



基于三维弹簧质点模型的实时可变形弹性体模拟

摘要

随着计算机的算力随着摩尔定律的预言逐年快速增长，人们逐渐对于做出更真实的游戏物理引擎，或是通过计算机进行模拟手术（模拟内脏）的有了更高的需求度，可变形弹性体的模拟正变得越来越应用广泛。在现存的算法中，一般无法兼顾物理上的真实性和模拟计算的速度。我的目标是建立一个符合物理学原理的通用可变形弹性体模型，这个模型应该易于使用，适合于各种不同形状的弹性体，同时也能够以相对精确的水平进行实时的模拟，以便用于上述的两个目的。所以，我选择了三维弹簧质点（MSS）的模型，使用三种具有不同力学特性的弹簧构建正六面体单元，使用这一单元组合成任意形状的质量均匀分布的各向同性物体。同时，考虑到仅有收缩和伸长两种状态的弹簧具有一定的局限性，当弹性体受力发生形变时发生的体积变化并不符合真实的情况。因此，在线性的弹簧弹力之外引入了额外的以弹性体体积变化为参数的体积修正力，通过快速的不规则六面体体积估算算法，实时控制模拟对象的体积变化，同时不影响整体的受力变形状况。通过以上两点，就可以达成具有实时性能，同时也比经典弹簧质点模型方法更精确的可变形弹性体模拟。为了可视化模拟的效果，本文使用了 OpenGL 来实时渲染输出模拟的模型并对可变形弹性体实现了碰撞检测。

关键词：可变形弹性体模拟，弹簧质点模型，实时，体积修正



Real-time Deformable Solid Simulation Based on 3D Mass Spring System

ABSTRACT

As the computing power of the computers grows rapidly like the prophecy of Moore's Law, people gradually get a higher demand for making more realistic game physics engines, or for virtual surgery (simulating internal organs) through computers, so the simulation of deformable elastomers is becoming more and more widely used. In the existing algorithms, it is generally impossible to get both physical accuracy and speed of simulation calculation at the same time. My goal is to build a universal deformable elastomer model that fits the principles of physics, which should be easy to use and suitable for elastomers of various shapes. It should calculate and show simulation result in real time. For the two purposes described above. I choose the model based on 3D Mass Spring System (MSS), using three kinds of springs with different mechanical properties to build a cubic unit. Then it's possible to use this kind of units to combine an isotropic object with uniform mass distribution in any shapes. At the same time, considering that the springs can only shrink and elongate, which may lead to some limitations. The volume change that occurs when an elastomer is deformed by some external force could not fit the real situation. Therefore, in addition to the linear spring elasticity, an additional volume correction force is introduced, and it takes the elastomer volume change as a parameter. The volume change of the simulated object is controlled in real time by a fast irregular-hexahedron volume estimation algorithm. It works without affecting the overall deformation condition. With the above two points, it is possible to achieve a deformable elastomer simulation with real-time performance, which is also more accurate than the classical Mass Spring System method. In order to visualize the effect of simulation, OpenGL is used to render the output of the simulation model in real time. A collision detection of the deformable elastomer method is attached in the simulation loop.

Key words: deformable solid simulation, mass spring system, real time, volume correction



目 录

第一章	绪论.....	1
1.1	可变形弹性体模拟研究现状.....	1
1.1.1	有限元法.....	1
1.1.2	无网格方法.....	1
1.1.3	弹簧质点模型.....	3
第二章	弹簧质点模型架构.....	4
2.1	弹簧网络构建.....	4
2.2	弹簧质点模型的物理参数.....	5
2.2.1	质点粒子质量.....	5
2.2.2	弹簧刚度.....	5
2.2.3	阻尼.....	6
2.3	模拟循环.....	6
2.3.1	质点粒子的运动.....	6
2.3.2	积分方法.....	7
2.3.3	模拟迭代算法.....	8
第三章	体积修正力.....	10
3.1	立方体单元体积估算.....	10
第四章	碰撞检测.....	12
4.1	与平面碰撞.....	12
第五章	结论.....	14
5.1	模拟效果.....	14
5.2	模拟效果分析.....	15
5.2.1	模拟效率.....	15
5.2.2	弹簧质点模型存在的不足.....	15
5.3	可行的改进.....	16



第一章 绪论

如今,对于现实生活中的时间和现象进行建模以进行计算机模拟正变得越来越普遍,通过建模与模拟计算,我们能够对特定的事件和行为进行低成本的预演,由此能够极大地辅助人类的决策过程,对于处理现实的问题具有很高的参考价值。物理模拟大致可以分为气体、液体、刚体和可变形固体,本文的内容集中于讨论可变形固体的模拟。一般来说,同样是可变形固体模拟的研究,出于计算机图形学方向的(例如游戏引擎)会更注重解决方案的轻量化,计算的实时性和视觉上的准确性,因此会使用简化的物理模型和数学工具,常用模型为弹簧质点模型和无连接点云模型;而出于专业模拟需求的模拟(常见于生物医学模拟,如虚拟手术)会更重视模拟的准确性和稳健性,因此往往会使用更复杂的仿真系统^[1],比如有限元模型以及有限元模型的一些变种。

1.1 可变形弹性体模拟研究现状

可变形弹性体的模拟计算是一个广阔的领域,涵盖了拓扑,线性约束和积分等等主题^[2]。因此也诞生了许多不同的模拟方法,除了本文使用的弹簧质点模型法,最常用的两种模拟方法分别是有限元法(FEM),和无网格(Meshless)方法。以下做一些基本的介绍。

1.1.1 有限元法

有限元法把单一的复杂系统分解成有限个易于解决的单元,再对结构中的每一个单元分别进行力学控制方程的求解,然后通过单元分析组装得到结构总刚度方程,再通过边界条件和其他约束解得结构总反应。解决了每个单元中的问题并收集解决方案后,就可以得到全局解决方案本身。这样就可以避免直接建立复杂结构的力学和数学模型了。

Sifakis 等人^[3]在 2012 年的文章中介绍了有限元模拟的实时技术,包括线性模态分析,模态扭曲,子空间模拟,域分解等。有限元方法的优势在于拥有许多通过真实物理实验获得的本构关系可供使用,这为有限元方法提供了极高的模拟真实性。不过也正因此,有限元法的计算复杂度相对较高。

