

# 软组织手术三维虚拟仿真系统的设计与实现

陈燕锋

指导教师：王贺升

## 摘 要：

虚拟手术是虚拟现实技术在医学领域的重要应用。本课题研究的虚拟手术仿真系统主要针对手术医生的培训。手术培训中强调操作者的手眼协调能力，手术仿真系统通过模拟人体的器官、手术器械及交互场景，让用户在视觉和触觉上得到反馈，进行手术操作的训练。这种培训方式可以大幅减少对医生的训练成本，且实验可重复。手术仿真系统所涉及的领域涵盖了计算机图形学、控制理论、机器人学、生物力学以及医学等。

本论文研究了软组织手术仿真系统中的一些关键技术，包括基于三角形面元和四面体体元的软组织几何建模，虚拟环境中的碰撞检测算法，软组织的切割分裂算法和 GPU 加速等等，都是非常基础且具备挑战性的问题。本文设计并实现了一个虚拟手术仿真系统，应用 OpenGL 库和 GPU 硬件加速技术对软组织的三维模型进行渲染，对肝脏和心脏等软组织器官进行仿真和切割，并应用了力反馈设备提供交互和触觉上的反馈。

**关键词：**虚拟手术；碰撞检测；四面体切割

目录

- 1 引言.....3
  - 1.1 课题研究背景.....3
  - 1.2 国内外研究现状.....3
- 2 系统概述.....4
  - 2.1 系统框架.....4
  - 2.2 硬件部分.....4
  - 2.3 软件部分.....4
- 3 软组织的几何模型及其渲染.....4
  - 3.1 三维几何建模技术.....5
  - 3.2 数据预处理模块.....5
  - 3.3 GPU 加速.....5
- 4 碰撞检测算法.....6
  - 4.1 碰撞检测算法概述.....6
  - 4.2 层次包围盒算法.....7
  - 4.3 算法优化.....9
- 5 切割分裂算法.....9
  - 5.1 表面网格切割算法.....9
  - 5.2 四面体切割算法.....11
  - 5.3 物理模型.....14
  - 5.4 切割效果.....15
- 6 系统的实现.....16
  - 6.1 数据结构.....16
  - 6.2 OpenGL 的应用.....16
  - 6.3 图形应用界面.....18
- 7 总结与展望.....19
- 8 致谢.....19
- 参考文献.....20

# 1 引言

## 1.1 课题研究背景

随着计算机图形图像技术与可视化技术的进步和发展,虚拟现实技术逐渐被应用到医疗领域,形成了计算机虚拟手术这个全新的领域。虚拟现实在医学领域的研究与应用发展得相当迅速。虚拟手术(Surgery Simulation,又称为 Medical VR)融合了计算机图形学、生物力学、传感器技术、现代医学、图像处理、机器人学、科学计算可视化等多学科交叉的研究内容。虚拟手术仿真系统是一个用来模拟在实际手术过程中可能遇到的各种情况的虚拟现实应用系统,所涉及的内容包括医学体数据的分割与可视化,虚拟手术刀切割操作模拟,以及对于软组织变形的模拟和各种感官反馈的模拟等。虚拟手术技术研究的最终目标是要给用户提供一个虚拟的手术环境,在视觉与触觉感官上为用户提供手术场景的真实再现。

传统的手术医生培训,是在人的尸体或动物身上进行的。利用动物进行培训,一方面,动物的解剖结构和人体的解剖结构不同,另一方面也会受到动物保护协会的谴责;而人的尸体又是不能多次使用的。虚拟手术仿真系统减少了对昂贵的教学实验对象的需求,降低了医生的培训费用。

为了达到较好的培训效果,手术仿真系统需要满足一定的真实性、实时性和稳定性。系统构建的软组织 3D 模型必须真实可靠,反馈必须及时,数据传输必须准确。找到优秀的模型构造和切割算法,提高系统的真实性和实时性,为手术医生提供极具真实感和沉浸感的模拟环境,便成为课题研究中重要的内容。

## 1.2 国内外研究现状

目前,世界各国都在致力于虚拟手术仿真系统的开发,日本欧美等国家的研究起步较早。针对各种手术的特点,研究人员从几何模型、物理模型、计算模型等方面,选取不同的方案进行仿真。瑞士工学院研发了 LASSO 虚拟腹腔镜系统<sup>[1,2]</sup>;德国卡尔斯鲁厄大学研制了 KISMET 虚拟手术开发环境<sup>[3]</sup>;法国 INRIA 的 Stephalle Cotin 等研究了肝脏在虚拟手术中的变形<sup>[4]</sup>;Stanfor 大学生物计算机中心开展了血管微手术的切割和缝合仿真研究;日本名古屋大学基于图形工作站建立三维手术模拟系统等。

2010 年,国际学术期刊 PBMB (Progress in Biophysics and Molecular Biology) 专门组织专刊发表了关于软组织建模的一系列最新学术论文,包括了软组织实时建模、GPU 并行计算、弹簧质点模型、无网格新型建模技术等等内容。

