

分类号：_____

密级：_____

UDC：_____

编号：_____

工学硕士学位论文

虚拟手术仿真中软体建模与碰撞检测技术的研究

硕士研究生： 黄乐

指导教师： 叶秀芬 教授

学科、专业： 控制科学与工程

论文主审人： 王科俊 教授

哈尔滨工程大学

2016 年 12 月

哈尔滨工程大学

学位论文原创性声明

本人郑重声明：本论文的所有工作，是在导师的指导下，由作者本人独立完成的。有关观点、方法、数据和文献的引用已在文中指出，并与参考文献相对应。除文中已注明引用的内容外，本论文不包含任何其他个人或集体已经公开发表的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

作者（签字）：

日期： 年 月 日

哈尔滨工程大学

学位论文授权使用声明

本人完全了解学校保护知识产权的有关规定，即研究生在校攻读学位期间论文工作的知识产权属于哈尔滨工程大学。哈尔滨工程大学有权保留并向国家有关部门或机构送交论文的复印件。本人允许哈尔滨工程大学将论文的部分或全部内容编入有关数据库进行检索，可采用影印、缩印或扫描等复制手段保存和汇编本学位论文，可以公布论文的全部内容。同时本人保证毕业后结合学位论文研究课题再撰写的论文一律注明作者第一署名单位为哈尔滨工程大学。涉密学位论文待解密后适用本声明。

本论文（☐在授予学位后即可 ☐在授予学位 12 个月后 ☐解密后）由哈尔滨工程大学送交有关部门进行保存、汇编等。

作者（签字）：

导师（签字）：

日期： 年 月 日

 年 月 日

摘 要

虚拟手术技术属于虚拟现实技术在医疗方面的分支,它是计算机科学、生物学、医学、力学等多学科的交叉研究领域,因其在手术训练、术前规划、术中导航等方面的突出作用,使其在现代医学发展中受到越来越多的重视。在虚拟手术技术的应用领域中,虚拟手术训练系统占比最大,它利用计算机建立一个虚拟环境来模拟真实世界中的手术训练场景,在这个虚拟环境中受训医护人员通过力交互设备控制手术器械对手术对象施行各种操作,虚拟环境为受训医护人员提供视觉和触觉等感官上的逼真反馈。这样的系统用于医务人员的培训,可以减少培训费用,提升培训效果。在这样的系统中最关键同时也是最难的技术就是软体的物理建模和虚拟环境中的碰撞检测技术,这两项技术左右着虚拟手术训练系统真实感和实时性两项重要性能指标,本文针对这两项技术展开研究。

首先,本文对软体物理建模技术进行了研究。传统虚拟手术训练系统中软体物理建模应用最广的是质点-弹簧(Mass-Spring Model, MSM)与有限元(Finite Element Model, FEM)等基于网格的方法。MSM具有实现简单、处理切割和缝合操作时,去除节点和增加节点较容易和实时性高的优点;但是其存在大变形失真、稳定性不好的缺点。FEM拥有仿真形变逼真度高的优点;但存在实现复杂度高、计算量大、实时性不佳的缺点。针对上述传统方法存在的问题,本文将基于形状匹配的无网格方法应用在软体的物理建模中,该方法不需要网格划分,粒子之间不需要连接信息、简单易实现,在软体变形实时性、精度、大变形、稳定性方面都能得到满意的效果。

其次,本文对如何提升碰撞检测的性能进行了一定的探索。根据不同包围体与虚拟手术仿真系统中发生碰撞情况的特性,在碰撞预检测阶段提出使用 Sphere-AABB 复合层次包围盒法,在碰撞详细检测阶段对于海燕等人提出的基于投影降维的用于三角形间的碰撞检测的算法改进后进行应用,实验结果证明在相交三角形数量任意数值下,本文算法相比传统单一层次包围盒结合 Möller 三角形相交测试的算法都有更高的检测效率、精确度和稳定性。

最后,运用上面的物理建模和碰撞检测技术搭建一个简易肝脏虚拟手术训练系统进行仿真实验,实验中场景 fps 值均值在 50 以上,视觉反馈流畅,满足系统实时性的要求;肝脏的按压、提拉变形效果接近于真实情况,没有出现塌陷和超弹性现象,同时手术器械没有穿越和远离软体组织,满足系统逼真性的要求。从而验证了本文基于形状匹配的无网格物理建模方法、复合层次包围盒碰撞检测方法及改进三角形间相交测试算法的有效性,实用性,能够应用到实际的虚拟手术训练系统中。

关键词：虚拟手术；三维重建；形状匹配；无网格；基于投影；复合层次包围盒

ABSTRACT

Virtual surgery technology is an important application of virtual reality technology in the medical field, it is the cross-over study in the field of computer science, biology, medicine, mechanics and other disciplines. Because of its prominent role in the training of surgery, preoperative planning, it gets more and more attention in the development of modern medicine. Virtual surgery training system is an important application branch of virtual surgical technique, which uses a computer to create a virtual environment of real surgery training scenarios, the medical trainer operates the surgical objects with surgical instruments controlled by the interactive device in the scene, virtual environments provide a realistic feedback to the trainer in the visual and tactile sensory to make the medical trainer feel like training in reality. Such systems can reduce training costs and improve training effectiveness in training of medical personnel. In such a system, the critical and difficult technology are the physical modeling of soft tissue and collision detection, these two technologies are the key factors affecting the real-time and realism of virtual surgical simulation training systems, this paper did depth research in both of them.

Firstly, we do some research in the physical modeling technology of soft tissue. Mass - spring model and the finite element model are the popular model used in traditional virtual surgery training system. Mass - spring model does not require continuous physical parameterization, easy to implement; when dealing with cutting and sewing operation, the removal and increase of nodes are easy; but its deformation accuracy and stability are poor. Finite element model has the advantage of high precision; but there is a huge amount of computation and have difficulty to handle the topology structure when it changes. For the above-mentioned problems, this paper will apply meshless method based on shape matching to build the physical model of the soft tissue, this method does not require meshing, there is no connection between the particle, it is simple and easy to implement, unconditionally stable, and can achieve satisfactory results on the deformation speed and accuracy aspects.

Secondly, this paper research in how to improve the efficiency of collision detection too. According to the characteristics of different bounding boxes and objects in the scene, this paper proposes to use Sphere-AABB complex bounding box in collision detection which overcomes the shortcomings that the accuracy and timeliness cannot be solved well when using a single defect. At the same time this paper improves the collision detection algorithm for base primitive (triangle patch) based on projection proposed by Yu Haiyan to enhance the

efficiency and accuracy of collision detection.

Finally, to build a simple liver virtual surgery training system by use of the above physical modeling and collision detection technology to do simulation experiments. In the scenario the mean value of fps is over 50 and visual feedback is smooth, which meets the real-time requirements. The deformation of pushing and pulling of the liver are closer to the real deformation effects, there is no collapse or super-elastic phenomenon, at the same time surgical instruments are not far from or cross the soft tissue, which meets the requirements of verisimilitude. So the technologiis of physical modeling method based on shape matching meshless theory and collision detection method based on complex hierarchical bounding box tree are effective and practical, they can be applied to the actual virtual surgery training system.

Key words: virtual surgery; three-dimensional reconstruction; shape matching; meshless; projection; complex hierarchical bounding boxes

目 录

| | |
|-------------------------|----|
| 第 1 章 绪论 | 1 |
| 1.1 研究背景及意义 | 1 |
| 1.2 国内外研究现状 | 4 |
| 1.2.1 软体物理建模技术的研究现状 | 5 |
| 1.2.2 碰撞检测技术的研究现状 | 9 |
| 1.3 虚拟手术仿真系统框架 | 12 |
| 1.4 本文的研究内容及结构安排 | 13 |
| 第 2 章 虚拟手术仿真中软体的几何建模方法 | 15 |
| 2.1 引言 | 15 |
| 2.2 软体几何建模方法分类 | 15 |
| 2.3 利用医学图像数据三维重建法 | 15 |
| 2.3.1 三维重建流程 | 15 |
| 2.3.2 面绘制三维重建法 | 16 |
| 2.4 利用三维建模软件绘制法 | 20 |
| 2.5 软体几何模型的建立 | 21 |
| 2.5.1 Amira 工具软件 | 22 |
| 2.5.2 利用 Amira 软件建立几何模型 | 22 |
| 2.6 本章小结 | 24 |
| 第 3 章 虚拟手术仿真中软体的物理建模方法 | 25 |
| 3.1 引言 | 25 |
| 3.2 基于网格的常用软体物理建模方法 | 25 |
| 3.2.1 质点-弹簧模型 | 25 |
| 3.3.2 有限元模型 | 27 |
| 3.4 基于形状匹配的无网格方法 | 33 |
| 3.4.1 算法基本原理 | 33 |
| 3.4.2 形状匹配原理 | 33 |
| 3.4.3 数值积分 | 35 |
| 3.4.4 算法扩展 | 36 |
| 3.4.5 实验 | 38 |
| 3.5 物理建模方法对比实验 | 38 |
| 3.6 本章小结 | 42 |

| | |
|------------------------------|----|
| 第 4 章 虚拟手术仿真中的碰撞检测算法..... | 43 |
| 4.1 引言..... | 43 |
| 4.2 碰撞检测算法概述..... | 43 |
| 4.3 复合层次包围盒..... | 45 |
| 4.3.1 复合层次包围盒树概述..... | 45 |
| 4.3.2 复合层次包围盒树算法流程..... | 45 |
| 4.3.3 复合层次包围盒的构建..... | 47 |
| 4.4 三角形间的相交测试..... | 56 |
| 4.4.1 常用三角形间相交测试算法..... | 57 |
| 4.4.2 基于投影降维的三角形间相交测试算法..... | 58 |
| 4.5 碰撞检测算法对比实验..... | 66 |
| 4.6 本章小结..... | 67 |
| 第 5 章 仿真验证实验..... | 69 |
| 5.1 引言..... | 69 |
| 5.2 软件环境概述..... | 69 |
| 5.2.1 OpenGL 概述..... | 69 |
| 5.2.2 OpenGL 工作原理..... | 69 |
| 5.3 虚拟手术系统的实现..... | 71 |
| 5.3.1 系统框架..... | 71 |
| 5.3.2 系统中重要的数据结构..... | 73 |
| 5.4 实验结果..... | 73 |
| 5.4.1 实验结果截图..... | 73 |
| 5.4.2 结果分析..... | 74 |
| 5.5 本章小结..... | 74 |
| 结 论..... | 75 |
| 参考文献..... | 77 |
| 攻读硕士学位期间发表的论文和取得的科研成果..... | 81 |
| 致 谢..... | 83 |

第1章 绪论

1.1 研究背景及意义

虚拟现实（Virtual Reality，简称VR）技术^[1]是一门正在被越来越多的人所熟知并且高速发展的技术，其利用计算机软硬件、软件和网络生成一个三维立体环境，借助一些穿戴式传感器设备（图1.1）人们可以与之交互，人可以对环境中的物体施加操作，同时虚拟环境提供给人视觉、听觉、触觉、嗅觉、味觉等感官的模拟，让人有种身临其境之感。



(a) 头盔式立体显示器



(b) 数据手套

图1.1 虚拟现实传感器设备

上世纪90年代以来，VR技术一路高歌猛进，在很多领域得到应用，如医疗、军事、游戏娱乐、古迹文物复原、城市规划、航天、虚拟工业仿真、教育等^[2]，如图1.2所示。



(a) 医疗应用



(b) 军事应用



(c) 娱乐应用



(d) 古迹文物复原

图1.2 虚拟现实技术在各个领域的应用

在过去的十多年间，VR技术在医疗领域的应用得到了稳健的发展^[3]，主要体现在以下方向：

1. 虚拟解剖

解剖是医学教育的基础，但到目前为止，大多数的解剖课程依然使用教科书和供解剖的尸体，而学生们利用尸体进行练习解剖的机会也很少。为了解决这个问题，在 20 世纪 90 年代，美国国立医学图书馆和科罗拉多大学通过将一个男性和女性尸体切成 1 毫米以下的薄片，然后对每个薄片进行光学照片采集和 CT、核磁共振断层扫描，获得了两个解剖数据库。然后基于这个数据库，加利福尼亚大学开发了 Anatomic VisualizerR 虚拟解剖室系统，学生们通过佩戴立体眼镜可以观看生成的立体交互模型，同时可以操作生成横截面图像、测量并确定组织结构、改变器官的透明度（为了观察被遮挡的后面的器官或部位）等操作，来研究人体器官（颅骨、耳朵、胸腔、腹部和神经系统等）的组织构造。通过这个系统可以观察到通过尸体解剖很难看到的细小的神经和血管，并且可以对某个部分可以随意生成切片进行观察，如图 1.3 所示。

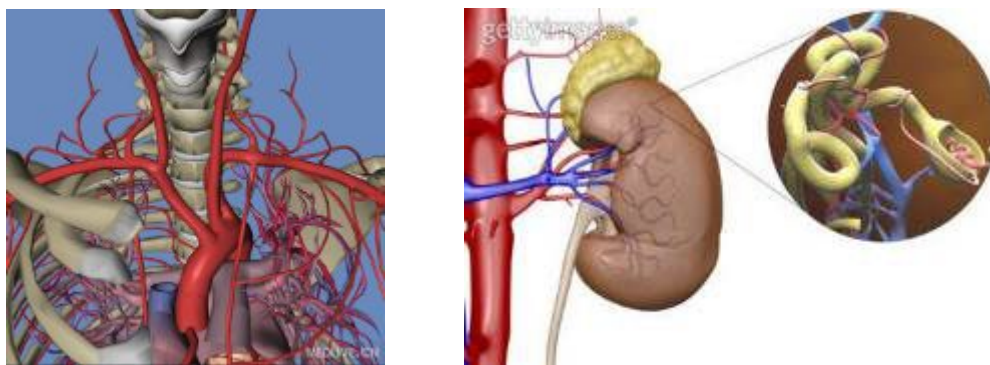


图 1.3 虚拟解剖

2. 疾病诊断

在疾病诊断方向的应用代表可以说是内窥镜检查。内窥镜检查是侵入式的，有一定的危险性，如果内窥镜划破腔壁会导致受伤、产生并发症甚至死亡，如果能有替代内窥镜检查的手段将会给病人带来很大的福音。

纽约州立大学开发了一个非侵入式虚拟结肠镜检查系统（图 1.4），检查过程只需病人吃下一定剂量的对照剂，然后做结肠 CT 扫描，将获得的 CT 图像进行图像分割，然后通过三维重构获得其立体模型，使用虚拟相机在模型中漫游进行结肠疾病检查。临床实验表明，对于直径大于 5mm 的息肉，虚拟结肠镜与光学结肠镜探测率相同，而对于 3mm 左右的息肉，虚拟结肠镜可以发现，但光学结肠镜很难做到。

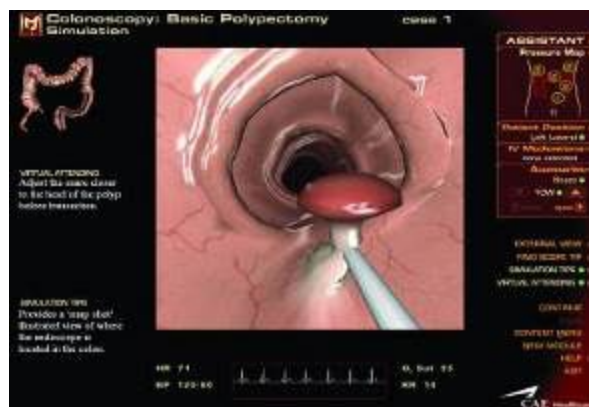


图 1.4 非侵入式虚拟结肠镜检查系统

3. 医务人员培训

现阶段医务人员的疾病诊断训练、手术训练等主要还是在塑胶模型、尸体、动物和病人身上。塑胶模型不能做出病人的反应，缺乏真实感，也无法记录操作人员的失误；尸体跟活体有些差别，同时数量少也无法重复利用；动物与人生理结构差别较大，同时也会遭到动物权益保护团体的谴责；在病人身上会使病人受到不必要的伤害，甚至导致严重的医疗事故，因此，虚拟医学训练系统由此诞生，如图 1.5 所示的支气管镜检仿真器，图 1.6 所示的静脉注射系统，图 1.7 所示的虚拟膝关节镜手术系统等。支气管镜检仿真器由可弯曲光纤内窥镜和 PC 机组成，内窥镜提供触觉反馈，PC 机提供具有真实感的腔道立体解剖模型，模型可以对受训人员的操作做出相应反应，例如，虚拟内窥镜碰到腔壁，腔道会发生变形并且病人会“咳嗽”，同时受训人员会感受到反馈的阻力。系统还可以模拟出血、腔道分泌物遮挡视线、多种支气管疾病病理特征，在训练中系统会记录受训人员完成特定操作花费的时间、出现的失误数、漏检腔道数等。将这些记录与专家完成这一过程的基准结果对比，有助于提高受训人员的培训效果和速度。实验证明该系统对提高医务人员内窥镜检查的速度、灵活性、准确性大有帮助。静脉注射仿真器可以模拟穿刺血管时的微小阻力，刺入静脉后针管中有血液回流，如果用力过大刺穿静脉在触觉上会感到突然的变化，同时手臂会出现血液溢出形成积血於肿，还会发出警告，病人开始抱怨等反应。系统可以模拟多种类型的病人，从新生婴儿到成年人，从肥胖者到老人等。



图 1.5 支气管镜检仿真器



图 1.6 静脉注射仿真系统



图 1.7 虚拟膝关节镜手术系统

在上述几个方面的应用中，医务人员培训方面的应用最为广泛，其中虚拟手术仿真系统是研究最多，困难也是最大的。

1.2 国内外研究现状

国外对虚拟手术仿真系统相关技术的研究相对较早，成果比较丰富，已经有很多实用系统被开发出来。除上文提到的成果外，现阶段还有许多，法国国家信息和自动化研究所（INRIA）于 2010 年做出一个虚拟肝脏切割手术仿真系统^[4]，如图 1.8 所示，软件采用有限元方法进行物理建模。

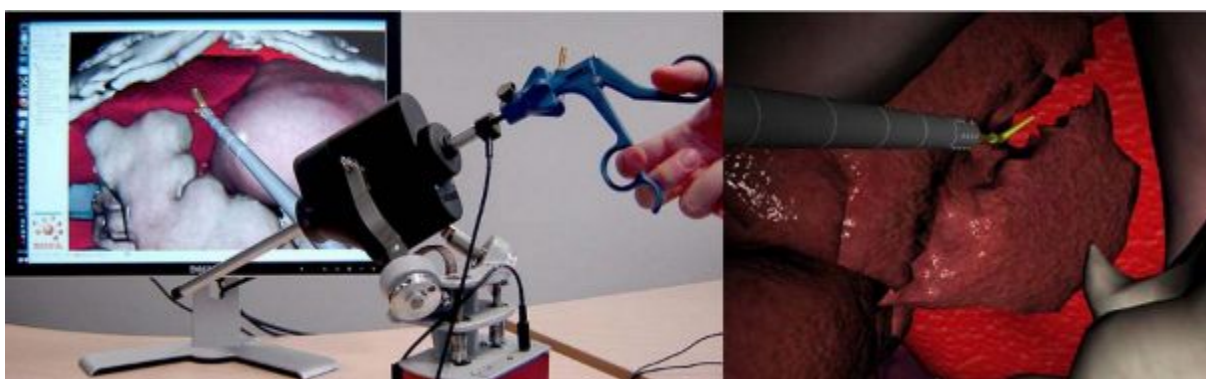


图 1.8 基于有限元物理形变模型的虚拟肝脏切割手术系统

斯坦福大学的 Brown B 等人^[5]做出了针对显微镜下1mm微细血管进行缝合操作的虚拟手术系统，其中血管的形变、缝合效果非常逼真，如图 1.9 所示。

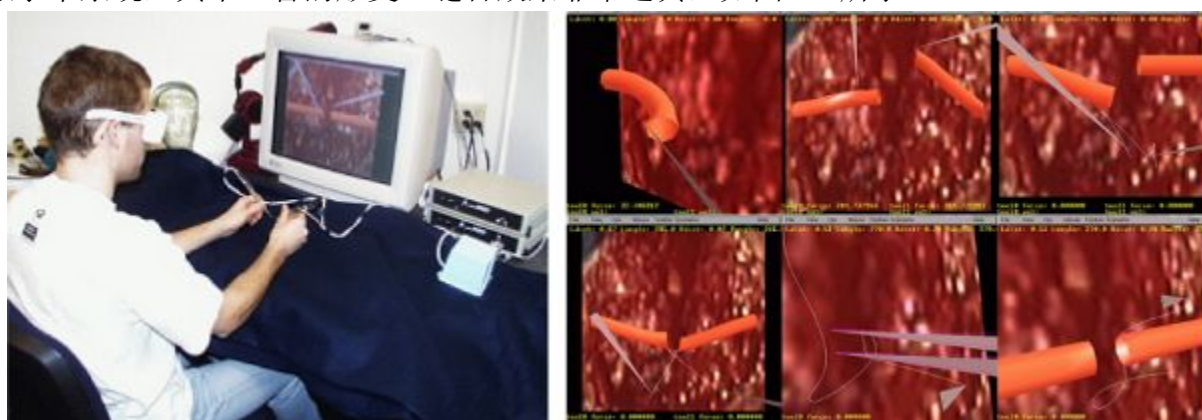


图 1.9 1mm 细小血管和神经缝合的虚拟手术系统

我国对虚拟手术仿真开始关注的时间比较晚，但也做出了很多成果。由第一军医大学、中科院计算技术研究所、首都医科大学、华中科技大学等多家单位合作，于 2003 年建立了国内第一例解剖虚拟人数据库，完成了国人自己的“数字化虚拟人”第一阶段——虚拟几何人阶段^[6]。2001 年，中国解放军总医院研制出鼻内镜系统，属国内首例^[7]。同时清华大学在虚拟手术系统交互实时性的改进，浙江大学在心脏建模，国防科技大学在虚拟膝关节镜手术^[8]，以及中国解放军总医院在鼻内窥镜手术等方面都做出了一定的成就。

1.2.1 软体物理建模技术的研究现状

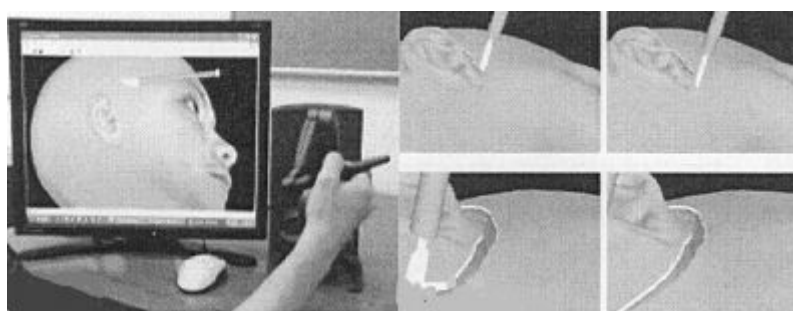
软体物理建模技术即软体的变形模拟技术，根据不同的分类标准可以进行不同的分类，依据是否基于网格可以分为基于网格的方法和基于无网格的方法。依据是否有物理基础可以分为基于几何和基于物理的方法。

基于网格的方法需要用到预先划分的网格信息，无网格方法把对象看做一个离散的粒子系统，不需要粒子间的连接信息。几何方法不考虑软体本身的生物力学特性、软体的受力、质量属性等，只是通过改变几何形状来模拟软体的变形，具有计算少，实时性

高的优点，但缺乏物理基础，形变仿真逼真度稍有欠缺。物理方法充分考虑了软体的力学特性，通过一种物理模型去模拟软体，考虑变形过程中软体的受力情况，利用了物理规律去模拟软体的变形，变形效果更逼真，但计算复杂度高^[6]。

1. 基于网格的方法

从几何角度对软体进行形变仿真的方法由 Sederberg^[9]最早提出，方法叫做自由形变（Free-form deformation，简称 FFD）方法，我们知道 B 样条曲线、贝塞尔曲线和非均匀有理 B 样条等可以用来描述物体的外部轮廓，而这些曲线、曲面的形状是通过相应的特征点控制的，FFD 方法就是直接控制这些特征点的位置从而控制物体形状变化的。利用 FFD 技术实现的头部手术切割模拟如图 1.10。之后，Gibson 等^[10]提出了 Chain-Mail 模型，如图 1.11 所示，它将对象离散化，表示为环环相连的链状结构，一个环运动会带动相连环。



(a) 虚拟手术系统

(b) 切割模拟

图 1.10 基于 FFD 几何方法的虚拟手术系统

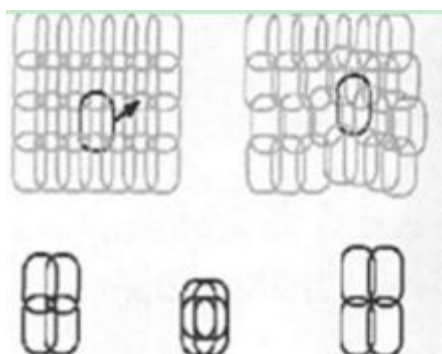


图 1.11 Chain-Mail 结构

基于物理的软体变形模拟方法主要有质点-弹簧模型、有限元模型、边界元模型、长单元模型、张量-质点模型等^[11]。

(1) 质点-弹簧模型

由 Terzopoulos^[12]提出的一个离散模型，它将连续的组织离散化为一系列的点，每个点都有质量，点与点之间通过弹簧连接。质点受到外力、其他弹簧给它的弹力、阻尼力的作用，变形通过牛顿运动学第二定律来计算。

因为其简单易求解，在手术仿真中应用很广。Waters K.^[13]等人使用质点-弹簧/阻

尼模型实现了人脸表情仿真,用非线性弹簧表现脸部肌肉、皮肤的力学特性,人脸几何面模型使用矩形拓扑结构。Breen 和 House^[14]将其用于模拟柔软织物的形变效果。

基本的质点-弹簧模型应用有限,研究人员在其基本形式基础上做了很多改进。Etheredge^[15]将并行运算应用到质点-弹簧模型中,极大的提高了形变模拟效率。鲍春波等人^[16]针对基本质点-弹簧模型会出现“超弹性”变形现象的问题,往基本模型中加入“体弹簧”进行了修正。Choi 等^[17]提出一种基于力传导的模型,外力从外到内逐层传导入软体内,从而产生变形。

质点-弹簧物理模型构建简单,计算量小,不必连续参数化;但是其各项参数需要人工精心调整,其变形精度和稳定性不足,大变形情况有失真的可能,一般适用于实时性要求较高、精度要求一般、中小变形的环境中。

(2) 有限元模型

有限元模型(Finite element model, FEM)方法是求解弹性力学问题的经典方法,它的基本思想是将连续的求解离散化为一个个小的单元上的求解,用简单的插值函数表示全域上的复杂函数,这些单元按一定方式联系在一起,单元交界处满足某些连续性条件。利用有限元模型可以精确划分复杂几何模型,因为一个模型可以划分为不同单元形状然后按不同的联结方式组合起来,从而使它模拟变形精度很高。

有限元法的突出优点是具有弹性力学基础,形变逼真度高,不存在失真情况,但其计算量庞大,实时性不佳。针对其效率不佳的问题,研究人员做了很多努力并将集合了其相应成果的有限元方法应用到了相应的虚拟手术系统中。常用的方法有预计算、凝聚、多尺度和 GPU 加速^[18]。预计算技术,弹性力学中弹性体在小变形范围内节点位移具有线性性质, Cotin 等人^[19]利用这一原理,对某些节点施加单位力,事先求出其他相关节点在此影响下的位移,仿真时根据受力情况利用事先计算的相关结果叠加即可得到要求的位移。Cotin 将此成果应用到肝脏的变形模拟中,实验表明计算效率甚至高于质点弹簧模型,精度也很好。凝聚技术,物体的变形通过其表面节点来表现出来, Bro-Nielsen 等利用凝聚技术把刚度矩阵中与内部节点相关的部分都约简掉,使得系统方程中只留下了与表面节点相关的部分,大大减少了计算复杂度。多尺度技术,在实际操作中手术区域一般只是软体对象的一小部分,根据这一点,孙剑等^[20]把对象划分为局部手术区和非手术区,对不同的区域进行不同的单元划分建立混合有限元模型,这样既保证了局部区域的精度,又减少了总的计算量。根据这一思想还出现了手术区域利用有限元模型,非手术区域采用质点-弹簧模型的混合模型,在实际系统中都有不错的表现。GPU 加速技术, Dick 等人^[21]对六面体网格的有限元模型进行了 GPU 加速试验,计算效率提升巨大,针对十万数量级的单元数,达到 11fps 的效率。

鉴于有限元法形变逼真度很高,但其计算量庞大,实时性不佳,处理拓扑逻辑结构变化困难的特点,一般将其应用在变形精度要求严格、实时性要求不高、局部小范围内

变形的环境中。

主流的基于网格的物理驱动软体形变模拟方法就是质点弹簧模型和有限元模型，其他的方法都是在二者基础上的变形。如边界元模型^[6] (Boundary Element Model, BEM)、长单元模型^[6] (Long Element Model, LEM)。