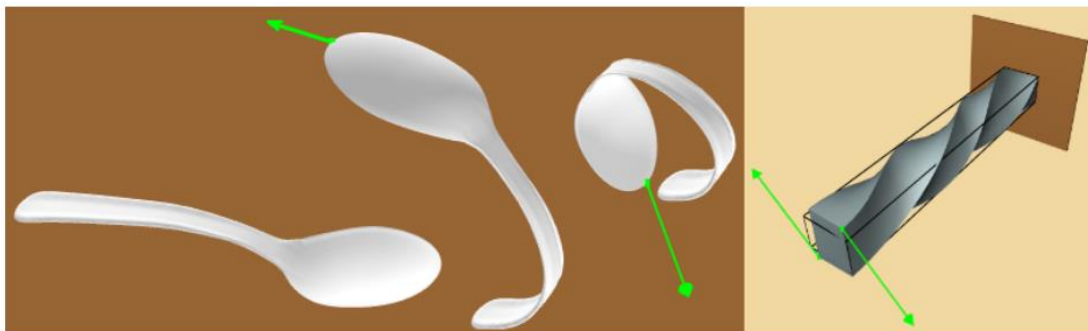


图 1-1 Sifakis 等人通过有限元方法,通过模态导数响应大变形

1.1.2 无网格方法

与可变形弹性体模拟领域经典的算法不同,无网格模拟中弹性体被抽象为点云,点云中的点之间不像经典网络中的那样会与周边的一些邻居相连。因此,传统的基于能量和作用力的几何约束不再适用,取而代之的是基于点云位置变化的形状匹配算法。这一算法通过将发



生位移的点拉向目标位置来模拟材料的弹性，在点云中的点运动的时候，施加线性或非线性的约束条件来使得模拟具有视觉上的真实感。这一方法的优点在于可以无条件地稳定处理非常大的变形，并且发生变形的物体可以完美地从剧烈变形中恢复原有的形状，而且此方法的计算效率是目前所有使用带质量粒子的模型中最高的，相比其他模型，可以在相同的时间内，模拟更多的粒子运动。这种方法的缺点在于：一般可变形物体都是符合连续介质力学的，以点云作为简化模型与之违背，因此虽然此方法具有令人满意的视觉效果，但是其模型的变形缺乏物理依据。

(1)

Müller 等人^[4]在 2005 年提出了这一方法，主要思路是将点云中的粒子的初始位置与其发生变形后的当前位置进行广义形状匹配来确定该粒子的运动方向和速度。在这个方案中，由于点总是被绘制到明确定义的位置，因此消除了显式积分方案的过冲问题。因为计算复杂性方面的高效以及无条件的稳定性使得该方法特别适合于游戏引擎。

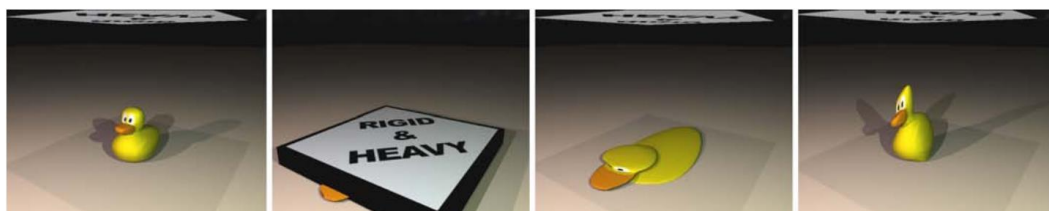


图 1-2 鸭子模型受重压发生巨大变形后，依旧能在撤去外力后恢复原形

(2)

Faure 等人^[5]在 2011 年改进了 (1) 中的无网格模拟方法，他们的模拟方案可以在给定材料的体积分布以及任意数量的控制节点之后，通过新的算法自动计算节点的分布以及相关的形状匹配参数。相比于与 (1) 中的原始方法，他们的方法适用于模拟异质材料特性的物体，文中给出的示例是 T 骨牛排，由坚硬的骨头与相对容易变形的牛肉两个部分组成。对于不同的材料属性，不同的节点会被不同的连续介质力学公式所控制，由于使用了自定义的材料感知形状匹配函数来代替无网格框架中使用的传统径向基函数，此文中所提到的方法能够有效地处理不均匀刚度材料的可变形体模拟。

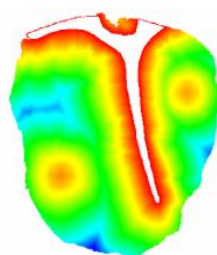


图 1-3 T 骨牛排中不同区域的不同刚度

(3)

Steinemann 等人^[6]在 2008 年提出了一种新的形状匹配变形模型，这种方法基于无网络方法中常用的格子形状匹配算法 (LSM)，但是通过重叠区域上的刚性模型匹配算子的卷积来计算模拟节点的位置。因此这种方法同时具有形状匹配算法的效率和稳健性，同时也具有算法简单性与线性的计算复杂度。这种方法消除了密集采样的要求，可以自适应地进行空间离散化。

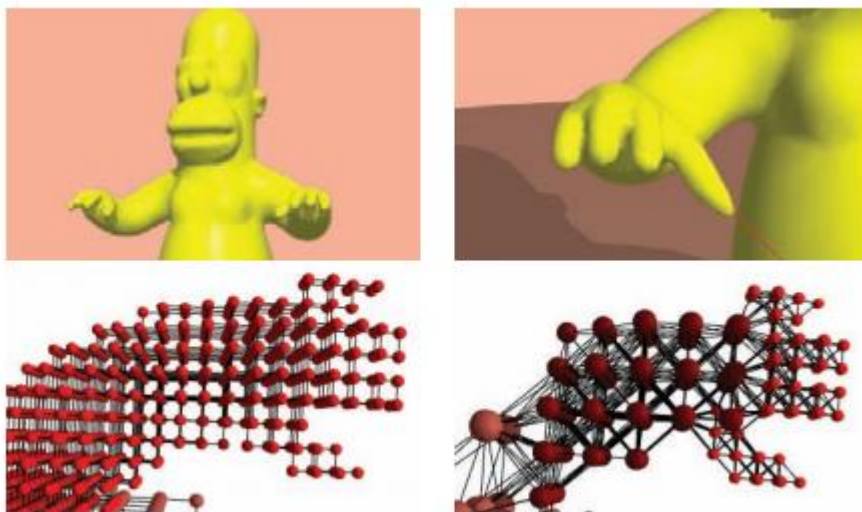


图 1-4 通过自适应空间离散化大大减少模型复杂度

1.1.3 弹簧质点模型

Mollemans^[7]等人在 2003 年建立了一种基于四面体单元的三维弹簧质点模型，在构建经典的弹簧质点网格之外，他们的解决方案还讨论到了弹性体变形时的体积控制。在他们的方案中，除了线性的弹簧弹力，还对每一个四面体单元附加了额外的体积修正力：

$$F_j^{vol} = \sum_{vi \in \Omega_j} (V_i^{cur} - V_i^{init}) \frac{p_j - p_{Bi}}{|p_j - p_{Bi}|} \quad (1-1)$$