国内虚拟手术方面的研究起步较晚,主要集中在科研院校和研究所。2001 年 12 月,中国解放军总医院研制了通过视觉、触觉体验的国内第一套虚拟鼻腔内窥镜手术仿真系统;2002 年 10 月,我国首例数字化可视人体在重庆第三军医大学完成。清华大学、香港中文大学等开展了基于有限元方法的实时力反馈交互技术研究<sup>[5,6]</sup>;上海交通大学 Med-X 研究院着重进行了手术导航和虚拟手术系统的研发,并且在理论研究领域上积累了一定的成果与经验<sup>[7,8]</sup>;中科院自动化所复杂系统与智能科学重点实验室开发的 MITK 类库(Medical Imaging ToolKit)<sup>[9]</sup>,已经成为国内医学影像领域的一种可供选择的优秀软件工具。浙江大学针对虚拟手术中的人体组织三维重构、碰撞检测方法、软组织

切割仿真等关键技术进行了系统研究<sup>[10]</sup>。虽然现在已经有许多科研院所开展了虚拟手术仿真方面的研究并取得了一定的成果，但是，大部分工作还停留在实验阶段。

## 2 系统概述

本章将从系统框架、系统硬件构成、软件构成等方面详细介绍本项目中所搭建的仿真系统以及系统中所遇到的关键技术问题。

### 2.1 系统框架

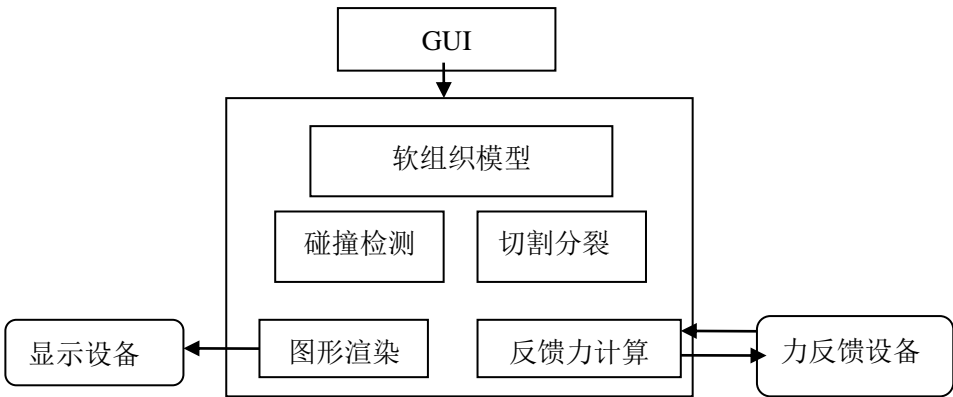


图 1 系统框架

图 1 为软组织手术三维仿真系统的系统框架示意图。根据仿真系统的功能，我们把系统模块化为以下几个部分：软组织模型数据导入模块、图形渲染模块、碰撞检测模块、切割分裂模块和反馈力计算模块等。

在手术仿真系统中，我们首先从医学影像中提取数据，绘制软组织的三维模型以及手术器械。用户主要通过力反馈设备来模拟手术器械，与虚拟场景进行交互。当碰撞检测模块检测到手术刀对软组织实施了切割动作时，切割分裂模块会对软组织模型进行分裂和更新，并重新渲染，而反馈力计算模块会通过力反馈设备对用户进行触觉上的反馈。

### 2.2 硬件部分

系统由显示器、处理器、力反馈设备等电子仪器组成。

显示器用于向用户显示三维的软组织和手术器械模型，提供视觉上的反馈。处理器用于系统中各模块的计算处理，包括中央处理器 CPU（Central Processing Unit）和图形处理器 GPU（Graphics Processing Unit），系统中使用了多线程处理和 GPU 硬件加速，因此这二者的处理性能对仿真系统的真实性和实时性有一定影响。力反馈设备用于模拟虚拟场景中的手术器械，如手术刀等，为用户提供交互，并实时进行触觉上的反馈，使用户在操作过程中感受到力的作用。在本系统中，我们使用的是 Novint 公司的产品 Falcon。此外，用户还可以通过键盘、鼠标等进行其他的辅助操作。

### 2.3 软件部分

本系统基于 C++ 语言进行实现，应用 Visual Studio 进行开发。在图形渲染方面，使用了 OpenGL 函数库；在力反馈设备交互方面，使用了 CHAI3D 函数库。这部分会在后面的系统实现中介绍。

## 3 软组织的几何模型及其渲染

### 3.1 三维几何建模技术

三维几何建模技术是虚拟手术系统研究的一个重要方面,虚拟手术环境中的一切操作都要在重建出的器官的三维模型上进行。三维几何建模就是从原始医学图像数据中(如 CT, M 刚等图像数据)提取出有用的信息,然后运用计算机图形学知识以及各种三维重建算法,在计算机中构建出逼真的三维器官模型。考虑到系统的真实性和实时性要求以及计算处理的需要,通常还需要重建出的网格模型进行简化和优化。

