronald Fedkiw, John Anderson. Robust treatment of collisions, contact and friction for cloth animation[J]. ACM Trans. Graph., 2002. 21(3):594–603.
- [34] David Baraff, Andrew Witkin, Michael Kass. Untangling cloth[J]. ACM Transactions on Graphics (SIGGRAPH 2003), July 2003. 22(3):862–870. ISSN 0730-0301.
- [35] Matthias Müller, Bruno Heidelberger, Marcus Hennix, John Ratcliff. Position based dynamics[J]. Journal of Visual Communication and Image Representation, 2007. 18(2):109–118.
- [36] Matthias Müller. Hierarchical Position Based Dynamics[C]//Francois Faure, Matthias Teschner. Workshop in Virtual Reality Interactions and Physical Simulation "VRIPHYS" (2008). The Eurographics Association, 2008.
- [37] Matthias Müller, Nuttapong Chentanez. Wrinkle meshes[C]//Proceedings of the 2010 ACM SIGGRAPH/Eurographics symposium on computer animation. Eurographics Association, 2010:85–92.
- [38] Edilson De Aguiar, Leonid Sigal, Adrien Treuille, Jessica K Hodgins. Stable spaces for real-time clothing[J]. ACM Transactions on Graphics (TOG), 2010. 29(4):106.
- [39] Wei-Wen Feng, Yizhou Yu, Byung-Uck Kim. A deformation transformer for real-time cloth animation[J]. ACM Transactions on Graphics (TOG), 2010. 29(4):108.
- [40] Huamin Wang, Florian Hecht, Ravi Ramamoorthi, James F O'Brien. Example-based wrinkle synthesis for clothing animation[J]. ACM Transactions on Graphics (TOG), 2010. 29(4):107.
- [41] Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, Eitan Grinspun. Discrete elastic rods[J]. ACM transactions on graphics (TOG), 2008. 27(3):63.
- [42] Jie Tan, Greg Turk, C. Karen Liu. Soft body locomotion[J]. ACM Trans. Graph., July 2012. 31(4):26:1–26:11. ISSN 0730-0301.
- [43] Eitan Grinspun, Anil N Hirani, Mathieu Desbrun, Peter Schröder. Discrete shells[C]//Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation. Eurographics Association, 2003:62–67.
- [44] Adam W. Bargteil, Chris Wojtan, Jessica K. Hodgins, Greg Turk. A finite element method for animating large viscoplastic flow[J]. ACM Trans. Graph., July 2007. 26(3).