在他们的解决方案中，体积修正力的大小与四面体单元的体积变化线性相关，方向平行于四面体单元的各个棱边。对于四面体来说，无论发生何种变形，其形状依旧为四面体，任选其中三点的空间坐标，可以得到底面三角形的面积，计算底面三角形的法向量并参考剩下一点的空间坐标，即可求出四面体的高。因此对于任意不规则的四面体，其体积的计算都是简单的，具有实时计算效率的。

最终，在他们的解决方案中，质点粒子的运动取决于外力，弹簧系统内力，与体积修正力的和，即：

$$F_j = F_j^{int} + F_j^{ext} + \alpha * F_j^{vol} \quad (1-2)$$



第二章 弹簧质点模型架构

弹簧质点模型因为其简单易用的特点在可变形弹性体的模拟中十分常见。简单地说，弹簧质点模型就是把一个物体按照预设的采样密度分为若干个离散的粒子，每一个粒子按照原物体的质量分布视作拥有那一部分空间所有物体质量的一个质点。然后再使用弹簧按一定的条件将相邻的部分都进行连接，形成一个质点和弹簧构成的网格系统。实际使用中，常用一维弹簧质点模型模拟头发或是绳索^[8]，用二维弹簧质点模型模拟布料^[9]，用三维弹簧质点模型模拟可变形弹性体。本文主要的讨论内容是三维弹簧质点模型。

2.1 弹簧网络构建

弹簧网络的构建是弹簧质点模型中对于模拟的稳定性和真实性影响最大的因素。在计算机图形学的领域中，任意复杂的模型对象在渲染时都会被分解为数量众多，边和边紧密连接的三角形或四边形面片。只有将模拟的进行可视化我们才能直观地了解到模拟计算的质量，因此，在二维弹簧质点模型中，最基本的网格单元是三角形和四边形，在三位弹簧质点模型中则是四面体与六面体。对于网格中的每一个弹簧，需要的参数有初始的长度和当前的长度，弹簧的刚度，以及阻尼大小。考虑到模型的轻量性以及模拟性能的需求，这些弹簧被视作是零质量的。对于质量分布均匀的各向同性材料，在三个方向轴方向都拥有轴对称特性的正六面体是最简单的模拟单元。于是处理方案的重点就在于如何在正六面体的八个顶点之间构建弹簧网络。

过多的约束条件不仅会增加模拟计算的复杂性，还会使得可变形弹性体的物理表现更接近刚体，因此连接六面体顶点的弹簧数量应该尽可能的少。在三维的六面体网格中，如果仅有构成六面体的 12 条棱边弹簧的话，六面体单元仅能够在棱边向量的六个方向进行压缩或伸长，任何轻微的扭转或是剪切力就会使得模拟单元奔溃。因此首先考虑添加 4 条体对角线弹簧使得六面体单元拥有对于扭转和弯折变形的模拟能力。但这种构建方法在弹簧网络发生变形并撤去外力之后依旧存在皱缩^[10]，无法完美回复到初始状态。

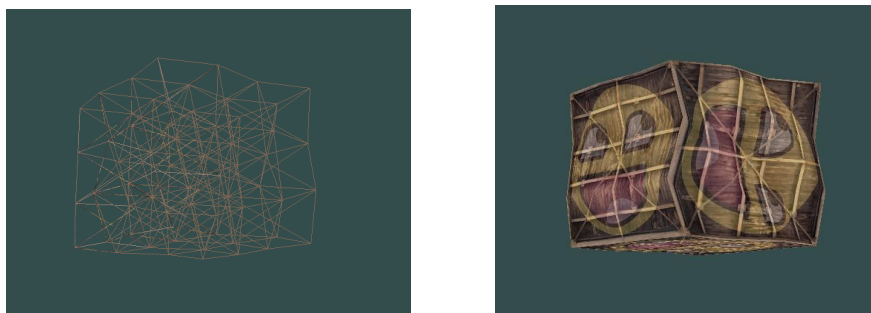


图 2-1 只有棱边弹簧和体对角线弹簧的 3*3*3 可变形弹性方块

于是进一步考虑添加面对角线弹簧，对于六面体单元中的每一个面，在此有两种可能的方案：

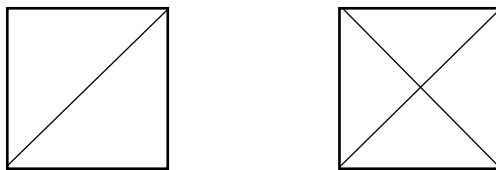


图 2-2 两种不同的面对角线分布

虽然左边的方案相比右边的可以为六面体的每个面节省一个弹簧，同时保持六面体单元的稳定性，但是考虑到我们建模的对象是各项同性的材料，左侧的材料会在朝向的不同同时对相同的外力无法保证产生一致的变形效果，所以还是采用右侧的方案^[12]。

综上，在我的弹簧质点模型中，一个正六面体最小单元的弹簧网络包含 12 根棱边弹簧，12 根面对角线弹簧，4 根体对角线弹簧，弹簧网络的构成如下图所示：

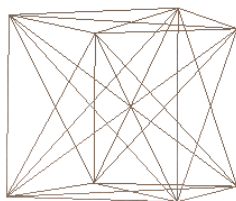


图 2-3 六面体单元中的弹簧分布

2.2 弹簧质点模型的物理参数

2.2.1 质点粒子质量

在构建弹簧质点模型的过程中，势必需要将模拟对象的质量离散到模型中的N个质点粒子中去，所以所有质点粒子的质量之和等于模拟的对象的质量，即：

$$m_{\text{object}} = \sum_i m_i, i \in [1, N] \quad (2-1)$$

虽然我们构建的是匀质物体的模型，但是不能简单地将整体质量平均地分配到每个质点粒子 p_i 中去。因为在离散化之后，显然每个质点粒子所代表的模拟对象的体积是不同的，位于物体内部的会代表更多的体积，而位于模拟对象棱角处的会代表更少的体积。因为我的模拟方案中的最小单元是正六面体，拥有 8 个顶点，可以推导出以下公式^[11]——对于每个质点粒子 p_i 的质量 m_i ：

$$m_i = \sum_j \frac{\rho}{8} v_j \quad (2-2)$$

其中， v_j 代表质点粒子 p_i 所能接触到六面体单元的体积，将所有 v_j 求和，得到质点粒子所占据的体积，再乘上匀质物体的密度，就可以相对准确地求出质点粒子的质量。