在虚拟手术系统中,几何模型主要分为面模型和体模型两类。面模型是用表面网格模型来表示研究对象,它只是表示了物体的表面外廓,无法表达内部信息。体模型则是用相应的充满整个模型空间的三维单元体来构造模型,不但表达了表面信息,也表达了物体内部信息。对面模型和体模型的选择主要考虑到计算效率和物理准确性的折衷。在构成模型的基元的选择上,结构化的基元,如三角形(面模型中)和四面体(体模型中),是手术仿真系统中几何建模的常用单元。

### 3.2 数据预处理模块

在本系统中,我们通过数据预处理模块导入构建好的软组织三维模型数据。利用三维建模软件,如 3DSMax 等,我们可以得到格式为\*.obj 的文件,存储了模型的原始数据,包括重构医学图像需要的所有点、线、面及其关系的数据。OBJ 文件格式支持直线(Line)、多边形(Polygon)、表面(Surface)和自由形态曲线(Free-form Curve),由一行行文本组成,有字的行都由一两个标记字母也就是关键字(Keyword)开头,关键字可以说明这一行是什么样的数据。

这种文件一般包含这样的数据结构:

顶点数据(Vertex data)

- v 几何体顶点(Geometric vertices)
- vt 贴图坐标点(Texture vertices)
- vn 顶点法线(Vertex normals)
- vp 参数空格顶点 (Parameter space vertices)

元素(Elements)

- p 点(Point)
- l 线(Line)
- f 面(Face)
- curv 曲线(Curve)
- curv2 2D 曲线(2D curve)
- surf 表面(Surface)

例如我们用来构建心脏模型的数据的一部分:

```
#  
v -0.936863 -0.332325 0.108849  
v -0.160347 -0.648645 -0.744008  
v -0.927743 0.367987 0.062274  
v 0.035066 0.576519 -0.816331#
```

v 表示几何体顶点,后面的数据分别为 (x, y, z) 形式的坐标。

对 obj 文件中的数据进行提取,我们就可以用 obj 文件生成应用 OpenGL 在仿真系统中绘制的 3D 模型了。

### 3.3 GPU 加速

系统中的图形渲染主要基于 OpenGL 库实现,这部分在后面系统实现部分介绍。此外,

为了提高实时性，系统运用了 GPU 硬件加速技术。

图形处理器（GPU），它是显卡的“心脏”，与 CPU 类似，只不过 GPU 是专为执行复杂的数学和几何计算而设计的，这些计算是图形渲染所必需的。时下的 GPU 多数拥有 2D 或 3D 图形加速功能。如果 CPU 想画一个二维图形，只需要发个指令给 GPU，GPU 就可以迅速计算出该图形的所有像素，并在显示器上指定位置画出相应的图形，画完后就通知 CPU，然后等待 CPU 发出下一条图形指令。有了 GPU，CPU 就从图形处理的任务中解放出来，可以执行其他更多的系统任务，这样可以大大提高计算机的整体性能。

GPU 现已得到了极大地发展,从最初局限于图形渲染的图形卡,发展为如今可编程的并行计算平台。GPU 可以应用在渲染之外的通用计算方法，如非线性最优化的并行计算、偏微分方程求解等。与 CPU 的串行计算模式不同，GPU 是一种高度并行的流处理器，具有更强的浮点计算能力，在物理模拟、信号分析等许多领域将高强度的计算任务以适当的方式转化为流数据计算模式，通过编程在 GPU 中进行加速计算，能获得一个数量级的速度提高。采用 GPU 来应用仿真算法，可以实现软组织变形的实时计算及渲染。

