

上海交通大学

SHANGHAI JIAO TONG UNIVERSITY

学士学位论文

BACHELOR'S THESIS



论文题目： 快速毛发仿真的方法研究

学生姓名： 郑诗阳

学生学号： 516021910753

专 业： 计算机科学与技术

指导教师： 刘天添

学院(系)： 电子信息与电气工程学院

快速毛发仿真的方法研究

摘要

毛发在随着身体运动时展现出复杂的运动状态和丰富的运动细节。快速且准确地仿真细节的毛发运动对于实时应用中呈现真实的人物动画来说至关重要。然而由于毛发数量大，内部碰撞密集，毛发仿真依然是个具有挑战性的课题。之前的方法在仿真大规模的毛发运动时由于碰撞求解过程很慢而耗时巨大。

为了加快毛发仿真中的碰撞求解，我们观察到，碰撞发生在局部的位置，而将碰撞冲量传播到整根毛发的是弹性力。基于这样的观察，本次毕设提出了一种基于牛顿法的迭代算法，将摩擦力解算集成到运动方程求解的迭代中。我们的方法能够快速求解摩擦力，并且结果遵循库仑摩擦力模型。在离散弹性杆模型下，我们的算法分为两个步骤，碰撞冲量更新将每个碰撞看作一个局部的问题求解，即将每对参与碰撞的杆看作刚体，解出与当前速度对应的摩擦力；全局速度更新利用一步牛顿迭代将局部的冲量信息传播到整根头发，交替进行全局更新和局部更新，当算法收敛时就达到一个无碰撞的状态。

最后，我们通过实验验证了算法的正确性，算法收敛后能够保证毛发之间没有穿透，最终速度和碰撞冲量之间满足库仑摩擦模型。与现有效果最好的毛发仿真算法对比后说明，我们的方法在不损失准确性的基础上大幅提升了仿真速度。

关键词：物理仿真，毛发仿真，摩擦力解算

FAST HAIR SIMULATION

ABSTRACT

Human hair exhibits complicated dynamic behaviors and rich motion details while moving with the head and interacting with other objects such as the human body. Faithfully simulating the detailed hair dynamics in real-time is important for realistic character animation in games and virtual reality. However, it is still a challenging task due to the large number of hair strands and rich self-interactions. Previous methods usually spend too much time resolving collisions when simulating full head of hair.

To accelerate collision response, we observe that the collision occurs in a local position, and it is the elastic force that propagates the collision impulse to the entire hair. Based on this observation, we propose an iterative algorithm based on Newton's method, which integrates friction calculation into every Newton iteration. Our method could solve friction force quickly and the result follows the Coulomb friction law without losing angular momentum. Under the discrete elastic rod model, our algorithm is divided into two steps. The local collision impulse update treats every rod involved in the collision as a rigid body and solves the collision impulse according to the current velocity. The global velocity update uses one Newton iteration to propagate the local impulse information to entire hair. Global updates and local updates are performed alternately until all the hair strands reach a collision-free state.

In the end, we use some well-designed experiments to verify the correctness of our algorithm. After the algorithm converges, it can ensure that there is no penetration between the hairs, and the final velocity and collision impulse follow the Coulomb friction law. Comparison with state-of-the-art hair simulation algorithms ADONIS shows that our method greatly improves the simulation speed without losing accuracy.

Key words: Physical Simulation, Hair Simulation, Collision Response

目 录

第一章	绪论.....	1
1.1	相关工作.....	2
1.1.1	时间积分.....	2
1.1.2	数值算法.....	2
1.1.3	毛发仿真.....	2
1.1.4	碰撞解算.....	3
1.2	本文组织结构.....	3
1.3	本章小结.....	3
第二章	毛发仿真基本理论及研究问题定义.....	4
2.1	离散弹性杆模型.....	4
2.2	运动方程.....	5
2.2.1	牛顿法求解优化问题.....	6
2.3	库伦摩擦模型.....	7
2.3.1	考虑碰撞后的运动方程.....	7
2.4	本章小结.....	8
第三章	基于牛顿法的快速碰撞求解器.....	9
3.1	算法总览.....	9
3.2	局部碰撞冲量求解.....	10
3.2.1	与外部物体之间的碰撞.....	10
3.2.2	毛发之间的碰撞.....	10
3.2.3	质量修正.....	11
3.3	全局更新.....	12
3.4	本章小结.....	12
第四章	实验验证与对比.....	13
4.1	实现细节.....	13
4.2	算法正确性验证.....	14
4.2.1	碰撞.....	14
4.2.2	摩擦.....	15
4.2.3	收敛特性.....	16
4.3	与现有方法对比.....	17
4.4	本章小结.....	19
第五章	结论.....	20
	参考文献.....	21
	谢辞.....	24

第一章 绪论

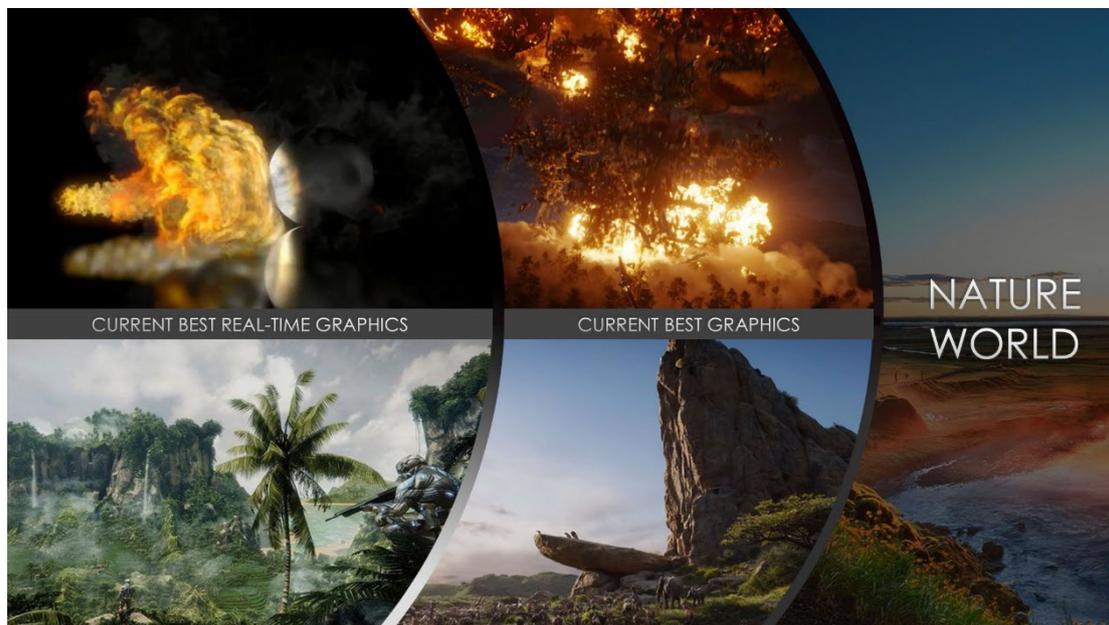


图 1-1 图形学现状

在图 1-1 中展现了图形学的现状，左边是现有最好的实时图形学带来的感官体验，中间是现有最好的影视级图形学所带来的感官体验，最右边是大自然本身。而图形学的两大目标存在于图的交界处，一是使影视级图形学和仿真技术尽可能更好的逼近大自然，二是使实时图形学技术尽可能逼近影视级质量。

物理仿真作为计算机图形学中的一个方向，目标是在计算机中再现真实世界各种物体的运动规律，常应用在影视游戏、医学仿真、VR 等领域。更详细的讲，物理仿真按照仿真物体可以分为刚体、流体、烟雾、可形变物体（布料，软体）等，如图 1-2 所示，按照仿真的效果还可以分为碰撞、撕裂、相变（熔化，结冰等）、多种类型的交互等，如图 1-3 所示。

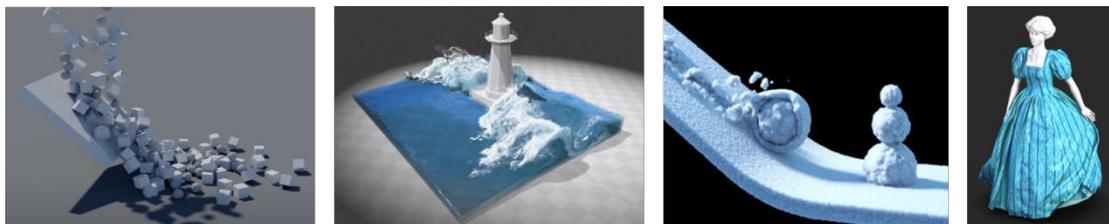


图 1-2 不同物质的物理仿真

其中，毛发仿真作为可形变物体的仿真的一种，拥有结构精细，运动复杂等特点。而且毛发涉及大量的内部碰撞，可以说碰撞实际上影响了的毛发整体形态，并且由于毛发的特殊形状（三维空间中的一维曲线），对于碰撞求解要求严格，因此仿真难度较大。现有的实时毛发仿真仅能做到看上去逼真的整体运动，而忽视了毛发之间的碰撞；能够仿真所有毛发及其碰撞交互的算法又耗时巨大，几秒的动画就需要仿真数小时甚至数天。因从，本次毕设的目标就是从算法层面加速毛发仿真，尤其针对之前最为耗时的摩擦力解算，大幅提升仿真效率。

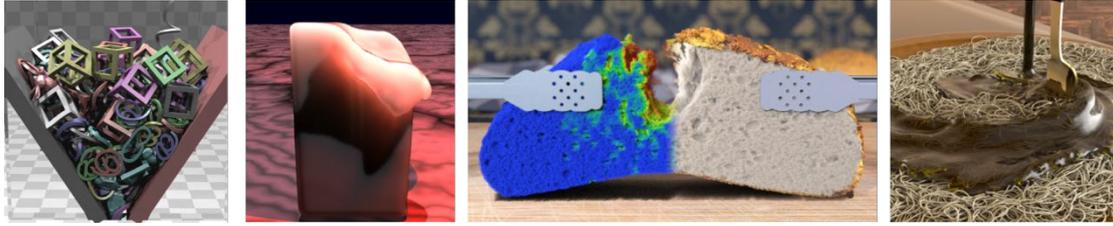


图 1-3 不同物理现象的仿真

1.1 相关工作

本节从四个方面介绍本次毕设涉及的相关工作。

1.1.1 时间积分

William H. Press 等人^[1]的书中提供了很多时间积分模型的综述，对于常见的 Explicit Integration Method 有非常多的描述。而大步长的时间积分通常依赖于稳定的 Implicit Integration Method, Terzopoulos 等人^[2]在其 1987 年的工作上提出了这方面的先驱工作。在计算机图形学上, Baraff 和 Witkin^[3]在弹簧质点上的隐式积分模型是目前最为使用的模型之一。如果对于时间积分器的能量和动量守恒要求非常高, Kharevych 等人^[4]和 Stern 等人^[5]还提出了基于辛几何的 Symplectic Integration Methods.