- [45] James F O'brien, Jessica K Hodgins. Graphical modeling and animation of brittle fracture[C]//Proceedings of the 26th annual conference on Computer graphics and interactive techniques. ACM Press/Addison-Wesley Publishing Co., 1999:137–146.
- [46] James F O'brien, Adam W Bargteil, Jessica K Hodgins. Graphical modeling and animation of ductile fracture[J]. ACM transactions on graphics (TOG), 2002. 21(3):291–294.
- [47] Neil Molino, Zhaosheng Bao, Ron Fedkiw. A virtual node algorithm for changing mesh topology during simulation[C]//ACM SIGGRAPH 2005 Courses. ACM, 2005:4.
- [48] Zhili Chen, Miaojun Yao, Renguo Feng, Huamin Wang. Physics-inspired adaptive fracture refinement[J]. ACM Transactions on Graphics (TOG), 2014. 33(4):113.
- [49] Rob Burgoon, Zoë J Wood, Eitan Grinspan. Discrete shells origami[C]//21st International Conference on Computers and their Applications (CATA 2006) Proceedings. 2006.
- [50] Damien Rohmer, Marie-Paule Cani, Stefanie Hahmann, Boris Thibert. Folded paper geometry from 2d pattern and 3d contour[C]//Eurographics 2011 (short paper). 2011:21–24.
- [51] Rahul Narain, Tobias Pfaff, James F O'Brien. Folding and crumpling adaptive sheets[J]. ACM Transactions on Graphics (TOG), 2013. 32(4):51.
- [52] Eder Miguel, Rasmus Tamstorf, Derek Bradley, Sara C Schwartzman, Bernhard Thomaszewski, Bernd Bickel, Wojciech Matusik, Steve Marschner, Miguel A Otaduy. Modeling and estimation of internal friction in cloth[J]. ACM Transactions on Graphics (TOG), 2013. 32(6):212.
- [53] Miklós Bergou, Basile Audoly, Etienne Vouga, Max Wardetzky, Eitan Grinspan. Discrete viscous threads[J]. ACM Transactions on Graphics (TOG), 2010. 29(4):116.
- [54] Christopher Batty, Andres Uribe, Basile Audoly, Eitan Grinspan. Discrete viscous sheets[J]. ACM Trans. Graph., July 2012. 31(4):113:1–113:7.
- [55] Mark Carlson, Peter J. Mucha, R. Brooks Van Horn, III, Greg Turk. Melting and flowing[C]//Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation (SCA 2002). ISBN 1-58113-573-4, 2002:167–174.
- [56] Tolga G Goktekin, Adam W Bargteil, James F O'Brien. A method for animating viscoelastic fluids[J]. ACM Transactions on Graphics (TOG), 2004. 23(3):463–468.
- [57] Christopher Batty, Robert Bridson. Accurate viscous free surfaces for buckling, coiling, and rotating liquids[C]//Proceedings of the 2008 ACM SIGGRAPH/Eurographics symposium on computer animation. Eurographics Association, 2008:219–228.
- [58] Alexey Stomakhin, Craig Schroeder, Lawrence Chai, Joseph Teran, Andrew Selle. A material point method for snow simulation[J]. ACM Transactions on Graphics (TOG), 2013. 32(4):102.
- [59] Afonso Paiva, Fabiano Petronetto, Thomas Lewiner, Geovan Tavares. Particle-based viscoplastic fluid/solid simulation[J]. Computer-Aided Design, 2009. 41(4):306–314.

- [60] Barbara Solenthaler, Markus Gross. Two-scale particle simulation[J]. ACM Transactions on Graphics (TOG), 2011. 30(4):81.
- [61] Nils Thürey, Chris Wojtan, Markus Gross, Greg Turk. A multiscale approach to mesh-based surface tension flows[J]. ACM Transactions on Graphics (SIGGRAPH 2010), July 2010. 29(4):48:1–48:10. ISSN 0730-0301.
- [62] Matthias Müller, David Charypar, Markus Gross. Particle-based fluid simulation for interactive applications[C]//Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation (SCA 2003). ISBN 1-58113-659-5, 2003:154–159.
- [63] Barbara Solenthaler, Renato Pajarola. Predictive-corrective incompressible sph[J]. ACM transactions on graphics (TOG), 2009. 28(3):40.
- [64] Hagit Schechter, Robert Bridson. Ghost sph for animating water[J]. ACM Transactions on Graphics (TOG), 2012. 31(4):61.
- [65] Olivier Mercier, Cynthia Beauchemin, Nils Thuerey, Theodore Kim, Derek Nowrouzezahrai. Surface turbulence for particle-based liquid simulations[J]. ACM Transactions on Graphics (TOG), 2015. 34(6):202.
- [66] Michael Kass, Gavin Miller. Rapid, stable fluid dynamics for computer graphics[C]//Proceedings of the 17th annual conference on Computer graphics and interactive techniques (SIGGRAPH '90). ISBN 0-89791-344-2, 1990:49–57.
- [67] Jim X Chen, Niels Da Vitoria Lobo. Toward interactive-rate simulation of fluids with moving obstacles using navier-stokes equations[J]. Graphical Models and Image Processing, 1995. 57(2):107–116.
- [68] James FO Brien, Jessica K Hodgins. Dynamic simulation of splashing fluids[C]//Computer Animation'95., Proceedings. IEEE, 1995:198–205.
- [69] Nick Foster, Dimitri Metaxas. Realistic animation of liquids[C]//Proc. of Graphics interface 1996. 1996:204–212.
- [70] Jos Stam. Stable fluids[C]//Proceedings of the 26th annual conference on Computer graphics and interactive techniques (SIGGRAPH '99). ISBN 0-201-48560-5, 1999:121–128.
- [71] Nick Foster, Ronald Fedkiw. Practical animation of liquids[C]//Proceedings of the 28th annual conference on Computer graphics and interactive techniques (SIGGRAPH 2001). ISBN 1-58113-374-X, 2001:23–30.
- [72] Douglas Enright, Stephen Marschner, Ronald Fedkiw. Animation and rendering of complex water surfaces[J]. ACM Transactions on Graphics (SIGGRAPH 2002), July 2002. 21(3):736–744. ISSN 0730-0301.
- [73] Frank Losasso, Frédéric Gibou, Ron Fedkiw. Simulating water and smoke with an octree data structure[J]. ACM Transactions on Graphics (SIGGRAPH 2004), August 2004. 23(3):457–462. ISSN