图 2 和图 3 分别显示了 GPU 加速前后肝脏模型的渲染效果。可以看出，图 2 边缘不平滑，可以明显看到边缘的锯齿，显示模型很粗糙。

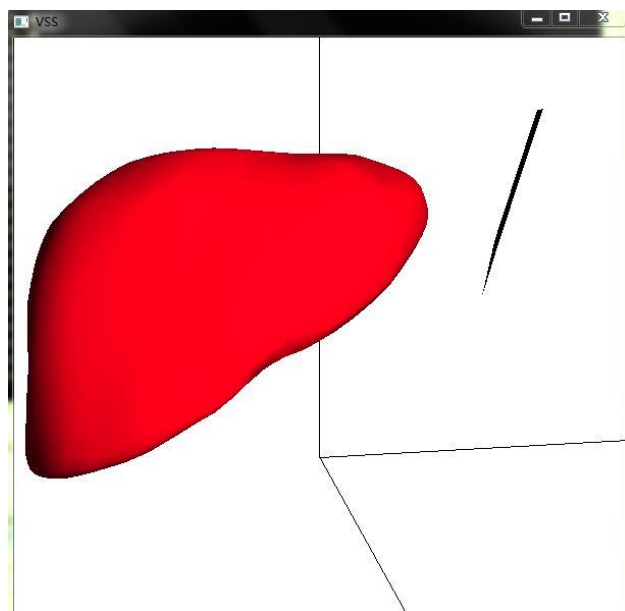


图 2 GPU 加速前

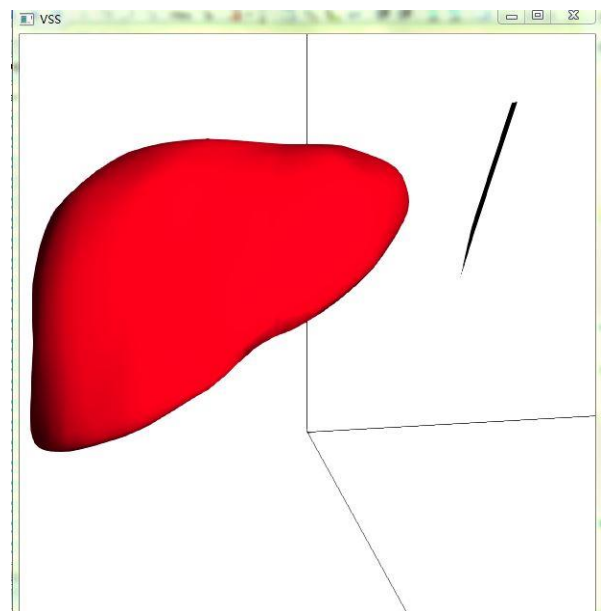


图 3 GPU 加速后

## 4 碰撞检测算法

### 4.1 碰撞检测算法概述

在虚拟手术系统中，为了增强虚拟操作的真实感与沉浸感，引入了碰撞检测技术，快速精确的碰撞检测是保证手术仿真真实性的必要条件。碰撞检测问题基于这样一个事实：两个不可穿透的对象不能同时占据同一空间区域。在虚拟环境中，由于手术者操作手术器械进行移动，手术器械与软组织模型之间会发生碰撞。在系统中，需要及时检测到这些碰撞的发生及发生的位置，为碰撞响应提供相关信息，以便碰撞物体做出正确的动作。

碰撞检测系统的输入模型是构成对象的基本几何元素的集合，最原始的碰撞检测算法是对两个输入模型中的所有基本元素进行两两相交测试，这种相交测试算法的时间复杂度是  $O(n^2)$ 。这种穷举法相交测试的效率是极其低下的。如何减少模型间基本元素两两相交测试的

数目，提高算法效率，以保证虚拟环境的实时交互性，是碰撞检测问题研究的核心。

碰撞检测方法大致可分为两类：空间分解法(space decomposition)和层次包围盒(hierarchical bounding volumes)方法。空间分解法是将整个虚拟场景划分成相等体积的小单元格，只对占据了同一单元格或相邻单元格的几何对象进行相交测试。层次包围盒方法的基本思想是用体积稍大且特性简单的几何体(称为包围盒)来包围对象并近似地代替复杂的几何对象进行相交测试，通过构造树状层次结构可以逐渐逼近对象的几何特性，进行相交测试时只需对有相交的包围盒进行进一步测试，从而大大减少了参与相交测试的基本几何元素的数目，提高了碰撞检测的效率。

下面详细介绍这种方法。

## 4.2 层次包围盒算法

### 4.2.1 层次包围和理论

对给定的  $n$  个基本几何元素的集合  $S$ ，定义  $S$  上的包围盒层次结构  $BVT(S)$  为一棵树，简称包围盒树，构造包围盒树通常有两种方法，自顶向下和自底向上。

(1) 自顶向下方法：从全集  $S$  出发，以全集  $S$  作为根节点，利用基于全集的信息递归的对这个结点进行划分以形成其子结点，直到到达叶结点，关键问题是如何把一个集合划分成若干个子集。

(2) 自底向上方法：从集合  $S$  的基本几何元素出发，每个元素对应一个叶结点，然后利用局部信息递归地对它们进行分组归并，形成父结点，直到得到一个单一的根结点（即集合  $S$ ），关键问题是如何把若干个子集合归并为一个父集。

包围盒树具有以下性质<sup>[13][14]</sup>：

- (1) 树中的每个结点  $v$  对应于  $S$  的一个子集  $S_v$ ， $S_v \subseteq S$ 。
- (2) 与每个结点  $v$  相关联的还有集合  $S_v$  的包围盒  $b(S_v)$ 。
- (3) 根结点对应全集  $S$  和  $S$  的包围盒  $b(S)$ 。
- (4) 树中的每个内部结点（非叶子结点）有两个以上的子结点，内部结点的最大子结点数称作树的度，记为  $\delta$ 。
- (5) 结点  $v$  的所有子结点所对应的基本几何元素子集是结点  $v$  的基本几何元素集合  $S_v$  的一个划分。