2.2.2 弹簧刚度

依据胡克定律，对于理想的轻质弹簧，在受到外力作用时，在其线性范围内，弹簧力的大小与弹簧的形变成正比，弹簧力的方向总是与形变方向相反，其比值即为弹簧的刚度：

$$k_{\text{spring}} = \frac{F_{\text{spring}}}{l - l_0} \quad (2-3)$$



2.2.3 阻尼

在现实世界中，能量的转换总是伴随着能量的损失，对于弹簧质点模型来说，为了避免系统中的弹簧无休止的震荡，必须在弹簧中引入阻尼。阻尼可以分成三种。

- (1) 欠阻尼——系统的震荡随时间指数衰减
- (2) 过阻尼——系统振幅未经震荡指数衰减到零
- (3) 临界阻尼——系统未震荡且最快恢复稳态

对于弹簧质点模型，一般需要尽量避免不必要的震荡，使系统尽快恢复稳态，过阻尼与临界阻尼是比较理想的情况，可以取^[13]：

$$d = 2 \frac{\sqrt{k(m_1 + m_2)}}{l_0} \quad (2-4)$$

其中， k 是弹簧的刚度系数， l_0 是弹簧的原长， m_1, m_2 分别是弹簧两端连接的质点粒子的质量。

或者可以构建一个阻尼力的大小与阻尼器端点的相对移动速度成正比的线性阻尼器，对于弹簧两端的质点粒子分别施加相反与其运动方向的阻尼力：

$$F_{\text{damping}} = k_d \Delta v \quad (2-5)$$

在模拟的过程中，往往扭转变形与线性变形同时存在，因此需要把弹簧两端的线速度差投影到弹簧的方向^[14]：

$$\Delta v = |v_1 - v_2| \times \frac{P_1 - P_2}{|P_1 - P_2|} \quad (2-6)$$

其中 P_1, P_2 是弹簧两端质点粒子的当前位置。

2.3 模拟循环

2.3.1 质点粒子的运动

物体的变形在弹簧质点模型中体现为组成物体的质点粒子在弹簧的力作用下分别独立地进行运动的过程。既然涉及到质点，力和运动，我们就需要使用到牛顿第二运动规律：

$$a \ m = f(v, x, t) \quad (2-7)$$

其中 a 是加速度， m 是质点粒子的质量， v 是质点粒子的速度， x 是物体在时刻 t 的世界坐标，以三维向量表示。 f 是一个以 v, x, t 为参数的函数，它的实际形式取决于用于模拟的具体物理模型。

在实际的模拟程序中，这一部分往往被分解成一下几个步骤：

$$\begin{aligned} a &= \frac{\sum f(v, x, t)}{m} \\ v &= \int a \, dt \\ x &= \int v \, dt \end{aligned} \quad (2-9)$$

这一过程也能表达为一个一阶微分方程组：

$$\begin{cases} \dot{v}_n = f(v_n, x_n, t_n) \\ \dot{x}_n = v_n \end{cases} \quad (2-10)$$

其中 $n \in \{1, 2, \dots, N\}$ ， n 是连续的时间的一刻， N 是模拟仿真的总步骤数。最终，我们需要选择一个积分方法来求解上述的常微分方程系统。



2.3.2 积分方法

(1) 显式欧拉法

显式欧拉法, 也被称为前向欧拉法(Forward Euler), 是本文所提及的积分法中最简单的一种, 它使用以下的公式计算下一时间刻每一个粒子的运动和位置信息:

$$\begin{cases} v_{n+1} = v(n) + \Delta t f(x_n, t_n) \\ x_{n+1} = x_n + \Delta t v_n \end{cases} \quad (2-11)$$

其中, $\Delta t = t_{n+1} - t_n$, 这一方法在每一次迭代的时候忽视了导数的变化, 只是在当前运动和位置信息的基础上进行了简单的迭代。显式欧拉法的优点在于在一个模拟循环中只需要计算一次 f 函数即可。

(2) 半隐式欧拉法

半隐式欧拉法能够给出比显式欧拉法更好的积分结果。该积分法采用以下形式:

$$\begin{cases} v_{n+1} = v(n) + \Delta t f(x_n, t_n) \\ x_{n+1} = x_n + \Delta t v_{n+1} \end{cases} \quad (2-12)$$

在我的模拟体系中, 这一方法和显式欧拉法最大的区别在于: 在 x_{n+1} 的计算过程中, 半隐式欧拉法采用了 v_{n+1} 作为参数, 而显式欧拉法简单的使用了 v_n 。这一方法比起显式欧拉法可以获得更精确的模拟结果。

(3) 隐式欧拉法

隐式欧拉法, 也称为后向欧拉法(Backward Euler), 正如其名称所示, 这一方法涉及到隐式给定的未知数。定义 $\Delta x = x_{n+1} - x_n$, $\Delta v = v_{n+1} - v_n$, 则我们的模拟计算可以表达为:

$$\begin{cases} \Delta v = \Delta t M^{-1} f(v_{n+1}, x_{n+1}, t_n) \\ \Delta x = \Delta t v_{n+1} \end{cases} \quad (2-13)$$

其中 M 是全局质量分布矩阵, 隐式欧拉法的优点在于可以在任意大小的步长保持模拟系统的稳定性^{[15][17]}, 但是由于其计算过程的复杂度(每次迭代都需遍历整个网络), 在需要实时性的模拟中不会采用此方法。

(4) Verlet 积分法

Verlet 积分法是一种二阶积分方法, 它使用中心差异近似二阶导数:

$$\frac{\Delta^2 x_n}{\Delta t^2} = \frac{\frac{x_{n+1} - x_n}{\Delta t} - \frac{x_n - x_{n-1}}{\Delta t}}{\Delta t} = \frac{x_{n+1} - 2x_n + x_{n-1}}{\Delta t^2} = a_n = A(x_n) \quad (2-14)$$

由此, 我们的模拟系统可以表达为以下形式:

$$\begin{cases} x_{n+1} = 2x_n - x_{n-1} + a_n \Delta t^2 \\ a_n = A(x_n) \end{cases} \quad (2-15)$$

可以看到使用此种积分计算方式我们不再需要在每个时刻迭代的时候计算质点粒子的速度。不过在我的弹簧质点模型中, 弹簧的阻尼力会使用到质点粒子的速度作为参数, 虽然无法直接求出准确的 v_n , 但可以通过一些曲折的步骤估算 $v_{n+\frac{1}{2}}$ 来近似计算。首先, 对 v_{n+1} 和 v_{n-1} 做泰勒展开:

$$v_{n+1} = v_n + \Delta t a_n + \frac{\Delta t^2}{2} a_n' + \frac{\Delta t^3}{6} a_n'' + \frac{\Delta t^4}{24} a_n''' + O(\Delta t^5) \quad (2-16)$$



$$v_{n-1} = v_n - \Delta t a_n + \frac{\Delta t^2}{2} x_n^3 - \frac{\Delta t^3}{6} x_n^4 + \frac{\Delta t^3}{24} x_n^5 + O(\Delta t^5) \quad (2-17)$$