0730-0301.

- [74] Andrew Selle, Nick Rasmussen, Ronald Fedkiw. A vortex particle method for smoke, water and explosions[J]. ACM Transactions on Graphics (TOG), 2005. 24(3):910–914.
- [75] Chris Wojtan, Nils Thürey, Markus Gross, Greg Turk. Deforming meshes that split and merge[J]. ACM Transactions on Graphics (SIGGRAPH 2009), July 2009. 28(3):76:1–76:10. ISSN 0730-0301.
- [76] Chris Wojtan, Nils Thurey, Markus Gross, Greg Turk. Physics-inspired topology changes for thin fluid features[J]. ACM Transactions on Graphics (SIGGRAPH 2010), July 2010. 29(4):50:1–50:8. ISSN 0730-0301.
- [77] Essex Edwards, Robert Bridson. Detailed water with coarse grids: Combining surface meshes and adaptive discontinuous galerkin[J]. ACM Transactions on Graphics (TOG), 2014. 33(4):136.
- [78] Tobias Pfaff, Nils Thuerey, Markus Gross. Lagrangian vortex sheets for animating fluids[J]. ACM Transactions on Graphics (TOG), 2012. 31(4):112.
- [79] Ronald Fedkiw, Jos Stam, Henrik Wann Jensen. Visual simulation of smoke[C]//Proceedings of the 28th annual conference on Computer graphics and interactive techniques. ACM, 2001:15–22.
- [80] Duc Quang Nguyen, Ronald Fedkiw, Henrik Wann Jensen. Physically based modeling and animation of fire[J]. ACM Transactions on Graphics (TOG), 2002. 21(3):721–728.
- [81] Bryan E Feldman, James F O'Brien, Okan Arikan. Animating suspended particle explosions[J]. ACM Transactions on Graphics (TOG), 2003. 22(3):708–715.
- [82] Bryan E Feldman, James F O'Brien, Bryan M Klingner. Animating gases with hybrid meshes[J]. ACM Transactions on Graphics (TOG), 2005. 24(3):904–909.
- [83] Theodore Kim, Nils Thürey, Doug James, Markus Gross. Wavelet turbulence for fluid simulation[J]. ACM Transactions on Graphics (TOG), 2008. 27(3):50.
- [84] Christopher Horvath, Willi Geiger. Directable, high-resolution simulation of fire on the gpu[J]. ACM Transactions on Graphics (TOG), 2009. 28(3):41.
- [85] Patrick Mullen, Keenan Crane, Dmitry Pavlov, Yiyi Tong, Mathieu Desbrun. Energy-preserving integrators for fluid animation[J]. ACM Transactions on Graphics (TOG), 2009. 28(3):38.
- [86] Huamin Wang, Peter J. Mucha, Greg Turk. Water drops on surfaces[J]. ACM Transactions on Graphics (SIGGRAPH 2005), July 2005. 24(3):921–929. ISSN 0730-0301.
- [87] Fang Da, Christopher Batty, Chris Wojtan, Eitan Grinspun. Double bubbles sans toil and trouble: discrete circulation-preserving vortex sheets for soap films and foams[J]. ACM Transactions on Graphics (TOG), 2015. 34(4):149.
- [88] Byungmoon Kim, Yingjie Liu, Ignacio LLamas, Xiangmin Jiao, Jarek Rossignac. Simulation of bubbles in foam with the volume control method[J]. ACM Transactions on Graphics (TOG), 2007. 26(3):98.
- [89] Paul W Cleary, Soon Hyoung Pyo, Mahesh Prakash, Bon Ki Koo. Bubbling and frothing liquids[J]. ACM