1.1.2 数值算法

Gast 等人^[6]和 Martin 等人^[7]把传统的隐式积分模型形式化成了一个最优化问题,从而可以更稳定的使用牛顿法求解。牛顿法本是针对凸优化问题设计的,对于非凸的问题, Smith 等人^[8]和 Teran 等人^[9]给出了两个修正海森矩阵至正定的方法。

但是牛顿法由于频繁的线性系统求解的需求,是一种比较笨重的数值方法,对于更高效的仿真系统而言是很大的负担,所以有一些更快的数值方法被发明出来。Position Based Dynamics (PBD)是非常流行的一种快速仿真方式: Müller 等人^[10]提出了对弹簧质点系统的一种快速仿真方法并被广为使用。Stam^[11]提出的 Nucleus Solver 也可以被看做 PBD 的一种变体。Macklin 等人^[12]提出的 XPBD 是 PBD 的一个直接的拓展,对于材料特性的支持上有了长足的补充。关于 PBD 的更多信息可以参考 Jan Bender 等人^[13, 14]的综述。

与 PBD 不同, Liu 等人^[15]和 Bouaziz 等人^[16]提出的 Projective Dynamics(PD)直接在数值积分的最优化问题出发,给出了一个非常快速的可以收敛到理论正确解的加速方法。之后,人们进一步发现 Projective Dynamics 可以被看做一种特殊的拟牛顿方法^[17]或者 ADMM 解法^[18],并对其做了进一步加速。

1.1.3 毛发仿真

毛发作为一种特殊且常见的弹性系统,被许多工作单独研究。早期对毛发仿真的研究可以参考 Hadap 等人^[19]和 Ward 等人^[20]的综述。

Petrovic 等人^[21]和 Selle 等人^[22]使用了弹簧质点系统来仿真头发,但是由于头发特殊的弯曲和扭转抗性,他们需要设立非常多的 ghost 质点和 ghost 弹簧,极大的增加了系统的自由度。但是因为其实现简单的缘故,还是被广为使用在各种常见的头发仿真软件中。另一方面, Grégoire 等人^[23]、Spillmann 等人^[24]和 Pai 等人^[25]分别使用了 Cosserat Model 对头发进行了建模; Bertails 等人^[25]建立了一种特殊的基于 Cosserat Rod 模型下的头发仿真,其名为 Super Helix。之后, Bergou 等人^[26, 27]则单独对头发的拉伸、弯曲和扭转进行建模,创造了名

散弹性杆 (Discrete Elastic Rod, DER) 的头发仿真模型。这种模型数值稳定性很高, 在其基础上做的头发仿真也很多, 我们的工作也是基于 DER 为基础的弹性模型建立的。

之后也有很多试图加速 Cosserat Rod 模型的方法。如 Deul 等人^[28]和 Kugelstadt 等人^[30]提出了 Cosserat Rod 的 PBD 版本, Soler 等人^[32]将 Cosserat Rod 与 PD 结合。Han 等人^[29]提出了一种构造了 tridiagonal matrix 的方法加速毛发仿真, Müller 等人^[31]使用一种动态的位置投影快速保证毛发的不可伸长性。

1.1.4 碰撞解算

碰撞在头发仿真是特别重要的一环, 因为碰撞几乎贯穿了所有的头发运动。

第一种常用的碰撞表示方法是 Cottle 等人^[33]提出的线性互补问题 (Linear Complementarity Problem, LCP)。最开始的时候, LCP 因被 Baraff 等人^[34]用在 rigid body dynamics 上而在计算机图形学领域传播开来。之后 Otaduy 等人^[35]把 LCP 的碰撞和柔性体仿真结合起来。另一种流行的方法是使用 Kaufman 等人^[36]提出的 Staggered Projection, 即使用一步无碰撞的弹性体解算器给出一个初解, 并慢慢将这个含碰撞的初解投影到没有碰撞的状态上。之后 Daviet 等人^[37]把 Staggered Projection 作为一种 Cone Programming 来解。Li 等人^[38]使用了一种快速的投影梯度下降方式去把一个有碰撞的状态投影到合理的没有碰撞的空间中。ADONIS^[39]在前人的投影方法基础上, 额外考虑了弹性材料的非线性因素, 提出了一个比较完备的带碰撞的头发仿真的框架。值得一提的是, 这个框架并没有限制投影的方式, 可以使用文章自身提出的 Gauss-Seidel 投影法, 也可以使用 Li 等人^[38]的 Projected Gradient 来进行更高速的投影。

这些基于投影的碰撞解决方式通常有一个共同的问题, 即大部分的解算时间都花在针对碰撞的后处理上了。我们观察到, 碰撞通常发生于非常局部的部位, 而碰撞产生的冲量在柔性物体之中的传递则是通过弹性材料本身进行。我们希望将碰撞解算与弹性解算放入同一个框架之中, 继而获得更高效准确的头发仿真碰撞解决方案。

1.2 本文组织结构

第二章介绍了毛发仿真的背景知识, 包括离散弹性杆模型, 如何将运动方程在时间上离散化并等价于优化问题, 如何用牛顿法解优化问题, 然后介绍了库仑摩擦模型的形式化表示, 并对之前提出的运动方程表示形式进行修正, 引出我们要解决的问题定义。

第三章是本文的核心章节, 详细介绍了本文提出的算法。首先给出算法提出的基本想法和算法总流程图, 然后对于算法中两个关键步骤进行详细描述, 包括如何局部的求解碰撞从良和如何全局更新速度, 以及为解决算法中的出现的问题提出的质量修正方法。

第四章为实验章节。首先通过实验验证了本文提出算法的正确性, 包括支持力正确性, 摩擦力正确性、收敛情况以及最终的结果是否符合库仑摩擦模型。接着与现有效果最好的毛发仿真算法 ADONIS 进行对比, 证明了本文算法在速度上的优越性。

1.3 本章小结

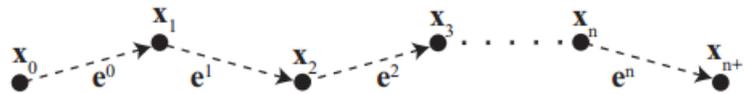
本章简要的介绍了物理仿真的背景, 引出本文的研究目标——快速毛发仿真, 并列出了相关工作。

第二章 毛发仿真基本理论及研究问题定义

毛发仿真属于可形变物体仿真的一种,因此在无约束的动力学方程的求解上与通用的仿真方法相同,不过毛发拥有其特有的模型表示方法,并且在运动过程中包含大量碰撞和交互。针对这些特点,本章将介绍一些毛发仿真所需的基本理论,2.1节介绍毛发仿真的常用模型——离散弹性杆模型,2.2节介绍在不考虑碰撞的情况下如何求解动力学方程,2.3介绍库仑摩擦力模型,以及考虑碰撞后如何修正运动方程,最后将每一帧的求解形式化表示为一个优化问题,2.4节对本章进行总结。

2.1 离散弹性杆模型

离散弹性杆模型将一根毛发离散化为 $n+2$ 个点和 $n+1$ 条可拉伸的杆,如下图所示。



其中,点由 x_0, \dots, x_{n+1} 表示,杆由 e^0, \dots, e^n 表示, $e^i = x_{i+1} - x_i$ (下标与点相关,上标与杆相关)。这样每根头发就有了 $3n + 6$ 自由度,但是只有点不能表示完整的头发位姿,想象在点的位置不变的情况下将相邻两边绕其轴线旋转。因此,离散弹性杆模型在每根杆上建立两个局部坐标系,如图 2-1 所示,其中 t^i 是沿杆方向的单位向量, $\{t^i, m_1^i, m_2^i\}$ 是自始至终固定在杆上的固定坐标系 (material frame), $\{t^i, u^i, v^i\}$ 是在杆之间无扭动传播参考坐标系 (Bishop frame)。固定坐标系和参考坐标系之间的夹角 θ^i 定义了杆沿轴线方向的转动。因此,每根头发可以表示为由所有点的坐标和杆的转动角度组成列向量 $X = \{x_0, \theta^0, x_1, \dots, \theta^n, x_{n+1}\}$ 。

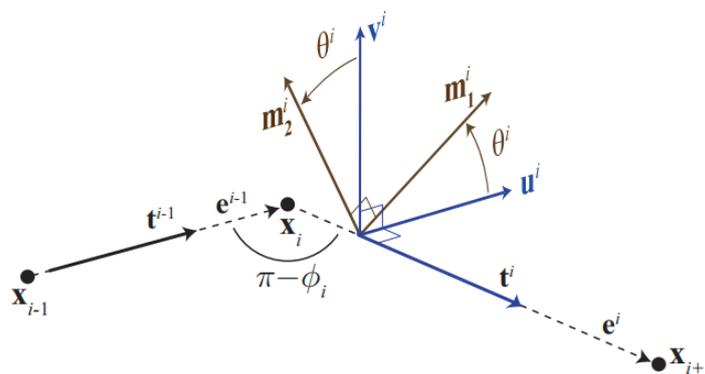


图 2-4 离散弹性杆定义^[26]

2.1.1 弹性势能定义

基尔霍夫杆理论定义了三种弹性势能,分别在拉伸方向、弯曲方向和扭转方向衡量杆偏离平衡位置所具有的能量,每种势能由对应物理量的线积分得出。而其离散化形式认为物理量的值在一个小的积分邻域中保持不变。对于在杆上定义的物理量,邻域大小等于杆的自由

长度 $\|\mathbf{e}^i\|$ ；而对于在点上定义的物理量，积分邻域为其 Voronoi region，即 $\bar{l}_i = \frac{1}{2}(\|\mathbf{e}^{i-1}\| + \|\mathbf{e}^i\|)$ 。由此可以定义三种弹性势能的离散形式：

拉伸势能 定义在每根杆上：

$$E_{\text{stretch}} = \sum_{i=0}^n \frac{1}{2} k_s \left(1 - \frac{\|\mathbf{e}^i\|}{\|\bar{\mathbf{e}}^i\|} \right)^2 \|\bar{\mathbf{e}}^i\| \quad (2.1)$$

弯曲势能 定义在点上，大小与杆之间的夹角有关，因此首先定义离散曲率：

$$\kappa_i = 2 \tan \frac{\phi_i}{2} \quad (2.2)$$

$$E_{\text{bend}} = \sum_{i=1}^n \frac{1}{2} k_b \left(\frac{\kappa_i - \bar{\kappa}_i}{\bar{l}_i} \right)^2 \bar{l}_i. \quad (2.3)$$