2. 基于无网格的方法

基于网格的方法研究的很多也相对成熟，但由于网格的存在使其存在固有缺陷：基于网格的方法都要进行网格划分，这是一个耗时且复杂的步骤，划分得到的网格的质量直接影响软体变形仿真效果；当组织变形过大或被切割时，拓扑结构发生变化，网格需要重新划分；对于虚拟手术中像血液等流体的模拟基于网格的模型也无能为力。此时，无网格方法应运而生^[22]。

无网格法是在建立问题域的系统代数方程时，不需要利用预先定义的网格信息进行问题域离散的方法^[23]。基于几何驱动无网格方法主要有粒子系统、基于点动画、填充球模型和基于形状匹配法等。基于物理驱动无网格方法主要有光滑粒子流体动力学法、Lennard-Jones 势能法^[24]、有限球配点法 (the point collocation-based method of finite spheres, PCMFS)^[25]、无单元伽辽金方法 (EFG)、DEM (Diffuse Element Method) 方法等。

(1) 粒子系统

粒子系统^[26]的组成和原理非常简单，它由很多拥有自身属性，并且某些属性随时间流逝而变化的粒子组成。这些粒子的属性一般包含质量、位置、速度、大小、颜色和生命周期等。每个粒子的属性都是根据某个规则随机产生的，规则由要模拟的对象决定，每个粒子都要经历诞生、属性更新、消亡三个阶段，这三个阶段的时间长度就是粒子的生命周期。它常被用来模拟像血流、爆炸、焰火、云等没有固定形态的对象。

(2) 基于形状匹配的无网格方法

该方法将对象离散为一个点云，点与点不需要连接信息。因为每个点的初始位置 and 实际位置间的对应关系是确定的，不同于标准的形状匹配，这里就不需要确定两种形状间的对应关系，只需求出初始位置 and 实际位置点云的最优变换，对求出的最优变换做些改动用于求出点云的目标位置，将每个点的目标位置用于积分中求取每个时间步点的位置，利用点的位置控制对象的外形。该方法实现简单，计算效率高，无条件稳定。

(3) PCMFS 无网格方法

该方法用到了连续介质理论和离散质点系统理论，采用的是二者混合的模型。可用于线性和非线性软体拓扑结构变化与大变形情形，具有较好的实时性与视觉逼真度。

基于无网格的方法在切割变形等拓扑结构改变或大变形的应用场合表现比基于网格的方法要好，但其在材料描述能力方面、视觉渲染速度和计算效率等方面仍需提高。

1.2.2 碰撞检测技术的研究现状

仿真系统的目的在于模拟真实的世界，所以其中的现象应该遵循真实世界中的物理规律。像手术器械接触到组织器官时，器械穿越了组织；不同的组织器官或是自身的不同部分接触时，彼此发生穿越的情形是不允许出现的。碰撞检测技术正是为了保证这一点而产生的。

国内外学者对碰撞检测技术已研究多年，开发出了许多碰撞检测系统，如 SOLID, RAPID, I-COLLIDE, V-COLLIDE, QucikCD 等。现有的碰撞检测技术在分类上大致可以做如图 1.12 所示的划分。静态检测方法主要针对位置、形状不随时间变动的物体，一般精度要求高，实时性无要求^[27]。对于随时间移动或变形的物体需要使用动态检测方法，一种实现方法是将连续时间离散为时间点，在时间点上执行静态碰撞检测，另一种实现方法是连续碰撞检测。前者运算速度快，但有漏检和穿刺现象产生，为了防止明显的穿刺现象出现，仿真中一般使用较小时间步长；后者实时性太差，无法应用在实际系统中。在动态检测的实现方法中，离散方法虽不完美，但其可以达到系统实时性的指标，因此离散碰撞检测方法成为现阶段研究的核心^[28]。



图 1.12 碰撞检测方法分类

离散碰撞检测方法可继续划分为基于图形和基于图像的。基于图形的又有很多分类，图中只列出了两种典型方法。

1. 基于图形的方法

(1) 空间分割法

其基本思想是将场景空间按照一定的规则划为小的单元，单元记录包含在其中的对象。常用的剖分规则有均匀剖分、八叉树剖分、Kd-树剖分和 BSP 树剖分^[29]，如图 1.13 所示。均匀剖分与对象复杂度无关，其关键问题是单元格大小的确定，针对这个问题李焱等人^[30]提出一种基于对象分布密度计算单元格大小的方法，在对象数量较多并且分布较均匀的环境下表现很好。其他剖分方法都与对象相关。判断两对象间或对象自身不同部分间是否发生交叠只检查同一个单元内或相邻单元内对象的两两相交情况即可。此方法一般用在对象分布比较均匀且稀疏的静态场景中。

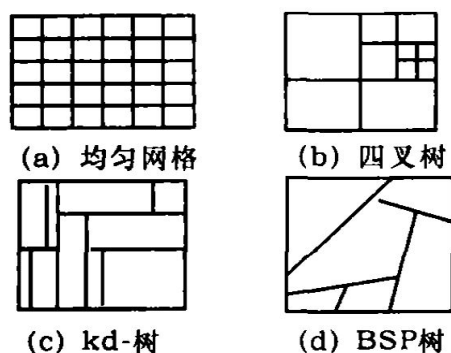


图 1.13 空间分割法剖分规则

(2) 掠扫和裁剪法 (Sweep and Prune)

此法由 Cohen 等人^[31]于 1995 年提出, 基本思想是如果两包围盒相交, 则它们在三个坐标轴上的投影肯定都相交, 在每个坐标轴上建立一个存储包围盒在轴上投影端点的表, 然后对这些端点进行排序, 根据三个轴上的投影端点信息判断投影的相交情况, 然后反推包围盒的相交情况。为了适应多对象运动的场景, Coming 等人^[32]提出动态掠扫与裁剪法 (Kinetic Sweep and Prune, KSP), 如图 1.14 所示。此法主要用于包围盒初期检测。

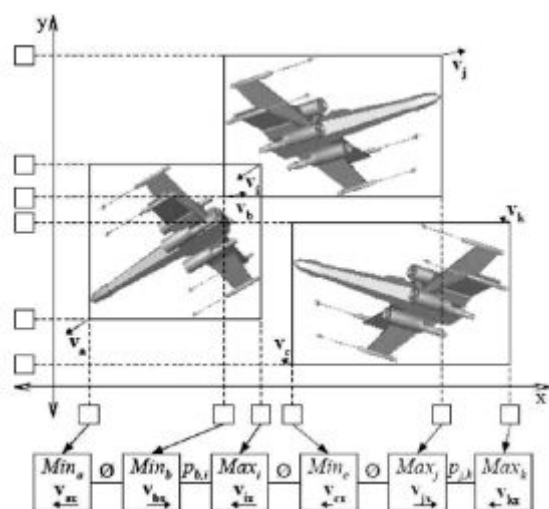
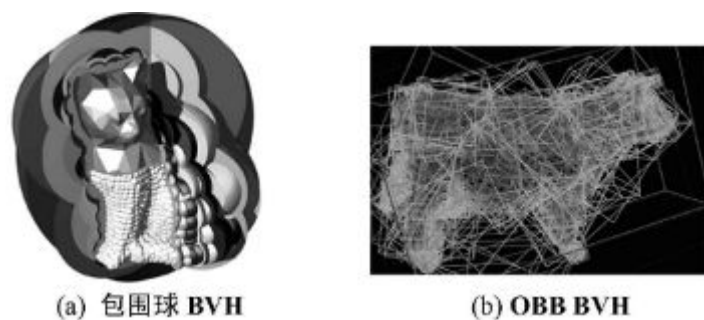
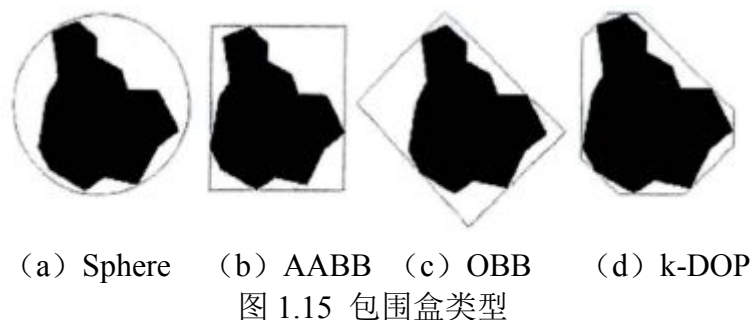


图 1.14 KSP 算法坐标轴投影端点排序

(3) 层次包围盒法 (Bounding Volume Hierarchy, BVH)

此方法在碰撞检测中运用较多, 它是包围盒与树形数据结构的结合, 包围盒是指几何特性简易且体积略大的用以包裹结构复杂的待检测对象的几何体, 常用的有包围球 (Sphere)、轴向包围盒 (Aligned Axis Bounding Box, AABB)、方向包围盒 (Oriented Bounding Box, OBB) 和 k-DOP 包围盒 (Discrete Oriented Polytope)^[33], 如图 1.15 所示。先用一个包围盒包围整个对象作为根节点, 然后将对象划分为两个或多个部分 (取决于使用的树的度数, 一般使用二叉树) 用同样的包围盒包围子部分作为上一节点的子节点, 重复上面的过程直到满足划分要求, 从而构建一个逐步逼近待检测对象的层次包

围盒，如图 1.16。判断两对象是否碰撞时，先遍历各自的层次包围盒，执行包围盒间的相交测试，如果包围盒相交且两棵树都遍历到了叶子节点，则通过叶子节点内包围的基本图元相交测试得到详细相交信息，其他情况下对象不相交。由于包围盒的相交测试容易，并且使用了树形层次结构，对于不相交的基本图元可以进行大量的前期排除，从而减少计算量提升碰撞检测效率。



除了上面应用最多的算法，还有很多基于其他思想的算法：

1993 年 Lin-Canny 提出最邻近特征碰撞检测法，利用了对象特征的 Voronoi 域的相关性质进行检测。Brain 等人^[34]于 1998 年在其基础上做了重大改进，提出 V-clip 算法，增加算法健壮性，能够有效控制渗透。

1997 年 Gilbert、Johnson、Keerthi 三人提出 GJK 算法^[35]，该方法是基于单纯形的算法，在判断出两对象碰撞的情况下还能给出刺穿的深度。Cameron 等人^[36]于 1999 年借鉴 Hill Climbing 思想对 GJK 算法做了改进，提升了算法效率。

2004 年 Guy 等人^[37]提出随机碰撞检测方法，该方法灵活性强、检测速度可控。

2006 年金汉均等人^[38]利用遗传算法求解碰撞检测中的最短距离问题。

2. 基于图像的方法

基于图像的碰撞检测方法原理一般比较简单，三维对象最终绘制到屏幕上都是像素，在这个转化过程中在各种不同的缓冲区存放了多种必要的信息，如深度，颜色等，通过这些信息来判别对象间的相交状况。其效率与精度与转化后的图像的分辨率有关，与对象的复杂度，运动和变形情况关系不大，适用于运动软体对象的碰撞检测，但其不能提供精确的碰撞信息。

Shinya 等人^[39]于 1991 年首先提出借助图形硬件进行碰撞检测的方法,但由于保存深度的空间复杂度和读取深度信息的时间复杂度都比较高,在当时并不实用。

2001 年, Hoff 等人^[40]提出了基于几何和基于图像相结合的方法,利用几何方法排除掉不相交的区域,对潜在相交区域利用基于图像的方法判断。

虽然将碰撞检测算法做了如上分类,但在实际使用中并不是只使用其中一种,有时会结合使用多种碰撞检测的思想,如 Bergen 等人于 2004 年开发了 SOLID (Software Library for Interference Detection) 库,其中就结合了 GJK 思想、掠扫和裁剪、包围盒等。

1.3 虚拟手术仿真系统框架

一个虚拟手术系统的功能大体可描述为:人借助外接力交互设备操纵系统中的虚拟手术工具对虚拟人体器官或组织进行操作(按压、提拉、针刺、切割、缝合等),当器械接触到器官并进行相应操作时系统要通过显示装置和力交互设备提供给人视觉和力触觉反馈,让人感觉如同在真实环境中操作。为了实现这一功能,构建如图所示的架构。

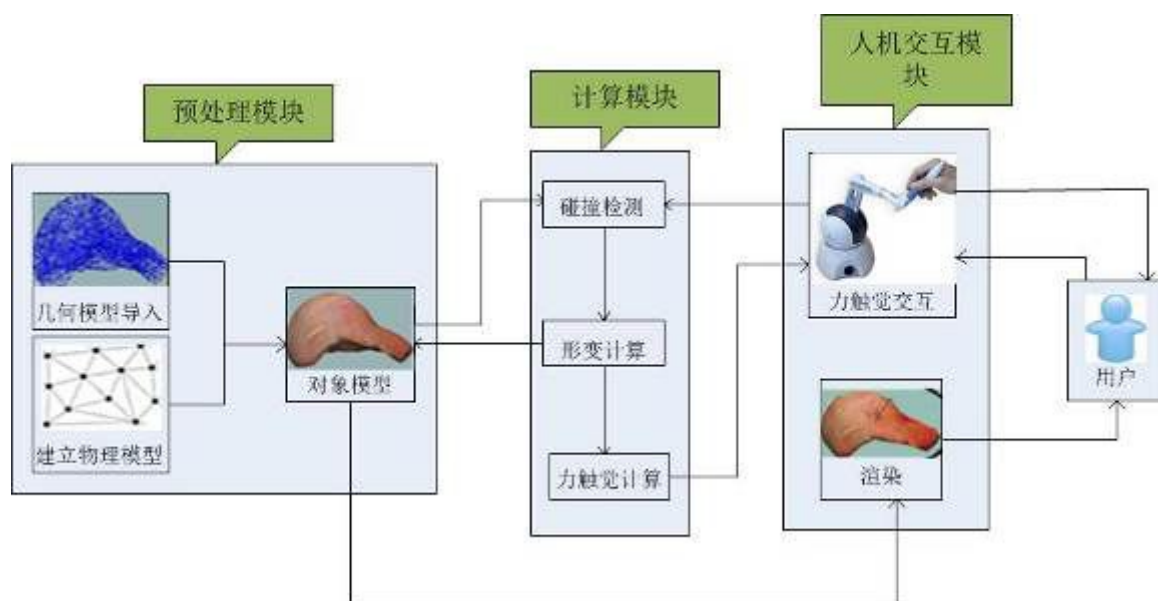


图 1.17 虚拟手术系统架构

(1) 预处理模块: 该模块是系统的初始化模块,主要处理几何模型的导入,物理模型的构建。

(2) 计算模块: 该模块是系统的核心,包含碰撞检测、形变计算、力触觉计算,其中形变计算和力触觉计算是耦合在一起的。碰撞检测模块负责虚拟手术工具与手术对象的相交测试,当二者接触时将碰撞信息和手术器工具加的力传递给形变计算部分,形变计算部分计算变形,然后调用人机交互模块的渲染部分渲染变形后的器官组织场景提供给人视觉反馈,同时力触觉部分计算力触觉并将结果提供给人机交互模块的力交互部

分施加给人手，软体变形的同时碰撞检测部分的相应信息也要更新为下一帧的碰撞检测做准备。

(3) 人机交互模块：该模块主要包含提供视觉反馈的渲染部分和提供力触觉反馈的力交互部分。渲染部分的交互是单向的，只由场景到人这个方向，力触觉反馈部分的交互是双向的，场景可以将力反馈给人，同时人也可以给场景施加力。

1.4 本文的研究内容及结构安排

软体模型和碰撞检测在虚拟手术仿真系统中处于核心地位，影响着视觉、力反馈效果，多年来，在国内外学者的多方努力下虚拟手术仿真技术获得了很大发展，但对软体建模和碰撞检测技术还没有建立一套标准的理论与方法，研究者们依然致力于不断提高软体变形和碰撞检测的速度与精度。所以本文以构建一个简单虚拟手术仿真系统为背景，对系统中软体几何建模技术、软体物理建模技术和碰撞检测技术做重点研究，最后搭建仿真系统验证前述研究技术的有效性。

本文组织如下：

第1章是绪论部分，介绍了论文的研究背景和意义、软体物理建模技术和碰撞检测技术的发展现状，设计了本文虚拟手术系统的总体架构并解释了各模块的作用，最后介绍了本文的研究内容和章节安排。

第2章对虚拟手术仿真中的软体几何建模方面的内容进行介绍。介绍几何模型的分类及相应特点，现有的几何建模方法的原理及特点，然后选择一种合适的方法建立本文所需的软体模型。

第3章是本文的重点章节之一，对软体的物理建模方法做深入研究，首先分别介绍目前最常用的基于网格理论的质点-弹簧模型和有限元模型。然后，为了一定程度上克服二者存在的稳定性、实时性、大变形方面的缺陷，将基于形状匹配的无网格方法引入软体物理建模中，对其原理做详细介绍。最后针对3种方法做对比实验说明本文算法的优势。

第4章也是本文的重点章节，针对虚拟手术系统中单层次包围盒碰撞检测算法存在的检测效率的问题，使用一种三层结构的 Sphere-AABB 复合层次包围盒碰撞检测算法，在不同基本图元相交数量下都能保持最佳检测效率。同时，针对三角形间的相交测试方法，采用于海燕等人提出的基于投影降维的方法，并对此法进行改进，进一步提高实时性、准确性和稳定性。最后分别对本文算法、Sphere 层次包围盒法、AABB 层次包围盒法做对比实验说明本文算法的优势。

第5章搭建系统并进行实验，验证第3章与第4章提出方法的有效性。

最后，总结概括本文的工作，同时对未来工作做展望。

第2章 虚拟手术仿真中软体的几何建模方法

2.1 引言

虚拟手术中的软体模型主要包含两个方面，一是几何模型，由基本几何图元--点、线、面等元素组成，最终通过 OpenGL 绘制到显示设备上，是有形可见的；二是物理模型，在几何模型提供的相应信息基础上，利用几何学或运动物理学建立相应运动学模型，使模型能产生运动、变形等，然后将运动、变形反应到几何模型上给人视觉反馈，是无形不可见的。几何模型和物理模型的关系类似于形与神的关系，形神兼备就可以表现软体在外力作用下的变形、切口形成等。同时几何模型也是物理模型和碰撞检测的基础，因此几何模型的建立也是虚拟手术系统中很关键的一环。本章首先对软体几何建模方法做分类；其次，介绍利用医学图像进行三维重建的方法；再次，介绍三维建模软件建立几何模型的过程；最后，利用 Amira 软件中的移动立方体法和相关医学图像建立本文的肝脏面模型。

2.2 软体几何建模方法分类

几何模型实际上是一个方便计算机表示和处理的数据文件，其中保存了对象的形状、大小、位置和结构关系等几何信息。人体不同的器官和组织结构具有不同的生物力学特性，为了真实模拟相应对象，应根据器官或组织的特点建立相应几何模型。像人体的胃、肠、血管等都是薄壁结构，肝脏、心脏、肾脏近似于实心结构，对于薄壁结构我们一般建立面模型，实心结构建立体模型。

建立几何模型的方法有两种，一是利用获取的医学图像（如 CT 图、MRI 图等）通过三维重建获得；二是通过建模软件（如 3ds Max、Maya 等）绘制。前者获得的几何模型更逼真，但方法复杂；后者得到的模型跟真实的器官组织会有差异，但建立简单。具体选择何种途径建立相应的软体几何模型取决于几何模型的应用环境。

2.3 利用医学图像数据三维重建法

2.3.1 三维重建流程

流程如图 2.1。



图 2.1 利用医学图像数据三维重建流程图

1. 数据获取

数据获取是指医学图像数据的获取，一般常用的有 CT 和 MRI 图像。

2. 预处理

因为获取的图像中包含很多信息，也许我们只对某一组织感兴趣，所以需要分割提取出我们所需要的部分，这个过程就是预处理。

3. 三维重建

利用断层图像序列数据获得具有三维立体感对象的过程就是三维重建，是整个流程的核心。面绘制和体绘制是其中的两种重要方法，面绘制法可以获取对象的几何模型，而体绘制法得到的是一个像素集合，主要用于医学图像数据的三维可视化，是实时绘制的，所以要利用医学图像三维重建获得几何模型一般采用面绘制法。接下来只介绍面绘制三维重建法。

4. 后期优化

对已经建立的模型不理想区域优化处理，同时精简模型。

2.3.2 面绘制三维重建法

面绘制法得到的是一个表面模型，利用多边形面片（一般为三角形）拼接拟合来逼近表示。边界轮廓线连接法和体素等值面提取法是常用方法^[41]。边界轮廓线连接法主要应用于早期的 CT 图像，因为早期的 CT 切片间距比较大，单张图像分辨率也不高。其主要步骤可分为：CT 切片轮廓线提取；相邻切片轮廓线对应并构造多边形面片进行连接。但是对于一个切片上存在多个轮廓线时处理比较复杂，尤其是对轮廓复杂的组织器官（如人脑）重建是很困难的。随着 CT、MRI 技术的进步，切片之间的距离大大缩小，同时图像分辨率也得到很大提升，体素等值面提取法得到广泛应用。主要算法有移动立方体法（Marching Cubes, MC）、Cuberille 法和 Dividing Cubes 法，最为经典、使用最多的当属 MC 算法。本章就重点介绍 MC 算法。

人体不同器官或组织在 CT 灰度图像上灰度值不同，同一器官或组织的边缘与其内部灰度值也有区别，从而利用灰度值就可以区分出一种器官或组织的边缘。将一种器官或组织边缘上的灰度值设为阈值，将体数据中与该阈值相等的点都提取出来，利用多边形面片将这些点连接起来就构成了器官或组织的面模型。MC 算法就是基于此原理，在体数据构成的体素中根据设定的阈值提取出等值面从而构建对象的表面模型。下面详细介绍 MC 算法。

1. 体素

如图 2.2 所示，相邻的 8 个采样点包围的中间区域就是移动立方体法中使用的体素， i, j 分别代表断层平面内两个方向的采样坐标， k 代表断层的采样坐标，假设图像空间（CT 图像序列组成的体数据空间）中一点为 x, y, z ，在三个方向上的采样间隔取

$\Delta x, \Delta y, \Delta z$, 则这一点的采样坐标 $i = x / \Delta x, j = y / \Delta y, k = z / \Delta z$ 。

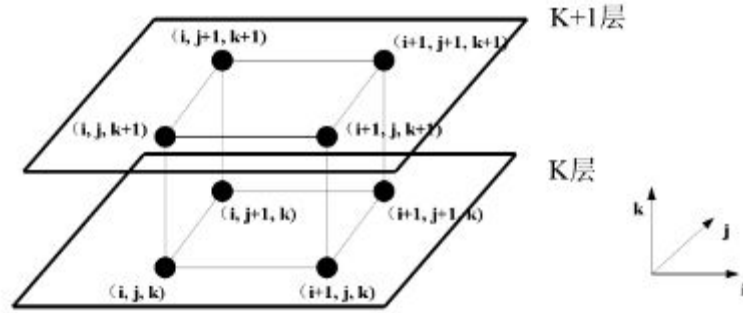


图 2.2 体素示意图

这里将体素内一点所具有的属性（这里是指图像灰度值）定义为此点的体素值，用 $f(p)$ 表示。假设在体素内有任意一点 p_6 ，如图 2.3 所示，其体素值可通过三线性插值求得，如式 2-1。

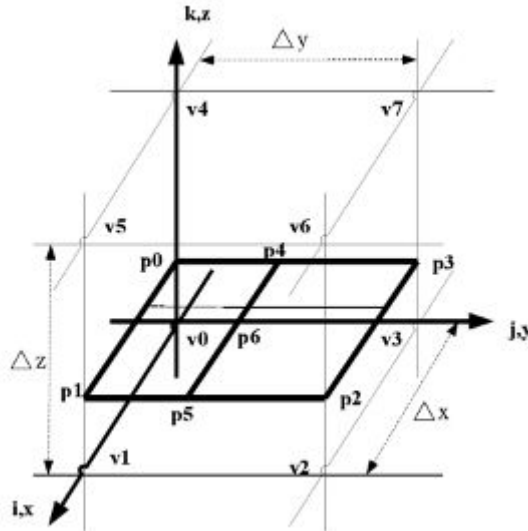


图 2.3 体素内任意一点体素值三线性插值模型

$$\begin{aligned} f(p_6) &= f(p_4)(i+1-i_6) + f(p_5)(i_6-i) \\ f(p_4) &= f(p_0)(j+1-j_6) + f(p_3)(j_6-j) \\ f(p_5) &= f(p_1)(j+1-j_6) + f(p_2)(j_6-j) \end{aligned} \quad (2-1)$$

上式中用到的 p_0, p_1, p_2, p_3 的灰度值依据相同的方法利用体素八个顶点的灰度值线性插值求出。

2. 等值面

$S = \{(x, y, z): f(x, y, z) = C, C \text{ 是固定值}\}$, S 曲面就叫等值面。由前面的介绍可知我们可以获取体数据所在空间中任意一点的体素值，此处 C 选择目标器官或组织边界的灰度值，假设灰度值在体素边上是线性均匀变化的，在体素的边上我们能够找到灰度值

与 C 相等的点，在相应体素中用这些点构建三角形面片近似表示等值面，将所有的近似等值面组合在一起就构建了目标器官或组织的三维面模型。

3.MC 算法

其输入的数据是排好序的一系列的 CT（或其他类型）切片图像，其步骤可归结为：
第一步：先导入前 2 幅图像作为第一层，此后每个循环导入紧挨着的下一幅切片图，用新导入的和紧挨着的上一幅作为下一层。

第二步：每一层的上下切片中相邻的八个采样点（一般为切片中的像素点，也有可能需重采样）组成一个长方体，该长方体即为体素。

第三步：遍历一层中所有的体素，根据阈值与体素角点的体素值关系判断每个体素中是否存在等值面，如果存在就构建等值面。此层处理结束后，返回到步骤一重复此算法流程，直到处理完全部的图像。

体素内部构建等值面：

（1）判断一个体素内是否包含等值面

通过体素 8 个角点的灰度值和灰度阈值即可判断，前者通过查询采样点的像素灰度值得到，后者是预先设置的。只要 8 个角点中既有灰度值比阈值大的也有灰度值比阈值小的，则体素内必存在等值面，否则不存在。

（2）等值面在一个体素中的分布情况

因为体素的每个角点灰度值与阈值的关系均有三种可能，所以等值面在体素内的分布会有 3^8 种，因为角点体素值与阈值相等时等值点就是角点，是利用线性插值法计算边上等值点的一个特例，所以可以只考虑每个角点灰度值比阈值大或小的两种情况，将相等的情形归于大于（或小于）的情形，这时体素内等值面的分布会有 2^8 种，再根据立方体旋转不影响等值面拓扑结构，角点灰度值大于阈值的情况与小于的情况完全对换也不影响等值面拓扑结构，所以体素内等值面的分布情形能够简化为如图 2.4 所示的 15 种，图中红色角点代表灰度值小于阈值，蓝色表示大于。根据体素角点灰度值与阈值的大小关系可以转化为图中的一种，根据图中所示求等值点并进行连接形成近似等值面。

（3）求取等值面的法向量

我们知道在计算机图形学中恰当地使用光照可以增强模型的真实感，但为了计算光照效果需要提供面片的法向量或组成面片的顶点的法向量。所以为了使用光照增强建立的模型的渲染效果，我们还需计算等值面顶点（等值点）的法向量。