### 4.2.2 层次包围盒的遍历

假定已分别为两个对象  $E$  和  $F$  建立了层次包围盒树。在包围盒树中，每个结点上的包围盒都对应于组成该对象的基本几何元素集合的一个子集，根结点为整个对象的包围盒。碰撞检测算法的核心就是通过有效的遍历这两棵树，以确定在当前位置下，对象  $F$  的某些部分是否与对象  $E$  的某些部分发生碰撞。这是一个双重遍历的过程。算法首先用对象  $F$  的包围盒树的根结点遍历对象  $E$  的包围盒树，如果到达叶结点，再用该叶结点遍历对象  $F$  的包围盒树。如果能到达对象  $F$  的叶结点，则进一步进行基本几何元素的相交测试。

考虑两个层次包围盒二叉树  $BVT_E$ ， $BVT_F$  的相交检测。算法输入： $E$ ， $F$  两物体的层次包围盒二叉树  $BVT_E$ ， $BVT_F$ 。输出：布尔值，true 为相交，false 为不相交。伪代码如下：

```
bool collisionDetect( $BVT_E$ ,  $BVT_F$ )
    if  $BVT_E$ ,  $BVT_F$  not cross
        return false
    if  $BVT_E$ ,  $BVT_F$  are leaves
        crossdetect all elements of ( $BVT_E$ ,  $BVT_F$ )
        return true
    else if  $BVT_E$  is leaf ,  $BVT_F$  is not leaf
```

```

collisionDetect (BVTE, BVTF->leftChild)
collisionDetect (BVTE, BVTF->rightChild)
else if BVTE is not leaf , BVTF is leaf
collisionDetect (BVTE ->leftChild, BVTF)
collisionDetect (BVTE->rightChild, BVTF)
else
collisionDetect (BVTE, BVTF->leftChild)
collisionDetect (BVTE, BVTF->rightChild)
collisionDetect (BVTE ->leftChild, BVTF)
collisionDetect (BVTE->rightChild, BVTF)

```

#### 4.2.3 包围盒的选择

包围盒根据需求的不同，可以有多种选择，大致有球包围盒、AABB 包围盒、OBB 包围盒等等。球包围盒除了对三个坐标轴上分布得比较均匀的几何体外，包围球会留下很大的空隙，紧密性不够理想，因此使用比较少，不多作介绍。

AABB 包围盒被称为轴向包围盒，这种算法在碰撞检测的研究中最早被采用。AABB 包围盒的计算非常简单，计算组成对象的几何元素集合之需要 6 个标量（每个轴向上需要 2 个标量）。AABB 的重叠测试也非常简单，2 个 AABB 重叠当且仅当 3 个坐标轴上的投影均重叠。因此，可以将 2 个包围盒各自的 8 个顶点找出，大多数情况下一次比较就可以断定两者是不重叠的。AABB 包围盒最大的优点就是速度快，测试简单。但其缺点是检测的精度很低，如果想要更加细分包围盒的话，就需要以级数增长包围盒的数量，这并不划算。

OBB 包围盒称作方向包围盒，这种算法的最大特点是它可以根据被包围对象的形状确定包围盒最佳方向，以使得其尽可能紧密的包围对象，但这样也使得它的相交测试变得复杂。OBB 的计算相对也比较复杂一些，关键是找出最佳方向，并确定在该方向上包围对象的包围盒的最小尺寸。OBB 的相交测试基于分离轴理论，其相交测试代价比较大，但它的紧密性比较好，可以大大减少参与相交测试的包围盒数目。OBB 法由于方向任意，不像 AABB 包围盒那样，只能跟随物体坐标轴的方向。这种方法作为替代 AABB 的方法，各方面特性都更为折中，因此应用的范围得以增加。

考虑到实现的难易程度，在系统中我采取简单快速的 AABB 包围盒法。算法如下：

（1）首先采用包围盒的方法进行检测，将模型用一个大的正方体先包围起来，大正方体会细分为八个正方体，依次再细分下去。

（2）切割判据。当碰撞发生时，我们根据手术器械与软组织模型碰撞所产生的力度进行一个阈值的判断。大于此阈值时，断定切割已经开始。

（3）遍历四面体单元，寻找到切割的四面体。这里采用树的数据结构，在搜索切割到的四面体时，优先搜索与当前被切割的四面体相邻的四面体，由此可加快搜索速度。

（4）对所有发生切割的四面体进行处理。应用后面会介绍的切割分裂操作实现。

（5）整理四面体的模型数据，更新链表。

算法描述如下：

```

if (collisionDetect (currentTetra))
currentTetra->activeTetra
currentTetra->doneTetra

while activeTetra is not NULL
activeTetra->collisionTetra

```



```

while currentTetra's neighbor is not NULL
    if (collisionDetect(currentTetra's neighbor)) currentTetra's neighbor->activeTetra
    currentTetra's neighbor->doneTetra
    currentTetra's next neighbor
activeTetra's next