若考虑弯曲方向各向异性，需要定义离散曲率副法线：

$$(\kappa \mathbf{b})_i = \frac{2 \mathbf{e}^{i-1} \times \mathbf{e}^i}{\|\bar{\mathbf{e}}^{i-1}\| \|\bar{\mathbf{e}}^i\| + \mathbf{e}^{i-1} \cdot \mathbf{e}^i} \quad (2.4)$$

$(\kappa \mathbf{b})_i$ 大小等于离散曲率，方向与相邻两边垂直，代表了弯曲方向。将 $(\kappa \mathbf{b})_i$ 分别投影到相邻两边的固定坐标系上可以得到杆在横截面上的弯曲向量：

$$\boldsymbol{\omega}_i^j = ((\kappa \mathbf{b})_i \cdot \mathbf{m}_1^j, (\kappa \mathbf{b})_i \cdot \mathbf{m}_2^j)^T, j \in \{i-1, i\}. \quad (2.5)$$

在已知弯曲刚度 $\bar{\mathbf{B}}^j$ 下，弯曲势能可以修正为：

$$E_{\text{bend}} = \sum_{i=1}^n \frac{1}{2 \bar{l}_i} \sum_{j=i-1}^i (\boldsymbol{\omega}_i^j - \bar{\boldsymbol{\omega}}_i^j)^T \bar{\mathbf{B}}^j (\boldsymbol{\omega}_i^j - \bar{\boldsymbol{\omega}}_i^j). \quad (2.6)$$

扭转势能 同样定义在点上，与相邻两遍转动角度的差值有关：

$$E_{\text{twist}} = \sum_{i=1}^n \frac{1}{2} k_t \left(\frac{\theta^i - \theta^{i-1}}{\bar{l}_i} \right)^2 \bar{l}_i \quad (2.7)$$

有了弹性势能的定义，我们就可以求出在毛发在任意形状下的内能和受到的内力。

2.2 运动方程

物理仿真的目标，是根据物体当前的位置和速度，计算出在已知的外力作用下物体的运动，本质是求解运动方程。假设物体的位置随时间变化的函数是 $\mathbf{x}(t)$ ，根据牛顿第二定律有：

$$\frac{d}{dt} \left(\frac{d\mathbf{x}(t)}{dt} \right) = \mathbf{M}^{-1} \mathbf{f}(\mathbf{x}(t)) \quad (2.8)$$

其中 $\mathbf{f} = \mathbf{f}_{\text{ext}} + \mathbf{f}_{\text{int}}(\mathbf{x}(t))$ 是物体受到的合力（不考虑碰撞）。那么，我们的目标是在给定初始值 $\mathbf{x}(t_0), \mathbf{x}'(t_0)$ 的情况下解出 $\mathbf{x}(t)$ 。通常这个二阶微分方程难以求得解析解，而是将其在时间上离散化为一系列点列 $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \dots$ 顺序求解，其中每个 $\mathbf{x}_k \approx \mathbf{x}(t_0 + kh)$ ， h 为时间步长，如图 2-2 所示。

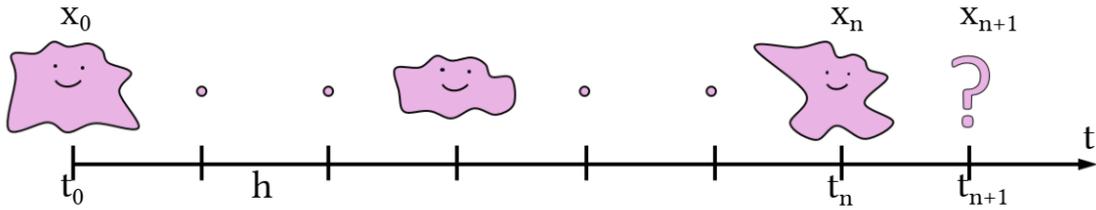


图 2-5 在离散的时间序列上可形变物体的位置随时间变化^[40]

用隐式欧拉法求解这些点列:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{v}_{n+1} \quad (2.9)$$

$$\mathbf{v}_{n+1} = \mathbf{v}_n + h\mathbf{M}^{-1}(\mathbf{f}_{ext} + \mathbf{f}_{int}(\mathbf{x}_{n+1})) \quad (2.10)$$

将式(2.10)带入式(2.9)得到关于 \mathbf{x}_{n+1} 的非线性等式:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{v}_n + h^2\mathbf{M}^{-1}\mathbf{f}_{ext} + h^2\mathbf{M}^{-1}\mathbf{f}_{int}(\mathbf{x}_{n+1}). \quad (2.11)$$

考虑到离散弹性杆的内力是保守力, 则有 $\mathbf{f}_{int}(\mathbf{x}) = -\nabla_{\mathbf{x}}E(\mathbf{x})$. 因此式(2.11)的解等价于如下优化问题的解:

$$\min_{\mathbf{x}} \frac{1}{2h^2} \|\mathbf{x} - \tilde{\mathbf{x}}\|_{\mathbf{M}}^2 + E(\mathbf{x}) \quad (2.12)$$

其中 $\tilde{\mathbf{x}} = \mathbf{x}_n + h\mathbf{v}_n + h^2\mathbf{M}^{-1}\mathbf{f}_{ext}$ 表示物体只受外力作用时下一时刻的位置, 所以式(2.12)的第一项希望物体尽量不违反牛顿第一定律, 第二项希望内能最小化。

这样的优化问题可以用梯度下降法或牛顿法等求解。其中牛顿法作为二阶优化算法, 收敛速度较快, 更为常用, 因此接下来简要的介绍一下牛顿法如何解式(2.12)。

2.2.1 牛顿法求解优化问题

牛顿法是一种用来求可导函数零点的迭代算法, 而在优化问题中, 求导函数的零点对应了目标函数的驻点, 在凸优化问题中就是目标函数的极值点。具体来讲, 设式(2.12)中我们要最小化的目标函数为 $g(\mathbf{x}): \mathbb{R}^n \rightarrow \mathbb{R}$, 牛顿法从一个初始的估计值 \mathbf{x}^0 开始, 寻找一些列使得目标函数值越来越小的点列 $\{\mathbf{x}^k\}$, 最终收敛到极值点 \mathbf{x}^* 。在每一步迭代中, 牛顿法将当前点的二阶泰勒展开作为目标函数的估计,

$$g(\mathbf{x}^k + \mathbf{t}) \approx g(\mathbf{x}^k) + \nabla g(\mathbf{x}^k)^T \mathbf{t} + \frac{1}{2} \mathbf{t}^T \nabla^2 g(\mathbf{x}^k) \mathbf{t}. \quad (2.13)$$

然后求出这个二次函数的极值点作为 \mathbf{x}^{k+1} ,

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \nabla^2 g(\mathbf{x}^k)^{-1} \nabla g(\mathbf{x}^k). \quad (2.14)$$

为保证收敛性, 还可以加入 line search 控制步长, 完整的算法见算法 1

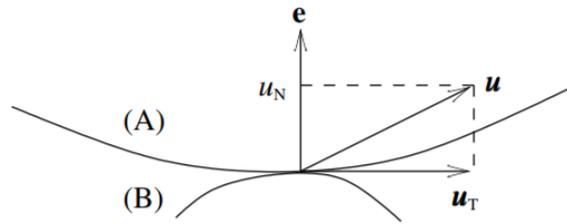
Algorithm 1: Newton Solver with Backtracking Line Search

```

1 Input: initial guess  $x_0$ 
2 for  $k \leftarrow 0$  to  $n$  do
3   compute  $\nabla g(x^k)$ 
4   compute  $\nabla^2 g(x^k)$ 
5    $\Delta x^k \leftarrow -\nabla^2 g(x^k)^{-1} \nabla g(x^k)$ 
6    $\alpha \leftarrow 1/\beta$ 
7   do
8      $\alpha \leftarrow \beta \alpha$ 
9      $x^{k+1} \leftarrow x^k + \alpha \Delta x^k$ 
10    compute  $g(x^{k+1})$ 
11    while  $g(x^{k+1}) \leq g(x^k) + \gamma \alpha (\nabla g(x^k))^T \Delta x^k$ ;
12 end

```

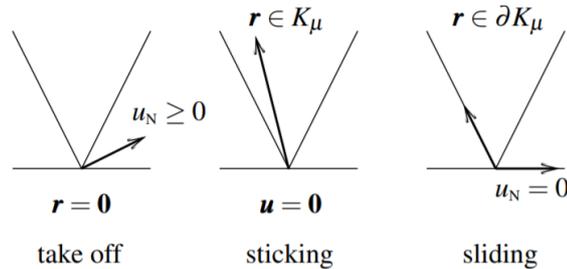
2.3 库伦摩擦模型



如上图所示，当物体(A)和(B)碰撞时，碰撞法向方向为 \mathbf{e} ，他们在碰撞点的相对速度为 \mathbf{u} ，碰撞冲量为 \mathbf{r} ， \mathbf{u} 在法向和切向的分量分别是 u_N 和 \mathbf{u}_T 。定义由摩擦力系数决定的二阶锥

$$K_\mu = \{x = [x_N, \mathbf{x}_T] \in \mathbb{R} \times \mathbb{R}^2, \mu x_N \geq \|\mathbf{x}_T\|\}, \quad (2.14)$$

则 \mathbf{u} 和 \mathbf{r} 应该满足库伦摩擦模型，记作 $(\mathbf{r}, \mathbf{u}) \in \mathcal{L}(\mu)$:



- $u_N \geq 0$, 两物体分离, $\mathbf{r} = \mathbf{0}$
- $\mathbf{u} = \mathbf{0}$, 物体受静摩擦, \mathbf{r} 落在 K_μ 内
- $u_N = 0, \|\mathbf{u}_T\| > 0$, 物体受滑动摩擦力, \mathbf{r} 在 K_μ 的边界上, 且在切平面上的方向与 \mathbf{u}_T 相反

2.3.1 考虑碰撞后的运动方程

有了摩擦模型，我们可以在 2.2 节中的运动方程中加入摩擦力，由于碰撞冲量与物体的速度有关，我们先将关于 \mathbf{x}_{n+1} 的方程修改为关于 \mathbf{v}_{n+1} 的：

$$\mathbf{M}\mathbf{v}_{n+1} = \mathbf{M}\mathbf{v}_n + h(\mathbf{f}_{ext} + \mathbf{f}_{int}(\mathbf{x}_n + h\mathbf{v}_{n+1})). \quad (2.15)$$

假设一个物体有 k 个碰撞点，则修正后的运动方程为：

$$\begin{cases} \mathbf{M}\mathbf{v}_{n+1} = \mathbf{M}\mathbf{v}_n + h(\mathbf{f}_{ext} + \mathbf{f}_{int}(\mathbf{x}_n + h\mathbf{v}_{n+1})) + \mathbf{I} \\ \forall i = 1, \dots, k, (\mathbf{r}_k, \mathbf{u}_k) \in \mathcal{L}(\mu_i) \end{cases} \quad (2.16)$$