以上两式相减可得：

$$v_{n+1} - v_{n-1} = 2\Delta t a_n + \frac{\Delta t^3}{3} x_n^4 + O(\Delta t^5) \quad (2-18)$$

由此我们可以近似地取：

$$v_{n+\frac{1}{2}} = v_{n-\frac{1}{2}} + \Delta t a_n \quad (2-19)$$

又因为：

$$x_{n+1} = x_n + \Delta t v_{n+\frac{1}{2}} \quad (2-20)$$

最终得到质点粒子的速度 v 在 $n - \frac{1}{2}$ 时刻的近似值：

$$v_{n-\frac{1}{2}} = \frac{x_n - x_{n-1}}{\Delta t} \quad (2-21)$$

2.3.3 模拟迭代算法

为了动态展示可变形弹性体的模拟，每一个离散的时间刻都对整个模型做如下操作：

算法 1 迭代计算更新弹簧质点模型

```

1  FOR  $p_i$  in all Particles DO
2      add external forces:  $F_{ext} \leftarrow F_{ext}$ 
3  END
4  FOR  $s_j$  in all Strings DO
5      computer spring force  $F_j$  for  $s_j$ 
6      add string force on the two particles
7       $F_{p_1} += +F_j$ 
8       $F_{p_2} += -F_j$ 
9      computer the speed difference  $\Delta v$  of two particles
10     computer the damping for  $s_j$  based on  $\Delta v$ 
11     add damping force on the two particles
12      $F_{p_1} += +F_{damping}$ 
13      $F_{p_2} += -F_{damping}$ 
14  END
15  FOR  $p_i$  in all Particles DO
16      compute the new positions  $x_i$  using an integration method
17  END

```

在每一次循环中，首先将所有的外力都施加于质点粒子之上（最常见的就是重力），然后对于弹簧质点模型中的每一条虚拟弹簧分别计算它们的弹力和阻尼力，并将这两个力合成到弹簧两端质点粒子的总合力中去。最后再遍历一遍所有的质点粒子，选择一种 2.3.2 中



提到的积分方法，通过 2.3.1 中的运动学公式计算出质点粒子的新位置。如果选择的是显式，或者半显式的积分方法的话，可能由于过长的时间步长导致计算的误差快速累积到使模型不再稳定的情况，这时可以选择调小迭代计算的步长，每迭代 3-5 次输出一帧图像来保持系统的稳定。

值得一提的是，对于上面展示的代码 1，只需要将 7、8、12、13 行代码处的合力计算（三维向量加法）使用原子操作，就可以达成多线程并行计算的安全条件。这很好地体现了弹簧质点模型简单易用，计算高效的优点。



第三章 体积修正力

在第二章中我们构建了一个经典的弹簧质点模型，使用正六面体形状的弹簧质点网格作为模拟各向同性材质的基本单元。在这个初步的构建中，仅靠基于胡克定律的线性弹簧力难以模拟真实的可变形弹性体情况。不过弹簧质点模型易用且灵活，我们可以在不引入额外模型元素的前提下，在弹簧质点模型中增加另一种类型的力，作为线性弹簧力的补充。在弹性体模型发生变形时，这些额外的力会依据弹性体材料的可压缩性（泊松比）来控制模型的体积，确保模型的体积变化符合物理规律的同时不会影响变形的整体形状。

为了计算体积修正力，对于弹簧质点模型中的每一个立方体单元，我们考虑将弹性体变形产生的体积变化计算一个能量值：

$$E_{vol} = \frac{1}{2} k_{vol} \frac{(v - v_0)^2}{v_0} \quad (3-1)$$

其中 E_{vol} 表示体积变化的能量， k_{vol} 是根据弹性体材料不同而不同的一个系数， v 和 v_0 分别表示模型中每一个立方体单元的当前体积和不受力状态下的原有体积。

由此，我们可以推算出对应区域的能量密度：

$$\tilde{E}_{vol} = \frac{E_{vol}}{v_0} = \frac{1}{2} k_{vol} \left(\frac{v - v_0}{v_0} \right)^2 \quad (3-2)$$

微分能量：

$$\delta E_{vol} = k_{vol} \frac{v - v_0}{v_0} \delta v \quad (3-3)$$

由此，我们可以推导出弹簧质点模型中每个质点粒子 p_i 的体积修正力：

$$F_{p_i} = - \frac{\partial E_{vol}}{\partial p_i} = -k_{vol} \frac{v - v_0}{v_0} \frac{\partial v}{\partial p_i} \quad (3-4)$$

至此，为了求出体积修正力，还需要在每一次模拟的循环中计算每个立方体模型单元的体积。

3.1 立方体单元体积估算

如前文所述，我的弹簧质点模型中，最小单元是八个质点粒子作为顶点的正六面体。在初始的正六面体表面，12条棱边可以分为三组，其中每一组之间都相互平行并且也平行于三个世界坐标方向向量中的一个。

随着模拟计算的进行，模型发生变形，其中的每个单元都会相应地扭转变形，不再是完美的正六面体。计算这一单元体积的传统方法是将其分解成一系列规则的六面体和四面体，分别求出体积后再求和。出于实时模拟的设计需求，这样复杂且需要大量计算时间的方案是不可接受的。因此需要使用快速的近似方法，将六面体单元的三组棱边的平均值作为笛卡尔坐标系中的方向向量，表达如下：



$$\begin{cases} X = \frac{1}{4}(\overrightarrow{p_1 p_2} + \overrightarrow{p_4 p_3} + \overrightarrow{p_6 p_5} + \overrightarrow{p_7 p_8}) \\ Y = \frac{1}{4}(\overrightarrow{p_2 p_3} + \overrightarrow{p_1 p_4} + \overrightarrow{p_5 p_8} + \overrightarrow{p_6 p_7}) \\ Z = \frac{1}{4}(\overrightarrow{p_1 p_6} + \overrightarrow{p_3 p_8} + \overrightarrow{p_2 p_5} + \overrightarrow{p_4 p_7}) \end{cases} \quad (3-5)$$