```

### 4.3 算法优化

基于 AABB 包围盒的基础上, 根据时空相关性, 我们在碰撞检测之前先进行状态的判断。如果上一时刻已经发生过切割, 则从上次切割树的父节点开始检测。这样能加快碰撞检测的速度。

## 5 切割分裂算法

虚拟切割是虚拟手术系统中的一个非常重要的操作。针对表面模型和体模型, 本文分别研究了相应的切割方法。在碰撞检测模块中, 需要用到相交测试, 相交测试包括线面相交测试和平面与四面体之间的相交测试。在相交测试中, 我们会更新被切割的点、线、面数据的相关状态, 为分裂动作提供必要的信息。

首先我们分析一下对手术刀模型的抽象。手术刀的刀片部位可以被抽象为一个三角形, 切割动作可以分两方面考虑: 刀尖轨迹在切割过程是一段线段, 与三角形面元发生相交; 刀刃轨迹在切割过程视为一个平面, 与四面体体元发生相交。因此, 切割过程的判断将分为线-面相交和面-体相交来介绍。

### 5.1 表面网格切割算法

#### 5.1.1 基本几何图形的位置关系判断

先介绍线段与三角形相交的算法。

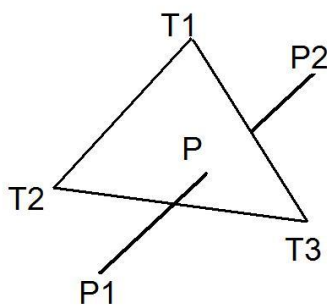


图 4 线面相交示意图

假定线段的端点是 P1 和 P2, 三角形的顶点是 T1, T2, 和 T3。这里点都是三维世界中的数据结构, 所以下面公式中都是一些向量计算。则相交的算法:

(1) 计算三角形所在平面的法线

$$N = (T2 - T1) \times (T3 - T1)$$

(5-1)

(2) 计算线段端点距离三角形所在平面的距离

$$u_0 = N(P1 - T1) \quad (5-2)$$

$$u_1 = N(P2 - T1) \quad (5-3)$$

(3) 若两个端点到三角形所在平面的距离同为正或同为负，则说明线段在平面的某一段，不相交。否则，计算线段与平面的交点 P 的参数坐标

$$u = \frac{u_0}{u_0 - u_1} \quad (5-4)$$

$$P = (1 - u)P1 + uP2 \quad (5-5)$$

(4) 判断 P 是否在三角形 T1 T2 T3 的内部

$$s1 = (T2 - P) \times (T3 - P) \cdot N \quad (5-6)$$

$$s2 = (T3 - P) \times (T1 - P) \cdot N \quad (5-7)$$

$$s3 = (T1 - P) \times (T2 - P) \cdot N \quad (5-8)$$

(5) 若 s0, s1, s2 中有负数，则线段与三角形不相交。否则，线段与三角形相交。

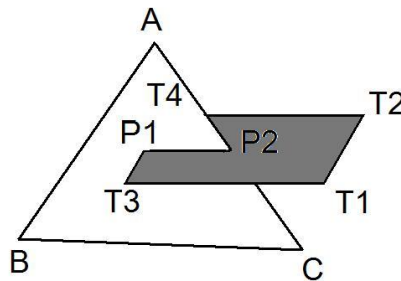


图 5 切面与三角形面元相交

根据上述方法求解可得到切面 T1 T2 T3 T4 与 AC 边的交点 P2，判断其是否在线段 AC 上；以及切面的一边 T3T4 与 ABC 的交点 P1，判断其是否在 ABC 内。满足上述两个条件，即认定平面与三角形发生了切割。