等值点的坐标和法向量都使用其所在体素的边的端点的相应值利用线性插值求取。体素角点的坐标是已知的，角点法向量可通过式 2-2 求得。

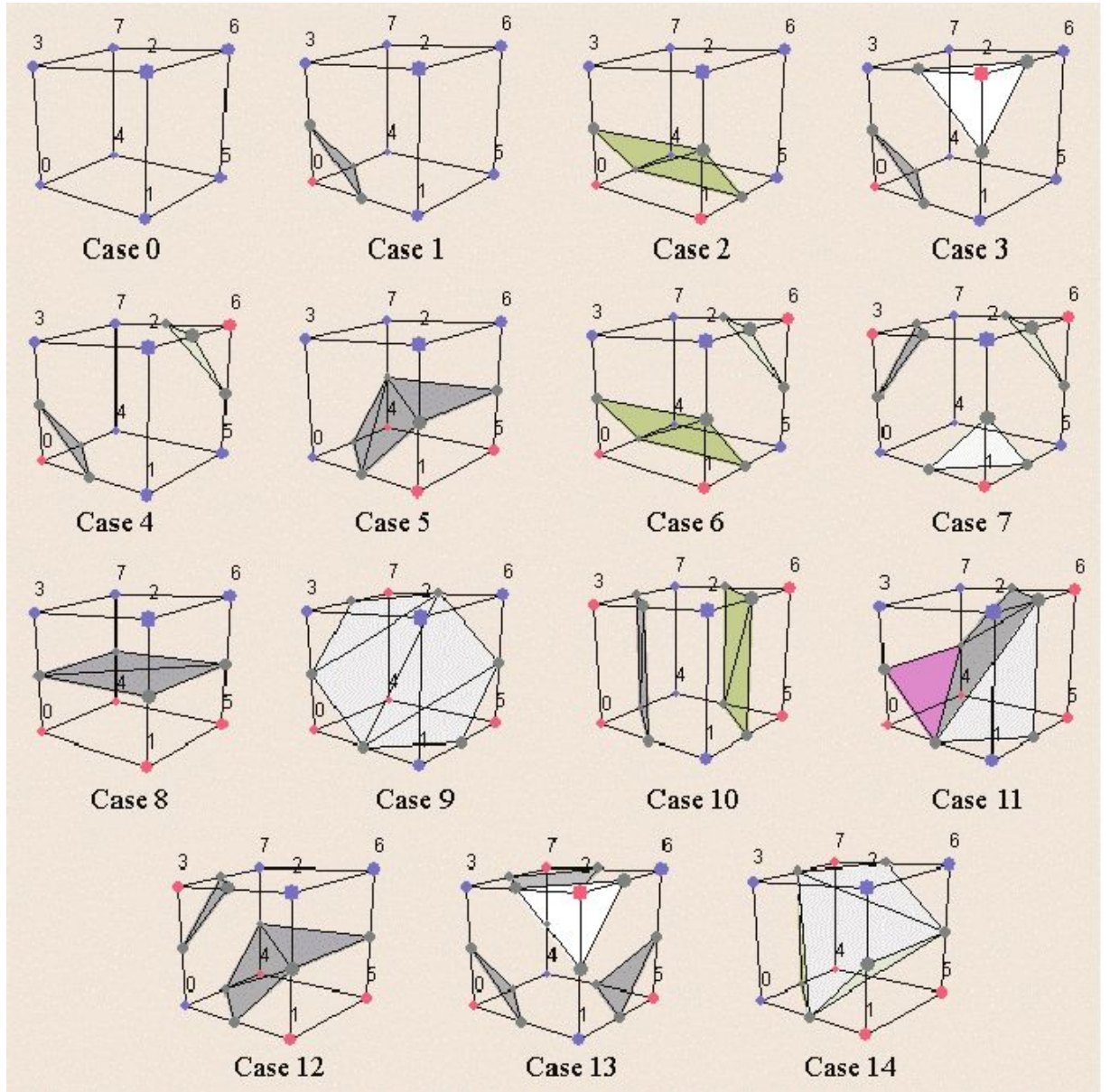


图 2.4 体素内等值面的分布情况

$$g(x, y, z) = \nabla f(x, y, z)$$

$$\begin{cases} g_x = \frac{f(x_{i+1}, y_j, z_k) - f(x_{i-1}, y_j, z_k)}{2\Delta x} \\ g_y = \frac{f(x_i, y_{j+1}, z_k) - f(x_i, y_{j-1}, z_k)}{2\Delta y} \\ g_z = \frac{f(x_i, y_j, z_{k+1}) - f(x_i, y_j, z_{k-1})}{2\Delta z} \end{cases} \quad (2-2)$$

等值点的相应坐标和法向量可通过式 (2-3) 求得。

$$P = P_1 + (P_2 - P_1)(isovalue - V_1)/(V_2 - V_1) \quad (2-3)$$

其中 P 表示等值点的相应量（坐标或法向量）， P_1, P_2 表示端点的相应量， V_1, V_2 代表端点的灰度值， $isovalue$ 表示等值点的灰度值。这里假定在边上灰度值是线性变化的。

2.4 利用三维建模软件绘制法

这种方法是指利用三维建模软件（如 3ds Max, Maya, Blender 等）绘制出具有三维数据的模型。

以 3dsMax 建模软件建立一个心脏模型为例简单介绍一下建模过程^[42]。

（1）首先参照相关图片或实体绘制出心脏的 6 个方向的位图，位图大小为 900x900 像素，如图 2.5 所示。

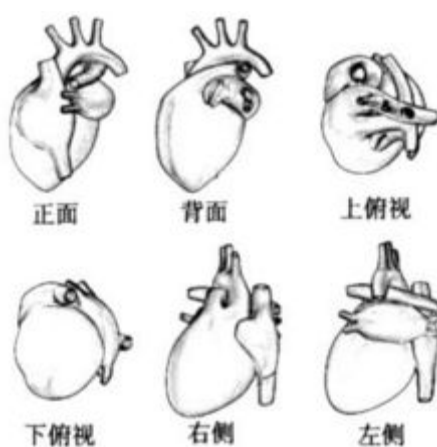


图 2.5 心脏 6 个方向的位图

（2）将上述位图导入到建模软件中，如图 2.6

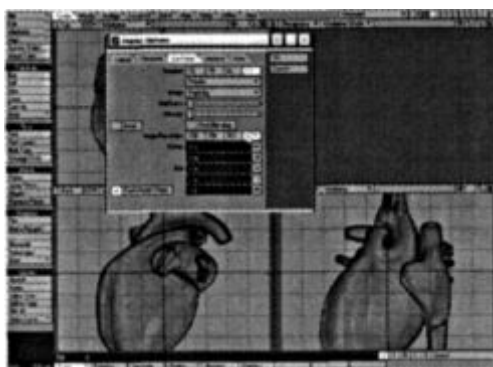
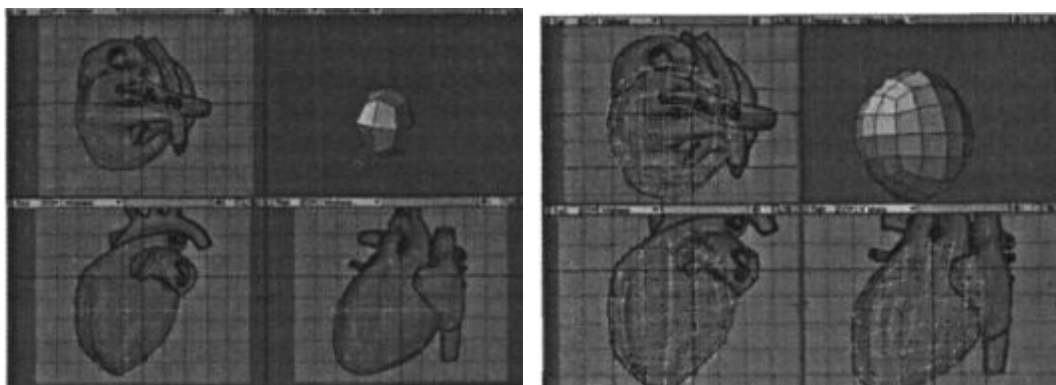


图 2.6 将位图导入软件

（3）将模型分解成多个部分分别来建立，下面只介绍建立模型中的一部分。

首先建立一个最简单的 Box 对象，利用 polygons 选项中的 metafrom 结合手动操作不断修正 Box 对象使其与导入的三个方向上的位图主体相吻合。如图 2.7 所示。



(a) 修正一

(b) 修正二

图 2.7 建立心脏主体

(4) 使用圆弧编辑、节点编辑、metafrom 等选项, 对照导入的图像进行动静脉血管模型的制作, 如图 2.8。

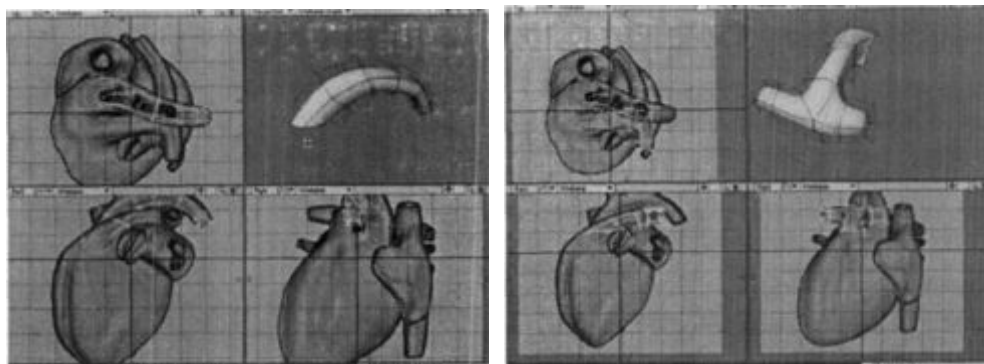


图 2.8 心脏动脉

(5) 将得到的主体和动静脉组合得到心脏 A 部分, 如图 2.9。

(6) 其他部分的建立步骤也是类似的, 分部分建立好后组合得到心脏的总体模型, 如图 2.10。

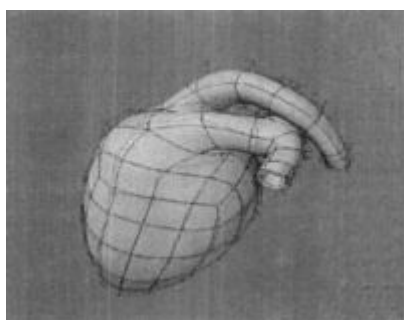


图 2.9 心脏 A 部分

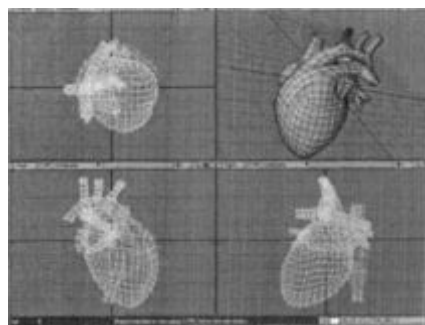


图 2.10 心脏模型

2.5 软体几何模型的建立

使用何种方法建立几何模型, 取决于几何模型的应用环境, 对于要求模型精细、真

实性高的系统如手术训练系统、术前规划系统、术中导航系统等，要利用医学图像通过三维重构方法建立几何模型；对于要求真实性不高的系统如演示系统，为了降低复杂性可以选择利用三维建模软件建立。本文采用医学图像三维重建法建立几何模型。

如今很多公司已研发出了成熟的医学图像三维可视化系统或软件，如美国 Stardent 公司的 AVS 系统，加拿大的 Allegro 系统，法国 TGS 公司的 Amira 软件等，同时也有很多类似 VTK、VIK、ITK 等的开源工具^[43]。本文选择集成了 MC 算法的成熟软件--Amira 建立所需的肝脏几何模型。

2.5.1 Amira 工具软件

Amira 是法国 TGS 公司开发的功能强大、多用途的工具软件，主要用于数据可视化及建模。

Amira 的用户界面如图 2.11 所示，主要分为四个区块^[44]：

- (1) 图形显示区，用于显示导入的 CT 图像和重建结果；
- (2) 对象池（object pool），用于显示数据对象和模块；
- (3) 工作区，用于显示当前模块的参数；
- (4) 控制台，用于显示系统信息和输入执行命令。

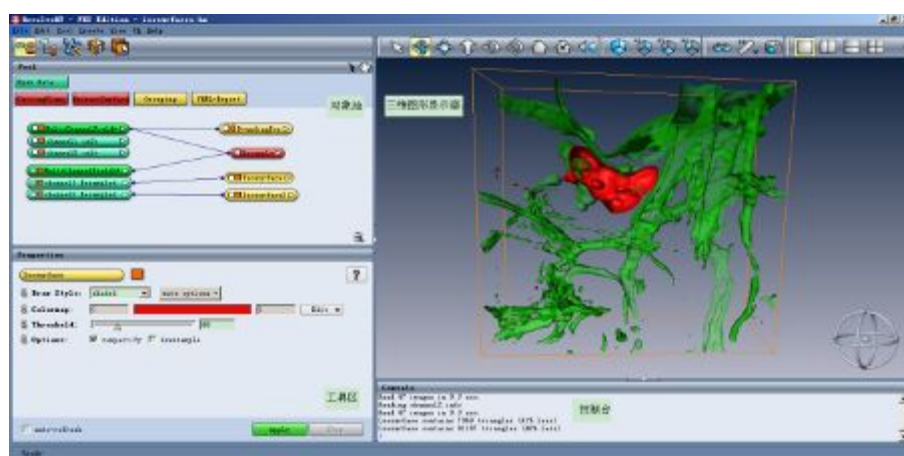


图 2.11 Amira 软件用户界面

2.5.2 利用 Amira 软件建立几何模型

其基本流程如图 2.12 所示

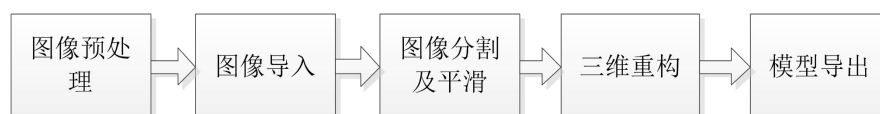


图 2.12 Amira 三维重建流程

1. 图像预处理

本文拟建立人体肝脏模型，CT 图像选用美国国立医学图书馆“可视人计划”中的腹部部分切片图像，因为切片间距只有 1mm，切片图像太多，为减少数据量，我们等间隔选用其中的包含肝脏部分的 200 张。总的的数据量大小为 $2048 \times 1216 \times 200$ 体素，其中只有肝脏部分是我们所需要的，为了减小数据量，我们对图像进行裁剪只保留需要的部分。裁剪前后对比如图 2.13。

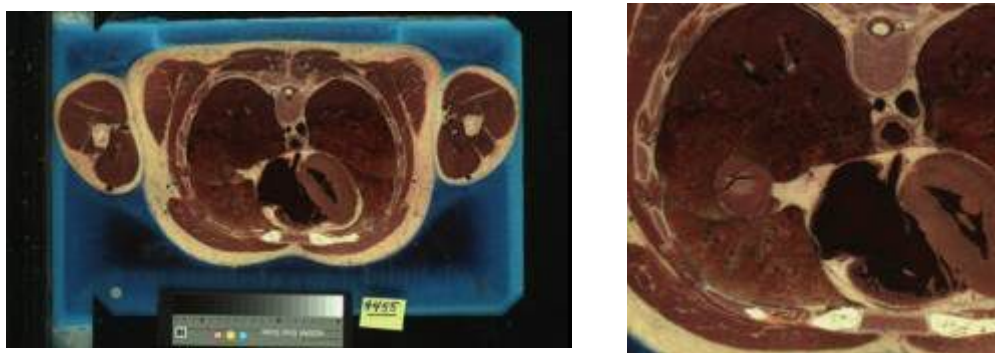
(a) 剪裁前 $2048 \times 1216 \times 200$ 体素(b) 剪裁后 $619 \times 635 \times 200$ 体素

图 2.13 剪裁前后数据量对比

2. 图像导入

通过 File/Load 菜单将图像全部导入并调整参数。导入后对象池中会显示其图标。

3. 图像分割及平滑

这一步是三维重建的基础。分割的质量和速度都很重要，现在分割的方法有手动分割和自动分割，在图像中我们可以看到肝脏组织与周边组织对比度不高，边界模糊，其他微细结构又对其有影响，使用自动分割很困难，而如果手动分割工作量又太大^[45]。所以我们采用二者结合的方法，取一个合适的间隔值将图片分组，间隔处采用手动分割，然后利用 Amira 的自动插值方法分割间隔之间的图片。分割后检查一遍，分割不好的地方手动修正，同时为每张图像作上标签。利用“smooth label”对边缘做光滑处理，“remove islands”删掉孤岛，增强分割效果。分割后并进行填充后的效果如图 2.14。

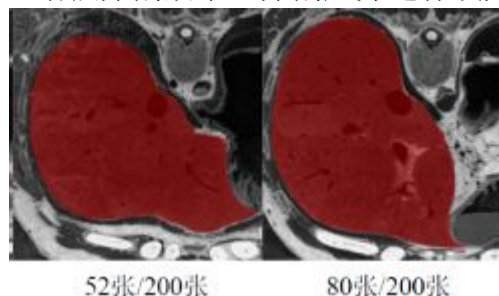


图 2.14 肝脏分割效果

4. 三维重构

使用 Amira 提供的移动立方体法构建肝脏面模型，并对其进行表面光滑处理。

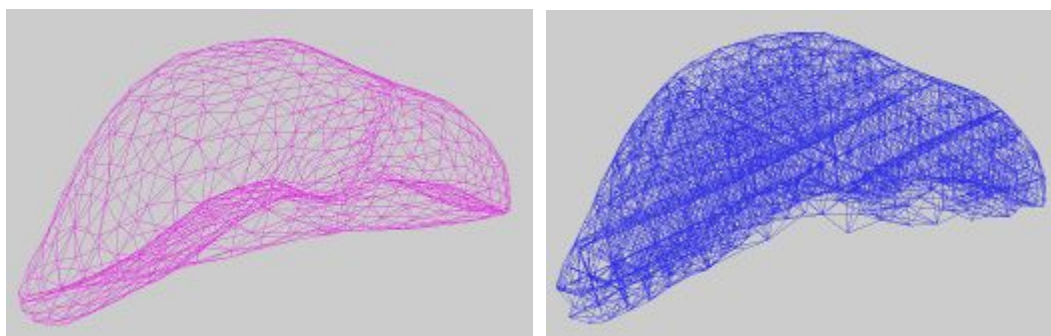
5. 模型导出

因为 Amira 导出的模型格式不能直接用于我们的系统,所以我们利用 3ds Max 软件先进行一下全局坐标调整,然后将其导出为.obj 格式文件,重建并加入纹理的肝脏面模型如图 2.15,共有 725 个顶点,1448 个三角形组成。



图 2.15 三维重建得到的肝脏面模型

因为肝脏为实体,应当建立其体模型,建立体模型的常用方法就是将面模型进行四面体剖分,本文利用 QTetraMesher (<http://code.google.com/p/qtetramesher/>) 软件来完成此工作,建立后的体模型如图 2.16 (b),共含有 1206 个顶点,4230 个四面体,9313 个三角形和 6288 条边。图 2.16 (a) 为四面体剖分前的网格表示的肝脏面模型。



(a) 网格表示形式的面模型 (b) 通过四面体剖分获得的体模型

图 2.16 肝脏几何模型

2.6 本章小结

本章主要对软体模型的几何建模方法做了总结,是本文的基础章节。首先介绍了医学图像三维重建中的面绘制法原理,然后以心脏模型为例介绍了利用 3ds Max 三维建模软件建立几何模型的过程。因为软体要用在虚拟手术训练系统中,对模型逼真度要求较高,所以,选择利用医学图像三维重建方法借助 Amira 软件建立了本文要用到的肝脏几何模型。

第3章 虚拟手术仿真中软体的物理建模方法

3.1 引言

虚拟手术仿真系统中几何模型和物理模型共同组成了软体组织形变的基础，在上一章我们介绍了软体的几何建模方法，本章重点介绍其物理建模方法，首先简单介绍常用的质点-弹簧模型，有限元模型，然后，针对质点-弹簧模型存在的大变形失真、不稳定性问题，有限元模型存在的实时性差问题，本章给出基于形状匹配的无网格方法，最后对上述方法进行仿真实验和对比分析。

3.2 基于网格的常用软体物理建模方法

软体物理建模的基本思想是将连续空间的问题转化为离散空间的问题进行简化，现在研究最多，应用最为广泛的模型依然是基于网格的质点-弹簧模型和有限元模型（FEM）。

3.2.1 质点-弹簧模型

质点-弹簧模型有面模型和体模型两种类型，对应的也都有很多种拓扑形式，简易拓扑结构如图3.1。一个对象可以表示为相互间通过虚拟弹簧连接的质点集合，一个质点上连接的弹簧数量由其在模型中所处的位置有关。

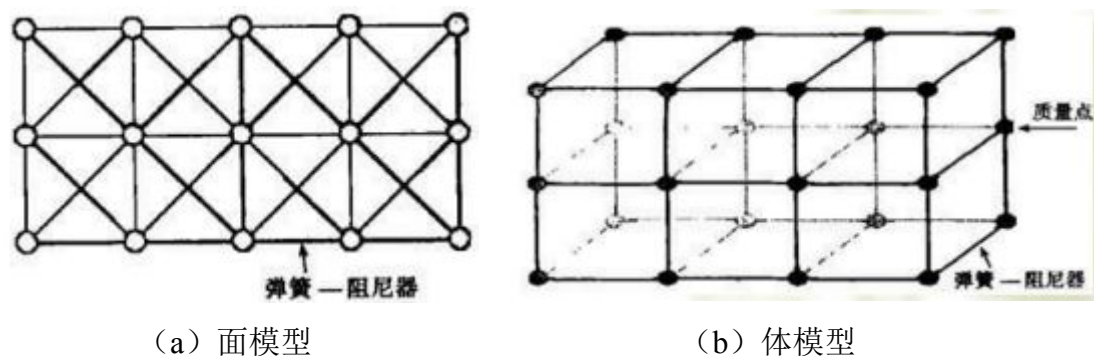


图 3.1 质点-弹簧模型拓扑结构

质点之间的连接弹簧一般有三种类型，结构弹簧、扭曲弹簧与拉伸弹簧，如图3.2所示。

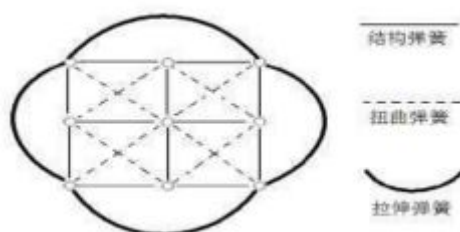


图 3.2 质点间连接弹簧分类

结构弹簧用于维持对象的整体结构；扭曲弹簧的作用是阻止对象发生扭曲形变；拉伸弹簧连接间隔质点，可以让对象弯曲时边缘光滑。在动态仿真系统中为了保证其稳定性和减少其达到新平衡态所需时间，一般都加入适当的阻尼力，通常使用的带阻尼弹簧模型有 Maxwell 和 Kelvin 两种，如图 3.3 所示，为了计算简单一般选用 Kelvin 模型。

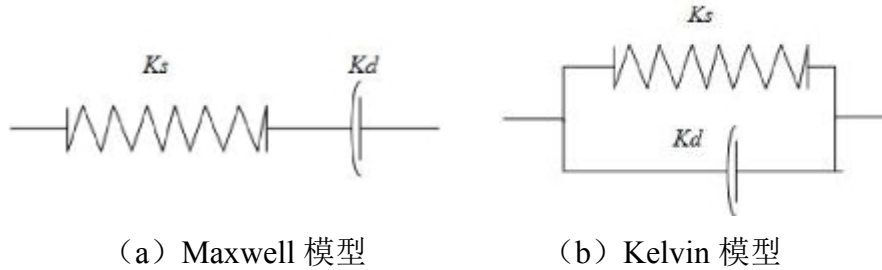


图 3.3 带阻尼弹簧模型

下面对质点-弹簧物理模型进行分析：

一个对象离散化为了质点集合，如果想知道对象在某一时刻的运动情况，只需知道每个质点在对应时刻的位移和速度即可。以质点 i 在 t 时刻为例对其做运动分析，根据牛顿运动第二定律要先求出质点 i 受到的合力 f_i ， f_i 可由质点 i 受到的内力（与之相连弹簧的弹性力、阻尼力）和外部力（重力、外界施加的力）进行合成得到，如式 3-1 所示。

$$f_i = f_{\text{int}} + f_{\text{ext}} \quad (3-1)$$

假设与之相邻的质点为 j ，则 j 施加给 i 的弹力、阻尼力合力为：

$$f_{ij} = k_s \left(|x_{ij}| - l_{ij} \right) \frac{x_{ij}}{|x_{ij}|} + k_d (v_j - v_i) \quad (3-2)$$

参数的含义： k_s 为弹簧刚度系数， x_{ij} 为相邻质点 i 与 j 的位置差（向量）， l_{ij} 是弹簧原长， k_d 表示阻尼系数， v_j 表示质点 j 的速度， v_i 表示质点 i 的速度^[46]。

则 f_{int} 为：

$$f_{\text{int}} = \sum_{j=0}^{n-1} f_{ij} \quad (3-3)$$

f_{ext} 易求，所以利用（3-1）式求出 i 质点所受合力。

根据牛顿第二定律：

$$m_i a_i = f_i \quad (3-4)$$

利用（3-5）（3-6）积分式可求出质点在下一时间步的速度和位置。

$$v_i(t+h) = v_i(t) + h \frac{f_i}{m_i} \quad (3-5)$$

$$x_i(t+h) = x_i(t) + hv_i(t+h) \quad (3-6)$$

通过以上所述原理在每一时间步求出每一质点的速度和位置就可以得到对象整体的运动状态。

3.3.2 有限元模型

有限元法^[47]把对象离散化为易分析的有限个子单元，单元与单元借助边界上的结点连接，通过求解子单元的相应特征，结合变形协调条件、边界条件求取问题的总体解。对连续体常用的子单元类型有三角形、四边形、四面体、六面体等。

1. 有限元方法的力学基础

有限元法用到的基本变量有位移、应力和应变。

在三维空间中它们之间存在的关系可以通过平衡方程、几何方程、物理方程、变形连续方程、应力边界条件、势能等来表示。

(1) 平衡微分方程--应力与体力关系：

$$\begin{aligned} \frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} + X &= 0 \\ \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + \frac{\partial \tau_{zy}}{\partial z} + Y &= 0 \\ \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial \sigma_z}{\partial z} + Z &= 0 \\ \tau_{yz} &= \tau_{zy}, \tau_{zx} = \tau_{xz}, \tau_{xy} = \tau_{yx} \end{aligned} \quad (3-7)$$

其中 $\sigma_x, \sigma_y, \sigma_z$ 表示 x, y, z 方向的正应力， $\tau_{xy}, \tau_{yx}, \tau_{xz}, \tau_{zx}, \tau_{yz}, \tau_{zy}$ 表示剪应力， X, Y, Z 表示作用在微元上的体积力向量。

(2) 几何方程--描述位移与应变的关系

$$\begin{aligned} \varepsilon_x &= \frac{\partial u}{\partial x}, \gamma_{xy} = \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \\ \varepsilon_y &= \frac{\partial v}{\partial y}, \gamma_{yz} = \frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} \\ \varepsilon_z &= \frac{\partial w}{\partial z}, \gamma_{zx} = \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \end{aligned} \quad (3-8)$$

其中 u, v, w 表示 x, y, z 方向的位移， $\varepsilon_x, \varepsilon_y, \varepsilon_z$ 表示 x, y, z 方向的正应变， $\gamma_{xy}, \gamma_{yz}, \gamma_{zx}$ 表示剪应变。

(3)物理方程--描述应力与应变关系

$$\begin{aligned}\varepsilon_x &= \frac{1}{E} [\sigma_x - \mu(\sigma_y + \sigma_z)] \gamma_{xy} = \frac{1}{G} \tau_{xy} \\ \varepsilon_y &= \frac{1}{E} [\sigma_y - \mu(\sigma_z + \sigma_x)] \gamma_{yz} = \frac{1}{G} \tau_{yz} \\ \varepsilon_z &= \frac{1}{E} [\sigma_z - \mu(\sigma_x + \sigma_y)] \gamma_{zx} = \frac{1}{G} \tau_{zx}\end{aligned}\quad (3-9)$$

其中 $\varepsilon_x, \varepsilon_y, \varepsilon_z$ 表示 x, y, z 方向的正应变, $\sigma_x, \sigma_y, \sigma_z$ 表示 x, y, z 方向的正应力, $\gamma_{xy}, \gamma_{yz}, \gamma_{zx}$ 表示剪应变, $\tau_{xy}, \tau_{yz}, \tau_{zx}$ 表示剪应力, E, μ, G 分别为材质的弹性模量, 泊松比和剪切弹性模量, 三者满足关系 $G = \frac{E}{2(1+\mu)}$ 。

(4)变形连续方程--描述应变分量间的关系

$$\begin{aligned}\frac{\partial^2 \varepsilon_y}{\partial z^2} + \frac{\partial^2 \varepsilon_z}{\partial y^2} &= \frac{\partial^2 \gamma_{yz}}{\partial y \partial z}, \frac{\partial}{\partial y} \left(\frac{\partial \gamma_{xy}}{\partial z} + \frac{\partial \gamma_{yz}}{\partial x} - \frac{\partial \gamma_{zx}}{\partial y} \right) = 2 \frac{\partial^2 \varepsilon_y}{\partial x \partial z} \\ \frac{\partial^2 \varepsilon_z}{\partial x^2} + \frac{\partial^2 \varepsilon_x}{\partial z^2} &= \frac{\partial^2 \gamma_{zx}}{\partial z \partial x}, \frac{\partial}{\partial x} \left(\frac{\partial \gamma_{zx}}{\partial y} + \frac{\partial \gamma_{xy}}{\partial z} - \frac{\partial \gamma_{yz}}{\partial x} \right) = 2 \frac{\partial^2 \varepsilon_x}{\partial y \partial z} \\ \frac{\partial^2 \varepsilon_x}{\partial y^2} + \frac{\partial^2 \varepsilon_y}{\partial x^2} &= \frac{\partial^2 \gamma_{xy}}{\partial x \partial y}, \frac{\partial}{\partial z} \left(\frac{\partial \gamma_{yz}}{\partial x} + \frac{\partial \gamma_{zx}}{\partial y} - \frac{\partial \gamma_{xy}}{\partial z} \right) = 2 \frac{\partial^2 \varepsilon_z}{\partial x \partial y}\end{aligned}\quad (3-10)$$

(5)应力边界条件

$$\begin{aligned}\tau_{xy}l + \sigma_y m + \tau_{zy}n &= \bar{Y} \\ \sigma_x l + \tau_{yx}m + \tau_{zx}n &= \bar{X} \\ \tau_{xz}l + \tau_{yz}m + \sigma_z n &= \bar{Z}\end{aligned}\quad (3-11)$$