然后我们可以依据平行六面体的体积公式近似求解变形后的不规则六面体单元的实时体积：

$$v = X \cdot (Y \times Z) = Y \cdot (Z \times X) = Z \cdot (X \times Y) \quad (3-6)$$

现在，终于所有的前置条件都得到了满足，对于组成一个六面体单元中的质点粒子 $p_i (i \in \{1, 2, \dots, 8\})$ ，其中， $\langle p_1, p_8 \rangle, \langle p_2, p_7 \rangle, \langle p_3, p_6 \rangle, \langle p_4, p_5 \rangle$ 互为体对角线的两端，我们可以求出作用于粒子端的体积修正力：

$$\begin{aligned} F_{p_1} &= -k_{vol} \frac{v - v_0}{4v_0} (- (Y \times Z) - (Z \times X) - (X \times Y)) \\ F_{p_2} &= -k_{vol} \frac{v - v_0}{4v_0} (+ (Y \times Z) - (Z \times X) - (X \times Y)) \\ F_{p_3} &= -k_{vol} \frac{v - v_0}{4v_0} (+ (Y \times Z) + (Z \times X) - (X \times Y)) \\ F_{p_4} &= -k_{vol} \frac{v - v_0}{4v_0} (- (Y \times Z) + (Z \times X) - (X \times Y)) \\ F_{p_5} &= -k_{vol} \frac{v - v_0}{4v_0} (+ (Y \times Z) - (Z \times X) + (X \times Y)) \\ F_{p_6} &= -k_{vol} \frac{v - v_0}{4v_0} (- (Y \times Z) - (Z \times X) + (X \times Y)) \\ F_{p_7} &= -k_{vol} \frac{v - v_0}{4v_0} (- (Y \times Z) + (Z \times X) + (X \times Y)) \\ F_{p_8} &= -k_{vol} \frac{v - v_0}{4v_0} (+ (Y \times Z) + (Z \times X) + (X \times Y)) \end{aligned} \quad (3-7)$$

第四章 碰撞检测

可变形弹性体的变形不会凭空产生，除了施加外力的情况，我们也应考虑到如何去模拟可变形弹性体与其他物体的交互反应，这就需要对我们的弹簧质点模型引入碰撞检测的功能。

4.1 与平面碰撞

物体与平面碰撞的情况很常见，其碰撞检测的计算也比较简单。对于三维空间中的任意一个平面：

$$f([x, y, z]) = 0 \quad (4-1)$$

对于模拟对象中的每一个质点粒子 i ， $[x_i, y_i, z_i]$ 是这个质点粒子当前时刻 t 的位置坐标， v_i 是这个粒子的运动速度，若有：

$$f([x_i, y_i, z_i]) \cdot f([x_i, y_i, z_i] + v_i) \leq 0 \quad (4-2)$$

则我们可以知道弹簧质点模型在 $[t, t+1]$ 的这段时间内与平面产生了碰撞。

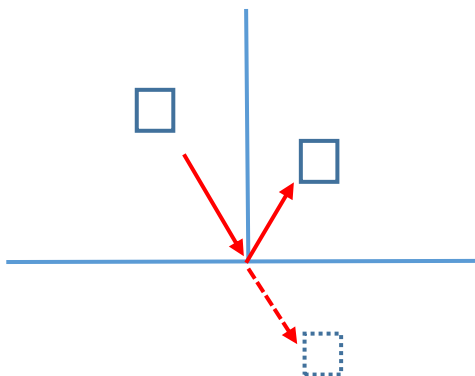


图 4-1 弹性体与刚性平面碰撞后反弹

如图 4-1 所示，右下方的虚线框表示未发生碰撞情况时粒子的预测位置 $[x_i, y_i, z_i] + v_i$ ，已知平面方程也就很容易可以算出平面的法向量，进而关于平面对称得到碰撞发生后，粒子应该处于的真正位置（右上方线框）。考虑到真实世界中的碰撞往往是非完全弹性碰撞，对物体的速度 v_i 关于平面做一次反射后，乘上一个衰减系数，使得系统能够最终回归静态。

当通过如上方法检测到碰撞且物体的速度 v_i 小于一个极小的界限时，将粒子重定位至碰撞点并且将速度归零，计算粒子此时对于平面的压力并由此计算出支持力和摩擦力，将这两个力附加到这个粒子所受的外力中并按算法 1 中所示进行模拟。如此，如果碰撞的平面是地面的话，几乎失去所有动能的物体会在地面稍作原地抖动，最终因为弹簧阻尼与地面施加给物体中粒子的摩擦力到达静止状态^[16]。

这个过程可以归纳为如下的伪代码：



算法 2 可变形弹性体与三维空间平面碰撞检测

```
1  FOR  $p_i$  in all Particles DO
2      computer position of  $p_i$  in next time tick  $t+1$ 
3      nextPositon  $\leftarrow$  currentPosition +  $v_i$ 
4      IF  $f(\text{nextPosition})f(\text{currentPosition}) \leq 0$ 
5          IF  $v_i = 0$ 
6               $N \leftarrow \text{downForce}$ 
7               $F_{\text{frac}} = \mu N$ 
8               $F_{\text{extern}} \leftarrow N + F_{\text{frac}}$ 
9          ELSE
10              $p_i.\text{position} \leftarrow \text{nextPosition}.\text{symmetry}$ 
11              $v_i \leftarrow v_i.\text{reflect}() \times \text{loss}$ 
12      END
13  END
```



第五章 结论

本文对于可变形弹性体建立了一个通用的三维弹簧质点模型，在这个模型中，使用了三种具有不同力学特性的弹簧构建正六面体单元，使用这一单元组合成任意形状的质量均匀分布的各向同性物体。弹簧质点模型在变形的时候往往兼具压缩，伸长，扭转，弯折这几种变形中的好几项，在此过程中，模型的体积会发生不小的改变。本文在线性的弹簧弹力之外引入了额外的以弹性体体积变化为参数的体积修正力，通过快速的不规则六面体体积估算算法，实时控制模拟对象的体积变化，进行比经典弹簧质点模型方法更精确的可变形弹性体模拟。

5.1 模拟效果

项目使用的模拟环境：

表5-1 模拟环境配置清单

项目	参数
CPU	Intel 4712mq 4C8T @2.3GHz
GPU	Nvidia GTX940M
RAM	8G @1600MHz
C++ tool set	Visual Studio 2019(v142)
OpenGL	Version 3.3