- Transactions on Graphics (TOG), 2007. 26(3):97.
- [90] Nils Thürey, Filip Sadlo, Simon Schirm, Matthias Müller-Fischer, Markus Gross. Real-time simulations of bubbles and foam within a shallow water framework[C]//Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation. Eurographics Association, 2007:191–198.
- [91] Jeong-Mo Hong, Ho-Young Lee, Jong-Chul Yoon, Chang-Hun Kim. Bubbles alive[J]. ACM Transactions on Graphics (TOG), 2008. 27(3):48.
- [92] Oleksiy Busaryev, Tamal K Dey, Huamin Wang, Zhong Ren. Animating bubble interactions in a liquid foam[J]. ACM Transactions on Graphics (TOG), 2012. 31(4):63.
- [93] Toon Lenaerts, Bart Adams, Philip Dutré. Porous flow in particle-based fluid simulations[J]. ACM Transactions on Graphics (TOG), 2008. 27(3):49.
- [94] Frank Losasso, Tamar Shinar, Andrew Selle, Ronald Fedkiw. Multiple interacting liquids[J]. ACM Transactions on Graphics (TOG), 2006. 25(3):812–819.
- [95] Olivier Génevaux, Arash Habibi, Jean-Michel Dischler. Simulating fluid-solid interaction.[C]//Graphics Interface. volume 2003. Citeseer, 2003:31–38.
- [96] Eran Guendelman, Andrew Selle, Frank Losasso, Ronald Fedkiw. Coupling water and smoke to thin deformable and rigid shells[J]. ACM Transactions on Graphics (TOG), 2005. 24(3):973–981.
- [97] Nadir Akinci, Markus Ihmsen, Gizem Akinci, Barbara Solenthaler, Matthias Teschner. Versatile rigid-fluid coupling for incompressible sph[J]. ACM Transactions on Graphics (TOG), 2012. 31(4):62.
- [98] Bo Ren, Chenfeng Li, Xiao Yan, Ming C Lin, Javier Bonet, Shi-Min Hu. Multiple-fluid sph simulation using a mixture model[J]. ACM Transactions on Graphics (TOG), 2014. 33(5):171.
- [99] Alan Brunton, Can Ates Arikán, Philipp Urban. Pushing the limits of 3d color printing: Error diffusion with translucent materials[J]. ACM Transactions on Graphics (TOG), 2015. 35(1):4.
- [100] Miloš Hašan, Martin Fuchs, Wojciech Matusik, Hanspeter Pfister, Szymon Rusinkiewicz. Physical reproduction of materials with specified subsurface scattering[J]. ACM Trans. Graph., 2010. 29(3):61.
- [101] Yanxiang Lan, Yue Dong, Fabio Pellacini, Xin Tong. Bi-scale appearance fabrication[J]. ACM Trans. Graph., 2013. 32(4):145.
- [102] Desai Chen, David IW Levin, Piotr Didyk, Pitchaya Sitthi-Amorn, Wojciech Matusik. Spec2fab: A reducer-tuner model for translating specifications to 3d prints[J]. ACM Transactions on Graphics (TOG), 2013. 32(4):135.
- [103] Kiril Vidimče, Szu-Po Wang, Jonathan Ragan-Kelley, Wojciech Matusik. Openfab: A programmable pipeline for multi-material fabrication[J]. ACM Transactions on Graphics (TOG), 2013. 32(4):136.
- [104] Pitchaya Sitthi-Amorn, Javier E Ramos, Yuwang Wangy, Joyce Kwan, Justin Lan, Wenshou Wang, Wojciech Matusik. Multifab: a machine vision assisted platform for multi-material 3d printing[J]. ACM Transactions on Graphics (TOG), 2015. 34(4):129.