### 5.1.2 相交算法描述

平面与三角网格切割之后，只可能出现三角形面，或者四边形。而四边形可再分解成若干个三角形。使得算法具有一致性。因此算法总结如下：

(1) 碰撞检测，将被切割到的面三角形存入链表中，更新三角形的边交点，面交点信息。

(2) 边交点为 2 的三角形，必然是被分成一个三角形加上一个四边形，四边形取其重心，再重构成四个三角形。

(3) 点与点复制分裂，边与边复制分裂。

下图为示意图。

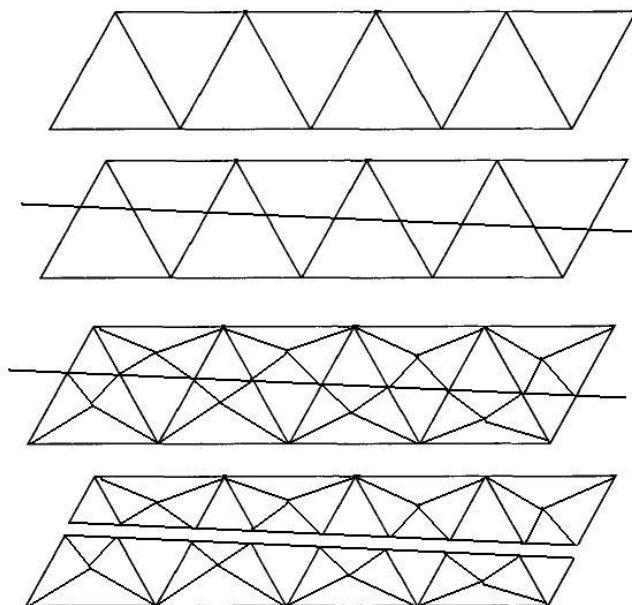


图 6 面模型切割示意图

## 5.2 四面体切割算法

体模型的切割算法相对于面模型的切割更加复杂。当前四面体模型的切割方法主要有两大类：去除法和分裂法。

去除法指的是，在模拟切割过程中，删除网格中所有与虚拟刀具相交的基本体元，1998年由 Morten Bro Nielsen 最早提出。该方法优点是实现简单，不对原有网格添加新的结点和单元，使得网格更新非常容易，便于在虚拟手术中实现实时切割。缺点是得到的网格边界不合理，会产生锯齿状的切割边界，尤其当网格模型不太精细的情况下这种缺陷尤为严重。此外由于接触到的体模型就去除掉，导致体模型的质量丢失，可能为其进行其他的运算带来困难。

分裂法顾名思义，是指分裂与虚拟刀具相交的单元体。该方法优点是得到的切割边界合理不会产生锯齿状走样。而分裂法又包括很多方法，下面进行简单介绍。

### 5.2.1 经典四面体切割算法介绍

考虑以一个平面来切割四面体的问题，经过数学的验证，共有五种拓扑结构，如图。

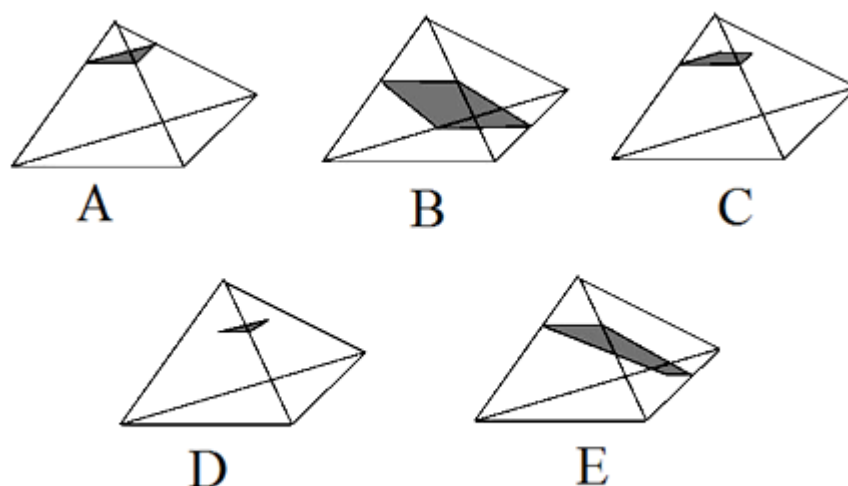


图 7 五种切割情况

上图给出了 A、B、C、D、E 五种不同的切割拓扑结构。通过旋转和对称性，这五种拓扑结构可咀表示所有的切割情况。A、B 为完全切割，将切割的四面体分成了两个部分。C、D、E 为部分切割，其中 C 为两条边被切割，D 为一条边被切割，E 为三条边被切割。

#### (1) 1: 17 细分法

1999 年 Daniel Bielser 提出 1: 17 的规则加减法<sup>[15]</sup>。即将模型用四面体网格构建，并且将每个四面体都预先分裂成 17 个小的四面体单元。这种方法大大提高了模型仿真的真实程度，使得仿真时的边缘锯齿和棱角较为光滑，对于单次的切割具有较好的效果。但如果你再次试图切割那些已经在上次切割过程中被分割的区域时，就会产生问题。这是因为经过 1: 17 细分之后的那些四面体网格不再是规则的，新产生的四面体与未被切割到的四面体间存在不一致性问题。此外，模型数量的级数增长，计算负担大，导致实时性出现问题。

#### (2) 状态机法

2004 年 Daniel Bielser 提出改进之后的状态机法。将四面体按面交点和边交点分裂，并将刀具与四面体的位置进行仔细的分辨。但这种方法产生大量的切割状态信息，也使得实时性难以实现。

#### (3) 最小子集法

2000 年 Andrew B Mor 提出最小子集法<sup>[11]</sup>，将四面体切割之后的拓扑结构分成五类，在构造新的四面体单元时，其顶点只使用切割产生的交点和原四面体的顶点，这样就使得在切割过程中产生的新顶点和新四面体的数量比较少。这样就使得四面体分裂之后的四面体单元达到最小的数量，控制了模型数量增长的程度，保证了计算的实时性。

表 5-1 传统的四面体切割方法比较

	优点	存在的问题
去除法	快速有效，实时性容易保证	模型切割后不真实，粗糙
1: 17 法	模型真实度有所增强	数据级数增长，运算负担大，实时性差
最小子集法	模型真实度高，数据增长速度可控，实时性好	不容易解决切割到点和边上的问题