其中 \mathbf{I} 表示物体所受的总碰撞冲量，在每个碰撞点，局部的相对速度和碰撞冲量之间都要满足库仑摩擦模型。与2.2节中类似，式(2.16)等价为一个带约束的优化问题：

$$\begin{aligned} \min_{\mathbf{v}, \mathbf{I}} \quad & \frac{1}{2} \|\mathbf{v} - \tilde{\mathbf{v}}\|_{\mathbf{M}}^2 + E(\mathbf{x}_t + \mathbf{v}h) \\ \text{subject to } \quad & \forall i = 1, \dots, k, (\mathbf{r}_k, \mathbf{u}_k) \in \mathcal{L}(\mu_i) \end{aligned} \quad (2.17)$$

其中， $\tilde{\mathbf{v}} = \mathbf{v}_t + \mathbf{M}^{-1}(h\mathbf{f}_{ext} + \mathbf{I})$ 。

2.4 本章小结

本章介绍了毛发仿真的背景知识，包括离散弹性杆模型的表示和能量定义，如何用牛顿法求解运动方程，以及库仑摩擦力模型的形式化表示，并在最后指出，加入了碰撞冲量的运动方程等价于一个带约束的优化问题。有了这些理论基础和形式化的问题表示，接下来的章节将介绍我们的算法如何快速求解碰撞冲量和速度。

第三章 基于牛顿法的快速碰撞求解器

本章将介绍一种基于牛顿法的迭代算法，可以快速求解 2.3.1 节提出的优化问题，在算法收敛时，可以得到满足库伦摩擦模型的碰撞冲量和对应的速度。本章结构为：3.1 节介绍算法的基本思想并给出算法总览，3.2 节和 3.3 节详细介绍如何求解局部碰撞冲量和将碰撞冲量应用到全局速度更新，最后 3.4 节对本章进行总结。

3.1 算法总览

考虑到解碰撞是一个局部的问题——一对碰撞点最对只涉及四个点，因此在有了一个可能发生碰撞的速度后，对于每一对碰撞点，我们都能根据库伦摩擦模型计算出将这个速度修正所需的冲量，解这个冲量只考虑了局部的速度信息。接着，我们把这个冲量带入运动方程，通过全局的速度求解将碰撞冲量的影响传播到整根头发上。这样得到一个新的速度，这个速度也许仍然不能在每个碰撞点上满足摩擦约束，于是我们就再次进行局部冲量计算。这样循环进行冲量计算和速度更新，当算法收敛时，就得到了一对同时满足运动方程和摩擦模型的速度和冲量。

因此，算法可以分为两个较为独立的子问题：(1)已知碰撞冲量求解速度，(2)已知速度求对应的碰撞冲量。(1)的计算可以使用 2.2.1 节提到的牛顿法，(2)的计算参照 2.3 节介绍库伦摩擦模型。由于一步牛顿迭代得到的解与最终解之间的差别很小，并且在算法没有收敛时每步(2)计算出的冲量都不是准确的，所以(1)的计算我们只使用了一步牛顿迭代，整个算法看上去像是在牛顿法中加入了额外的冲量更新步骤。完整算法见算法 2。

Algorithm 2: Collision handling system based on Newton's method

```

1 Input: initial position  $x_0$  and velocity  $v_0$ 
2 // Simulation loop
3 for  $t \leftarrow 0$  to  $n - 1$  do
4    $v^0 \leftarrow v_t, I^0 \leftarrow 0$ 
5    $k \leftarrow 0$ 
6   do
7      $g(v) = \frac{1}{2} \|v - \bar{v}^k\|_M^2 + E(x_t + vh)$ , where  $\bar{v}^k = v_t + M^{-1}(hf_{ext} + I^k)$ 
8      $\Delta v^k \leftarrow -\nabla^2 g(v^k)^{-1} \nabla g(v^k)$ 
9      $v_{k+1} \leftarrow v^k + \alpha \Delta v^k$ 
10    if  $k = 0$  then
11       $C \leftarrow \text{ContinuousCollisionDetection}(v^k, v^{k+1})$ 
12    end
13     $I^{k+1} \leftarrow \text{ComputeCollisionImpulse}(C, v^{k+1}, I^k)$ 
14     $k \leftarrow k + 1$ 
15    while  $k \leq N_{max}$  and  $\|\Delta v\| < tol$ ;
16     $v_{t+1} \leftarrow v^k$ 
17 end

```

算法 2 的第 4 行将初始的碰撞冲量设为 0，因此第一步牛顿迭代与没有考虑碰撞时完全相同，有了这个可能发生碰撞的速度，第 10~12 行做连续碰撞检测，找出发生碰撞的边，

然后第 14~17 行遍历所有的碰撞点计算冲量，这个冲量将在之后的循环中用于更新速度（第 7~9 行），当算法达到最大迭代次数或速度变化量小于某个设定值时跳出迭代（第 19 行）。接下来的两节将详细介绍如何解碰撞冲量和更新速度。

3.2 局部碰撞冲量求解

在毛发仿真中发生的碰撞分为两种：毛发与外部物体的碰撞，如梳子、头部等，和毛发之间的碰撞。第一种碰撞较为简单，我们只考虑点与外部物体的碰撞，算出的冲量也直接作用在点上；而第二种碰撞是一种双向交互，需要考虑毛发本身的模型，并且需要保证碰撞过程中动量守恒。所以我们将两种碰撞分开计算，下面将分别详细介绍。

3.2.1 与外部物体之间的碰撞

假设外部物体均为刚体，且运动速度已知。毛发上的结点和外部物体之间的相对速度 $\mathbf{u} = \mathbf{v}_i - \mathbf{u}_{free}$ ，其中 \mathbf{v}_i 是毛发结点的速度， \mathbf{u}_{free} 是已知的外部物体的速度， \mathbf{u} 在碰撞法向上的投影为 $u_N = \mathbf{u}^T \mathbf{N}$ （法向方向由外部物体指向毛发，与 \mathbf{u} 的定义对应），在切平面上的投影为 $\mathbf{u}_T = \mathbf{u} - u_N \mathbf{N}$ ，碰撞冲量可以分 3 种情况计算，如图 3-1 所示。

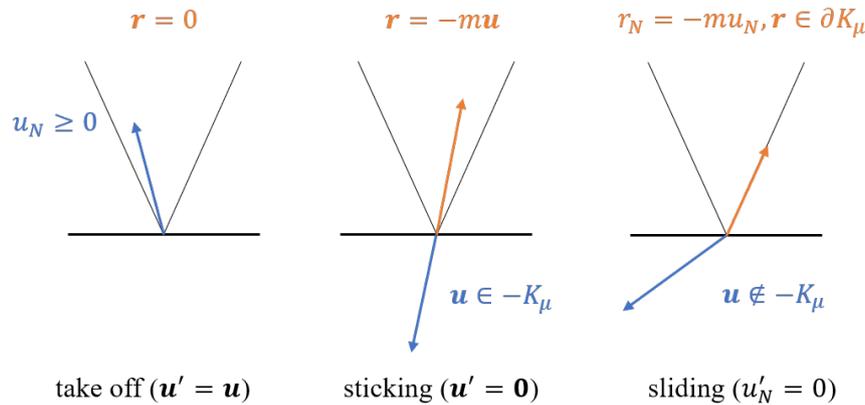


图 6-1 与外部物体碰撞的冲量计算

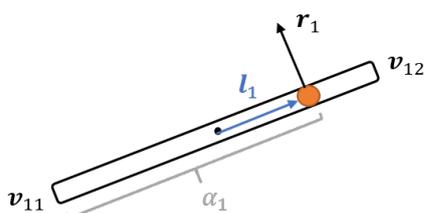
- 当 $u_N \geq 0$ 时，这对碰撞已经不再发生，所以 $\mathbf{r} = \mathbf{0}$
- 当 $u_N < 0$ 时，冲量需要使两物体在法向上相对速度为 0，并且不能落在 K_μ 外

$$\mathbf{r}_N = -mu_N, \mathbf{r}_T = -\min\left(1, \frac{\mu u_N}{\|\mathbf{u}_T\|}\right) m \mathbf{u}_T \quad (3.1)$$

其中 m 是该结点的质量。若 \mathbf{u} 直接受到冲量影响变为 \mathbf{u}' ，则 $(\mathbf{r}, \mathbf{u}') \in \mathcal{L}(\mu)$ 。

3.2.2 毛发之间的碰撞

在离散弹性杆模型下，毛发之间的碰撞发生在两根杆之间。在每个小的碰撞问题中，我们将参与碰撞的杆看作刚体，考虑冲量对杆线速度和角速度的影响，从而求得使碰撞点法向相对速度为 0 的冲量。



设两根杆质量分别为 m_1, m_2 ，受到的冲量分别为 $\mathbf{r}_1, \mathbf{r}_2$ ， $\mathbf{r}_2 = -\mathbf{r}_1$ ，两杆质心到碰撞点的向量分别为 $\mathbf{l}_1, \mathbf{l}_2$ ，两杆端点的速度分别问 $\mathbf{v}_{11}, \mathbf{v}_{12}$ 和 $\mathbf{v}_{21}, \mathbf{v}_{22}$ ，碰撞点到第一个端点的比例分别为 α_1, α_2 ，碰撞点的速度由两端速度线性插值得到。相对速度

$$\mathbf{u} = (1 - \alpha_1)\mathbf{v}_{11} + \alpha_1\mathbf{v}_{12} - ((1 - \alpha_2)\mathbf{v}_{21} + \alpha_2\mathbf{v}_{22}) \quad (3.2)$$

杆 1 的碰撞点在冲量的影响下速度改变量为:

$$\Delta\mathbf{u}_1 = \frac{\mathbf{r}_1}{m_1} + (\mathbf{I}_1^{-1}\mathbf{l}_1 \times \mathbf{r}_1) \times \mathbf{l}_1 \quad (3.3)$$

其中 $\mathbf{I}_1 = \frac{1}{12}m_1\|\mathbf{e}_1\|^2$ 为杆的转动惯量。定义 $\mathbf{K}_1 = \frac{\delta}{m_1} + [\mathbf{l}_1]_x^T\mathbf{I}_1^{-1}[\mathbf{l}_1]_x$, $\Delta\mathbf{u}_1$ 可以表示为 $\mathbf{K}_1\mathbf{r}_1$.