模拟程序的运行效果如下：



图 5-1 5*5*5 单元组成的弹性体的伸长，压缩与扭转

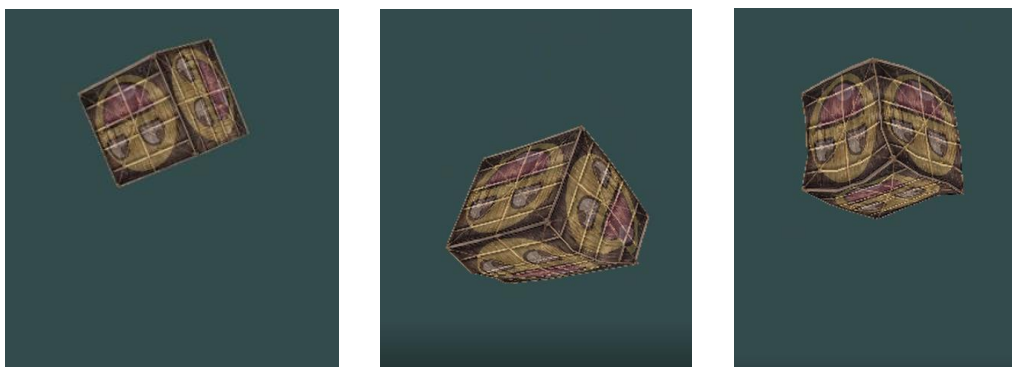


图 5-2 弹性体与平面的碰撞

图 5-2 展现了弹性体与平面（这里是地面）发生碰撞的模拟，从左到右分别是：

- (1) 弹性方块一边旋转一边斜抛而出



(2) 弹性方块接触地面，发生形变，一部分动能转化为弹性势能

(3) 弹性方块完成从地面弹起的过程，在空中仍有一些震荡

此后弹性方块会重复(2)，(3)中的过程，最终在地面滑行、震荡一段时间后，由于地面摩擦力与弹簧阻尼力的作用而静止于地面上。

图 5-1, 5-2 中，棱边弹簧与面对角线弹簧的弹性系数 $k_e = k_f = 100$ ，体对角线 $k_c = 300$ ，体积修正力系数 $k_{vol} = 0.1$ ， $5*5*5$ 解析度的弹性方块每一条边的边长为 1。

5.2 模拟效果分析

5.2.1 模拟效率

通过模拟三维物体中最简单的正方体进行模拟效率的测试，我们得到了如下的模拟时间与模拟精度的关系图标（测试时使用了 8 线程的 CPU 并行计算）：

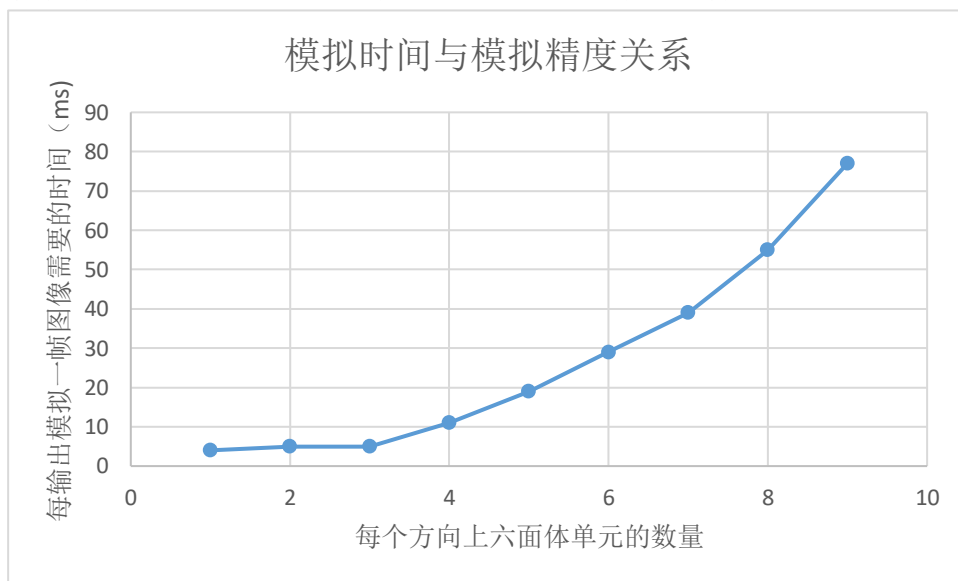


图 5-3 模拟时间与模拟精度关系

5.2.2 弹簧质点模型存在的不足

(1) 实时性

由图 5-3 我们可以看到即使弹簧质点模型已经是相对简单的模型，但是当每个方向上的六面体单元数量达到 7 时，每一次更新模型数据需要花费的时间正好为 40ms，即输出渲染视频的帧率为 24fps。因此如果需要更高精度的采样的话，目前的解决方案无法提供足够的实时性。

(2) 弹簧反向

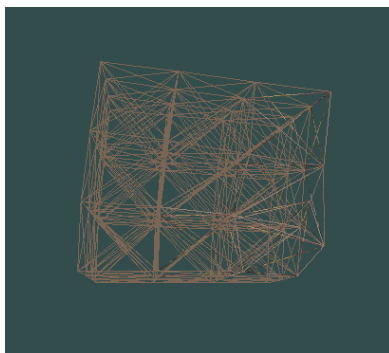


图 5-4 碰撞后发生永久变形的 $3*3*3$ 弹性方块



如图 5-4 所示, 图中是一个经过剧烈碰撞的弹簧质点模型, 它的右下角由于反弹后的粒子速度过快, 阻尼力与弹力不足以快速限制其位移。于是六面体单元的一个棱角发生溃缩, 弹簧到达新的稳定状态。在没有外力作用的情况下, 整个弹簧质点模型无法自发地回到初始形状。这种情况可以通过参数的调整得到一定程度上的避免, 例如设置较小的模拟步长或是较大的弹簧弹性系数, 尤其是弹簧密度较低的体对角线弹簧。但只要碰撞的速度够大就一定会触发这种期望之外的永久变形。

5.3 可行的改进

对比我的模拟结果与一些现存的解决方案, 有以下几个方向可以作为改进的目标:

(1) GPU 计算^[18]

即使是三维物体也只有表面需要进行渲染, 因此渲染的计算复杂度比物理模拟的计算复杂度小一个量级。考虑到弹簧质点模型的计算都可以很容易地并行化, 这就意味着 GPU 的算力未被充分利用, 可以考虑使用 GPU 加速物理计算。

(2) 复合四面体单元^[19]

在计算机图形的渲染中, 复杂模型最常用的就是三角形网络, 现有的弹簧质点模型中的最小模拟单位是六面体, 不可能精确地填充模型所用地三角形网络。但是六面体单元确实在模拟均质各向同性材料时具有很大的优势。目前能够想到更好的解决方案是在保留原有的六面体单元的基础上, 新增四面体单元。在进行弹性体建模时首先使用四面体单元紧密填满模型三角形网络的内部, 形成一个气泡状的模型。此外, 还需要一个能够将四面体单元与六面体单元完全贴合的算法, 填满上述气泡的内部。如此便能以最简单的两种模型单元去进行复杂模型的精确模拟。