其中 $\sigma_x, \sigma_y, \sigma_z$ 代表边界上 x, y, z 方向的正应力, $\tau_{xy}, \tau_{yz}, \tau_{zx}$ 代表边界上剪应力, l, m, n 代表边界面法线与各坐标轴夹角的余弦值, $\bar{X}, \bar{Y}, \bar{Z}$ 为作用在边界上的面力在各坐标轴上的分量。

(6)系统势能

$$\Pi = U - V = \frac{1}{2} \int_{\Omega} \sigma_{ij} \varepsilon_{ij} d\Omega - \left[\int_{\Omega} \bar{b}_i u_i d\Omega + \int_{S_p} \bar{p}_i u_i dA \right] \quad (3-12)$$

U, V 分别为系统的应变能和外力功, 外力功包含体力功和面力功两部分。

2.有限元法求解问题一般步骤

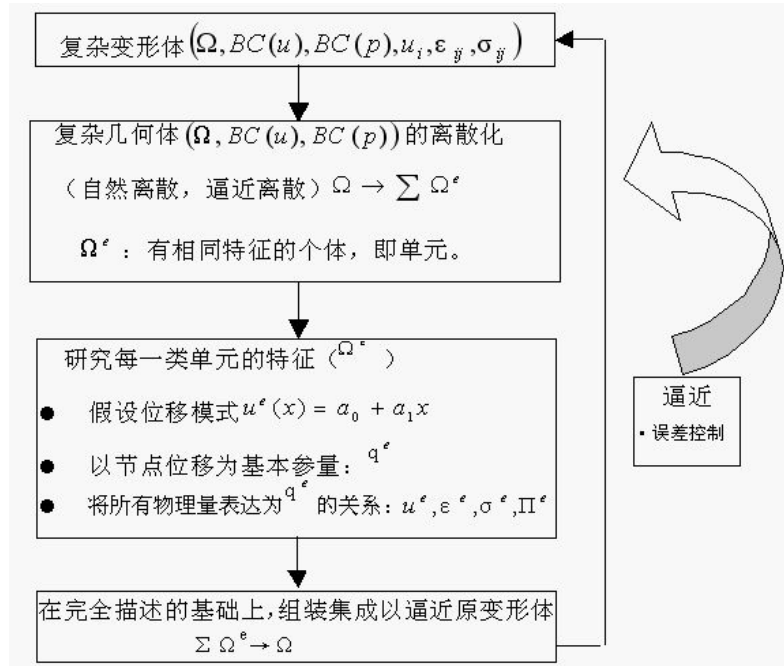


图 3.4 有限元法求解问题的一般步骤

(1) 离散化: 把几何模型通过自然离散和逼近离散方法 (针对连续体) 划分为有限个子单元, $\Omega = \sum \Omega^e$ 。

(2) 单元研究: 根据简单、完备、连续及待定系数唯一确定原则选择单元的位移场模式, 利用单元结点条件求出位移场模式中的待定系数, 根据几何方程、物理方程、势能公式等将单元的位移、应变、应力、势能等力学量都用结点的位移来表示。

(3) 利用应最小势能原理建立单元的平衡关系。

(4) 单元装配得到整体平衡关系。

(5) 利用边界条件求取未知结点位移, 然后求取未知外力和其他力学量。

3. 以平面三角形单元为例做有限元分析

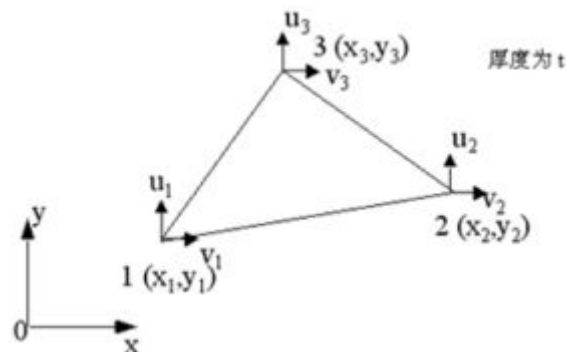


图 3.5 平面三角形单元模型

如图 3.5 所示, 三个结点编号为 1, 2, 3, 位置坐标为 $(x_i, y_i), i=1, 2, 3$, 结点位移为 $(u_i, v_i), i=1, 2, 3$ 。

结点位移向量与力向量分别用 q^e 和 P^e 来表示，如式 (3-13)：

$$\begin{aligned} q^e &= [u_1 \quad v_1 \quad u_2 \quad v_2 \quad u_3 \quad v_3]^T \\ P^e &= [P_{x1} \quad P_{y1} \quad P_{x2} \quad P_{y2} \quad P_{x3} \quad P_{y3}]^T \end{aligned} \quad (3-13)$$

(1) 单元位移场

根据位移场模式选取原则选取如下模式：

$$\begin{aligned} u^e(x, y) &= a_0 + a_1x + a_2y \\ v^e(x, y) &= b_0 + b_1x + b_2y \end{aligned} \quad (3-14)$$

由结点条件知：

$$\begin{aligned} u^e(x_i, y_i) &= u_i \quad i=1,2,3 \\ v^e(x_i, y_i) &= v_i \end{aligned} \quad (3-15)$$

上述两式结合可求出式 (3-14) 中的 6 个系数，

$$\begin{aligned} a_0 &= \frac{1}{2A} \begin{vmatrix} u_1 & x_1 & y_1 \\ u_2 & x_2 & y_2 \\ u_3 & x_3 & y_3 \end{vmatrix} = \frac{1}{2A} (\alpha_1 u_1 + \alpha_2 u_2 + \alpha_3 u_3), \quad b_0 = \frac{1}{2A} (\alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3) \\ a_1 &= \frac{1}{2A} \begin{vmatrix} 1 & u_1 & y_1 \\ 1 & u_2 & y_2 \\ 1 & u_3 & y_3 \end{vmatrix} = \frac{1}{2A} (\beta_1 u_1 + \beta_2 u_2 + \beta_3 u_3), \quad b_1 = \frac{1}{2A} (\beta_1 v_1 + \beta_2 v_2 + \beta_3 v_3) \\ a_2 &= \frac{1}{2A} \begin{vmatrix} 1 & x_1 & u_1 \\ 1 & x_2 & u_2 \\ 1 & x_3 & u_3 \end{vmatrix} = \frac{1}{2A} (\gamma_1 u_1 + \gamma_2 u_2 + \gamma_3 u_3), \quad b_2 = \frac{1}{2A} (\gamma_1 v_1 + \gamma_2 v_2 + \gamma_3 v_3) \end{aligned} \quad (3-16)$$

其中 A 的值为：

$$A = \frac{1}{2} \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} = \frac{1}{2} (\alpha_1 + \alpha_2 + \alpha_3) = \frac{1}{2} (\beta_1 \gamma_2 - \beta_2 \gamma_1) \quad (3-17)$$

$\alpha_i, \beta_i, \gamma_i (i=1,2,3)$ 的值为：

$$\begin{aligned} \alpha_1 &= \begin{vmatrix} x_2 & y_2 \\ x_3 & y_3 \end{vmatrix} = x_2 y_3 - x_3 y_2 \\ \beta_1 &= - \begin{vmatrix} 1 & y_2 \\ 1 & y_3 \end{vmatrix} = y_2 - y_3 \\ \gamma_1 &= - \begin{vmatrix} 1 & x_2 \\ 1 & x_3 \end{vmatrix} = -x_2 + x_3 \end{aligned} \quad \left\{ \begin{matrix} 1,2,3 \end{matrix} \right\} \quad (3-18)$$

式中 (1,2,3) 表示下标轮换，如 $1 \rightarrow 2, 2 \rightarrow 3, 3 \rightarrow 1$ 。

然后用结点位移的形式来表达位移函数，表达形式如 (3-19)。

$$\begin{aligned} u^e(x, y) &= N_1(x, y) \cdot u_1 + N_2(x, y) \cdot u_2 + N_3(x, y) \cdot u_3 \\ v^e(x, y) &= N_1(x, y) \cdot v_1 + N_2(x, y) \cdot v_2 + N_3(x, y) \cdot v_3 \end{aligned} \quad (3-19)$$

用矩阵表示,

$$u^e(x, y) = \begin{bmatrix} u^e(x, y) \\ v^e(x, y) \end{bmatrix} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{bmatrix} = N^e(x, y) \cdot q^e \quad (3-20)$$

式中 $N^e(2 \times 6)$ 称为形状函数矩阵,

$$\begin{aligned} N^e(x, y) &= \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 \end{bmatrix} \\ N_i &= \frac{1}{2A}(\alpha_i + \beta_i x + \gamma_i y), i=1,2,3 \end{aligned} \quad (3-21)$$

(2) 单元应变场

由平面问题的几何方程得:

$$\varepsilon^e(x, y) = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} \end{bmatrix} \begin{bmatrix} u^e(x, y) \\ v^e(x, y) \end{bmatrix} = [\partial] u^e = [\partial] N^e(x, y) \cdot q^e = B^e(x, y) \cdot q^e \quad (3-22)$$

其中 $B^e(x, y)$ 称为几何函数矩阵,

$$\begin{aligned} B^e(x, y) &= [\partial] N^e(x, y) = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} \end{bmatrix} \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 \end{bmatrix} \\ &= \frac{1}{2A} \begin{bmatrix} \beta_1 & 0 & \beta_2 & 0 & \beta_3 & 0 \\ 0 & \gamma_1 & 0 & \gamma_2 & 0 & \gamma_3 \\ \beta_1 & \gamma_1 & \beta_2 & \gamma_2 & \beta_3 & \gamma_3 \end{bmatrix} = [B_1 \quad B_2 \quad B_3] \\ B_i &= \frac{1}{2A} \begin{bmatrix} \beta_i & 0 \\ 0 & \gamma_i \\ \beta_i & \gamma_i \end{bmatrix} \quad i=1,2,3 \end{aligned} \quad (3-23)$$

(3) 单元应力场

由平面问题的物理方程得：

$$\sigma^e(x, y) = \begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix} = \frac{E}{1-\mu^2} \begin{bmatrix} 1 & \mu & 0 \\ \mu & 1 & 0 \\ 0 & 0 & \frac{1-\mu}{2} \end{bmatrix} \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix} = D^e \cdot \varepsilon^e \quad (3-24)$$

D^e 称为弹性系数矩阵，

$$D^e = \frac{E}{1-\mu^2} \begin{bmatrix} 1 & \mu & 0 \\ \mu & 1 & 0 \\ 0 & 0 & \frac{1-\mu}{2} \end{bmatrix} \quad (3-25)$$

将(2)中求出的单元应变场代入式 (3-24) 中有

$$\sigma^e = D^e \cdot B^e \cdot q^e = S^e \cdot q^e \quad (3-26)$$

$S^e = D^e \cdot B^e$ ，称为单元应力矩阵。

(4) 单元势能

由势能公式得：

$$\begin{aligned} \Pi^e &= \frac{1}{2} \int_{\Omega^e} \sigma^{eT} \cdot \varepsilon^e d\Omega - \left[\int_{\Omega^e} \bar{b}^T \cdot u^e d\Omega + \int_{S_p^e} \bar{b}^T \cdot u^e d\Omega \right] \\ &= \frac{1}{2} q^{eT} K^e q^e - P^{eT} q^e \end{aligned} \quad (3-27)$$

K^e 称为单元刚度矩阵，

$$K^e = \int_{\Omega^e} B^{eT} D^e B^e d\Omega = \int_{A^e} B^{eT} D^e B^e dA \cdot t, \quad t \text{ 为三角形的厚度} \quad (3-28)$$

P^e 为单元结点等效载荷，

$$P^e = \int_{A^e} N^{eT} \bar{b} t dA + \int_{l_p^e} N^{eT} \bar{p} t dl \quad (3-29)$$

l_p^e 为有外力作用的单元边。

(5) 单元刚度方程

根据最小势能原理求出单元体的刚度方程

$$K_e q^e = P^e \quad (3-30)$$

(6×6) (6×1) (6×1)

(6) 装配集成

上面所求仅仅是一个三角形单元的力学量，要求问题的解，需要将所有划分得到的三角形单元装配在一起，得到整体刚度方程 $Kq = P$ 。

$$q = \sum q^e, K = \sum K^e, P = \sum P^e \quad (3-31)$$

(7) 利用位移和力边界条件求结点位移

$$\begin{bmatrix} K_1 & K_2 \\ K_3 & K_4 \end{bmatrix} \begin{bmatrix} q_u \\ q_k \end{bmatrix} = \begin{bmatrix} P_k \\ P_u \end{bmatrix} \quad (3-32)$$

将整体刚度方程分块处理, q_u 和 P_u 为待求的结点位移和结点力, q_k 和 P_k 为已知的结点位移和结点力。将矩阵形式表示的方程展开, 可以求出未知的结点位移和结点力, 然后根据结点位移场可以求取单元的或整体的其他力学量。

3.4 基于形状匹配的无网格方法

我们知道质点-弹簧模型和有限元模型方法都要进行网格划分, 这是一个耗时且复杂的步骤, 划分得到的网格的质量还直接影响组织变形仿真效果, 同时质点-弹簧模型有不稳定、大变形失真的缺陷, 有限元法算法复杂、实时性较差。本文引入一种基于形状匹配的无网格方法^[48]用于软体的物理建模, 该方法不需要网格划分, 简单易实现、变形逼真度与质点弹簧模型相当但实时性更强、无条件稳定、适用于大变形。

3.4.1 算法基本原理

算法将对象几何模型的每个顶点作为一个粒子, 将对象看做是一个粒子系统, 粒子之间不存在力和连接信息, 每个粒子有单独的质量, 可以受外力、重力、碰撞的作用。

算法所需的输入很简单, 只需粒子的质量与原始位置。其原理很简单, 如图 3.8 所示, 分为两步, 第一步, 根据粒子初始位置 x_i^0 与外界因素作用下当前位置 x_i , 求得最优变换矩阵 (这个过程就是形状匹配), 然后利用初始位置和变换矩阵就可以求出目标位置 g_i ; 第二步, x_i 与 g_i 之间存在弹力, x_i 会被拉向 g_i , 使其运动到目标位置。

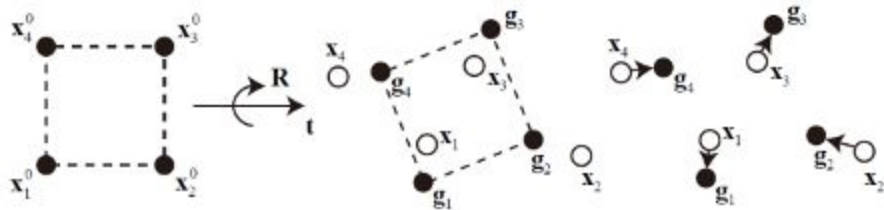


图 3.6 原理图

3.4.2 形状匹配原理

标准形状匹配方法^[49]重点关心如何建立两形状间的对应关系, 但在我们的这个应用环境中形状对应关系是已知的, 我们需要做的是根据已知的对应关系在三维空间中求解出前后两个点云间基于最小二乘法的最佳变换矩阵。

给定两组点云初始位置和变形后的位置，如图 3.6 所示，使式 (3-33) 取得最小值的 R, t, t_0 即为基于最小二乘法的最佳刚性变换。

$$\sum_i w_i (R(x_i^0 - t_0) + t - x_i)^2 \quad (3-33)$$

式中 x_i^0 和 x_i 分别是粒子的初始位置和当前变形位置， w_i 是每个粒子的权重，通常选择粒子的质量作为权重，即 $w_i = m_i$ 。这个公式的含义就是一个物体粒子集合从初始位置经旋转、平移后到达一个新位置，使新位置与实际的粒子集合位置的误差最小的旋转矩阵和平移矩阵作为此物体从初始位置到达目标位置的变换，从而根据物体初始位置和变换矩阵求出物体的目标位置（求出的粒子集合的目标位置和现在实际的位置不是重合的）。

容易求得满足条件的 t, t_0 为：

$$\begin{aligned} t_0 = x_{cm}^0 &= \frac{\sum_i m_i x_i^0}{\sum_i m_i} \\ t = x_{cm} &= \frac{\sum_i m_i x_i}{\sum_i m_i} \end{aligned} \quad (3-34)$$

即 t_0 是初始位置下粒子集合的质心， t 是位置改变后粒子集合的质心。

下面来求最优旋转矩阵 R 。设变量 $q_i = x_i^0 - x_{cm}^0$, $p_i = x_i - x_{cm}$ ，求最优旋转变换 R 可转化为求最优线性变换矩阵 A ，(3-33) 式可化为 (3-35) 式。

$$\sum_i m_i (Aq_i - p_i)^2 \quad (3-35)$$

式 (3-35) 对 A 求偏导，并令 A 的所有偏导系数都为 0，可求得 A ，如 (3-36) 式。

$$A = (\sum_i m_i p_i q_i^T) (\sum_i m_i q_i q_i^T)^{-1} = A_{pq} A_{qq} \quad (3-36)$$

(3-36) 中 A_{qq} 是一个对称矩阵，由对称阵的性质可得其只包含缩放。所以最优旋转矩阵 R 只能包含在 A_{pq} 中，对 A_{pq} 进行极分解得 $A_{pq} = RS$ ，其中 $S = \sqrt{A_{pq}^T A_{pq}}$ ，是对称矩阵，故旋转部分为 $R = A_{pq} S^{-1}$ 。最后，每个粒子的目标位置就可以通过式(3-37)求得。

$$g_i = R(x_i^0 - x_{cm}^0) + x_{cm} \quad (3-37)$$

3.4.3 数值积分

为了获得物体在每一时刻的位置，需要对相关的加速度和速度进行积分运算。在说明本文算法使用的积分方法前，先看一个质点-弹簧系统的例子，因为粒子在实际位置被拉向目标位置的过程跟下面要看到的质点-弹簧系统的运动相像。

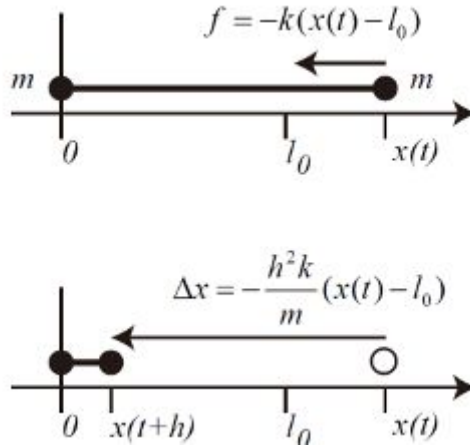


图 3.7 线性质点-弹簧系统

如图 3.7 所示是一个线性无阻尼质点-弹簧系统,弹簧初始长度为 l_0 ，刚度系数为 k 。弹簧连着 2 个点，一个在原点位置，不能运动；另一个在 $x(t)$ 位置，是自由的,质量大小为 m 。自由点将受到大小为 $f = -k(x(t) - l_0)$ 的力向平衡位置 $x(t) = l_0$ 运动。

为求取自由点在每一时间步的位置，需进行如下欧拉积分，

$$\begin{aligned} v(t+h) &= v(t) + h \frac{-k(x(t) - l_0)}{m} \\ x(t+h) &= x(t) + hv(t+h) \end{aligned} \quad (3-38)$$

设状态向量为 $[v \ x]^T$ ，由式 (3-38) 可推导出系统矩阵为

$$A = \begin{bmatrix} 1 & -\frac{kh}{m} \\ h & 1 - \frac{h^2 k}{m} \end{bmatrix} \quad (3-39)$$

经过计算可得其特征值为

$$\begin{aligned} e_0 &= 1 - \frac{1}{2m}(h^2 k - \sqrt{-4mh^2 k + h^4 k^2}) \\ e_1 &= 1 - \frac{1}{2m}(h^2 k + \sqrt{-4mh^2 k + h^4 k^2}) \end{aligned} \quad (3-40)$$

A 表示一个离散系统，A 的谱半径（特征值 e_0 、 e_1 模值大者）只有不大于 1 才能保证系统是稳定的。当 $|e_0| < 1, h^2 k \rightarrow \infty$ 时特征值 e_0 的模值趋于 1，然而，特征值 e_1 的模值只

有当 $h < 2\sqrt{\frac{m}{k}}$ 时才会小于 1。如果积分时间步选取的较大，系统可能就不稳定，所以，上述积分式是有条件稳定的。

用 (3-37) 式计算得出的目标位置构造本文算法要采用的积分方程如下（该积分方程就说明了粒子在实际位置是如何被拉向目标位置的）：

$$\begin{aligned} v_i(t+h) &= v_i(t) + \alpha \frac{g_i(t) - x_i(t)}{h} + h \frac{f_{ext}(t)}{m_i} \\ x_i(t+h) &= x_i(t) + h v_i(t+h) \end{aligned} \quad (3-41)$$

式中 $\alpha = [0...1]$ 是模拟物体刚度的参数，与积分方程 (3-38) 对比可知，二者的差别在于这里用实际位置与目标位置之间的距离代替了力。对于 $\alpha = 1$ ， $\frac{g_i(t) - x_i(t)}{h}$ 项被添加到质点的速度项上，从而 $g_i(t) - x_i(t)$ 被直接增加到位移项上，使得质点从实际位置直接移动到目标位置，表现为刚体；对于 $\alpha < 1$ ，质点将逐渐朝目标位置移动，表现为软体。

对于图 3.9 中显示的弹簧例子，将 (3-41) 式写成状态方程的形式如式 (3-42)

$$\begin{bmatrix} v(t+h) \\ x(t+h) \end{bmatrix} = \begin{bmatrix} 1 & -\alpha/h \\ h & 1-\alpha \end{bmatrix} \begin{bmatrix} v(t) \\ x(t) \end{bmatrix} + \begin{bmatrix} \alpha l_0/h \\ \alpha l_0 \end{bmatrix} \quad (3-42)$$

则系统矩阵为 $\begin{bmatrix} 1 & -\alpha/h \\ h & 1-\alpha \end{bmatrix}$ ，其特征值为 $(1-\alpha/2) \pm i\sqrt{4\alpha-\alpha^2}/2$ 。二者的最大模值均

为 1，与 α 和 h 无关。这说明此积分式是无条件稳定的，从而说明此无网格方法是无条件稳定的，同时未引入阻尼。这同样适用于无外力作用的一般三维实体的情况。物体受到的外力只要与质点所在位置无关（如重力），或者作用时间很短（如碰撞力），则系统矩阵就不会改变，系统也是稳定的。

3.4.4 算法扩展

上面描述的方法的基本形式适用于刚体和小变形对象的模拟，还不适用于大变形对象，下面对基本形式进行变形扩展其适用性。

1. 线性变形

将 (3-37) 中求 g_i 的旋转矩阵替换为 $\beta A + (1-\beta)R$ ，其中 β 是小于 1 的控制参数。因为 A 为求得的最小二乘意义上的初始形状匹配实际形状的最优线性变换矩阵，所以利用上述替换后，目标形状允许出现线性变形，从而可以模拟线性形变，同时由于 R 的存在可以保证可以恢复到初始形状。为了确保变形后体积守恒，将 A 除以 $\sqrt[3]{\det(A)}$ ，从而保证

$\det(A)=1$ 。

2. 二次变形

上述的扩展只允许对象出现剪切和拉伸形变，为了使其能进行扭曲和弯曲变形，构造二次变换矩阵。通过下面的式子定义二次变换矩阵并求粒子的目标位置

$$g_i = [A \quad Q \quad M] \tilde{q}_i \quad (3-43)$$

其中 $g_i \in R^3$, $\tilde{q} = [q_x, q_y, q_z, q_x^2, q_y^2, q_z^2, q_x q_y, q_y q_z, q_z q_x]^T \in R^9$, $A \in R^{3 \times 3}$ 中是线性项的系数, $Q \in R^{3 \times 3}$ 是完全二次项的系数, $M \in R^{3 \times 3}$ 是复合项的系数。现在要使式 (3-44) 的值最小,

$$\sum_i m_i (\tilde{A} \tilde{q}_i - p_i)^2 \quad (3-44)$$

经计算得最优二次变换矩阵为

$$\tilde{A} = \left(\sum_i m_i p_i \tilde{q}_i^T \right) \left(\sum_i m_i \tilde{q}_i \tilde{q}_i^T \right)^{-1} = \tilde{A}_{pq} \tilde{A}_{qq} \quad (3-45)$$

类似于前述的线性变形情况，利用 $\beta \tilde{A} + (1-\beta) \tilde{R}$ 代换原来的 R ，其中 $\tilde{R} \in R^{3 \times 9} = [R \quad 0 \quad 0]$ 。

利用改进后的积分式可以得到的线性和二次变形效果如图 3.8。

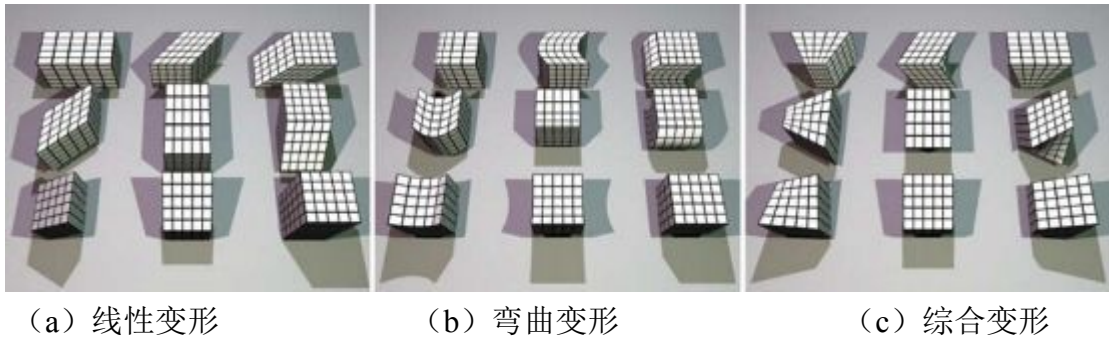


图 3.8 变换矩阵变形后的应用效果

3. 基于簇的变形

为了进一步扩展变形的幅度，可以将粒子划分为簇，彼此之间存在交叠部分。对于体模型对象，将模型中的各个顶点作为粒子，将与同一个单元（如四面体）相邻接的粒子作为一个簇。如果我们使用面模型，我们将面模型内部划分为彼此有重叠部分的立方体，将每个立方体包含的所有顶点作为一个簇。

在每个积分步，每个簇独立的用其自身的初始形状匹配其实际形状，然后将求出的变换矩阵和平移向量应用于这个簇中的所有粒子，单独计算每个簇的变形，整合在一起形成总体的变形。

图 3.12 显示了三个拥有 60 个质点但划分为不同数量簇的柱形体二次变形情况，三个图中划分的簇数依次为 1、2 和 5，施加的力的位置、方向相同，从图中可看出划分的

簇数越多，变形细节体现的就越好，变形幅度也越大。



(a) 1 个簇 (b) 2 个簇 (c) 5 个簇

图 3.9 不同簇数对对象变形灵活性的影响

3.4.5 实验

文中方法的时间性能主要取决于形状匹配的粒子数量和划分的簇数。为了定性说明进行如下实验，对一定数量的粒子随机安排其初始位置及目标位置，然后选取不同的粒子数，不同的簇数，进行线性变形和二次变形实验。图 3.13 显示了实验结果，从图中可以看出不论采用哪种方案，计算复杂度与参与形状匹配的质点数都是呈线性关系。同时划分簇数越多，因为需要的极分解越多，所以计算代价也越大。此外，二次变形比线性变形复杂，计算代价也相应增加。在实验中，100 个包含 100 个参与形状匹配粒子，划分成 8 个簇的对象进行二次变形 fps 值可以达到 50。

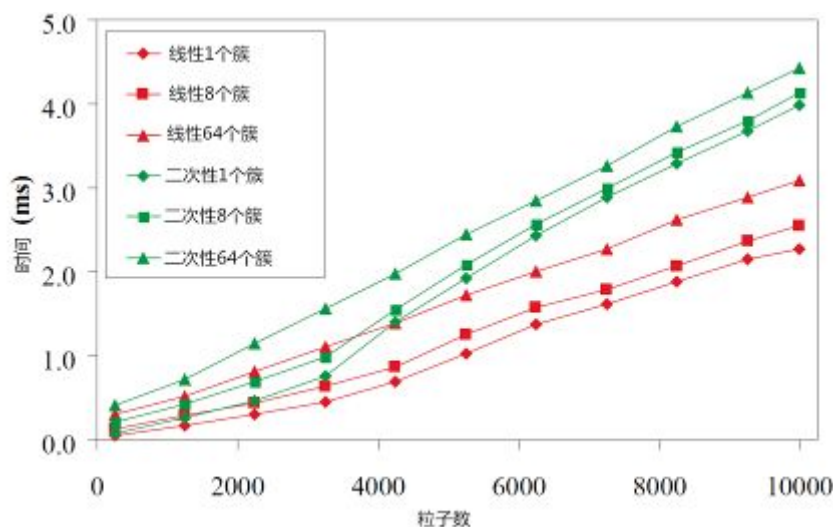


图 3.10 不同实验方案下参与形状匹配粒子数与每一帧计算耗费时间关系

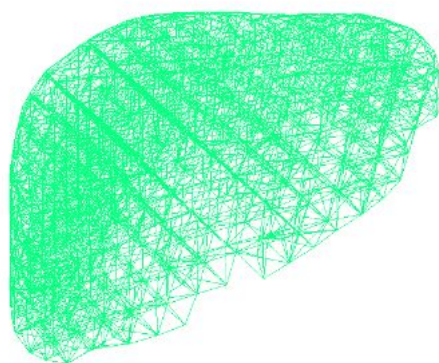
3.5 物理建模方法对比实验

对物理建模方法效果的评价主要集中在实时性、逼真度、稳定性方面。我们利用第二章建立的肝脏体模型分别针对质点-弹簧、有限元和形状匹配无网格法进行变形实验。实验硬件环境为个人电脑，配置为 CPU: Intel Core i3-2100 @ 3.1GHz、GPU: NVIDIA