同理 $\Delta\mathbf{u}_2 = \mathbf{K}_2\mathbf{r}_2 = -\mathbf{K}_2\mathbf{r}_1$. 首先假设在冲量作用下碰撞点相对速度为零:

$$\mathbf{u}' = \mathbf{u} + (\mathbf{K}_1 + \mathbf{K}_2)\mathbf{r}_1 = 0 \quad (3.4)$$

解这个三维的线性等式可以得出 \mathbf{r}_1 , 然后与 3.2.1 节类似将 \mathbf{r}_1 投影到 K_μ 中:

$$\mathbf{r}_T = \min\left(1, \frac{\mu r_N}{\|\mathbf{r}_T\|}\right)\mathbf{r}_T \quad (3.5)$$

3.2.3 质量修正

在速度更新将局部碰撞信息传播到整根毛发上后, 下一次冲量计算可以获得上次迭代的全局信息, 但是由于冲量的计算在所有碰撞中并行执行, 他们无法知晓其他碰撞在当前迭代产生的冲量。这个问题在初次迭代中尤为明显, 如果一根杆受到 n 个同向的碰撞, 每个碰撞不知道其他碰撞的存在, 于是都给出生能使相对速度为 0 的冲量, 这些冲量叠加起来就会变成实际所需的冲量的 n 倍, 而我们的算法又无法修正这个过大的冲量, 就会出现毛发被弹开的错误结果, 如图 3-2 (a)所示。

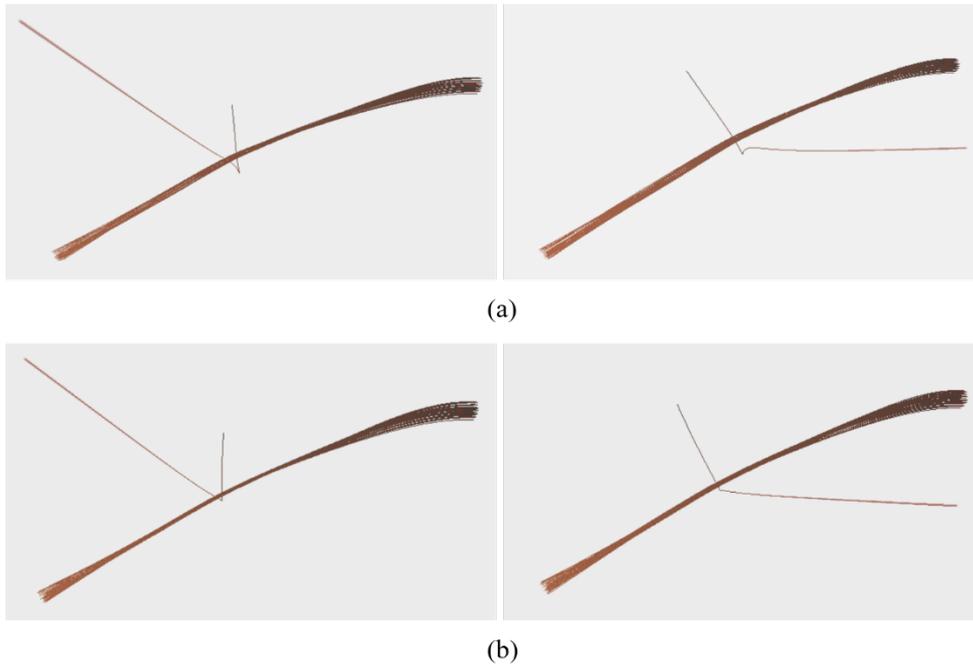


图 3-2 一束一端固定的头发到一根两端固定的头发上。(a)没有质量修正下面的一根头发由于受到过大的冲量被弹开, (b)有了质量修正可以稳定的搭在上面

为了修正冲量过大的问题, 我们对碰撞点的质量进行修正。如果一根杆受到 n 个碰撞, 我们假设碰撞点均匀的分布在杆上, 然后将杆划分为 n 个质量相等小段, 这样每个碰撞点只需将一个质量更小的小段推出去, 计算出的冲量就会变为原来的 $1/n$, 修正结果如图 3-2 (b)所示。在实现中, 我们只对激活的碰撞进行计数 ($\|\mathbf{r}\| > 0$), 完整的碰撞冲量计算流程见算

法 3.

Algorithm 3: Compute collision impulse function

```

1 Input: collision set  $C$ , velocity  $v^{k+1}$ , impulse  $I^k$ 
2 Function ComputeCollisionImpulse( $C, v^{k+1}, I^k$ ):
3   colNum  $\leftarrow \emptyset$ 
4   foreach collision  $i$  in  $C$  do
5     foreach vertex  $j$  in  $c_i$  do
6       colNum( $j$ )  $\leftarrow$  colNum( $j$ ) + 1
7     end
8   end
9    $I^{k+1} \leftarrow I^k$ 
10   $C' \leftarrow \emptyset$ 
11  foreach collision  $i$  in  $C$  do
12    foreach vertex  $j$  in  $c_i$  do
13       $m'_j \leftarrow m_j / \text{colNum}(j)$ 
14    end
15    compute  $r$  with  $v^{k+1}$  and  $m'$ 
16     $I^{k+1} \leftarrow I^{k+1} + r$ 
17    if  $\|r\| > 0$  then
18      inset  $c_i$  into  $C'$ 
19    end
20  end
21   $C \leftarrow C'$ 
22  return  $I^{k+1}$ 

```

3.3 全局更新

有了碰撞产生的冲量,可以用一步牛顿法迭代将局部冲量作用到全局的速度。具体来讲,只需要求出目标函数的一阶导和二阶导:

$$\nabla g(\mathbf{v}) = \mathbf{M}(\mathbf{v} - \mathbf{v}_t) - h(\mathbf{f}_{ext} - \nabla E(\mathbf{x}_t + \mathbf{v}h)) - \mathbf{I} \quad (3.6)$$

$$\nabla^2 g(\mathbf{v}) = \mathbf{M} + h^2 \nabla^2 E(\mathbf{x}_t + \mathbf{v}h) \quad (3.7)$$

由于计算离散弹性杆模型势能的海森矩阵比较耗时,这一步也可以用拟牛顿法来代替:每一次迭代的海森矩阵都与第一次迭代相同,这样只需计算梯度,并且可以事先将海森矩阵分解,方便求逆。我们的实验表明,用拟牛顿法和用牛顿法的视觉效果在绝大部分例子上基本没有差别,但是拟牛顿法速度快很多。

3.4 本章小结

在将原优化问题分解为两个子问题后,每一个子问题都可以快速求解,我们的算法交替进行全局的速度更新和局部的冲量更新,在收敛时得到原优化问题的解。在计算局部冲量时,我们将碰撞分为两类,毛发与外部物体的碰撞只考虑发生在节点上,毛发之间的碰撞将两根参与碰撞的杆看作刚体,在利用碰撞计数对质量进行修正后求出满足库仑摩擦模型的碰撞冲量。全局速度更新采用牛顿法,也可以使用拟牛顿法进一步加快算法速度。在下一章中,我们将通过实验验证算法的准确性,并与前人的方法进行比较。

第四章 实验验证与对比

本章将用实验验证算法的正确性并与前人的方法进行对比。首先，4.1 节介绍了代码实现上的一些细节。接着，4.2 节用一些精心设计的实验证明了算法输出结果能够保证碰撞点之间不发生相互穿透以及摩擦力计算的正确性，并且我们还分析了算法在解单个碰撞时的收敛特性。最后，4.3 节呈现了大规模的仿真测试，在视觉效果和运行速度上与现有方法进行比较。

4.1 实现细节

算法实现使用 C++，项目链接：<https://github.com/pielet/creamystrand>。实现基于开源项目 creamystrand¹，在碰撞检测和模型输入输出部分调用了原项目代码。下面简要介绍一下实现中非算法部分的一些细节：

- 连续碰撞检测

在已知物体相邻两帧的位置后可以进行连续碰撞检测，如图 4-1 所示，假设物体在这一小段时间做匀速运动，在两帧都不发生穿透的情况下仍可能在运动过程中相撞。

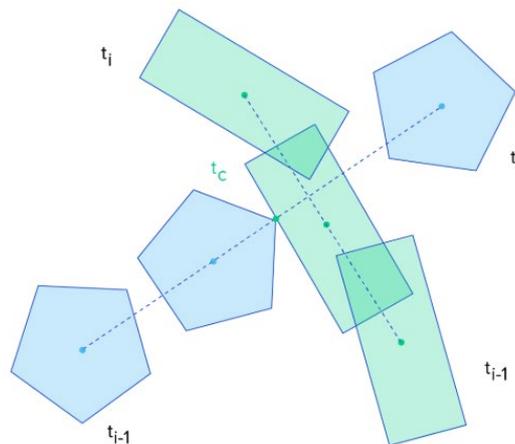


图 7-1 连续碰撞检测：两物体在 $t_{i-1} \sim t_i$ 时间段内匀速运动，在 t_c 时刻发生碰撞²

我们的算法中需要检测毛发之间的边边碰撞和毛发与外部物体之间的点面碰撞，对于这两种情况，若中间出现碰撞，则会发生四点共面的情况，根据共面条件可以解出共面时刻 t ：

$$(x_1 + v_1 t) \times (x_2 + v_2 t) \cdot (x_3 + v_3 t) = 0, \quad (4.1)$$

其中， x_k, v_k ($k = 1, 2, 3$) 是后三个点相对于第一个点的位置和速度。有了共面时刻我们可以算出共面时两物体的距离，若这个距离小于预设的碰撞半径就认为发生了碰撞。

在实现中，我们还使用了 BVH 过滤需要进行碰撞检测的物体。

¹ <https://github.com/nepluno/creamystrand>

² <https://www.toptal.com/game/video-game-physics-part-ii-collision-detection-for-solid-objects>

- 固定点软约束

在处理毛发的固定点时，如毛发根部与头部的连接，我们使用一个自由长度为 0 的弹簧连接需要固定的点和目标位置点，并将这个弹簧的系数设置的很大。这个弹簧的能量被算作内能的一部分：

$$E_{\text{fix}} = \frac{1}{2} k_{\text{fix}} \|\mathbf{x} - \hat{\mathbf{x}}\|^2 \quad (4.2)$$

其中， \mathbf{x} 是毛发上固定点的位置， $\hat{\mathbf{x}}$ 是目标位置。

- 并行计算

算法中有很多可以并行执行的部分，比如速度更新可以在毛发之间并行计算，碰撞检测可以在不同的 BVH grid 中并行执行，局部速度更新也可以在碰撞点之间并行（后两种需要读写一部分共享内存，并行效率不太高）。代码中使用 OpenMP 进行并行计算。

4.2 算法正确性验证

本节前两个小节将验证算法能否保证物体之间不相互穿透，和不同的摩擦系数下摩擦力计算的正确性。最后一个小节验证算法收敛速度。

4.2.1 碰撞

- 毛发之间的碰撞



图 4-2 一束一端固定的头发掉到一束两端固定的头发上。最大迭代次数 $N_{\text{max}} = 20$ ，平均速度变化阈值 $\text{tol} = 10^{-6} \text{cm/s}$ 。

- 毛发与外部物体的碰撞



图 4-3 一束一端固定的头发掉到一个固定的圆柱形刚体上。最大迭代次数 $N_{\text{max}} = 20$ ，平均速度变化阈值 $\text{tol} = 10^{-6} \text{cm/s}$ 。