(3) 局部细化^[20]

如果有一个均值各向同性材料组成的复杂模型, 为了精确的模拟, 也不必在全球都采用高密度的采样。对于一个实心的复杂模型来说, 只需要对外表的部分需要精细的变形的局部区域高密度采样就足以保证模拟的效果, 更低的全局采样数可以极大的降低物理计算的时间, 因为弹簧质点模型的模拟的时间复杂度大致在 $O(n^3)$ 的量级。



参考文献

- [1] Pham Q C, Vincent F, Clarysse P, et al. A FEM-based deformable model for the 3D segmentation and tracking of the heart in cardiac MRI[C]//ISPA 2001. Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis. In conjunction with 23rd International Conference on Information Technology Interfaces (IEEE Cat. IEEE, 2001: 250-254.
- [2] Nealen A, Müller M, Keiser R, et al. Physically based deformable models in computer graphics[C]//EUROGRAPHICS 2005 STAR-STATE OF THE ART REPORT. 2005.
- [3] Sifakis, E., & Barbic, J. (2012, August). FEM simulation of 3D deformable solids: a practitioner's guide to theory, discretization and model reduction. In *Acm siggraph 2012 courses* (p. 20). ACM.
- [4] Müller, M., Heidelberger, B., Teschner, M., & Gross, M. (2005, July). Meshless deformations based on shape matching. In *ACM transactions on graphics (TOG)* (Vol. 24, No. 3, pp. 471-478). ACM.
- [5] Faure F, Gilles B, Bousquet G, et al. Sparse meshless models of complex deformable solids[C]//ACM transactions on graphics (TOG). ACM, 2011, 30(4): 73.
- [6] Steinemann D, Otaduy M A, Gross M. Fast adaptive shape matching deformations[C]//Proceedings of the 2008 ACM SIGGRAPH/eurographics symposium on computer animation. Eurographics Association, 2008: 87-94.
- [7] Mollemans W, Schutyser F, Van Cleynenbreugel J, et al. Tetrahedral mass spring model for fast soft tissue deformation[C]//International Symposium on Surgery Simulation and Soft Tissue Modeling. Springer, Berlin, Heidelberg, 2003: 145-154.
- [8] Yao S, Zhou X, Hu G. Experimental study on negative effective mass in a 1D mass-spring system[J]. *New Journal of Physics*, 2008, 10(4): 043020.
- [9] Jensen J S. Phononic band gaps and vibrations in one-and two-dimensional mass-spring structures[J]. *Journal of Sound and Vibration*, 2003, 266(5): 1053-1078.
- [10] V. Baudet, M. Beuve, F. Jaillet, B. Shariat, and F. Zara. "Integrating Tensile Parameters in Hexahedral Mass-Spring System for Simulation". In: *WSCG'2009*. 2009. ISBN: 9788086943930.
- [11] Oliver Deussen, Leif Kobbelt, and Peter Tücke. "Using Simulated Annealing to Obtain Good Nodal Approximations of Deformable Bodies". In: *Computer Animation and Simulation '95: Proceedings of the Eurographics Workshop in Maastricht, The Netherlands, September 2-3, 1995*. Ed. by Demetri Terzopoulos and Daniel Thalmann. Vienna: Springer Vienna, 1995, pp. 30-43. ISBN: 978-3-7091-9435-5. DOI: 10.1007/978-3-7091-9435-5_3.
- [12] Gaizka San-Vicente, Iker Aguinaga, and Juan Tomas Celigueta. "Cubical Mass Spring Model Design Based on a Tensile Deformation Test and Nonlinear Material Model". In:



- IEEE Transactions on Visualization and Computer Graphics 18.2 (Feb. 2012), pp. 228–241.
ISSN: 1077-2626. DOI: 10.1109/TVCG.2011.32.
- [13] Richard Evelyn Donohue Bishop and DC Johnson. The Mechanics of Vibration. Cambridge University Press, 1979.
- [14] Bourguignon D, Cani M P. Controlling anisotropy in mass-spring systems[M]//Computer Animation and Simulation 2000. Springer, Vienna, 2000: 113-123.
- [15] David Baraff and Andrew Witkin. “Large Steps in Cloth Simulation”. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '98. New York, NY, USA: ACM, 1998, pp. 43–54. ISBN: 0-89791-999-8. DOI: 10.1145/280814.280821.
- [16] Teschner M, Kimmerle S, Heidelberger B, et al. Collision detection for deformable objects[C]//Computer graphics forum. 9600 Garsington Road, Oxford, OX4 2DQ, UK.: Blackwell Publishing Ltd., 2005, 24(1): 61-81.
- [17] Klaus-Jürgen Bathe. “Conserving energy and momentum in nonlinear dynamics: a simple implicit time integration scheme”. In: Computers & structures 85.7 (2007), pp. 437–445.
- [18] Burger B, Bettinghausen S, Radle M, et al. Real-time GPU-based ultrasound simulation using deformable mesh models[J]. IEEE transactions on medical imaging, 2013, 32(3): 609-618..
- [19] Mollemans W, Schutyser F, Van Cleynenbreugel J, et al. Tetrahedral mass spring model for fast soft tissue deformation[C]//International Symposium on Surgery Simulation and Soft Tissue Modeling. Springer, Berlin, Heidelberg, 2003: 145-154.
- [20] Hutchinson D, Preston M, Hewitt T. Adaptive refinement for mass/spring simulations[M]//Computer Animation and Simulation'96. Springer, Vienna, 1996: 31-45.



致谢

毕业设计的编程和论文写作都是极具挑战性的，很适合作为大学四年的学习生涯的总结。感谢这次毕业设计使我从毕业季的浮躁心态中解脱出来，可以有一段时间沉浸地回顾这四年所学到点点滴滴，并且又一次快速地学习掌握到了与时俱进的新技术，让我在一次次感慨终于入门之后体会到了茅塞顿开的快乐。

当然，最需要感谢的还是我的毕设导师——刘晓培教授，正是因为之前选修了刘晓培教授的计算机图形学课程，对图形学产生了一定的兴趣，因此才促使我选择了这个与图形学有关的毕业设计项目，甚至对择业方向有了一些影响。也感谢教授在我的毕设陷入瓶颈之时，多次向我指点可行的探索方向。

本人的学术水平有限，本篇论文难免有各种不足之处，恳请各位审阅者多多指导纠正！