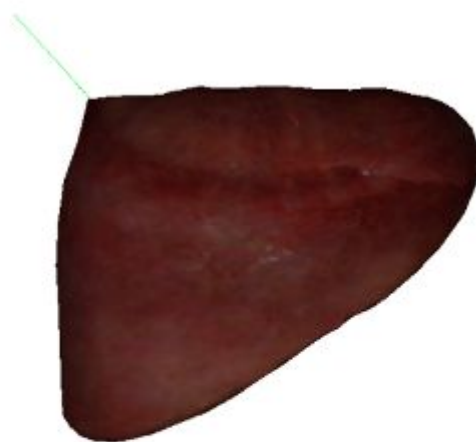
GeForce GTS450、内存：6G。软件环境：VS2010 集成开发环境，C++和 OpenGL 编程语言。

1. 实时性对比

质点弹簧刚度系数设为 180，阻尼系数设为 5，弹簧使用四面体形式的连接方式；有限元方法的弹性模量设为 3000，泊松比设为 0.3，单元使用四面体单元；基于形状匹配的无网格方法的簇数设为 50，区域半径设为 1，参与形状匹配粒子数为 96， α 设为 0.72， β 为 0.63，使用二次变形方法。参数进行如上设置后可以比较好地模拟肝脏的变形，实验截图如下：



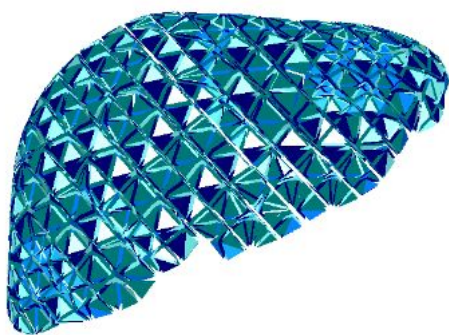
(a) 弹簧连接网络



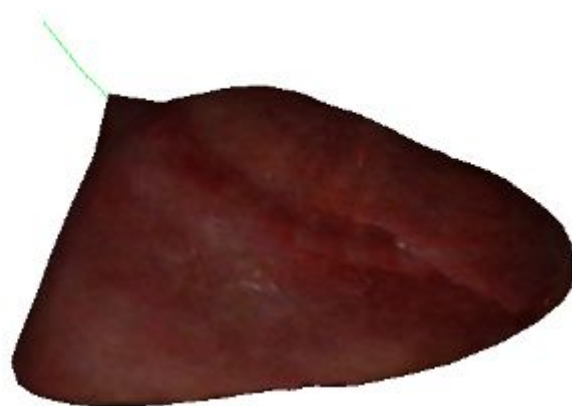
335.7 FPS

(b) 拉伸变形效果及 FPS 值

图 3.11 质点--弹簧模型方法效果



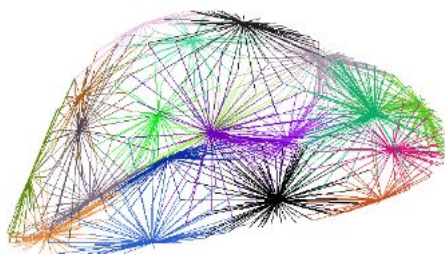
(a) 有限元四面体单元剖分



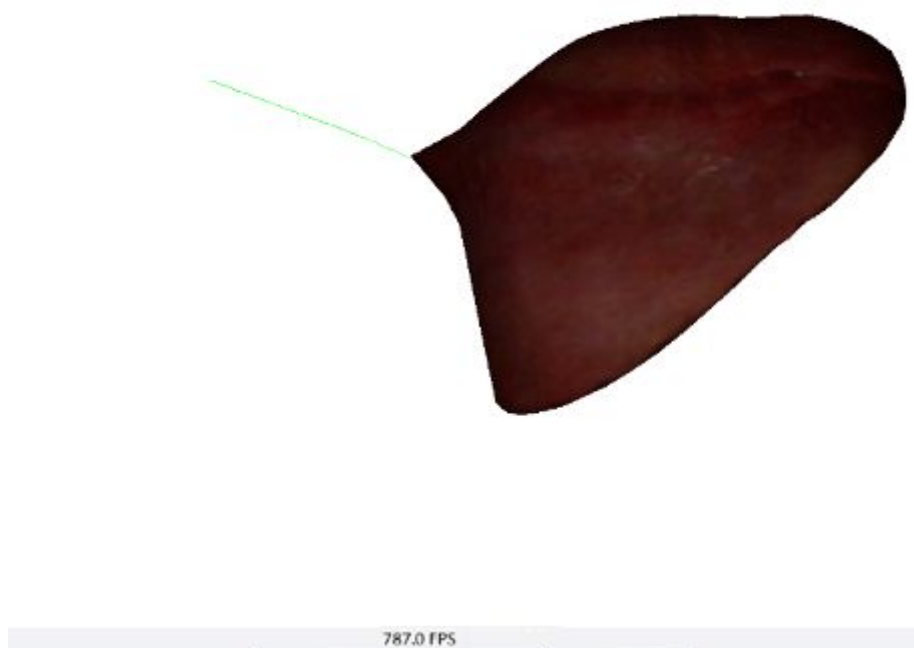
36.5 FPS

(b) 拉伸变形效果及 FPS 值

图 3.12 有限元方法效果



(a) 簇划分



(b) 拉伸变形效果及 FPS 值

图 3.13 基于形状匹配无网格方法效果

从实验结果可看出，在具有相似的变形逼真度的情况下，有限元法的 FPS 值为 36.5，质点-弹簧法的为 335.7，而形状匹配无网格法的为 787，本文算法的实时性占很大优势。

2. 稳定性、大变形对比

在实验过程中使用本文形状匹配无网格法对肝脏无论如何变形，变形效果在视觉上都不会出现畸变，并且都能恢复原形，在上面的原理介绍中也在理论上说明了其无条件稳定性。



图 3.14 形状匹配无网格任意变形效果

而质点弹簧模型在大变形情况下出现变形效果不真实，并且无法恢复，如图



图 3.15 质点-弹簧方法变形失真并且无法恢复

由于有限元模型是基于小变形假定下的方法，不适用于大变形情况。所以形状匹配无网格法在大变形、稳定性方面有优势。

综上所述，本文基于形状匹配的无网格方法在实时性、稳定性和大变形适用性方面表现很好。

3.6 本章小结

本章主要对软体物理建模方法做了研究。先对基于网格的质点-弹簧模型和有限元模型做了介绍，然后重点对本文选用的基于形状匹配的无网格方法做了阐述。虽然此模型是基于几何驱动而不是基于物理属性驱动的，但其在表现变形的逼真度方面的能力与质点-弹簧方法还是相当的。相对于质点-弹簧模型和有限元模型方法，其原理简单、易实现，不需要网格划分，不需要复杂的数据结构，通过对比实验说明，本文方法的仿真效率很高；同时算法针对任何的变形操作都具有无条件稳定性，适用于大变形。

第4章 虚拟手术仿真中的碰撞检测算法

4.1 引言

虚拟手术系统要给人以真实感,就要在视觉和交互操作上做到当人控制手术器械接近操作对象时器械不能穿越组织,对对象做出提拉、按压、切割、缝合操作时组织要立即做出相应反应,这就需要有一种方法来判断器械与组织是否接触,这就是碰撞检测算法要做的事情。碰撞检测在虚拟手术系统中主要完成手术器械与软体之间,软体自身之间的碰撞检测。准确性与实时性是其关键指标。涉及软体的碰撞检测不同于刚体与刚体之间的碰撞检测^[50],因为软体本身结构复杂,在外力作用有形变产生,甚至拓扑结构的变化,这都增加了软体碰撞检测的难度。本章首先对碰撞检测算法做了概述,然后,根据单一层次包围盒法在相交三角形数量变化情况下碰撞检测速度变化特点,提出 Sphere-AABB 复合层次包围盒法用于本文碰撞检测算法的预检测阶段,同时,对于海燕等人^[51]提出的基于投影降维的三角形间相交测试算法中的投影情况划分、三角形相离情况排除、三角形空间位置奇异情况处理和三角形线裁剪算法方面做了改进,并用于本文碰撞检测算法的详细检测阶段,最后通过实验来验证本文算法的效率。

4.2 碰撞检测算法概述

碰撞检测过程通常可划分为预处理和实时检测两个阶段,实时检测阶段又可分为预检测和详细检测^[52]。

预处理阶段,依据不同的应用环境选择适当的碰撞检测方法对待检测对象建立相应的数据结构。实时检测阶段执行检测任务,预检测阶段的主要负责排除整个场景中不可能发生碰撞的区域或对象,逐步缩小进行检测的范围。详细检测进行基本图元间的碰撞测试,求取碰撞的位置、深度等具体信息。

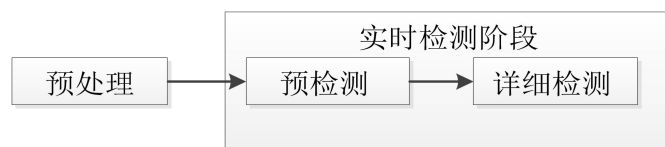


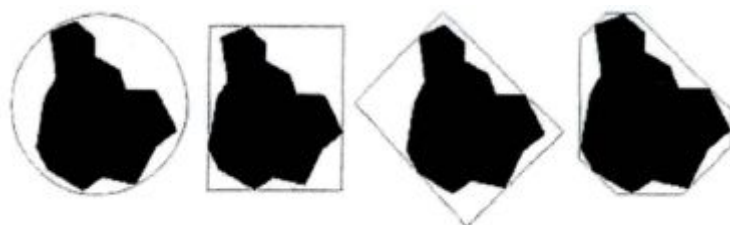
图 4.1 碰撞检测流程

为了增强碰撞检测的实时性主要有 2 种方案,一是在预检测阶段尽量多的排除不相交的对象,降低需要测试的基本图元的数量,二是在详细检测阶段提升基本图元间碰撞测试的效率。

层次包围盒法是减少参加测试的基本图元数量的一种常用方法,其基本思想就是利用大包围盒包围整个对象,然后逐步细分对象用相应包围盒包围,直到对象细分到一定

程度或到达基本图元，包围盒组织成树形结构，通过简单几何体间的碰撞检测排除不相交对象，同时缩小检测范围，最终利用叶子节点包围盒的碰撞检测结果代替目标对象间的碰撞结果或者对双方叶子节点包围盒内部的基本图元进行相交测试以求取精确碰撞结果。

包围盒是指形状简单体积略大用于包围目标对象的几何体，常用包围盒如下：



(a) Sphere (b) AABB (c) OBB (d) k-DOP

图 4.2 包围盒类型

包围球（Sphere）是能包围目标对象的体积最小的球体，其优点是构建和相交测试简单，相交测试只需计算两球心距离跟两球半径和的大小关系即可，对于彼此分离的对象，进行分离排除实时性很高；对象进行旋转时，包围球不必更新；但其包围紧密性较差。

轴向包围盒（Axis Aligned Bounding Boxs, AABB）是包围目标对象的体积最小的与坐标轴对齐的六面体，在碰撞检测中应用最多，其优点是构造简单，计算量小，碰撞检测、对象运动或变形后的更新都相对容易，在软体碰撞检测中多有应用；但包围紧密性不是很好。

方向包围盒（Oriented Bounding Boxs, OBB）是指包含目标对象的相对于坐标轴方向任意的体积最小的正六面体^[53]，因其轴向与坐标轴方向没有对应关系，可以按照对象的外形最大限度的包紧对象，其紧密程度比 Sphere 和 AABB 要强一些，从而可以减少参与碰撞检测的包围盒数量，对象发生旋转后，OBB 只需完成相同的旋转就能更新，在刚体的碰撞检测应用中，实时性优于 Sphere 和 AABB。但其构造是比较复杂的，如果对象发生变形，包围盒需要重新构建，这个过程所花费的时间抵消了其紧密性所带来的优势，所以在软体碰撞检测中应用较少。

k-DOP（Discrete Orientation Polytopes）包围盒是选取 k 个固定方向来构建的包围目标对象的体积最小的凸多面体，能够把它看成是 AABB 的延伸，它采用 $k/2$ 对平面替换 AABB 中的三对来包围对象，这样做的结果就是弥补了 AABB 紧密性差的缺陷，但其构建和相交测试相对 AABB 复杂。

软体碰撞检测中因为软体组织结构复杂，基本图元数多，并且会发生变形，所以要求包围盒一定要构建和更新简单，这有利于构建大量包围盒和变形后包围盒重构或更新，根据上述包围盒的特点，满足这一点的只有 Sphere 和 AABB。Sphere 相交测试比

AABB 简单, 但包围紧密性不如 Sphere, 分离排除效率很高, 所以在相交的基本图元数较少的环境中 Sphere 的效率高, 而在相交的基本图元数多的环境中就不如 AABB 了。而在软体碰撞环境中因为软体的变形相交基本图元数量是变化的, 并且变化范围可能还很大, 单独使用上述中的一种包围盒无法在任意相交数量下保持最高的效率 (虽然单独使用 AABB 可以满足系统实时性要求), 为了在任意基本图元相交数量下保持最高的检测效率, 本文提出 Sphere-AABB 复合层次包围盒, 利用 Sphere 和 AABB 各自的优点而规避各自的缺点, 以提高碰撞检测的总体性能。

4.3 复合层次包围盒

Sphere-AABB 复合层次包围盒将两种包围盒集中在一棵树中, 利用在不同碰撞概率下使用不同包围盒的思想, 来提升实时性。

4.3.1 复合层次包围盒树概述

将构建的层次包围盒分为三层, 如图 4.3。上层用 Sphere, 中间层用 Sphere-AABB 复合包围盒, 下层用 AABB 包围盒。之所以插入中间层 Sphere-AABB 使层数为三层, 是为了使层次间包围盒的紧密程度渐近变化, 避免从上层到下层相交检测时间复杂度剧变。复合层次包围盒法的意义在于, 综合了 Sphere 快速排除不相交对象和 AABB 相对紧密包围对象、相对准确判断对象相交的性能。

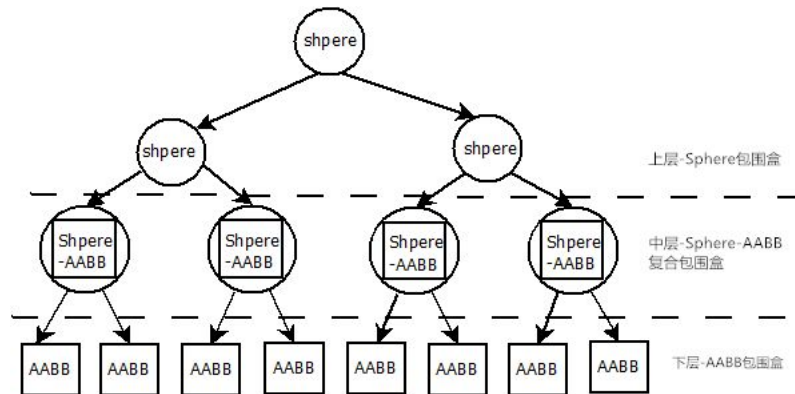


图 4.3 复合层次包围盒结构

因为每层包围盒各有特点, 根据应用环境通过调整上中下各层的层数, 可以达到不同的碰撞检测速度。

4.3.2 复合层次包围盒树算法流程

算法流程如图 4.4 所示, 其中 a_i, b_j 分别是对象 A 和 B 层次包围盒树上的结点号, Flag 为是否碰撞的标识, Sphere_Sphere, Sphere_AABB, AABB_AABB 是不同的包围盒相交测试函数, Box() 是参数结点对应的包围盒, Child() 是参数结点对应的子节点。

当遍历对象 A 的上层节点, 对象 B 的上层节点时, 由于包围盒均为 Sphere, 紧密

性较差, 采用 Sphere_Sphere 检测函数做相交测试, 当遍历对象 A 的上层节点, 对象 B 的中层节点, 或者遍历对象 A 的中层节点, 对象 B 的上层节点, 或者遍历对象 A 的中层节点, 对象 B 的中层节点时, 依然使用 Sphere_Sphere 检测函数进行相交测试, 因为这两层的盒子紧实性都不太好, 并且中间层拥有混合包围盒结构, 可以选取 Sphere 包围盒, 如果对象 A、B 分离可以快速排除。

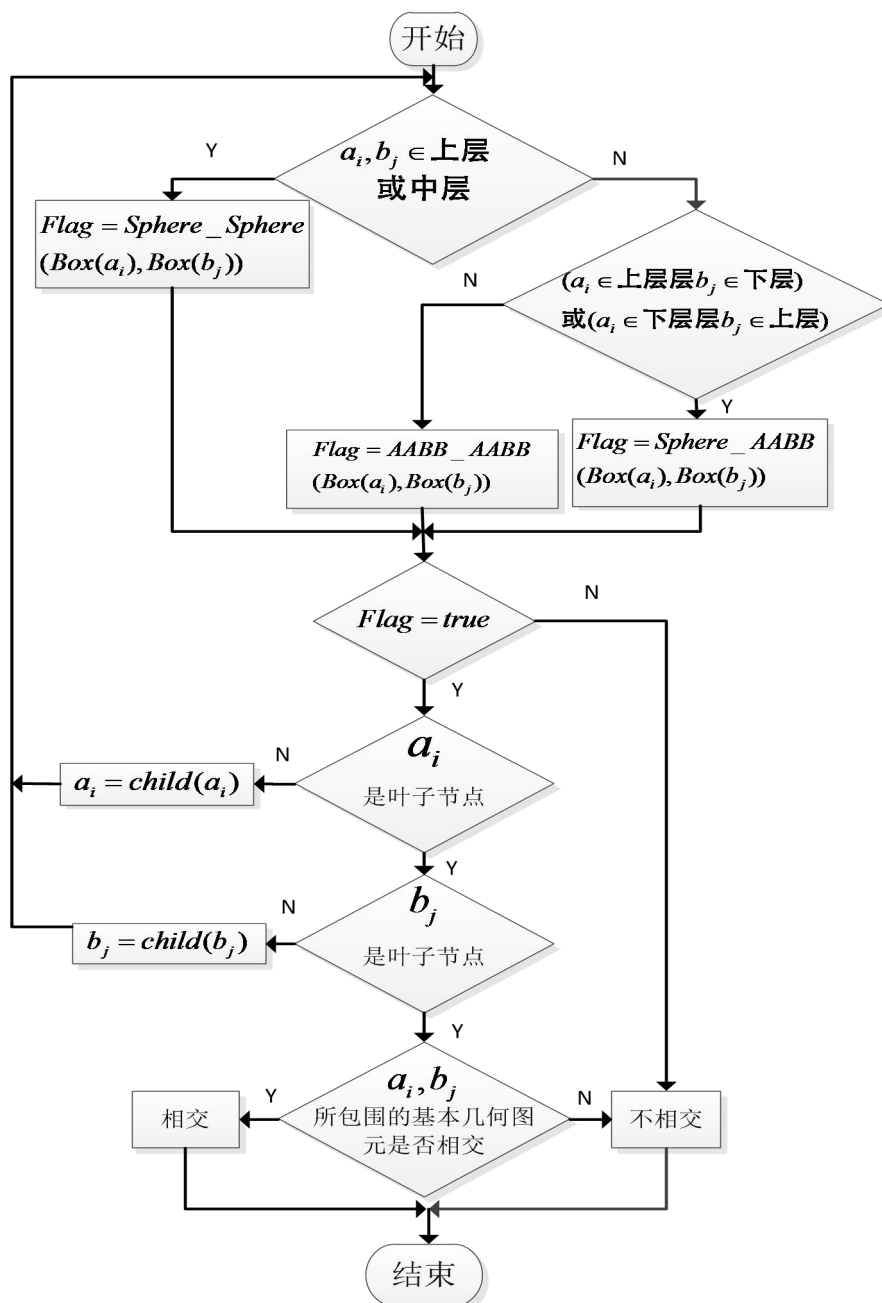


图 4.4 复合层次包围盒算法流程

当遍历对象 A 的上层节点, 对象 B 的下层节点, 或者遍历对象 A 的下层节点, 对象 B 的上层节点时, 使用 Sphere_AABB 检测方法。

当遍历对象 A 的中层、下层节点, 对象 B 的下层节点, 或者遍历对象 A 的下层

节点, 对象B的中层、下层节点时, 使用AABB_AABB检测方法。因为中层是复合结构, 能够取AABB包围盒, 在这一层碰撞的可能性也提高了。

当遍历到两棵树的叶子节点包围盒相交, 则对叶子节点内基本图元集(三角形)做相交检测, 如果存在两三角形对相交, 则A和B相交, 否则相离。对于三角形之间的碰撞测试将在后面进行介绍。

4.3.3 复合层次包围盒的构建

复合层次包围盒法为给定的对象上层建立Sphere包围盒, 中层建立Sphere-AABB包围盒, 下层建立AABB包围盒。因为考虑到Sphere包围盒剖分时无法确定分离轴, 需要借助AABB包围盒作为辅助结构, 所以上层和中层都建立Sphere-AABB复合结构, 下层只建立AABB结构。每层的深度根据对象的特点, 碰撞检测的效果要求进行预设, 在调试时进行调整, 使碰撞检测达到良好的效果。

1. Sphere 和 AABB 的构建

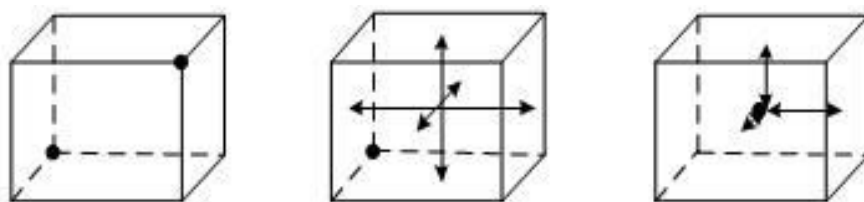
Sphere-AABB复合结构是在AABB建立完后, AABB再外接一个Sphere。Sphere的球心位于AABB的中心, 半径为 $R = \sqrt{a^2 + b^2 + c^2} / 2$, 其中 a, b, c 是AABB的边长^[54]。

AABB有3种表达方式:

Min-Max表示法: 使用坐标最小与最大的两个角点来唯一的定义一个包围体。

Min-Width表示法: Min表示坐标值最小的角点, width存储AABB在x, y, z坐标轴方向上的边长。

Center-Radius表示法: Center表示AABB中心, radius存储AABB在x, y, z坐标轴方向上的半径。



(a) Min-Max

(b) Min-Width

(c) Center-Radius

图4.5 AABB包围盒的常用表示方法

不同的表示方法的碰撞检测算法有差异, 同时适用情形也不一样。文中选用“Min-Max”方式, 因为这种表示法求取和碰撞检测都比较简单。

对一个基本图元集合建立一个包围盒步骤: 集合中图元的顶点向各个方向轴(x, y, z)上投影(用向量的点乘运算即可), 获取各个轴向上投影最小值(min_x, min_y, min_z)和最大值 max_x, max_y, max_z, 这样就获得了“Min-Max”表示方式的AABB包围盒。

2. Sphere、AABB 相交测试

(1) Sphere 与 Sphere 间的测试最为简单，只需比较球心之间的距离与两 Sphere 的半径和大小即可。

(2) AABB 与 AABB 间的测试也相对容易，只要 3 个轴向上都相交，AABB 就相交。

(3) Sphere 与 AABB 的碰撞检测

首先要求解点到 AABB 的最近点问题：假设 B 是一个 AABB 包围盒， P 为空间内一点，求 B 上离 P 最近的点 Q 。

Q 可能的情况如下，这里用到了 Voronoi 域概念，详情请参考文献[54]，一个 AABB 包围盒的相关 Voronoi 域如图 4.6 所示。

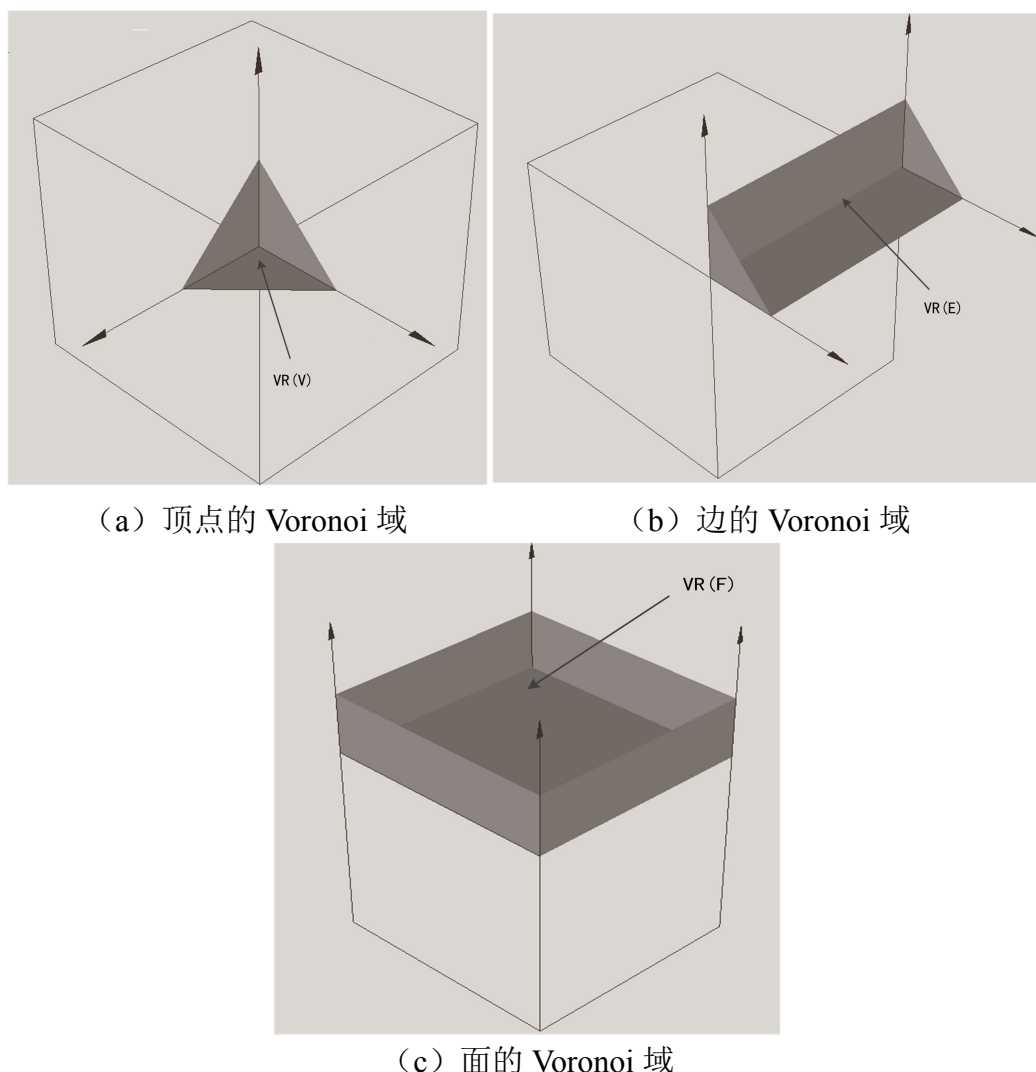


图 4.6 一个 AABB 包围盒的相关 Voronoi 域

- (1) 如果 P 在 B 内部，则 Q 即为 P 。
- (2) 如果 P 在 B 的某一顶点 Voronoi 域内，则此顶点即为 Q 。
- (3) 如果 P 在 B 的某一边 Voronoi 域内，则 P 在此边上的正交投影即为 Q 。
- (4) 如果 P 在 B 的某一面 Voronoi 域内，则 P 在此面上的正交投影即为 Q 。

以“Min-Max”法表示的 AABB 包围盒 B ，求在 B 上距离空间内一点 P 最近的点 Q 代码：

```
// Given point p, return the point q on or in AABB b that is closest to p
void ClosestPtPointAABB(Point p, AABB b, Point &q)
{
    // For each coordinate axis, if the point coordinate value is
    // outside box, clamp it to the box, else keep it as is
    for (int i = 0; i < 3; i++)
    {
        float v = p[i];
        if (v < b.min[i]) v = b.min[i];
        if (v > b.max[i]) v = b.max[i];
        q[i] = v;
    }
}
```