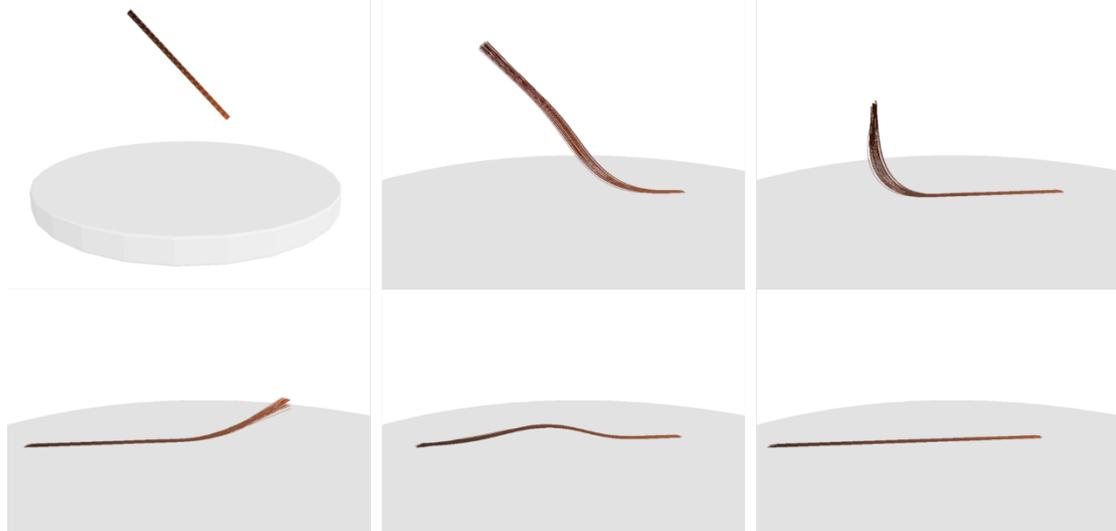


图 4-4 一束倾斜的头发自由掉落到地上。可以看到隆起在头发两端传播，期间涉及大量头发与地面以及头发之间的碰撞，最终这束头发能够稳定的静止在地面上。最大迭代次数 $N_{max} = 20$ ，平均速度变化阈值 $tol = 10^{-6}cm/s$ 。

4.2.2 摩擦

- 毛发之间的摩擦

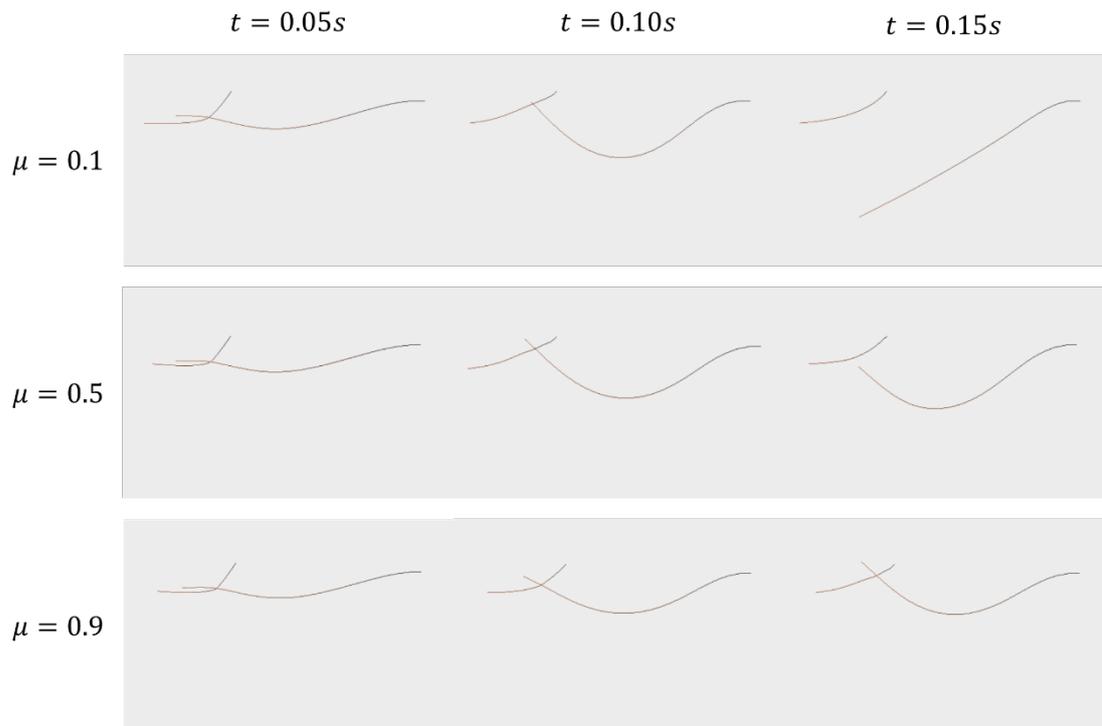


图 4-8 一根一端固定的头发掉到一根两端固定的头发上。每一行对应不同摩擦系数下的仿真结果，每一列处于仿真序列的同一时刻，可以看到摩擦系数越大头发越难滑掉。

- 毛发与外部物体之间的摩擦

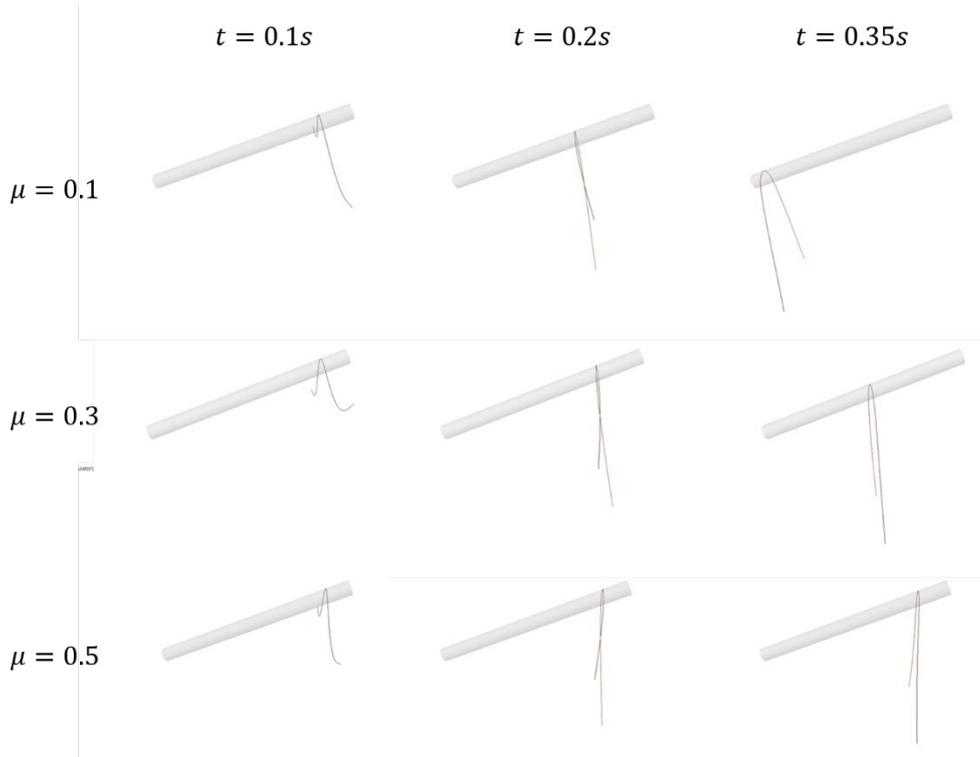


图 4-9 一个头发自由掉落到一根倾斜的钢管上。每一行对应不同摩擦系数下的仿真结果，每一列处于仿真序列的同一时刻，可以看到摩擦系数越小头发滑落越快。

4.2.3 收敛特性

首先定义三个衡量收敛性的指标：

- 平均速度变化量——两次迭代速度变化量的平均值，其中 n 为毛发数， k 为当前迭代数：

$$\|\Delta v\| = \frac{1}{n} \sum_{i=1}^n \frac{\|v_i^{k+1} - v_i^k\|}{\dim v_i}. \quad (4.3)$$

- 平均碰撞冲量变化量——每次迭代算出的碰撞冲量的平均值，其中 m 为碰撞数

$$\|\Delta r\| = \frac{1}{m} \sum_{j=1}^m \|r_j^k\|. \quad (4.4)$$

- Fischer-Burmeister 互补函数

库伦摩擦定律可以等价为一个二阶锥互补问题^[37] (SOCCP, second-order cone complementarity problem)

$$(\mathbf{r}, \mathbf{u}) \in \mathcal{L}(\mu) \Leftrightarrow K \ni \hat{\mathbf{u}} \perp \hat{\mathbf{r}} \in K \quad (4.5)$$

其中 $\hat{\mathbf{u}}, \hat{\mathbf{r}}$ 由 \mathbf{u}, \mathbf{r} 变换得到，变换关系如图 4-7 所示。而 SOCCP 又可等价于 Fischer-Burmeister 互补函数求零点问题：

$$K \ni \mathbf{x} \perp \mathbf{y} \in K \Leftrightarrow f_{SOCCP}^{FB}(\mathbf{x}, \mathbf{y}) = \mathbf{0}, \quad (4.6)$$

其中, $f_{SOC}^{FB}(\mathbf{x}, \mathbf{y}) = \mathbf{x} + \mathbf{y} - (\mathbf{x} \circ \mathbf{x} + \mathbf{y} \circ \mathbf{y})^{\frac{1}{2}}$, \circ 运算符表示 Jordan product. 因此可以用 f_{SOC}^{FB} 的值衡量速度和冲量满足库仑摩擦定律的情况。

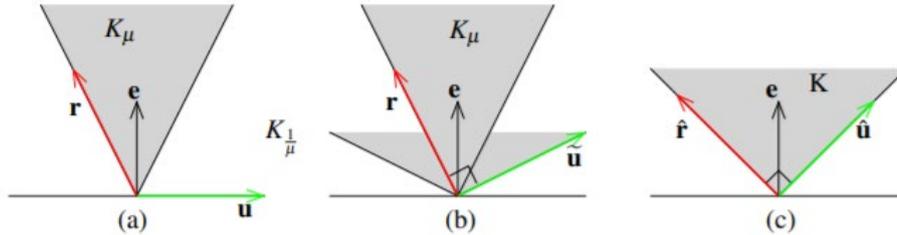


图 4-10 遵循库仑摩擦模型的 \mathbf{u} 和 \mathbf{r} 经过变换在 \mathbf{K} 内互补^[37]