- [105] Yue Dong, Jiaping Wang, Fabio Pellacini, Xin Tong, Baining Guo. Fabricating spatially-varying subsurface scattering[J]. ACM Trans. Graph., 2010. 29(3):62.
- [106] Ondrej Stava, Juraj Vanek, Bedrich Benes, Nathan Carr, Radomír Měch. Stress relief: Improving structural strength of 3d printable objects[J]. ACM Trans. Graph., 2012. 31(4):48. ISSN 0730-0301.
- [107] Xiang Chen, Changxi Zheng, Weiwei Xu, Kun Zhou. An asymptotic numerical method for inverse elastic shape design[J]. ACM Trans. Graph., 2014. 33(4):95.
- [108] Christian Schumacher, Bernd Bickel, Jan Rys, Steve Marschner, Chiara Daraio, Markus Gross. Microstructures to control elasticity in 3d printing[J]. ACM Trans. on Graph., 2015. 34(4):136.
- [109] Qingnan Zhou, Julian Panetta, Denis Zorin. Worst-case structural analysis[J]. ACM Transactions on Graphics (TOG), 2013. 32(4):137.
- [110] Lin Lu, Andrei Sharf, Haisen Zhao, Yuan Wei, Qingnan Fan, Xuelin Chen, Yann Savoye, Changhe Tu, Daniel Cohen-Or, Baoquan Chen. Build-to-last: Strength to weight 3d printed objects[J]. ACM Transactions on Graphics (TOG), 2014. 33(4):97.
- [111] Mélina Skouras, Bernhard Thomaszewski, Stelian Coros, Bernd Bickel, Markus Gross. Computational design of actuated deformable characters[J]. ACM Trans. Graph., 2013. 32(4):82.
- [112] Jesús Pérez, Bernhard Thomaszewski, Stelian Coros, Bernd Bickel, José A Canabal, Robert Sumner, Miguel A Otaduy. Design and fabrication of flexible rod meshes[J]. ACM Transactions on Graphics (TOG), 2015. 34(4):138.
- [113] Bernd Bickel, Moritz Bächer, Miguel A Otaduy, Hyunho Richard Lee, Hanspeter Pfister, Markus Gross, Wojciech Matusik. Design and fabrication of materials with desired deformation behavior[J]. ACM Trans. Graph., 2010. 29(4):63.
- [114] Julian Panetta, Qingnan Zhou, Luigi Malomo, Nico Pietroni, Paolo Cignoni, Denis Zorin. Elastic textures for additive fabrication[J]. ACM Trans. Graph., 2015. 34(4):135.
- [115] Linjie Luo, Ilya Baran, Szymon Rusinkiewicz, Wojciech Matusik. Chopper: Partitioning models into 3d-printable parts[J]. ACM Trans. Graph., November 2012. 31(6):129:1–129:9.
- [116] Miaojun Yao, Zhili Chen, Linjie Luo, Rui Wang, Huamin Wang. Level-set-based partitioning and packing optimization of a printable model[J]. ACM Transactions on Graphics (TOG), 2015. 34(6):214.
- [117] Weiming Wang, Tuanfeng Y Wang, Zhouwang Yang, Ligang Liu, Xin Tong, Weihua Tong, Jiansong Deng, Falai Chen, Xiuping Liu. Cost-effective printing of 3d objects with skin-frame structures[J]. ACM Transactions on Graphics (TOG), 2013. 32(6):177.
- [118] Jérémie Dumas, An Lu, Sylvain Lefebvre, Jun Wu, TU München, Christian Dick, TU München. By-example synthesis of structurally sound patterns[J]. ACM Transactions on Graphics (TOG), 2015. 34(4):137.
- [119] Stelian Coros, Bernhard Thomaszewski, Gioacchino Noris, Shinjiro Sueda, Moira Forberg, Robert W