然后就可以求出点 P 到 AABB 的最近距离，求此距离时 Q 其实不必显示求出，当然为了减少计算代价，这里距离采用平方形式，代码如下：

```
// Computes the square distance between a point p and an AABB b
float SqDistPointAABB(Point p, AABB b)
{
    float sqDist = 0.0f;
    for (int i = 0; i < 3; i++) {
        // For each axis count any excess distance outside box extents
        float v = p[i];
        if (v < b.min[i]) sqDist += (b.min[i] - v) * (b.min[i] - v);
        if (v > b.max[i]) sqDist += (v - b.max[i]) * (v - b.max[i]);
    }
    return sqDist;
}
```

最后就可以解释 Sphere 与 AABB 如何进行碰撞检测了，利用前两步求出 Sphere 球心到 AABB 的最近距离与 Sphere 的半径相比较就可以得到判断结果，如果距离比半径小，二者相交。代码如下：

```
// Returns true if sphere s intersects AABB b, false otherwise
int TestSphereAABB(Sphere s, AABB b)
```

```

{
// Compute squared distance between sphere center and AABB
float sqDist = SqDistPointAABB(s.c, b);
// Sphere and AABB intersect if the (squared) distance
// between them is less than the (squared) sphere radius
return sqDist <= s.r * s.r;
}

```

3.包围盒的计算与更新

包围体一般在对象的局部模型空间中建立，为了进行两个包围体间的碰撞检测，一定要将它们变换到一个相同的坐标系下。方法一，把二者均转换到世界空间坐标系；方法二，把一个包围体转换到另一个包围体所在的局部空间坐标系^[55]。

下面介绍一下当对象运动时 **AABB** 的更新。

对象的运动可以分解为平移运动和绕自身中心的旋转运动，针对平动只需对包围体的 **Min** 和 **Max** 两个点的坐标做相应平移，针对转动常用的更新方法有下面几种：

（1）基于原点的 **AABB** 重构方式

此方法的思想是通过遍历包围体内的基本几何图元的顶点，找到在三个坐标轴方向上的最值点（最小值和最大值），从而重新确定包围体的尺寸。

（2）爬山法 **AABB** 重构方式

该方法利用 6 个顶点指针指向各轴向上物体的端点，当物体发生旋转后，将原来轴向上的端点与它的邻接顶点进行比较，判断其是否仍为此轴向上的端点，如果不是，则用相应的邻接顶点替换原来的端点，重复该过程直至得到此轴上的端点。为了避免在局部最小值处阻塞，使用爬山法的物体必须是凸体，如果对非凸体运用必须先进行预处理。此法时间复杂度为 $O(1)$ 。

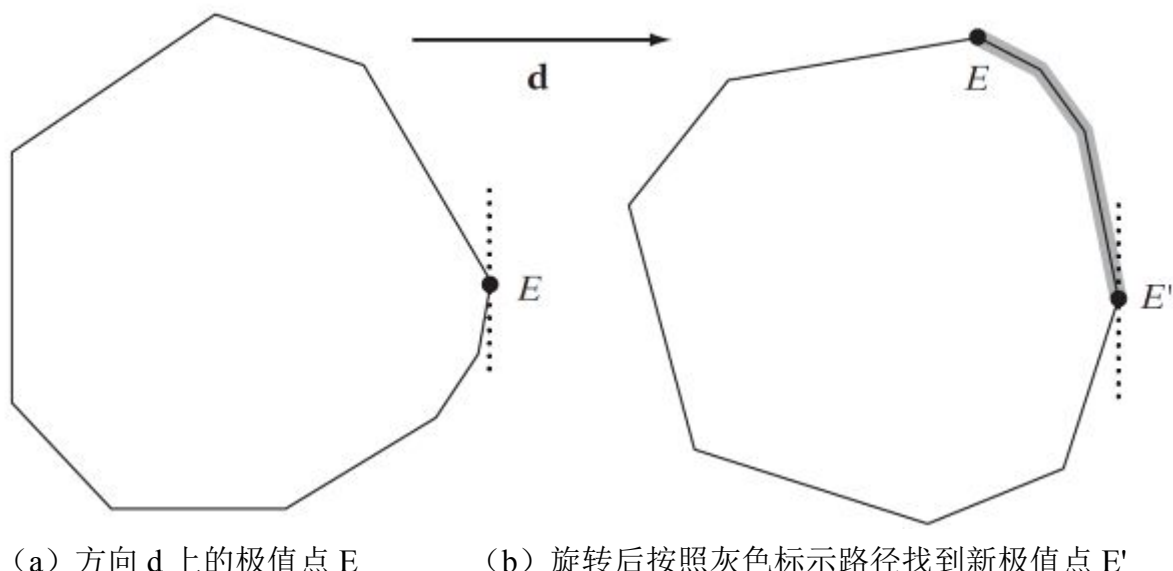


图 4.7 爬山法 AABB 重构方式

(3) 基于旋转后的 AABB 的重构方式

此法的思想是构建一个新 AABB 去包围旋转后的 AABB，这样重构出的包围盒仅仅是逼近包围盒^[55]。需要注意的是，每次重构必须在初始 AABB 所在的坐标系下针对初始 AABB 计算旋转，否则，当结果 AABB 大于初始 AABB 时，针对上一个时间步内的、旋转后的 AABB，重复旋转计算将使得 AABB 无限增长。

若包围盒采用"Min-Max"表示形式，一个 AABB 包围盒为 A ，一个旋转矩阵 M 作用其上， M 为列矩阵形式（ M 的列向量表示 A' 所在的局部坐标系的坐标轴），旋转后的包围盒记为 A' ，包围 A' 的重构包围盒记为 B 。 B 的 Min 和 Max 顶点由 A' 顶点在 A 所在的坐标系的各个坐标轴上的最小和最大投影确定。将 A 的顶点表示为列向量 α ，则对应 A' 的顶点为 $M \times \alpha$ ，则对于 A' 的顶点坐标中的 x 分量，只有矩阵 M 列向量中的 x 分量对其产生影响。 B 的 Min 和 Max 顶点的 x , y , z 分量都是一个组合值，是 3 个最小值或最大值的组合。例如 B 的 Max 顶点的 x 分量求法如下：

$$\begin{aligned} B.\text{max}[0] = & \max(m[0][0] * A.\text{min}[0], m[0][0] * A.\text{max}[0]) \\ & + \max(m[0][1] * A.\text{min}[1], m[0][1] * A.\text{max}[1]) \\ & + \max(m[0][2] * A.\text{min}[2], m[0][2] * A.\text{max}[2]) + t[0]; // t[0] \text{ is translation on } x\text{-axis} \end{aligned}$$

类似的可以得到 B 的求取方法如下：

```
// Transform AABB a by the matrix m and translation t,
// find maximum extents, and store result into AABB b.
void UpdateAABB(AABB a, float m[3][3], float t[3], AABB &b)
{
    // For all three axes
    for (int i = 0; i < 3; i++) {
```

```

// Start by adding in translation
b.min[i] = b.max[i] = t[i];
// Form extent by summing smaller and larger terms respectively
for (int j = 0; j < 3; j++) {
    float e = m[i][j] * a.min[j];
    float f = m[i][j] * a.max[j];
    if (e < f) {
        b.min[i] += e;
        b.max[i] += f;
    }
    else {
        b.min[i] += f;
        b.max[i] += e;
    }
}
}
}
}
}

```

如果对象只运动而不变形(如手术器械,软体与手术器械接触前发生运动)其 AABB 包围盒的更新一般选择方法(3), 如果对象有变形发生时(如软体与手术器械接触发生形变), 其 AABB 包围盒的更新一般选择方法(1)。Sphere 的更新则只需在新的 AABB 的基础上计算中心与半径。

4. 树的构建

常用构建方法有两种, 一是自顶向下, 二是自底向上。自顶向下方法, 以构建二叉树结构为例, 首先构建一个包围整个对象的包围盒作为根结点, 然后将整个对象分为俩部分, 对这两个部分分别建立包围盒(同构建根节点那样), 作为根结点的左右子结点, 然后把左右子结点当做根节点那样重复上面的处理过程, 直到满足要求不再分割。自底向上方法, 先对对象的基本图元(单个或多个)建立包围盒作为叶子结点, 再依据基本图元所处的空间位置进行递归划分组建上层包围盒, 直到根节点的包围盒包围整个对象。本文采用构建技术更成熟的自顶向下方法。

层次包围盒树的度数(父节点能够拥有的最大子节点数量)是影响碰撞检测效率的一个重要因素, 一般情况下, 树的度数和树的高度是对立关系, 树的度数高, 高度就小, 向下搜索子节点就越复杂, 相反, 树的度数小, 向下搜索子节点会容易, 但树的高度会相应增加。本文选用二叉树, 因为其最容易构建, 节点剖分简单; 从根节点访问到叶节点其访问速度也最快。证明如下, 对具有 N 个叶节点度数为 μ 的完全树而言, 从根节点

到叶节点平均访问次数可由式(4-1)计算。

$$f(\mu) = \mu \log_{\mu} N \quad (4-1)$$

由相关数学知识可知, 对于 $\mu \in (1, +\infty)$, 当 $\mu = e$ 时函数可取到最小值, 因为度数要取整数值, 所以当 $\mu = 2$ 时函数取最小值。

在建立包围盒树的过程中需要不断剖分父节点, 如图 4.7, 这需要遵循一定的原则: 原则一, 剖分后子节点中拥有的基本图元数要比父节点中的数量少, 否则剖分没有意义。原则二, 剖分后子节点所处深度要小于系统预设的最大深度。原则三, 剖分后子节点中包含的基本图元数要大于等于系统预设的最小个数。设定原则二和三是因为剖分不必过细, 有时候剖分过细反而会增加碰撞检测时间。每次剖分时要检查是否满足此三原则, 不满足时停止剖分。

本文使用自顶向下的方法建立二叉树形式的复合层次包围盒, 这需要不断剖分构成对象的图元(三角形)集合, 如何确定剖分平面是关键问题。剖分平面由分裂点和面法线就可确定, 面法线所在的直线称为分离轴, 下面分别对二者做介绍。

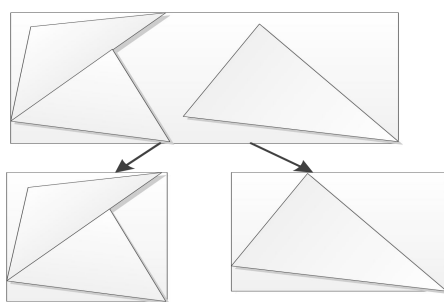


图 4.8 节点剖分

(1) 分离轴的选择

为了使分裂后两个子节点重叠部分尽可能小, 选取的分离轴应该使父节点内的基本图元在其上尽可能的分布均匀。本文中涉及的节点分裂是有关 AABB 包围盒的, 以其为例, 常用的方法有:

- 1) 最长方向轴法: 三个坐标轴方向中基本图元分布最长的方向选作分裂轴。
- 2) 最大方差轴法: 将节点内的基本图元的重心向三个坐标轴进行投影, 分别计算每个轴上投影的方差情况, 方差最大的轴选作分裂轴。
- 3) 子包围盒体积和法: 选取的轴应当使分裂后两子节点包围盒的体积和最小。

其中, 最长方向轴法计算最为容易, 方法 2) 3) 计算量依次增大。本文选择最长方向轴法。

最长方向轴法, 假设父节点中基本图元集合为 S , 图元 $T_k \in S (k=1,2,3...)$, G_k 记为图元的重心, 将 G_k 向坐标轴投影, 投影记为 R_k , 如图 4.9。

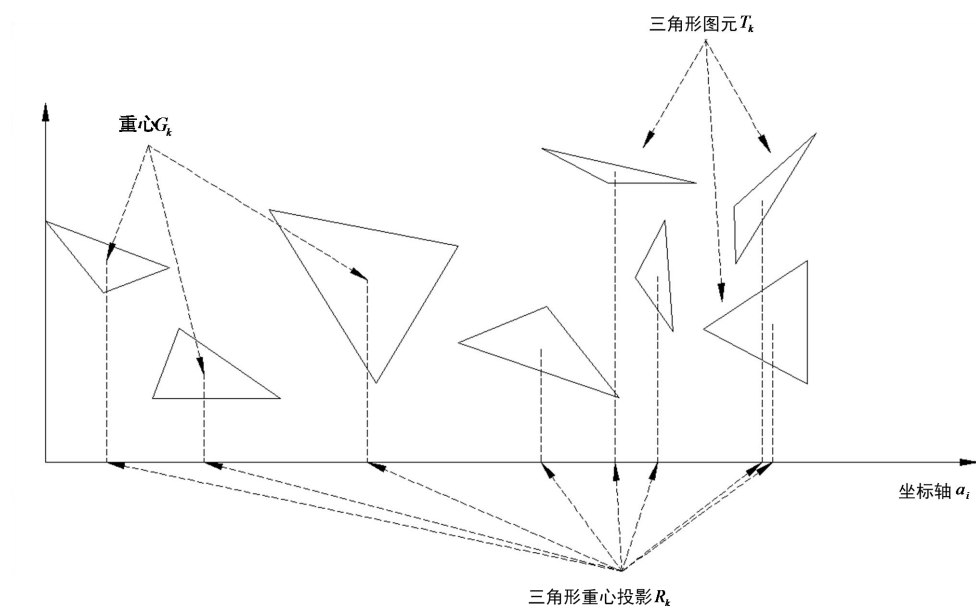


图 4.9 三角形重心向坐标轴的投影

在 R_k 中找到最大值 $Max(R_k)$ 和最小值 $Min(R_k)$ ，从而可以得到投影长度 l_i ：

$$l_i = Max(R_k) - Min(R_k) \quad (i = 1, 2, 3) \quad (4-2)$$

l_i 取最大值所在的轴向即为分离轴。

(2) 分裂点的选择

分裂点用于定位分裂平面，分裂点的确定对包围盒树的对称性有一定影响，继而对节点的搜索效率产生影响。常用的分裂点选取方法有均值法与中值法。

1) 均值法

均值法选择分离轴上父节点的所有图元质心投影 R_k 的均值作为分裂点，投影位于分裂点左边的图元划归左子节点，位于右边的划归右子节点，位于分裂点上的划归图元数目少的一边，如果数目相同划归左子节点。此方法计算简单，但分割后左右子节点中图元数目可能有差异，影响树的对称性。

2) 中值法

中值法选择分离轴上父节点的所有图元质心投影 R_k 的中值作为分裂点，投影位于分裂点左边的图元划归左子节点，位于右边的划归右子节点，位于分裂点上的划归左子节点。此方法可以使树具有绝对的对称性。

下面对节点的剖分试举一例。假设父节点中的图元如图 4.10 所示，图中投影轴即为按照最长方向轴法选取的分离轴，基本图元已经编号，也标记出了图元质心投影的均值点和中值点。

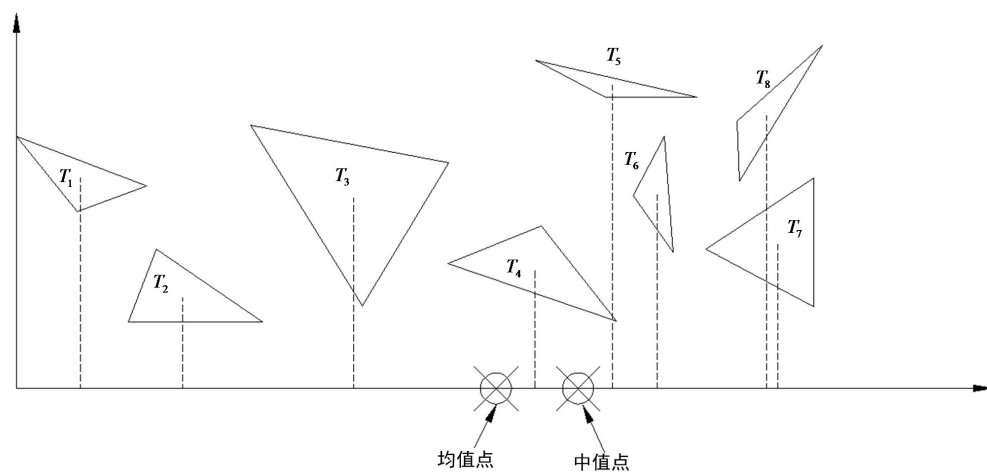
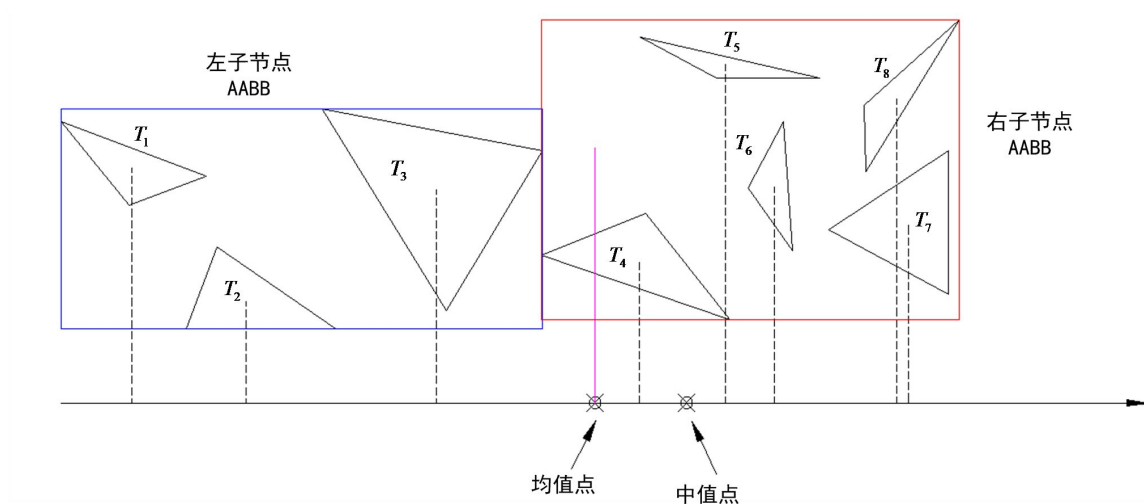


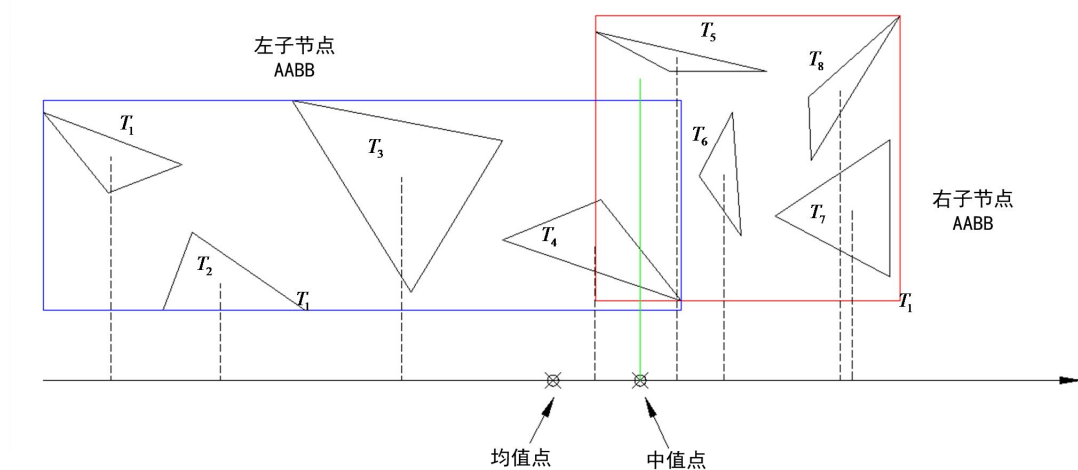
图 4.10 图元重心投影的均值点和中值点

按照均值法确定分裂点方法剖分父节点结果如图 4.11 (a) 所示, $T_1 \sim T_3$ 划归左子节点, 其他划归右子节点。

按照中值法确定分裂点方法剖分父节点结果如图 4.11 (b) 所示, $T_1 \sim T_4$ 划归左子节点, 其他划归右子节点, 可以看出划分后左右子节点包含的图元是完全相等的, 保证了树的平衡性。



(a) 均值点剖分



(b) 中值点剖分

图 4.11 采用均值法和中值法剖分的区别

5. 树的更新

当对象发生运动或变形后，其原有的包围盒树已不能很好的逼近对象，就需要重建或更新，因为仿真中时间步都很小，根据时空相关性，前后时间步中对象运动或变形幅度都不大，采用重建方法没有必要，同时耗时又太多，效率不高，一般选择包围盒更新的方法，同时采用某种策略判断某一部分是否需要更新来减少更新量，提高效率。

本文采用自底向上的更新策略来进行对象发生运动或变形后包围盒树的更新。大体步骤如下：

- (1) 对上一时间步内所有叶子节点包围盒包围的基本图元集合（可能只有一个基本图元，也可能有多个），根据整体对象是否发生变形选择更新包围盒（无变形情况）或重建包围盒（有变形情况）作为当前时间步的叶子节点包围盒，用 $\text{Min}'\text{-Max}'$ 形式表示。
- (2) 假设与 (1) 中对应的上一时间步的包围盒以 Min-Max 表示。
- (3) 如果 $\|\text{Min}' - \text{Min}\| + \|\text{Max}' - \text{Max}\| > \delta$ （ δ 是衡量包围盒变形程度的阈值， δ 越大树更新速度越高但碰撞测试精准度下降），则用当前时间步的包围盒替换上一时间步的对应包围盒，同时对其父节点进行更新，即根据子节点包围盒重新构建父节点包围盒。如果不满足条件则仍使用上一时间步的包围盒。
- (4) 此层更新完后，按照同样的思路处理其上一层，直到某一层所有节点都不满足 (3) 中的条件则结束。

4.4 三角形间的相交测试

利用复合层次包围盒进行碰撞检测，当检索到叶子节点时包围盒间仍显示相交，此时碰撞检测转化为叶子节点包围的基本图元集合间的相交测试，如果基本图元集合间不存在相交的三角形对，则两对象不相交，否则求出碰撞具体信息。三角形间的相交测

试也是碰撞检测中很关键的环节，其影响着碰撞检测的精度与实时性。

4.4.1 常用三角形间相交测试算法

三角形间相交测试算法发展到现在已经有很多种，大体上可以归为两种，标量判别类型和矢量判别类型。

标量判别类型，由 Möller, Held 和 Tropp 等人提出的算法是其中的典型代表。Möller 算法^[56]：如图 4.12 所示，判断三角形 A 和 B 是否相交。首先，计算三角形各自所在的平面 Π_A 和 Π_B ；然后，进行相离排除，计算三角形 A 的三个顶点是否在 Π_B 的同一侧，如果是，则三角形 A 和 B 肯定相离，同理判断三角形 B 的三个顶点是否在 Π_A 的同一侧（判断方法是求取点到平面的距离，根据距离的正负号判断点在平面的哪一侧）；如果不能排除二者相离的可能性，则求取 A 与 Π_B 的交线 L_A ， B 与 Π_A 的交线 L_B ， L_A 与 L_B 肯定共线，最后，判断 L_A 与 L_B 是否有重叠部分即可判断，存在重叠，三角形相交，（图 4.12(a)），否则分离，（图 4.12(b)）。

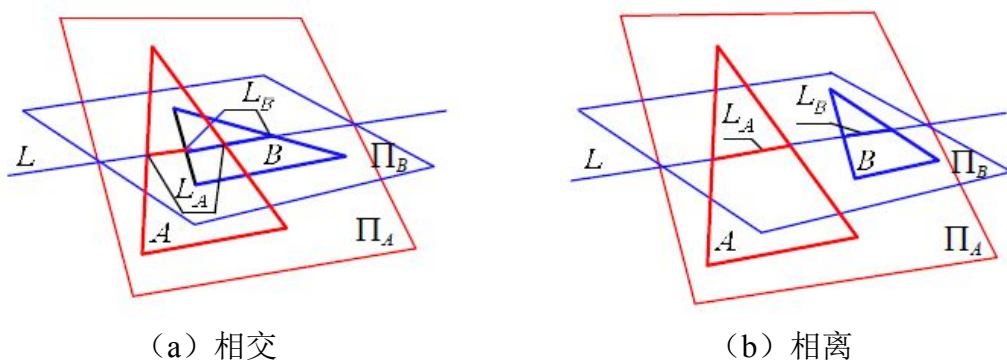


图 4.12 Möller 三角形相交测试算法

Held 算法与 Möller 算法相似，都是从几何观点出发，把三维问题降维为二维问题，解决三角形相交测试问题。而 Tropp 算法另辟蹊径，从代数观点出发，利用求解线性方程组解决此问题。

矢量判别类型，由 Devillers & Guigue、Shen 等提出的算法是其中的典型代表。Devillers & Guigue 算法基于三角形顶点组成的行列式的几何意义，Shen 提出的算法基于分离平面。

一般而言，标量判别法能够直接获取相交点的坐标，但是由于计算累积误差的存在，使精度会受到一些干扰；矢量判别法相对来说在检测结果的判定上更健壮，因为其判定不依赖于计算精度，但是如果想要获取碰撞的详细信息需要做额外的处理。然而，以上方法都存在这样的缺陷，在算法的稳定性方面都只是在代数角度上进行了说明，没有在理论上阐述针对奇异情况的处理。同时，在算法的评估上偏重于速度，没有对算法的稳定性进行分析测试，而在实际应用中针对奇异情况的处理失误有可能使系统崩溃。

针对上述算法稳定性方面的问题，于海燕等人^[57]提出了基于投影理论的三角形间相

交测试方法。算法关注了两三角形间相对位置的奇异情形，根据投影降维思想转化为平面上的几种简单情形，在理论上对几何奇异情形（如点碰撞、线碰撞等）对算法稳定性的影响进行了说明。所以本文选用此方法作为三角形间的相交测试方法，然而该算法还不尽完善，本文在投影分类，奇异情况处理、分离情况排除和三角形线裁剪算法方面对算法进行改进以提升其实时性、稳定性和准确性，然后应用于本文系统。

4.4.2 基于投影降维的三角形间相交测试算法

如图 4.13，两三角形 A 、 B 。

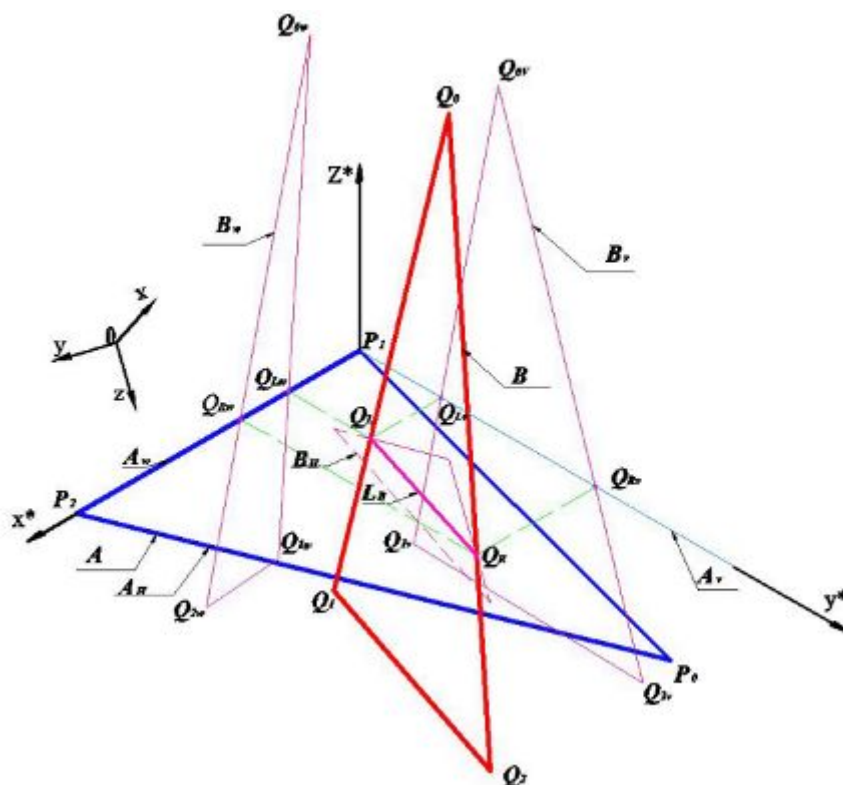


图 4.13 计算坐标系下的三角形

图中标记符号的含义如下： xyz 和 $x^*y^*z^*$ 分别代表世界坐标系和计算坐标系，我们将 x^*-y^* 平面记为 H 平面， x^*-z^* 平面记为 W ， y^*-z^* 记为 V 。 P_0 、 P_1 、 P_2 、 Q_0 、 Q_1 、 Q_2 分别是 A 、 B 的顶点； Q_{0v} 、 Q_{1v} 、 Q_{2v} 、 Q_{0w} 、 Q_{1w} 、 Q_{2w} 分别是三角形 B 的顶点在 V 、 W 平面上的投影； A_v 、 A_H 、 A_W 、 B_v 、 B_H 、 B_W 分别是三角形 A 、 B 在 V 、 H 、 W 平面上的投影； L_B 是三角形 B 与 H 平面的交线； Q_L 、 Q_R 是 L_B 的端点； Q_{Lv} 、 Q_{Rv} 、 Q_{Lw} 、 Q_{Rw} 分别是 Q_L 、 Q_R 在 V 、 W 平面上的投影。

我们知道，如果两三角形有相交点，那它们三个坐标平面的投影肯定都会存在交点。若两三角形三个坐标平面的投影存在没有交点的情况，则两空间三角形必定不相交，基于投影降维三角形相交检测改进算法基本原理就在于此。