为了方便观察, 本节实验使用两根头发相撞, 只有一个碰撞点, 迭代 50 次, 在两个不同的时刻收敛曲线如图 4-8 所示。

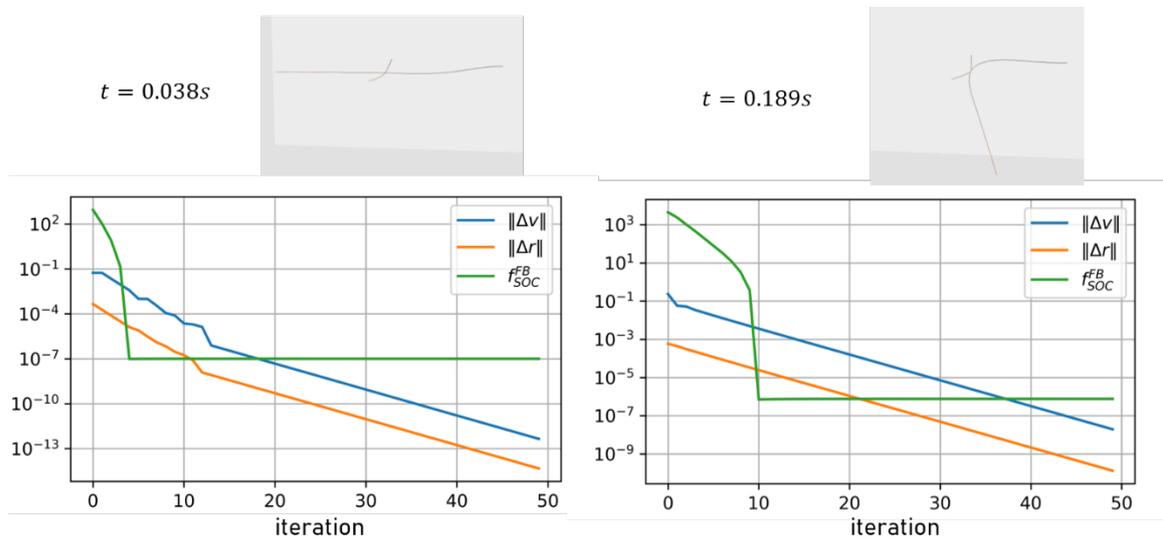


图 4-11 两根头发相撞, 左边是刚好相撞时的收敛曲线, 右边是弯曲最大时刻的收敛曲线, 横轴为迭代数, 纵轴为对数尺度下的变化量大小。可以看到算法收敛速度很快, 在 20 个迭代内变化量可以减少到 10^{-3} , 并且结果很好的遵循了库仑摩擦定律。

4.3 与现有方法对比

本节对比的方法为 ADONIS^[39], 测试用的模型是一个上半表面长有 1K 根头发的球体, 从如右图所示的静止状态开始逆时针旋转一周, 仿真步长为 1ms, 总时长 5s。我们的算法使用拟牛顿法更新速度, 最大迭代次数为 50, 速度变化量跳出值为 10^{-6} ; ADONIS 使用论文中提到的 Gauss-Seidel 解碰撞, 最大迭代次数 200, 跳出条件为平均 f_{SOC}^{FB} 值小于 10^{-6} , 最大牛顿迭代次数为 50。两种方法中头发与球体的摩擦系数和头发之间的摩擦系数均为 0.3。



我们的算法和 ADONIS 的仿真结果分别见图 4-9 和图 4-10，每帧平均用时分布见表格 4-1。可以看到我们的方法在视觉效果上和 ADONIS 几乎没有差别，但是用时远小于 ADONIS。ADONIS 仿真 95% 的时间都在解碰撞，而解碰撞在我们的算法中只占不到 5%，因此虽然我们的算法每次迭代都需要重新计算内能梯度，速度依然比 ADONIS 快一个数量级还多，尤其在碰撞数量多的帧，我们的算法的加速效果更加明显。

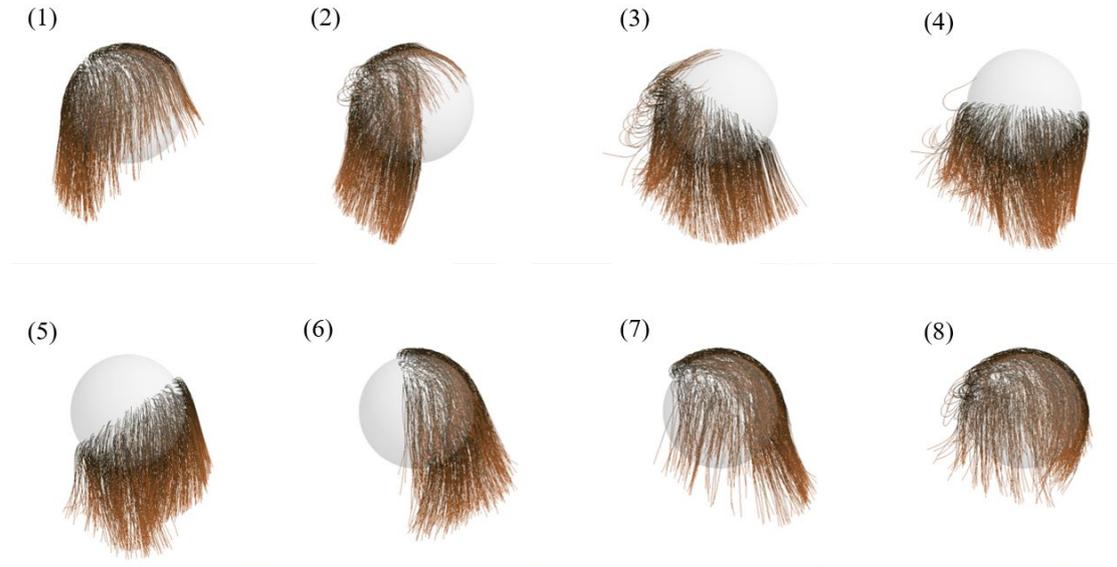


图 4-12 我们的算法仿真结果

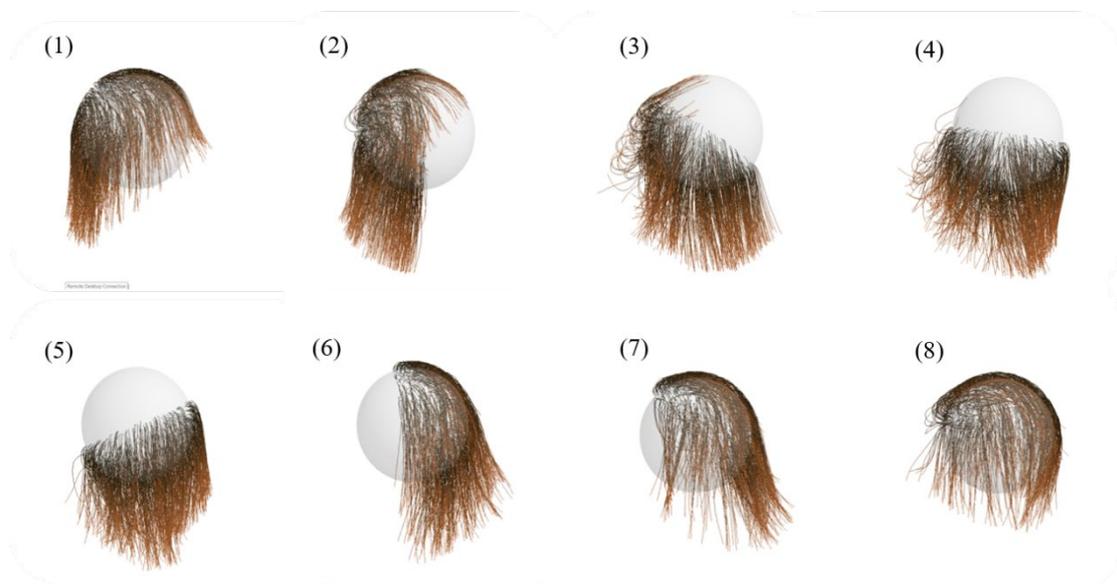


图 4-13 ADONIS 仿真结果

表格 4-1 我们的算法和 ADONIS 仿真 hairball 时的一帧的平均用时，模型总自由度 40k，平均每帧碰撞数约为 5500。Prepare 包含更新外部物体位置和一些预计算（我们的 Prepare 包含了计算初始海森矩阵和预分解），Dynamic 表示牛顿迭代，CCD 表示连续碰撞检测，PC 表示处理碰撞，Solve 为解碰撞冲量

Method	Prepare	Dynamic	CCD	PC	Solve	Total
ours	57.25 ms	418.86 ms	102.54 ms	1.24 ms	29.24 ms	609.22 ms
ADONIS	8.87 ms	206.99 ms	218.48 ms	7.07 ms	8490.51 ms	8930.93 ms

4.4 本章小结

本章通过实验验证了算法的正确性，证明了我们的算法能以较快的速度收敛，并且在收敛时得到正确的摩擦力。在与 ADONIS 对比后证明我们的算法在视觉效果上无明显差异的情况下运行速度大幅提升。

第五章 结论

本次毕设探索了如何快速处理毛发之间的碰撞，提出了一种分步迭代算法，将碰撞求解与弹性力求解结合。其中，冲量更新将每个碰撞点看作局部的刚体系统，速度更新将局部的碰撞冲量通过弹性力求解传播到整根头发上。交替进行冲量更新与速度更新，在算法收敛时可以达到无碰撞的状态。实验表明，我们的算法在各种碰撞情境下都能收敛到正确状态，并且算法收敛速度很快，每步计算耗时小，因而在大规模的仿真中速度远超目前效果最好的毛发仿真算法。

接下来的工作我们将探索如何进一步加速毛发仿真，可行的方向有，如何修正因迭代次数较少而发生穿透的情况，如何在实现层面提升代码的可并行性。并且我们还想提升算法的通用性，将这种想法应用到更广泛的仿真材质上。