- Sumner, Wojciech Matusik, Bernd Bickel. Computational design of mechanical characters[J]. ACM Trans. Graph., 2013. 32(4):83.
- [120] Moritz Bächer, Emily Whiting, Bernd Bickel, Olga Sorkine-Hornung. Spin-it: Optimizing moment of inertia for spinnable objects[J]. ACM Trans. Graph., July 2014. 33(4):96:1–96:10.
- [121] Romain Prévost, Emily Whiting, Sylvain Lefebvre, Olga Sorkine-Hornung. Make it stand: balancing shapes for 3d fabrication[J]. ACM Trans. Graph., 2013. 32(4):81.
- [122] Przemyslaw Musalski, Thomas Auzinger, Michael Birsak, Michael Wimmer, Leif Kobbelt. Reduced-order shape optimization using offset surfaces[J]. ACM Trans. Graph, 2015. 34(4):102.
- [123] Lifeng Zhu, Weiwei Xu, John Snyder, Yang Liu, Guoping Wang, Baining Guo. Motion-guided mechanical toy modeling.[J]. ACM Trans. Graph., 2012. 31(6):127.
- [124] Duygu Ceylan, Wilmot Li, Niloy J. Mitra, Maneesh Agrawala, Mark Pauly. Designing and fabricating mechanical automata from mocap sequences[J]. ACM Trans. Graph., 2013. 32(6):186.
- [125] Vittorio Megaro, Bernhard Thomaszewski, Maurizio Nitti, Otmar Hilliges, Markus Gross, Stelian Coros. Interactive design of 3d-printable robotic creatures[J]. ACM Transactions on Graphics (TOG), 2015. 34(6):216.
- [126] Moritz Bächer, Stelian Coros, Bernhard Thomaszewski. Linkedit: interactive linkage editing using symbolic kinematics[J]. ACM Transactions on Graphics (TOG), 2015. 34(4):99.
- [127] Huamin Wang, Gavin Miller, Greg Turk. Solving general shallow wave equations on surfaces[C]//Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation (SCA 2007). ISBN 978-1-59593-624-0, 2007:229–238.
- [128] Daniele Panozzo, Olga Diamanti, Sylvain Paris, Marco Tarini, Evgeni Sorkine, Olga Sorkine-Hornung. Texture mapping real-world objects with hydrographics[J]. Comput. Graph. Forum, 2015. 34(5):65–75.
- [129] S Junk, J Sämann-Sun, M Niederhofer. Application of 3d printing for the rapid tooling of thermoforming moulds[C]//Proceedings of the 36th International MATADOR Conference. Springer, 2010:369–372.
- [130] Jerry Dees. FEA Simulations of Thermoforming—Using Hyperelastic Material Properties[M]. Engineering simulations LLC, scientific article, 2009.
- [131] HF Nied, CA Taylor, HG Delorenzi. Three-dimensional finite element simulation of thermoforming[J]. Polymer Engineering & Science, 1990. 30(20):1314–1322.
- [132] Accuform, 2015. [Http://www.t-sim.com/](http://www.t-sim.com/).
- [133] Christian Schüller, Daniele Panozzo, Anselm Grundhöfer, Henning Zimmer, Evgeni Sorkine, Olga Sorkine-Hornung. Computational thermoforming[J]. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH), 2016. 35(4).
- [134] JP Mellor, M Oder, J Starr, J Meador. An image-based three-dimensional digitizer for pre-decorating thermoformed parts.[C]//BMVC. Citeseer, 2003:1–10.