1. 坐标系转换

一个合适的坐标系可以简化几何表述与代数运算，首先建立一个计算坐标系，方法与原算法类似。

建立计算坐标系：

如图 4.13，以三角形 A 所在的平面作为 x^*-y^* 平面，此平面的法向作为 z^* 轴，并以 A 的一条边（如 P_1P_2 边）作为 x^* 轴。

(1) 取 P_1P_2 向量的单位向量作为 $n_1(a_1, a_2, a_3)$ 。

(2) 对向量 P_1P_2 ， P_1P_0 求叉乘可以得到一个垂直于三角形 A 的向量，取单位向量为 $n_3(c_1, c_2, c_3)$ 。

(3) 利用 $n_3 \times n_1$ 然后取单位向量可得 $n_2(b_1, b_2, b_3)$ 。

以 P_1 为原点，以 n_1 、 n_2 、 n_3 为轴向建立计算坐标系 $x^*y^*z^*$ 。

$$\begin{pmatrix} n_1 & n_2 & n_3 \end{pmatrix} = \begin{pmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{pmatrix} \quad (4-3)$$

坐标变换：

因为 P_1 为计算坐标系 $x^*y^*z^*$ 的原点，我们可以得到 xyz 坐标系到 $x^*y^*z^*$ 的平移矩阵：

$$T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -P_{1x} & -P_{1y} & -P_{1z} & 1 \end{pmatrix} \quad (4-4)$$

同时可以得到旋转矩阵：

$$R = \begin{pmatrix} a_1 & a_2 & a_3 & 0 \\ b_1 & b_2 & b_3 & 0 \\ c_1 & c_2 & c_3 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4-5)$$

从而坐标变换公式为：

$$(x^*, y^*, z^*, 1) = (x, y, z, 1) \cdot T \cdot R \quad (4-6)$$

$$(x, y, z, 1) = (x^*, y^*, z^*, 1) \cdot (T \cdot R)^{-1} \quad (4-7)$$

将两三角形的坐标都转换到计算坐标系下。

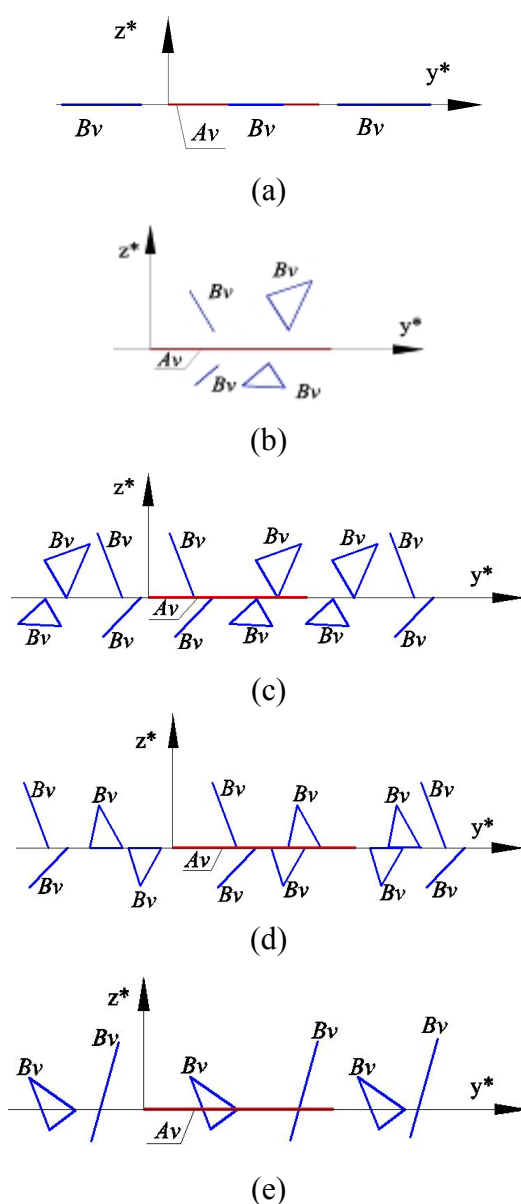
2. 基于投影理论对三角形空间位置关系的分析

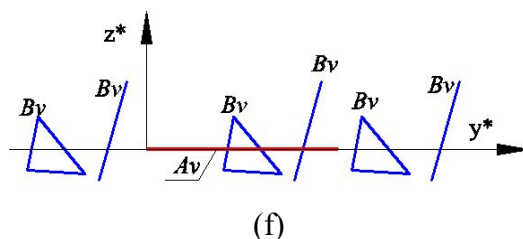
这一部分是算法的关键之处，也是我们对原算法改进之处。原算法中只用到了三角形在 V 面和 H 面投影，当三角形 A 与 B 不相交时，如果 B_V 与 A_V 没有交点，可以排除这种相离的情况，但如果 B_V 与 A_V 存在交点，而 B_W 与 A_W 没有交点，这种情况下显然 A 与 B 是相离的，而原算法针对此种情况无法进行前期排除，只能经过后续的一些计算后才判断出相离，显然会花费比较高的代价。针对这个问题，本文利用 V 、 W 两个平面的投影

进行分离情况的排除来解决。当两三角形在 V 面上的投影情形跟图 4.14(a)(b) 中所示不同时，原算法需要计算 B_v (B 的投影三角形) 的三条边与 A_v 所在的坐标轴的交点，根据交点的个数不同做不同的处理，这增加了很多计算。针对这个问题，本文通过顶点交换实现 B_v 的特定边与 A_v 所在的坐标轴相交，使奇异情况处理更简单，减少计算。同时本文对三角形投影分类更加细化，使相交测试更条理。

因为计算坐标系基于三角形 A 所建，下面的结论是显而易见的： $A_H = A$ ； A_v 和 A_w 分别为位于 y^* 和 x^* 轴上的线段；尽管三角形 B 在三个坐标平面上的投影形状不确定，但也只有线段和三角形的可能。投影为线段的情形实际是投影为三角形的特例，针对这种情形相关参数的求解处理都按三角形处理。

以三角形 A 和 B 在 V 平面上的投影为例 (W 平面上的类似)，所有可能的投影形状和相对位置关系如图 4.14 所示。



图 4.14 三角形 A 和 B 在 V 平面上的投影

(红线代表三角形 A 的投影, 蓝线或蓝色三角形代表三角形 B 的投影)

投影分类及对应相交测试:

(1) 如图 4.14 (a) 所示, B_v 是 y^* 轴上的线段, 当三角形 B 所有顶点的 z^* 坐标均为 0 时出现这种情形。如果 B_v 与 A_v 没有重叠部分, 则三角形 A 与 B 相离; 如果有重叠部分, 则继续判断 B_w 与 A_w 是否有重叠部分, 没有则三角形间相离; 如果有则需要判断 H 平面内二维三角形 A_H 与 B_H 是否相交, 不相交则 A 与 B 相离, 否则 A 与 B 相交。如何判断两共线线段是否存在重叠部分和如何判断两共面三角形是否相交的方法将在后面详述。

(2) 如图 4.14 (b) 所示, B_v 为位于 A_v 之上或之下的线段或三角形, 当三角形 B 所有顶点的 z^* 坐标均大于或小于 0 时出现这种情况。显然此种情况下三角形 A 与 B 相离。

(3) 如图 4.14 (c) 所示, B_v 与 y^* 轴有一个交点, 当三角形 B 的一个顶点的 z^* 坐标为 0 而另两个坐标同时大于或小于 0 时出现这种情况。我们把 z^* 坐标为 0 的顶点标记为 Q_0 (利用坐标轮换)。显然三角形 B 与 H 平面间只有一个交点-- Q_0 , 即 $Q_L = Q_R = Q_0$ 。 Q_L 在 V 平面上的投影为 $Q_{LV}(0, y_{Q_0}^*, 0)$, 如果 Q_{LV} 没在 A_v 上 (这很容易判断), 三角形 A 与 B 相离; 如果在, 则判断 $Q_{LW}(x_{Q_0}^*, 0, 0)$ 是否在 A_w 上, 如果不在, 三角形 A 与 B 相离; 如果在, 则需要判断在 H 平面内, Q_L 点是否在三角形 A_H 内, 如果在, 在三角形 A 与 B 相交于点 Q_L , 是点碰撞, 否则二者相离。如何判断在同一个平面内点与三角形的关系的方法在后面详述。

(4) 如图 4.14 (d) 所示, B_v 与 y^* 轴有两个交点, 当三角形 B 的两个顶点的 z^* 坐标为 0, 剩余的顶点 z^* 坐标不为 0 时出现这种情况。我们把 z^* 坐标为 0 的顶点标记为 Q_0 、 Q_1 (利用坐标轮换)。显然三角形 B 与 H 平面间有两个交点-- Q_0 、 Q_1 , 即 $Q_L = Q_0$ 、 $Q_R = Q_1$ 。 Q_L 、 Q_R 在 V 平面上的投影分别为 $Q_{LV}(0, y_{Q_0}^*, 0)$ 、 $Q_{RV}(0, y_{Q_1}^*, 0)$ 。如果以 Q_{LV} 、 Q_{RV} 为端点的线段 L_{B_v} (L_B 在 V 平面上的投影) 与 A_v 没有重叠部分, 三角形 A 与 B 相离; 如果有, 则判断 L_{B_w} (L_B 在 W 平面上的投影) 与 A_w 是否有重叠部分, 如果没有, 三角形 A 与 B 相离; 如果有, 则需要判断在 H 平面内, 线段 L_B 与三角形 A_H 的相交情况, 如果相交, 三角形 A 与 B 相交, 属于线碰撞, 交线为 $L_B \cap A_H$, 否则二者相离。如何判断在同一个平面内线段与三角形的关系的方法在后面详述。

(5) 如图 4.14 (e) 所示, B_v 与 y^* 轴有两个交点, 当三角形 B 的一个顶点的 z^* 坐

标为 0，一个顶点 z^* 坐标大于 0，一个顶点 z^* 坐标小于 0 时出现这种情况。我们把 z^* 坐标为 0 的顶点标记为 Q_0 （利用坐标轮换）。显然三角形 B 与 H 平面间有两个交点，一个是 Q_0 ，记 $Q_L = Q_0$ ，另一个是三角形 B 的边 Q_1Q_2 与 H 平面的交点，记为 Q_R 。后续步骤跟 4) 中相关步骤相同。关于 Q_R 的求取在后面介绍。

(6) 如图 4.14 (f) 所示， B_V 与 y^* 轴有两个交点，当三角形 B 的一个顶点的 z^* 坐标大于 0（小于 0），另两个顶点的 z^* 坐标均小于 0（大于 0）时出现这种情况。我们把 z^* 坐标大于 0（小于 0）的那个顶点标记为 Q_0 （利用坐标轮换）。显然三角形 B 与 H 平面间的两个交点一个是三角形 B 的边 Q_0Q_1 与 H 平面的交点，记为 Q_L ，另一个是三角形 B 的边 Q_0Q_2 与 H 平面的交点，记为 Q_R 。后续步骤参考 5)。

3. 相关详细算法

(1) 两共线线段相交检测

如图 4.15 所示，两共线线段 L_1 （端点为 P 、 Q ）、 L_2 （端点为 M 、 N ），根据端点坐标判断，如果端点 Q 位于端点 M 左侧或端点 P 位于端点 N 右侧，两线段无重叠部分，否则，有重叠部分。

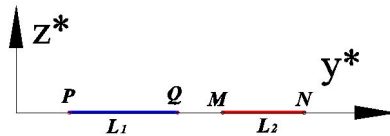


图 4.15 两共线线段

(2) 同一平面内线段与三角形的相交测试

如图 4.16 所示，三角形 A （顶点 P_0 、 P_1 、 P_2 ），线段 L_1 。二者的相交测试利用孙燮华提出的凸多边形线裁剪算法^[58]解决，这也是本文与原算法不同之处，本文采用的方法更简易，同时能处理所有出现的裁剪情况。首先，求取线段 L_1 的一般方程，如式 (4-8) 所示；其次，将三角形顶点 $P_i (i=0,1,2; P_3 = P_1)$ 的坐标代入式 (4-8) 求取特征值 e_i ；再次，计算线段与三角形的交点及交点数量，下面所示的伪代码表述了这一求解过程；最后，如果得到的交点数为 0 或者是 1 个但交点不在 L_1 线段上或者 2 个但交点组成的线段 $L_{Q_1Q_2}$ 与线段 L_1 没有重叠部分，线段 L_1 与三角形 A 相离，否则相交。利用上述方法针对图 4.16 中所示的线与三角形的相交情况都能判别出。

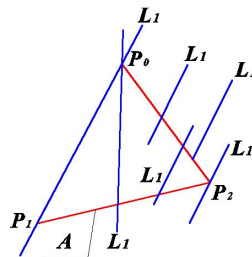


图 4.16 平面内线段与三角形位置关系

$$ax + by + c = 0 \quad (4-8)$$

线段与三角形的交点及交点数量求取伪代码:

```

int n = 0; //counts of intersection points
Point Q1, Q2, Q; //define two intersection points
float e[4];
if (e[0] == 0)
{
    n++;
    Q1 = P[0]; //P[i] is the vertex of triangle
}
for (i = 0; i <= 3; i++)
{
    if (e[i + 1] == 0)
    {
        n++;
        if (n == 1)
            Q1 = P[i + 1];
        if (n == 2)
        {
            Q2 = P[i + 1];
            break;
        }
    }
    else if (e[i] * e[i + 1] < 0)
    {
        n++;
        Q = CallInterPoint(); //calculate intersect point
        if (n == 1)
            Q1 = Q;
        if (n == 2)
        {
            Q2 = Q;
            break;
        }
    }
}

```


}
}
}

(3) 同一平面内两三角形的相交检测

假设有两三角形 A 、 B 在同一平面内，首先，利用上述 2) 中的方法判断 B 的边中是否存在与三角形 A 相交的，如果存在，则两三角形相交；如果不存在，不能得出二者相离的结论^[59]，如图 4.17 所示的情形，三角形 A 位于 B 之内，显然二者是相交情况，这需要判断 A 的顶点是否存在位于 B 之内的，如果有则三角形 A 位于 B 之内，否则， A 与 B 相离。

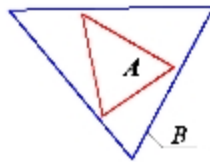


图 4.17 同一平面内三角形 A 在三角形 B 内部

(4) 同一平面点与三角形的位置关系

如图 4.18 所示，有一点 P 和三角形 A 在同一平面内。为了计算方便，对点 P 和 A 的顶点进行平移变换，使点 P 位于原点。如果 P 位于 A 之内，则 P 一定位于 A 各边的同一侧（左侧或右侧）否则， P 位于 A 之外。详细计算参考文献[52]。

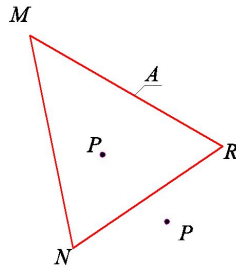


图 4.18 同一平面点与三角形的位置关系

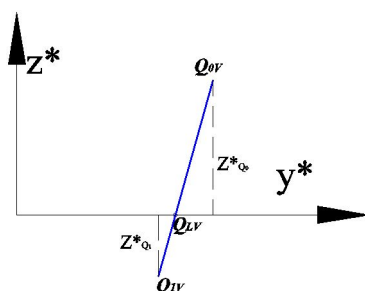
(5) 线与面交点计算

如图 4.13 所示，三角形 B 的 Q_0Q_1 边与 H 平面相交于点 Q_L ， Q_0Q_1 在 V 平面的投影 $Q_{0V}Q_{1V}$ 与 y^* 轴的相交于点 Q_{LV} 。我们可以容易的得到点 Q_L 和 Q_{LV} 的参数方程如式 (4-9) 所示，显然 t 与 t_V 相等。

$$\begin{aligned} Q_L &= Q_1 + t(Q_0 - Q_1) \\ Q_{LV} &= Q_{1V} + t_V(Q_{0V} - Q_{1V}) \end{aligned} \quad (4-9)$$

由图 4.19 可得：

$$t_V = \frac{Q_{LV} - Q_{1V}}{Q_{0V} - Q_{1V}} = \frac{|Z_{Q_1}^*|}{|Z_{Q_1}^*| + |Z_{Q_0}^*|} \quad (4-10)$$

图 4.19 Q_0Q_1 在 V 平面的投影 $Q_{0V}Q_{1V}$

4. 本文改进算法与原算法对比实验

实验条件同第三章，为了使算法的每一步都至少执行一次并测试算法的稳定性，本文专门设计了 40 对三角形的位置关系，包含了所有可能的三角形相交奇异情况，部分示例如图 4.20，利用原算法和本文改进后的算法对这 40 对三角形进行测试并且执行不同的次数，测试结果如表 4.1 所示。

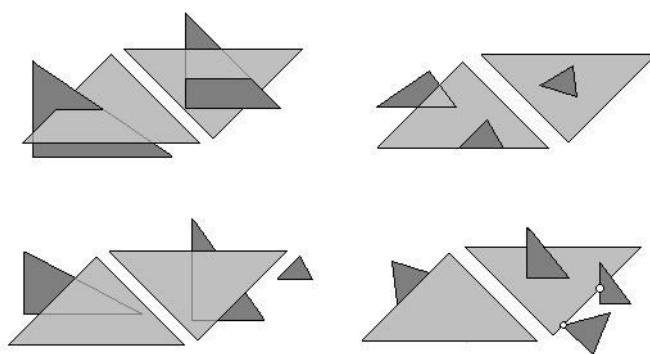


图 4.20 两三角形空间位置关系举例

表 4.1 改进算法与原算法性能对比

| 参加测试的三角形对数量 | 测试时间 | |
|-------------|------|--------|
| | ms | |
| 百万对 | 原算法 | 本文改进算法 |
| 0.1 | 70 | 42 |
| 0.5 | 392 | 231 |
| 1 | 654 | 384 |
| 1.5 | 947 | 557 |
| 2 | 1256 | 739 |

改进后算法的时间效率提升约 41%，同时设计的三角形对的位置关系碰撞检测结果跟预设情况对比无误，并且针对奇异情况处理程序运行稳定。说明了本文改进算法在效率、准确性、稳定性方面有提升。

4.5 碰撞检测算法对比实验

实验平台与上一章相同，碰撞对象分别采用第 2 章建立的肝脏体模型和用 3ds Max 建立的手术器械，手术器械有 628 个顶点，1248 个三角形。实验方案为固定肝脏面模型，控制手术器械运动，仿真二者的碰撞过程；进行三组实验，第一组采用本文的复合层次包围盒法和基于投影降维的三角形间相交测试法方案，因为肝脏体模型表面共有 9313 个三角形，让每个叶子结点包围盒最多包围 10 个三角形单元且最多划分 11 层包围盒即可，其中上面三层用 Sphere，中间三层用 Sphere-AABB，下面五层用 AABB，手术器械三角形较少，划分为 9 层即可，上面两层用 Sphere，中间两层用 Sphere-AABB，下面五层用 AABB（这里之所以对手术器械也同样采用这样的包围盒而不是传统的只对手术器械尖端建立一个 Sphere 或 AABB 包围盒，是为了能够精确检测出手术器械与软体对象的精确接触点从而可以对软体精确施加不同方向的力，后续还可以考虑手术器械相对软体不同角度施加力对软体变形效果的影响）。第二组采用 Sphere 层次包围盒法和 Möller 三角形相交测试法方案，第三组采用 AABB 层次包围盒法和 Möller 三角形相交测试法方案。

碰撞实验截图如图 4.21。



图 4.21 肝脏碰撞过程

每组实验中两对象的相对位置和手术器械的运动速度是相同的，记录每组实验中相交三角形数目相同时的 FPS 值，具体实验测试数据采样值如下。

表 4.2 不同碰撞检测方案下 FPS 值与相交三角形数量实验数据

| 相交三角形对数 | 帧速率 (FPS) | | |
|---------|---------------------------|-------------------|-----------------|
| | 复合层次包围盒法 (Sphere-AABB) | 球形包围盒 (Sphere) | 轴向包围盒 (AABB) |
| 0 | 263.2 | 263.2 | 254.7 |
| 32 | 260.3 | 259.7 | 251.6 |
| 97 | 253.6 | 250.1 | 244.8 |

| | | | |
|------|-------|-------|-------|
| 279 | 217.9 | 211.2 | 214.6 |
| 534 | 163.2 | 148.4 | 158.9 |
| 733 | 129.4 | 95.6 | 119.4 |
| 925 | 89.6 | 47.3 | 73.1 |
| 1106 | 53.2 | 17.2 | 42.7 |

将上述表格内数据绘制成曲线图，如图 4.22。

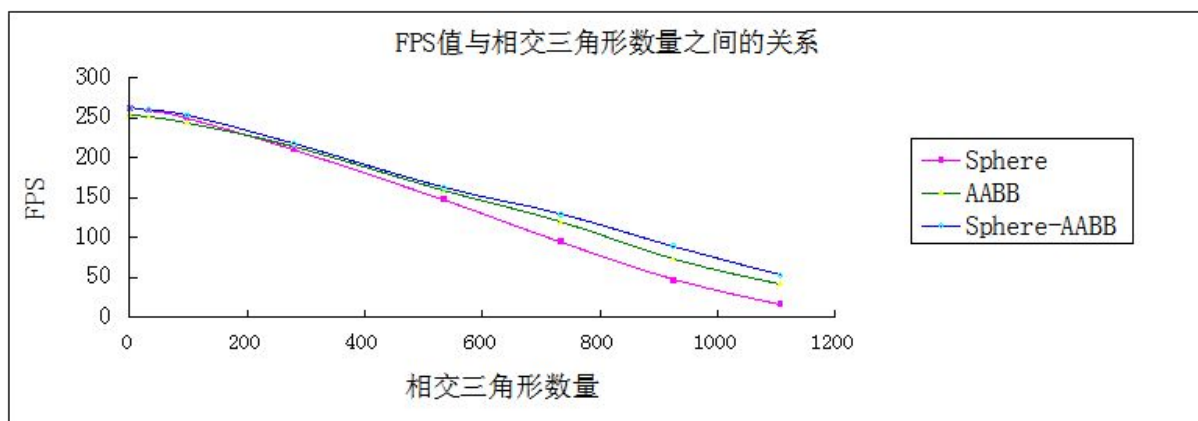


图 4.22 不同碰撞检测方法下 FPS 值与相交三角形数量的关系

从上述表格和折线图中可以看出，当对象没有相交或相交的三角形数量较少时，Sphere 层次包围盒法速度快，这是因为大部分三角形对不相交，通过 Sphere 相离排除简单，当相交数量增多时，其包围紧密性差的缺点暴露出来，速度迅速下降，当相交数量超过 1000 时，FPS 降到 30 以下。AABB 层次包围盒法在相交的三角形数量较少时速度小于 Sphere 方法，因为其相交测试比 Sphere 慢，分离排除就会慢一些，当相交数量增多时，其速度下降较慢，因为其包围紧密性相对较好。Sphere-AABB 复合方法结合了二者的优点，随相交数量的增加，速度变化平稳，性能均在单独使用 Sphere 和 AABB 方法之上，当三角形相交数量超过 1106 时，FPS 值仍为 53.2，满足视觉反馈实时性要求。

4.6 本章小结

本章主要针对虚拟手术系统中的碰撞检测方法进行了研究。根据不同包围盒的优缺点及适用场景，提出 Sphere-AABB 复合层次包围盒方法作为本文的碰撞检测方法，对该方法涉及到的基本包围盒的构建，相交测试，计算与更新，树的构建及更新做了详细介绍。对于海燕等人提出的基于投影降维的三角形间的碰撞检测算法做了改进并应用在基本图元间的相交测试中。最后对不同碰撞检测方法做了实验对比，说明了复合层次包围盒结合基于投影降维三角形碰撞检测方法在实时性方面的优势。

第5章 仿真验证实验

5.1 引言

本文在绪论中构建了虚拟手术系统的整体架构,在前面章节中分别对软体的几何建模、物理建模、碰撞检测技术做了详细研究和介绍。本章将综合前面的内容搭建一个简易虚拟手术系统来验证所研究技术的有效性。搭建系统需要用到的硬件环境为实验室的台式电脑,配置为 CPU: Intel Core i3-2100 @ 3.1GHz、GPU: NVIDIA GeForce GTS450、Memory: 6G; 需要的软件环境为 OS: Win7、IDE: VS2010、Language: C++、Library or Software: OpenGL 及相关扩展库或插件、Amira 软件。

5.2 软件环境概述

系统构建所需的硬件条件无需解释,就是普通配置的台式机,这里针对软件环境做简单介绍。因为虚拟手术系统需要进行大量的包围盒计算,碰撞检测计算,积分计算,还有大量的读取、存储操作等,同时为了显示对象的运动和变形需要进行三维图形的绘制,而虚拟手术系统效率要求比较高,所以需要选择运行高效的编程语言和图形绘制库,C++和 OpenGL 组合正是最理想的选择。Amira 是本文在第二章中提到的几何建模的工具软件,下面仅对 OpenGL 做简单介绍。

5.2.1 OpenGL 概述

OpenGL(Open Graphics Library)^[60]是“图形硬件的一种软件接口”,提供给显示和处理 3D 图形的计算机硬件使用。它是一个“开放”的标准,具有高度的可移植性和非常快的速度。

OpenGL 以两种形式存在。一种是“OpenGL 规范”,这个规范定义了行业标准,用非常完整和明确的术语描述了 OpenGL。它完整地定义了 OpenGL API、OpenGL 的整个状态机,以及各种特性是如何共同工作和运行的。然后 NVIDIA、ATI、Intel 或 Apple 这样的硬件厂商获取这个规范,并实现它。另一种就是 OpenGL 的实现,软件开发人员和顾客可以使用它生产实时图形。例如,PC 上的软件驱动程序和图形卡就共同组成了一个 OpenGL 实现。

5.2.2 OpenGL 工作原理

介绍 OpenGL,不得不提到的两个最为重要的关键点就是 OpenGL 的渲染管线和 OpenGL 的矩阵变换。渲染管线描述了 OpenGL 的工作流程,矩阵变换描述了一个空间的图形是如何显示在二维像素平面的,当然,矩阵变换只是渲染管线中的一个小的环节。

1.OpenGL渲染管线

OpenGL自从3.0版本以后就重点推荐使用可编程渲染管线，在核心版本中已经去除了固定管线。如图5.1所示为一个简化的可编程渲染管线^[59]，以绘制一个三角形为例对其简要介绍。

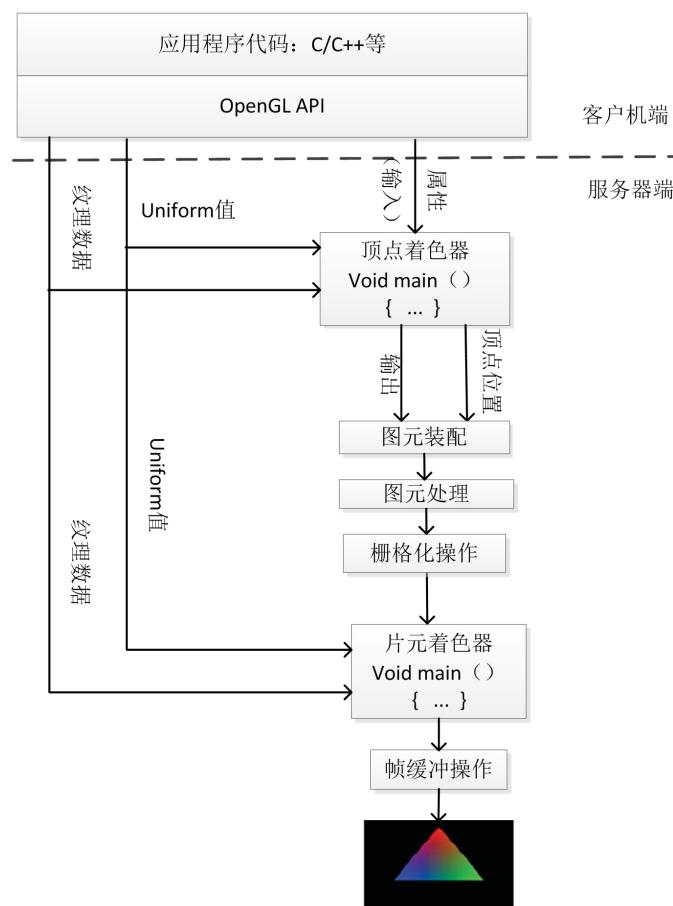


图 5.1 可编程渲染管线

管线按照功能分成了客户端和服务端，二者是异步工作的，客户端是指CPU端，服务器端指GPU端，客户端负责运行我们的应用程序代码并把相应数据块和命令块组合后送入缓冲区，服务器端负责接收缓冲区的数据和命令完成图形的渲染。

在渲染管线中最重要的两个环节是顶点着色器和片元着色器，它们是用GLSL语言编写的程序。为了使着色器工作我们需要为其提供数据，提供数据的方式有如图中所示的属性、uniform值和纹理三种。顶点着色器主要负责接收客户端传来的几何图元的顶点数据，利用矩阵变换或其他处理方式对顶点的光照、位移、颜色等属性进行计算。顶点着色器对每个顶点都会执行一次，可以并行执行。顶点处理完后根据顶点的处理命令进行图元组装，形成基本图元（点、线、面）。然后进行图元处理，包括根据裁剪平面和投影矩阵进行的裁剪操作、坐标透视除法、视口变换（从顶点坐标变换到窗口坐标）、