参考文献

- [1] William H Press. Numerical recipes 3rd edition: The art of scientific computing. Cambridge university press, 2007. 5, 21, 3.
- [2] Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. Elastically deformable models. In Computer Graphics (Proceedings of SIGGRAPH), volume 21, pages 205–214, 1987. 4, 6.
- [3] David Baraff and Andrew Witkin. Large steps in cloth simulation. In Proc. of ACM SIGGRAPH, pages 43–54, 1998. 6, 36, 41.
- [4] Liliya Kharevych, Weiwei Yang, Yiyong Tong, Eva Kanso, Jerrold E Marsden, Peter Schroder, and Matthieu Desbrun. Geometric, variational integrators for computer animation. pages 43–51, 2006. 6, 21.
- [5] Ari Stern and Matthieu Desbrun. Discrete geometric mechanics for variational time integrators. In ACM SIGGRAPH Courses, pages 75–80. ACM, 2006. 6, 21.
- [6] Theodore F Gast, Craig Schroeder, Alexey Stomakhin, Chenfanfu Jiang, and Joseph M Teran. 2015. Optimization integrator for large time steps. IEEE transactions on visualization and computer graphics 21, 10 (2015), 1103–1115..
- [7] Sebastian Martin, Bernhard Thomaszewski, Eitan Grinspun, and Markus Gross. 2011. Example-based elastic materials. In ACM Trans. Graph., Vol. 30. ACM, 72.
- [8] Breannan Smith, Fernando De Goes, and Theodore Kim. 2019. Analytic Eigensystems for Isotropic Distortion Energies. ACM Trans. Graph. 38, 1 (2019), 3.
- [9] Joseph Teran, Eftychios Sifakis, Geoffrey Irving, and Ronald Fedkiw. 2005. Robust quasistatic finite elements and flesh simulation. In Proc. EG/ACM Symp. Computer Animation. ACM, 181–190.
- [10] Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. 2007. Position based dynamics. Journal of Visual Communication and Image Representation 18, 2 (2007), 109–118.
- [11] Jos Stam. 2009. Nucleus: towards a Unified Dynamics Solver for Computer Graphics. In IEEE Int. Conf. on CAD and Comput. Graph. 1–11.
- [12] Miles Macklin, Matthias Müller, and Nuttapong Chentanez. 2016. XPBD: position-based simulation of compliant constrained dynamics. In Proceedings of the 9th International Conference on Motion in Games. ACM, 49–54.
- [13] Jan Bender, Matthias Müller, and Miles Macklin. 2015. Position-Based Simulation Methods in Computer Graphics.. In Eurographics (Tutorials).
- [14] Jan Bender, Matthias Müller, Miguel A Otaduy, Matthias Teschner, and Miles Macklin. 2014. A survey on position-based simulation methods in computer graphics. In Comput. Graph. Forum, Vol. 33. Wiley Online Library, 228–251.
- [15] Tiantian Liu, Adam W Bargteil, James F O’Brien, and Ladislav Kavan. 2013. Fast simulation of mass-spring systems. ACM Trans. Graph. 32, 6 (2013), 214.
- [16] Sofien Bouaziz, Sebastian Martin, Tiantian Liu, Ladislav Kavan, and Mark Pauly. 2014. Projective dynamics: fusing constraint projections for fast simulation. ACM Trans.
- [17] Tiantian Liu, Sofien Bouaziz, and Ladislav Kavan. 2017. Quasi-newton methods for real-time simulation of hyperelastic materials. ACM Trans. Graph. 36, 4 (2017), 116a.

- [18] Matthew Overby, George E Brown, Jie Li, and Rahul Narain. 2017. ADMM Projective Dynamics: Fast Simulation of Hyperelastic Models with Dynamic Constraints. *IEEE TVCG* 23, 10 (2017), 2222–2234
- [19] Sunil Hadap et al. “Strands and hair - Modeling, simulation and rendering.” In: *SIGGRAPH Courses*. 2007, pp. 1–150.
- [20] Kelly Ward et al. “A survey on hair modeling: Styling, simulation, and rendering.” In: *IEEE TVCG 13.2* (2007), pp. 213–234.
- [21] Lena Petrovic, Mark Henne, and John Anderson. “Volumetric methods for simulation and rendering of hair.” In: *Pixar Animation Studios 2.4* (2005).
- [22] Andrew Selle, Michael Lentine, and Ronald Fedkiw. “A mass spring model for hair simulation.” In: *ACM Transactions on Graphics (TOG)* 27.3 (2008), p. 64.
- [23] Mireille Grégoire and Elmar Schömer. “Interactive simulation of onedimensional flexible parts.” In: *Computer-Aided Design* 39.8 (2007), pp. 694–707.
- [24] Jonas Spillmann and Matthias Teschner. “CoRdE: Cosserat rod elements for the dynamic simulation of one-dimensional elastic objects.” In: *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer animation*. Eurographics Association. 2007, pp. 63–72.
- [25] Florence Bertails et al. “Super-helices for predicting the dynamics of natural hair.” In: *ACM Transactions on Graphics (TOG)* 25.3 (2006), pp. 1180–1187.
- [26] Miklós Bergou et al. “Discrete Elastic Rods.” In: *ACM Transactions on Graphics (SIGGRAPH)* 27.3 (2008), 63:1–63:12.
- [27] Miklós Bergou et al. “Discrete viscous threads.” In: *ACM Transactions on Graphics (TOG)* 29.4 (2010), p. 116.
- [28] Crispin Deul et al. “Direct Position-Based Solver for Stiff Rods.” In: *Computer Graphics Forum* 37.6 (2018), pp. 313–324.
- [29] Dongsoo Han and Takahiro Harada. “Tridiagonal matrix formulation for inextensible hair strand simulation.” In: *Proceedings of the Workshop on Virtual Reality Interaction and Physical Simulation*. The Eurographics Association, 2013.
- [30] Tassilo Kugelstadt and Elmar Schömer. “Position and orientation based Cosserat rods.” In: *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Eurographics Association. 2016, pp. 169–178.
- [31] Matthias Müller, Tae-Yong Kim, and Nuttapon Chentanez. “Fast simulation of inextensible hair and fur.” In: *Proceedings of the Workshop on Virtual Reality Interaction and Physical Simulation*. Vol. 12. 2012, pp. 39–44.
- [32] Carlota Soler, Tobias Martin, and Olga Sorkine-Hornung. “Cosserat Rods with Projective Dynamics.” In: *Computer Graphics Forum* 37.8 (2018), pp. 137–147.
- [33] Richard Cottle, Jong-Shi Pang, and Richard E. Stone. 1992. *The Linear Complementarity Problem*. Academic Press.
- [34] David Baraff. 1996. Linear-time dynamics using Lagrange multipliers. In *Proc. of the 23rd Annual Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH'96)*. ACM, New York, NY, 137–146.
- [35] Miguel A. Otaduy, Rasmus Tamstorf, Denis Steinemann, and Markus Gross. 2009. Implicit contact handling for deformable objects. *Comput. Graph. Forum* 28, 2 (2009), 559–568.
- [36] Danny M. Kaufman, Shinjiro Sueda, Doug L. James, and Dinesh K. Pai. 2008. Staggered

- projections for frictional contact in multibody systems. *ACM Trans. Graph.* 27, 5 (2008), 164:1–164:11.
- [37] Gilles Daviet, Florence Bertails-Descoubes, and Laurence Boissieux. 2011. A hybrid iterative solver for robustly capturing Coulomb friction in hair dynamics. *ACM Trans. Graph.* 30, 6 (2011), 139:1–139:12.
- [38] Siwang Li, Zherong Pan, Jin Huang, Hujun Bao, and Xiaogang Jin. 2015. Deformable objects collision handling with fast convergence. *Comput. Graph. Forum* 34, 7 (2015), 269–278.
- [39] Danny M. Kaufman, Rasmus Tamstorf, Breannan Smith, Jean-Marie Aubry, and Eitan Grinspun. 2014. Adaptive Nonlinearity for Collisions in Complex Rod Assemblies. *ACM Trans. on Graph.* 33, 4, Article 123 (July 2014).
- [40] Tiantian Liu. TOWARDS REAL-TIME SIMULATION OF HYPERELASTIC MATERIALS. 2018

谢辞

本次毕设是我大四在微软亚洲研究院实习期间的项目，从刚来时对于物理仿真一无所知，到之后阅读各种文献、实现各种算法，这个过程使我成长了许多。虽然在物理仿真或者说图形学方面我依然是个新手，但我非常愿意继续在这个领域学习更多的知识，做更多有趣的研究。

在实习期间最感谢的就是我的导师刘天添，是他真正带我走进图形学科研，经常与我一起讨论，回答我各种弱智问题。除了科研上，他在生活中和申请学校时也给予了我巨大的帮助。

然后是要感谢交大已经毕业的学长蒋健巍，他在做助教时经常花几个小时与我讨论问题，在之后的暑期实践中带我实现了初步的毛发仿真框架，在我来实习后也持续帮我解决问题。还要感谢我之前实验室的汪旻学长和谭鑫学长，在计算机视觉领域帮助我入门，在我迷茫的时候帮我规划未来方向。

最后还要感谢我的父母，为我提供良好的生活条件得以使我完成大学生涯和这篇毕业论文。

FAST HAIR SIMULATION

Human hair exhibits complicated dynamic behaviors and rich motion details while moving with head and interacting with other objects such as the human body. Faithfully simulating the detailed hair dynamics in real-time is important for realistic character animation in games and virtual reality. However, it is still a challenging task due to the large number of hair strands and rich self-interactions.

Over the past years, researchers have developed methods for hair simulation based on a variety of hair models. Petrovic et al.^[21] and Selle et al.^[22] use the mass-spring model to represent hair. Bertails et al.^[25] created the “Super-Helix” model based on Cosserat theory of rods. In their model, a hair strand is represented as a serial of helix connected. Bergou et al.^[26, 27] defined stretching, bending, and twisting energy based on Kirchhoff’s theory and proposed the Discrete Elastic Rods model (DER). Because of its robustness and accuracy, DER became one of the most popular models in hair simulation.

The interactions among strands is another important factor for achieving realistic hair animations. Daviet et al.^[37] combined the motion equation with Coulomb friction law and formulated it as a cone complementarity problem. Li et al.^[38] proposed a projected gradient descent method to project unconstrained position into a collision-free manifold. ADONIS^[39] considered the nonlinearity in the hair model and proposed an iterative hair simulation algorithm based on Daviet et al.’s solver. However, the methods mentioned above are extremely time-consuming when a large amount of collisions happened.

We observe that the collision response is a local problem, they only affect a small number of points near them. It is the elastic force that propagates the collision impulse to entire hairs. Based on this observation, we propose an iterative algorithm based on Newton’s method, which integrates friction calculation into every Newton iteration. Under the discrete elastic rod model, our algorithm is divided into two steps. The local step updates collision impulse. It treats every rod involved in the collision as a rigid body and solves the collision impulse according to the current velocity. Specifically, we calculate collision impulse needed to keep the rigid body rods unmoved relative to each other, then project the tangential impulse to the friction cone to make them follow the Coulomb friction law. The global step updates velocities through one Newton iteration to propagate the local impulse information to entire hair. Besides, we divide the mass of rods by the number of collisions they involved to correct collision impulse. Global updates and local updates are performed alternately until all the hair strands reach a collision-free state. Due to the simplicity of the local collision problem, our method could solve contact and friction force quickly and the result follows the Coulomb friction law without losing angular momentum.

In the experiment part, we use some well-designed experiments to verify the correctness of our algorithm. After the algorithm converges, it can ensure that there is no penetration between the hairs, and the final velocity and collision impulse follow the Coulomb friction law. Comparison

with state-of-the-art hair simulation algorithms ADONIS shows that our method greatly improves the simulation speed without losing accuracy.