- [135] Mark Pollack, Anthony Jollett. Reproducible three dimensional vacuum forming technique[M]. US Patent 8406508B2, 2013.
- [136] Matthias Müller. Fast and robust tracking of fluid surfaces[C]//Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA 2009). ISBN 978-1-60558-610-6, 2009:237–245.
- [137] Tyson Brochu. Fluid Animation with Explicit Surface Meshes and Boundary-Only Dynamics[D]. :University of British Columbia, 2006.
- [138] Mark Sussman, Mitsuhiro Ohta. A stable and efficient method for treating surface tension in incompressible two-phase flow[J]. Journal of Scientific Computing, June 2009. 31(4):2447–2471.
- [139] Mathieu Desbrun, Mark Meyer, Peter Schröder, Alan H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow[C]//Proceedings of the 26th annual conference on Computer graphics and interactive techniques (SIGGRAPH '99). ISBN 0-201-48560-5, 1999:317–324.
- [140] I. Eckstein, J.-P Pons, Y. Tong, C.-C. J. Kuo, M. Desbrun. Generalized surface flows for mesh processing[C]//Proceedings of the fifth Eurographics symposium on Geometry processing (SGP 2007). 2007:183–192.
- [141] Desbrun Mathieu, Meyer Mark, Schröder Peter, Alan H. Barr. Discrete differential-geometry operators in nD[C]. Springer-Verlag, 2000:35–57.
- [142] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, Werner Stuetzle. Mesh optimization[C]//Proceedings of the 20th annual conference on Computer graphics and interactive techniques (SIGGRAPH '93). ACM, 1993:19–26.
- [143] Tyson Brochu, Robert Bridson. Robust topological operations for dynamic explicit surfaces[J]. SIAM J. Sci. Comput., 2009. 31:2472–2493.
- [144] Tim Weyrich, Pieter Peers, Wojciech Matusik, Szymon Rusinkiewicz. Fabricating microgeometry for custom surface reflectance[J]. ACM Trans. Graph., 2009. 28(3):32.
- [145] Mordechay Schlesinger, Milan Paunovic. Modern electroplating[M], volume 55. John Wiley & Sons, 2011.
- [146] Robin Hopper. Making Marks: Discovering the Ceramic Surface[M]. Krause Publications Craft, 2004.
- [147] Linda Darty. The art of enameling: techniques, projects, inspiration[M]. Lark Books, 2004.
- [148] Wikipedia. Hydrographics (printing), 2014.
[http://en.wikipedia.org/wiki/Hydrographics_\(printing\)](http://en.wikipedia.org/wiki/Hydrographics_(printing)).
- [149] Joseph Katz, Allen Plotkin. Low-speed aerodynamics[M], volume 13. Cambridge University Press, 2001.
- [150] Frank M White, Isla Corfield. Viscous fluid flow[M], volume 3. McGraw-Hill New York, 1991.
- [151] Teodor Sandu, Andrei Sarbu, Celina Maria Damian, Adela Marin, Silviu Vulpe, Temenuzhka Budi-

- nova, Boyko Tsyntsarski, M. Ferhat Yardim, Ahmet Sirkecioglu. Preparation and Characterization of Membranes Obtained from Blends of Acrylonitrile Copolymers with Poly(vinyl alcohol)[J]. Journal of Applied Polymer Science, November 2014. 131(21).
- [152] Yotam Gingold, Adrian Secord, Jefferson Y Han, Eitan Grinspun, Denis Zorin. A discrete model for inelastic deformation of thin shells[C]//Proceedings of SCA'04. 2004.
- [153] Jonathan Richard Shewchuk. Triangle: Engineering a 2d quality mesh generator and delaunay triangulator. //Applied computational geometry towards geometric engineering. Springer, 1996. 203–222.
- [154] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera[C]//Proceedings of UIST'11. ACM, 2011:559–568.
- [155] Paul J Besl, Neil D McKay. Method for registration of 3-d shapes[C]//Robotics-DL tentative. International Society for Optics and Photonics, 1992:586–606.
- [156] Pedro V. Sander, John Snyder, Steven J. Gortler, Hugues Hoppe. Texture mapping progressive meshes[C]//Proceedings of SIGGRAPH '01. New York, NY, USA:ACM, 2001:409–416.
- [157] Chunghee Park, Hoon Huh, Jinsung Kim, Changnam Ahn. Determination of true stress–true strain curves of polymers at various strain rates using force equilibrium grid method[J]. Journal of Composite Materials, 2012. 46(17):2065–2077.
- [158] R JT Park, MJN Priestley, WR Walpole. The seismic performance of steel encased reinforced concrete bridge piles, 1982.
- [159] Marc A Meyers, Krishan Kumar Chawla. Mechanical behavior of materials[M], volume 2. Cambridge university press Cambridge, 2009.
- [160] MK Warby, JR Whiteman, W-G Jiang, P Warwick, T Wright. Finite element simulation of thermoforming processes for polymer sheets[J]. Mathematics and computers in simulation, 2003. 61(3):209–218.
- [161] Yonghao Yue, Breannan Smith, Christopher Batty, Changxi Zheng, Eitan Grinspun. Continuum foam: A material point method for shear-dependent flows[J]. ACM Trans. Graph., November 2015. 34(5):160:1–160:20. ISSN 0730-0301.
- [162] Peter Klein. Fundamentals of plastics thermoforming[J]. Synthesis Lectures on Materials Engineering, 2009. 1(1):1–97.
- [163] Arnulph Fuhrmann, Gerrit Sobotka, Clemens Groß. Distance fields for rapid collision detection in physically based modeling[C]//Proceedings of GraphiCon 2003. 2003:58–65.
- [164] S Craig. Direct digital uv imaging for deep draw thermoforming[J]. Thermoforming Quarterly, 2013. 32:24–26.