消隐操作。然后进行栅格化处理，将图元分解成与缓冲区各个像素对应的片元，每个片元都有各自的位置坐标、纹理坐标和深度等属性。然后片元着色器为每个片元进行纹理颜色采用，雾化计算，然后将纹理、主定义颜色、雾化颜色、次颜色、光照颜色叠加得出每个片元的最终颜色作为输出。然后片元经过一些测试（深度测试，模板测试等）后执行帧缓冲区操作，一般就是写入帧缓冲区，通过刷新缓冲区或交换前后缓冲区我们就看到了显示在屏幕上的像素。

2..OpenGL中的坐标变换

利用 OpenGL 绘制 3D 图形的过程中涉及很多坐标变换，其目的就是把三维图形转换成二维的平面图形进行显示，图 5.2 显示了这个变换过程。

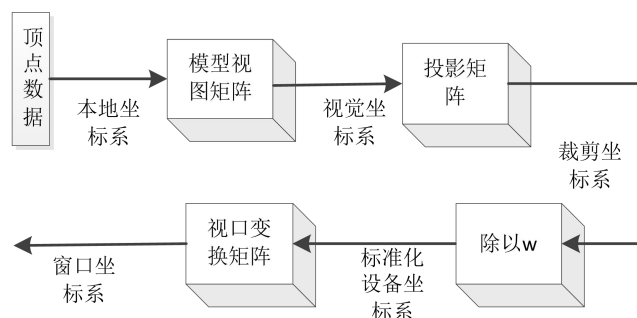


图 5.2 顶点变换过程

我们绘制对象时都是在对象所在的局部坐标系进行绘制，为了使多个对象不发生重叠，并如真实世界中那样有自己的位置，就要利用其本地坐标乘上一个模型变换矩阵变换到世界坐标系下。所有的对象都处在世界坐标系后，我们如果用眼睛去观察它们，它们相对于我们的眼睛又有不同的位置，用它们的世界坐标乘以一个视图矩阵变换到视觉坐标系下。图中将模型矩阵和视图矩阵相乘在一起了，直接进行了从本地坐标系到视觉坐标系的转换。为了减少计算量只显示在视景体内的对象，执行投影变换，将视景体外的对象裁剪掉，变换到裁剪坐标系；为了使图形的显示不依赖于硬件，执行归一化处理，变换到标准化设备坐标系；最后通过视口变换将三维的坐标转化为二维坐标，变换到窗口坐标系。

5.3 虚拟手术系统的实现

5.3.1 系统框架

实现一个系统前要先设计好其总体框架，理清逻辑结构，本文系统总体框架如图 5.3 所示。

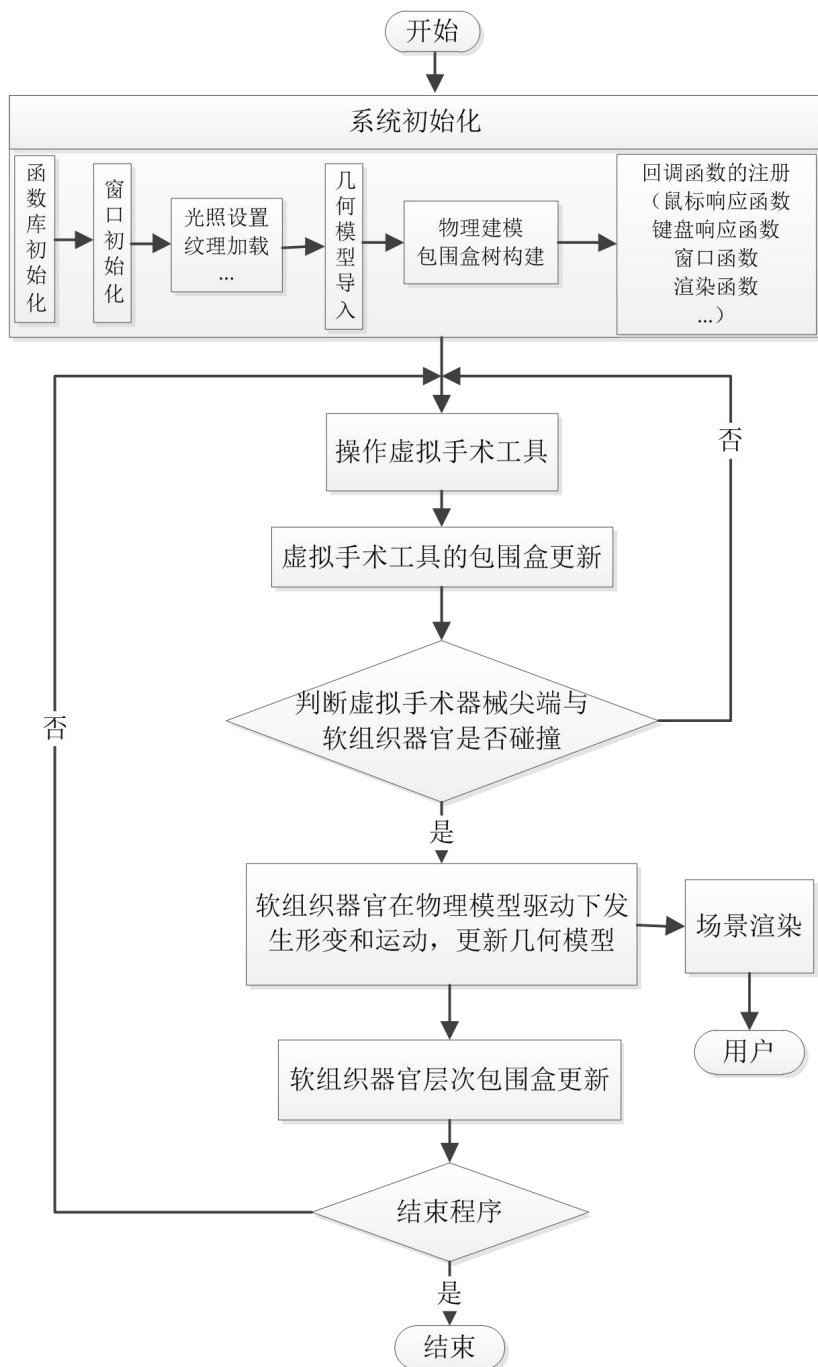


图 5.3 仿真系统框架流程图

系统是按模块设计实现的，包括光照设置计算模块，纹理加载模块，几何模型加载模块，物理建模模块，碰撞检测模块，软体形变计算模块，渲染模块等很多模块，通过上面框架中所示的逻辑将它们组织起来协同工作来完成系统的功能。系统首先需要进行初始化，为后面的操作做准备工作，初始化之后的操作都是处在一个循环中的，人为控制循环的结束与否。

5.3.2 系统中重要的数据结构

在系统的实现中，数据结构是非常重要的，因为，程序就是算法与数据结构的融合，数据结构相当于骨架，是算法实现的基础，良好的几何模型数据结构不仅可以提高系统的整体性能还有助于计算的简化。设计数据结构就是选择存储方式，在这里介绍一下系统中用到的重要的数据结构的思想。

如图 5.4 所示为简化的的数据结构，它能够清晰地表示点、面、对象之间的相互关系，从而使数据的访问和更新更快捷，方便。当然图中只是用简化的形式表示了系统中相关数据结构的思想，实际的数据结构比这复杂的多。

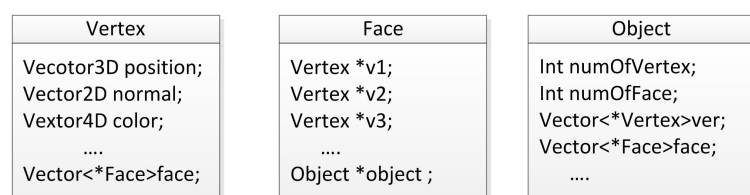


图 5.4 系统中重要的数据结构

5.4 实验结果

5.4.1 实验结果截图

1、肝脏在不同角度观察下的变形前状态，如图 5.5 所示。

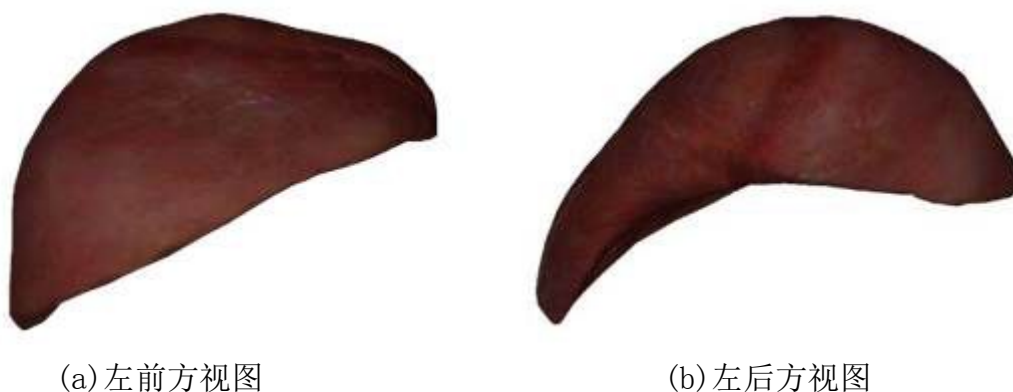


图 5.5 不同角度观察下的变形前状态

2、按压变形效果，如图 5.6 所示。

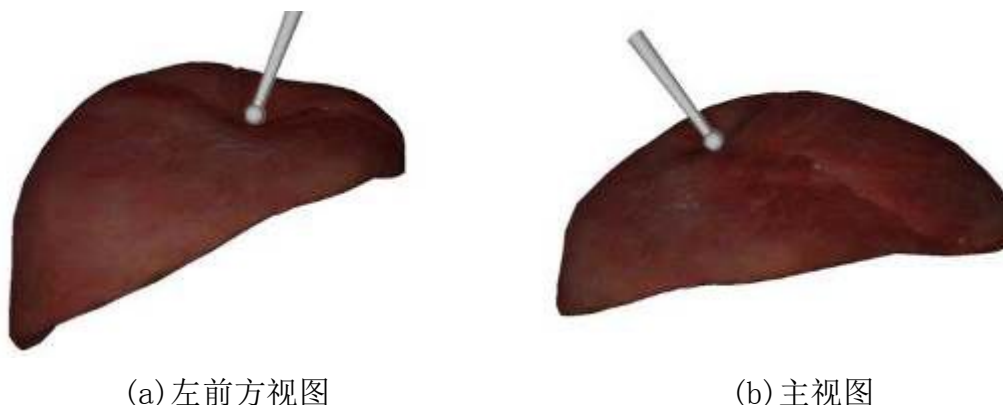


图 5.6 肝脏按压形变效果

3、提拉变形效果，如图 5.7 所示。

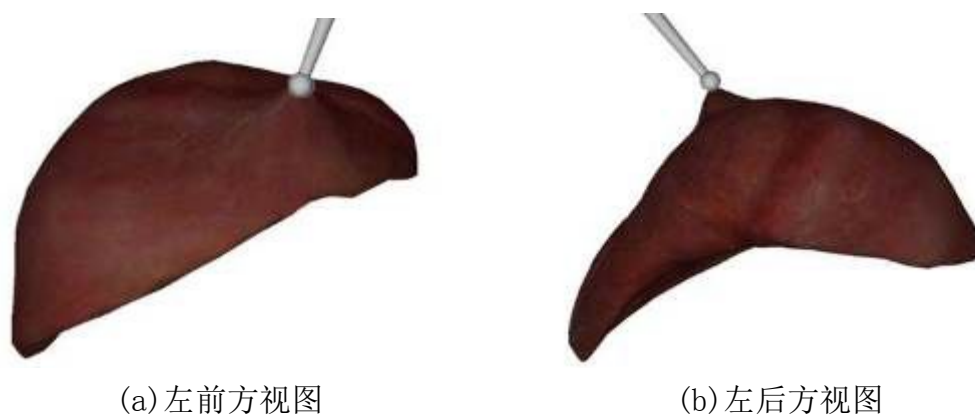


图 5.8 肝脏提拉形变效果

5.4.2 结果分析

在整个实验中平均 fps 值均在 50 以上，在虚拟手术系统中视觉反馈 fps 至少要达到 24 以上^[61]，实验中的 fps 值远高于此值，视觉反馈流畅，满足了实时性的要求。从图 5.7 和 5.8 中可看出肝脏的按压、提拉变形效果接近于真实情况，没有出现塌陷和超弹性现象，同时手术器械没有穿越和远离软体组织，满足逼真性的要求。从而验证了本文基于形状匹配的无网格物理建模方法和复合层次包围盒树碰撞检测方法的有效性。

5.5 本章小结

本章在 VS2010 集成开发环境利用 C++ 语言、OpenGL 三维图形库和文中的算法搭建了一个简易虚拟手术系统，进行了肝脏模型的按压、提拉变形实验。分别对 OpenGL 的核心原理、虚拟手术系统的框架和重要的数据结构做了简要介绍，同时展示了实验的结果并做了分析。实验结果说明系统满足实时性和逼真性两个方面的要求，验证了本文算法的有效性。

结 论

虚拟手术仿真系统虽然从提出至今得到了国内外众多研究者的关注并取得了一系列进展,但对于其中的关键技术至今还尚未建立一套成熟的理论与方法,特别是针对软体的形变模拟和软体的碰撞检测,在虚拟手术中制约着系统的实时性与真实性,依然是虚拟手术技术的瓶颈。在这一背景下,本文针对这两个方面展开研究并做了如下工作:

1、肝脏几何模型的建立。常用的几何模型建立方法有医学图像三维重建法和建模软件绘制法,鉴于医学图学三维重建方法获取的几何模型逼真度高,因此适用于虚拟手术仿真系统,借助于 Amira 三维重建软件利用医学图像数据重建出肝脏面模型。

2、利用基于形状匹配的无网格方法对肝脏进行物理建模。目前最常用的软体物理建模方法就是质点-弹簧模型和有限元模型,但质点-弹簧模型存在稳定性不足、大变形失真的缺点;有限元模型存在实时性不佳的缺点。为了解决上述问题,本文将基于形状匹配的无网格方法应用到软体的物理建模中,经仿真实验验证本文方法在实时性、大变形、稳定性方面都优于质点-弹簧模型和有限元模型,变形逼真度方面跟质点-弹簧模型相当。

3、碰撞检测算法的改进。在软体的碰撞检测中最常采用的粗略检测方法就是层次包围盒法,但单独使用一种包围盒在相交三角形数量不同时其时间性能不能一直保持很好的状态,所以本文提出三层结构的 Sphere-AABB 复合层次包围盒方法克服这一缺陷。同时本文针对于海燕等人提出的基于投影降维的三角形间碰撞检测算法做了改进。然后将二者结合与传统单一层次包围盒结合传统三角形间碰撞检测方法做对比实验,实验结果表明本文算法在任何情况下实时性都优于单一层次包围盒结合传统三角形间碰撞检测算法。

4、仿真验证实验。在 VS2010 集成开发环境下利用 C++ 语言编程实现了文中的物理建模方法和碰撞检测方法,调用 OpenGL 库实现场景渲染,然后导入利用 Amira 建立的肝脏几何模型进行了形变实验。实验中视觉渲染 fps 平均在 50 以上,达到了实时性的要求;按压、提拉变形逼真度也很高,不会出现模型表面塌陷、大变形产生超弹性现象,系统运行稳定,符合实用性的要求,验证了文中算法的有效性,实用性。

实用的虚拟手术系统是很复杂的,场景要包含手术台,手术灯光,多种手术器械,周边环境等,手术对象也不是独立存在的,与其他组织是在一起的,手术操作也是多样的,包括按压、提拉、切割、缝合等,同时组织在被切割的同时也会伴随流血现象等,远比比本文的简易系统要复杂的多。经过很长时间的的努力,本文取得了一定的成果,但对于构建成熟系统还远远不够,还需要很多的工作要做,在今后的研究中重点要做的工作有:

- 1、研究不同器官如胃、肝脏组织的切割和缝合操作，使手术操作更全面。
- 2、研究切割操作时伴随的流血现象，增强手术模拟的真实性。
- 3、由于切割、缝合操作会改变原来几何模型的拓扑结构，势必造成包围盒的更新不同于文中采用的方法，需要对这种操作时包围盒的更新进行研究。
- 4、构建整个人体几何模型，丰富手术环境，使操作环境更逼近于真实，增强系统使用人员的临场感。

参考文献

- [1] 陈定方主编.五彩缤纷的虚拟现实世界[M].北京:中国水利水电出版社,2015:4-14 页
- [2] 申蔚, 曾文琪编著. 虚拟现实技术. 北京: 清华大学出版社, 2009: 1-5 页
- [3] (美) Grigore C. Burdea, (法) Philippe Coiffet 著, 魏迎梅, 栾悉道等译.虚拟现实技[M]. 北京:电子工业出版社,2005:232-254 页
- [4] Courtecuisse H, Jung H, Allard J, et al. GPU-based real-time soft tissue deformation with cutting and haptic feedback. Progress in Biophysics and Molecular Biology, 2010, 103(2-3): 159-168P
- [5] Brown B, Sorkin S, Bruyns C et al. Real-Time Simulation of Deformable Objects: Tools and Application. Computer Animation-The Fourteenth Conference on Computer Animation. 2001: 228-258P
- [6] 钟世镇,原林,唐雷,黄文华,戴景兴,李鉴轶,刘畅,王兴海,洪辉文,李华,罗述谦,秦笃烈,曾绍群,吴涛,张美超,吴坤成,焦培峰,陆云涛,陈浩,李培良,郜元,王彤,樊继宏. 数字化虚拟中国人女性一号(VCH-F1)实验数据集研究报告(英文)[J]. 第一军医大学学报,2003,03:196-200+209.
- [7] 谭珂,郭光友,王大君等.虚拟现实技术在鼻腔镜手术仿真训练系统中的应用研究.系统仿真学报, 2001, 13(2): 347-349 页
- [8] 熊岳山, 徐凯, 王彦臻等. 虚拟膝关节镜手术仿真系统的关键技术.国防科技大学学报, 2007, 1(1): 16-21P
- [9] Sederberg T W, Parry S R. Free-form deformation of solid geometric models[C].ACM SIGGRAPH computer graphics. ACM, 1986, 20(4): 151-160.
- [10] Gibson S,Fyock C.Volumetric object modeling for surgical simulation[J].Medical Image Analysis,1990, 2(2):121-132P
- [11] 朱玲.虚拟手中软体形变与切割技术研究.哈尔滨工程大学博士学位论文,2012:1-3, 17 页
- [12] Terzopoulos D, Platt J, Barr A, et al. Elastically deformable models[C].ACM Siggraph Computer Graphics. ACM, 1987, 21(4): 205-214.
- [13] Waters K, Terzopoulos D. A physical model of facial tissue and muscle articulation[C].Visualization in Biomedical Computing, 1990., Proceedings of the First Conference on. IEEE, 1990: 77-82.
- [14] Breen D E, House D H, Wozny M J. Predicting the drape of woven cloth using interacting particles[C].Proceedings of the 21st annual conference on Computer graphics

- and interactive techniques. ACM, 1994: 365-372.
- [15] Etheredge C E. A Parallel mass-spring model for soft tissue simulation with haptic rendering in CUDA [D]. Enschede:University of Twente, 2011
- [16] 鲍春波, 王博亮, 刘卓, 程明. 一种用于软组织变形仿真的动态质点弹簧模型[J]. 系统仿真学报, 2006, 18 (4): 847-851 页
- [17] Choi K S. Interactive cutting of deformable objects using force propagation approach and digital design analogy[J]. Computers & Graphics, 2006, 30(2): 233-243.
- [18] 徐少平, 刘小平, 张华, 罗洁. 虚拟手术中软组织实时形变模型的研究进展[J]. 生物医学工程学杂志, 2010, 02: 435-439.
- [19] Delingette H, Cotin S, Ayache N. A hybrid elastic model allowing real-time cutting, deformations and force-feedback for surgery training and simulation[C]//Computer animation, 1999. Proceedings. IEEE, 1999: 70-81.
- [20] 孙剑, 柴建云, 唐泽圣. 手术模拟中软组织变形的实时有限元新算法[J]. 系统仿真学报, 2000, 05: 553-556.
- [21] Dick C, Georgii J, Westermann R. A real-time multigrid finite hexahedra method for elasticity simulation using CUDA [J]. Simulation Modelling Practice and Theory, 2011, 19(2): 801-816.
- [22] 毛磊. 基于无网格方法的软组织形变模型[D]. 华北水利水电学院, 2011: 9-10 页.
- [23] G. R. Liu, Y. T. Gu 著 王建明, 周学军译. 无网格法理论及程序设计[M]. 济南: 山东大学出版社, 2007: 4 页
- [24] Nealen A, Müller M, Keiser R, et al. Physically based deformable models in computer graphics[C]//Computer Graphics Forum. Blackwell Publishing Ltd, 2006, 25(4): 809-836.
- [25] Lim Y J, De S. Real time simulation of nonlinear tissue response in virtual surgery using the point collocation-based method of finite spheres[J]. Computer Methods in Applied Mechanics and Engineering, 2007, 196(31): 3011-3024.
- [26] Müller M, Chentanez N. Solid simulation with oriented particles[C]. ACM transactions on graphics (TOG). ACM, 2011, 30(4): 92.
- [27] Chazelle B. An optimal algorithm for intersecting three-dimensional convex polyhedra[J]. SIAM Journal on Computing, 1992, 21(4): 671-696.
- [28] 王祎. 虚拟现实中碰撞检测关键技术研究[D]. 吉林大学, 2009: 1-12 页
- [29] 梁小红, 刘少强. 虚拟环境中的软体碰撞检测技术综述[J]. 计算机与数字工程, 2007, 03: 24-28+120.
- [30] 李焱, 卢晓军, 贺汉根. USSCD: 一个基于均匀空间分割的快速碰撞检测算法[J]. 中国

- 图象图形学报,2003,12:84-89.
- [31] Cohen J D, Lin M C, Manocha D, et al. I-collide: An interactive and exact collision detection system for large-scale environments[C]//Proceedings of the 1995 symposium on Interactive 3D graphics. ACM, 1995: 189-ff.
- [32] Coming D S, Staadt O G. Kinetic sweep and prune for multi-body continuous motion[J]. Computers & Graphics, 2006, 30(3): 439-449.
- [33] 谢倩茹,耿国华. 虚拟手术中的快速碰撞检测算法[J]. 计算机应用,2012,03:719-721.
- [34] Mirtich B. V-Clip: Fast and robust polyhedral collision detection[J]. ACM Transactions On Graphics (TOG), 1998, 17(3): 177-208.
- [35] Gilbert E G, Johnson D W, Keerthi S S. A fast procedure for computing the distance between complex objects in three-dimensional space[J]. Robotics and Automation, IEEE Journal of, 1988, 4(2): 193-203.
- [36] Cameron S. Enhancing GJK: Computing minimum and penetration distances between convex polyhedra[C].ICRA. 1997, 97: 20-25.
- [37] Guy S, Debunne G. Monte-carlo collision detection[R]. Technical Report RR-5136, INRIA, 2004.
- [38] 金汉均,李朝晖,张晓亮,郭亚军. 基于遗传算法的凸多面体间碰撞检测算法研究[J]. 华中师范大学学报(自然科学版),2006,01:25-28.
- [39] Shinya M, Fergie M C. Interference detection through rasterization[J]. The Journal of Visualization and Computer Animation, 1991, 2(4): 132-134.
- [40] Hoff III K E, Zaferakis A, Lin M, et al. Fast and simple 2D geometric proximity queries using graphics hardware[C].Proceedings of the 2001 symposium on Interactive 3D graphics. ACM, 2001: 145-148.
- [41] 王召伟. 医学图像分割与三维重建[D].电子科技大学,2009: 27-41 页
- [42] 唐闻. 制作心脏模型的三维模型的过程和方法 [J]. 科技信息(学术研究),2008,32:693-694.
- [43] 何晓乾. 医学图像三维重建技术研究[D].电子科技大学,2006: 5 页
- [44] 薛艳青. 融合结构关联信息的帕金森病靶区自动识别技术研究[D].北京工业大学,2013.
- [45] 薛艳青,杨春兰,高宏建,卢旺盛,吴水才. 利用 Amira 进行帕金森病靶点核团的分割及三维重建[J]. 医疗卫生装备,2012,08:1-3.
- [46] 靳碧鹏. 基于物理的计算机实时动画研究[D].电子科技大学,2007:43 页
- [47] 胡于进,王璋奇编著.有限元分析及应用[M].北京:清华大学出版社,2009:20-53 页
- [48] Müller M, Heidelberger B, Teschner M, et al. Meshless deformations based on shape

- p>matching[C]//ACM Transactions on Graphics (TOG). ACM, 2005, 24(3): 471-478.
- [49] Shoemake K, Duff T. Matrix animation and polar decomposition[C]//Proceedings of the conference on Graphics interface. 1992, 92(1992): 258-264.
- [50] 魏迎梅,王涌,吴泉源,石教英. 刚体在软体对象环境中的碰撞检测的研究[J]. 计算机学报,2001,08:802-808.
- [51] 于海燕,何援军. 空间两三角形的相交问题[J]. 图学学报,2013,04:54-62.
- [52] 朱元峰, 孟军, 谢光华, 马文娟. 基于复合层次包围盒的实时碰撞检测研究[J]. 系统仿真学报, 2008, 02:372-377.
- [53] 王晓荣. 基于 AABB 包围盒的碰撞检测算法的研究[D].华中师范大学,2007.
- [54] Christer Ericson 著, 刘天慧译.实时碰撞检测算法技术[M].清华大学出版社.2010:1-2, 46、50-53 页
- [55] 王志芳. 碰撞检测技术的研究及应用[D].太原科技大学,2012: 32-35.
- [56] Möller T.A fast triangle-triangle intersection test[J].Journal of Graphics Tools,1997,2(2): 25-30.
- [57] 于海燕,何援军. 空间两三角形的相交问题[J]. 图学学报,2013,04:54-62.
- [58] SUN Xie-hua. A New Algorithm of Line Clipping for Convex Polygons[J]. Journal of Image and Graphics,2003,12:115-117.
- [59] HE Da-hua , CHEN Chuan-bo. A Sufficient and Necessary Condition of Two Triangles' Non-Overlapping[J]. MATHEMATICA APPLICATA,2002,S1:28-30.
- [60] Richard S. Wright 著,付飞, 李艳辉译. OpenGL 超级宝典[M].北京:人民邮电出版社,2012: 20-23、56 页
- [61] 魏迎梅, 王涌, 吴泉源, 等. 虚拟手术仿真中碰撞检测问题的研究[J]. 系统仿真学报, 2000, 12(5): 572~ 575

攻读硕士学位期间发表的论文和取得的科研成果

- [1] Xiufen Ye, Le Huang, Lin Wang, Huiming Xing. An Improved Algorithm for Triangle to Triangle Intersection Test. Proceeding of the 2015 IEEE International Conference on Information and Automation: 2689-2694P, Lijiang, China, 2015. 8.
- [2] 叶秀芬, 黄乐. 基于形状匹配的软体变形算法. 哈尔滨工程大学自动化学院 2015 年学术年会. 2015. 8.

致 谢

充实快乐的时光总是过的很快，转眼间硕士研究生学习生活即将结束，在论文即将完成之际，回望过去的点点滴滴，心里满是感激。

首先，感谢我的导师叶秀芬教授，叶老师严谨的治学态度、渊博的专业知识、活跃的学术思维、忘我的工作热情和平易近人的风格在研究生阶段一直深深地感染和激励着我，让我在科研的道路找到了榜样；在论文的选题、研究、写作过程中叶老师都进行悉心指导，论文中所获得的成果都凝聚了叶老师大量的心血；在生活中叶老师给予了我大量的关心和照顾，让我少去了很多忧虑，在此对叶老师的辛勤付出表示由衷的感谢。

其次，特别感谢 408 教研室的王科俊、冯伟兴、莫宏伟、付斌老师，感谢他们在论文开题时提出的宝贵建议，使我的论文内容和架构更加充实合理。

再次，感谢实验室的师兄师弟师姐师妹们，共同营造了一个宽容有趣的学习生活氛围，让我感受到彼此间的深厚情谊，带给我学习和生活上的快乐。在这里尤其感谢张建国、李鹏两位博士师兄，在查找文献、论文写作和论文程序编写、调试过程中给予的大量指导和帮助。感谢王璘、邢会明、袁方同窗在两年半的学习生活中的鼓励与陪伴。

最后，衷心感谢我的父母，姐姐和哥哥，感谢你们在背后的支持与鼓励，你们的支持是我坚强、不断奋进的动力。马上就要离开学校了，在工作中我会更加努力，不辜负你们的期望，成就一番事业，好好回报你们。

两年半的时光，匆匆而过，为我留下了很多宝贵的回忆，在未来的时光里我会继续努力，不断前行。在离开之际，祝愿母校未来更加光明美好！