- [165] Menglei Chai, Linjie Luo, Kalyan Sunkavalli, Nathan Carr, Sunil Hadap, Kun Zhou. High-quality hair modeling from a single portrait photo[J]. ACM Trans. Graph., October 2015. 34(6):204:1–204:10.

攻读博士学位期间主要研究成果

1. **Yizhong Zhang**, Huamin Wang, Shuai Wang, Yiyi Tong, Kun Zhou.
A Deformable Surface Model for Real-Time Water Drop Animation.
IEEE Transactions on Visualization and Computer Graphics, 2012, 18(8), 1281-1289
2. **Yizhong Zhang**, Chunji Yin, Changxi Zheng, Kun Zhou.
Computational Hydrographic Printing.
ACM Transactions on Graphics(SIGGRAPH), 2015, 34(4), 131:1-131:11
3. **Yizhong Zhang**, Weiwei Xu, Yiyi Tong, Kun Zhou.
Online Structure Analysis for Real-time Indoor Scene Reconstruction.
ACM Transactions on Graphics, 2015, 34(5), 159:1-159:13
4. **Yizhong Zhang**, Yiyi Tong, Kun Zhou.
Coloring 3D Printed Surfaces by Thermoforming.
Submitted to IEEE Transactions on Visualization and Computer Graphics. Minor Revision.

致 谢

在即将完成自己六年的博士阶段学习之际，回想起刚刚进入实验室的情景，仿佛还在昨天。六年间，我有过论文被录用的喜悦，也经历过论文多次被拒的挫折。能够磕磕绊绊地读完博士，离不开大家的帮助，有好多人需要感谢。

首先，感谢我的导师周昆教授，在这六年中对我的用心指导。我第一次见到周昆老师是大四保研的时候。那时周昆老师已经成为中国计算机图形学年轻一代的领军人物，作为长江学者全职回到浙大。于是我从本科所在的机械工程学系慕名而来，加入周昆老师的实验室。很难想象一个实验室的“大老板”还能够每天亲自来实验室指导学生，有时甚至还会亲自帮学生写代码。在朝夕相处的六年中，周昆老师不仅教给我如何做科研，也教给我如何与他人相处与合作，在方方面面对我都产生了深远的影响。

其次，感谢我博士期间科研项目的合作者，包括童一颖老师、郑昌熙老师、王华民老师、许威威老师、王帅博士，以及尹淳骥同学。每一个课题的完成，每一篇论文的发表，都离不开大家共同的努力。特别要感谢童一颖老师，我有三篇论文都是与他合作完成的。童一颖老师学问渊博、待人谦和，跟他讨论问题的时候会有很大的收获。

再次，感谢曾经与我并肩奋斗的同学们。在赶SIGGRAPH deadline的日子里，我们一起熬夜写代码，后半夜一起跑去15公里外的24h永和大王吃夜宵，有时甚至还睡在实验室、过着黑白颠倒的日子。同学们对我的科研项目也提供了许多技术支持。曹晨同学在水滴仿真项目中帮助我实现了水滴的实时渲染，林敏敏同学、朱萌同学、王超同学在室内场景扫描项目中帮我写了部分代码，龚旭超同学、贺明明同学在多个项目中帮我实现了高质量离线渲染，另外和许多同学都有过建设性的讨论，在此一并表示感谢。

最后，感谢我父母的养育之恩，一路陪伴着我成长。还有我心爱的J.H.，我所做的一切，都是为了和你在一起。