状态机法	模型真实度高，数据增长速度可控	没有分裂的问题，但状态多，运算复杂
------	-----------------	-------------------

### 5.2.2 最小子集法介绍

基于上述分析，本系统采用 Andrew 提出的最小子集法，并在切割上进行了一定处理。

刀尖与刀的尾部划过的轨迹，可简化为一个平面，一般说来是平行四边形。这样就可以用两个三角形来代替一个四边形。问题即可简化为，三角形和四面体之间的切割问题。

当虚拟刀具扫略过四面体单元时，会产生两种交点：边交点和面交点。边交点为刀刃扫略面与四面体边的交点，面交点为刀刃尖端与四面体面的交点，体现了切割的深度信息。这些相交点信息是进行切割处理的前提，相交信息可由碰撞检测阶段取得。在最小子集法中，四面体切割之后的拓扑结构分成五类，如图所示。

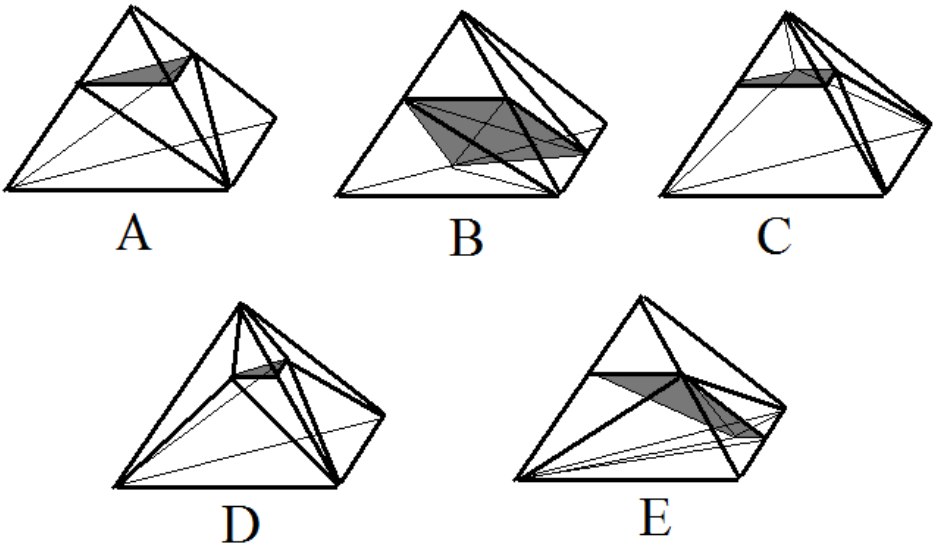


图 8 最小子集法切割

表 5-2 五种拓扑结构面交点与边交点一览

	新四面体数量	切割种类	可再次分割种类	面交点数量	边交点数量
A	4	完全切割	无	0	3
B	6	完全切割	无	0	4
C	8	不完全切割	A+A	2	2
D	6	不完全切割	A	2	1
E	9	不完全切割	A+B	1	3

Andrew 算法即最小子集法能解决大量的切割种类和问题。但是由图 7 即可知，如果平面切割到四面体的点或边上，最小子集法是无能为力的。如图 9 即为三种未考虑到的情况。

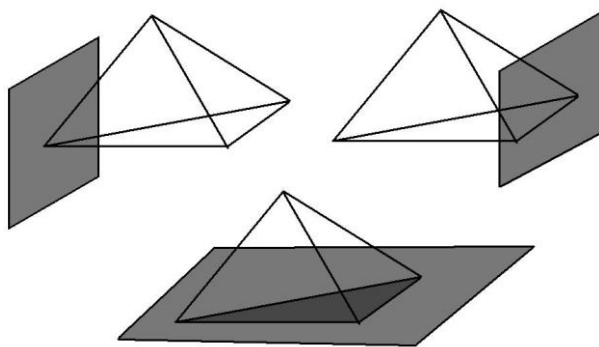


图 9 三种未考虑到的情况

当切割平面与四面体的一点、一边或一面重合时，我们的解决方法是对相交的点、边或面进行复制分裂。

因此四面体切割的算法如下：

(1) 检测到碰撞后，遍历所有切割到的四面体，计算这些四面体中被切割到的点、边、面的数量。

(2) 根据边交点和面交点的数量，如表 5-2，依照 Andrew 法进行归类。A 和 B 属于完全切割，按 Andrew 法分裂成最小子集即可。

C、D、E 属于不完全切割，按照上述的边交点和面交点的方式先行分裂，之后得到的四面体必然属于 A、B 之中的一种，再按照 A、B 的方式切割。

(3) 根据点交点、边交点和面交点的数量，属于特殊切割的，对相应的点、边、面进行复制分裂。否则，出现其他交点数量情况的不予切割。

(4) 对新生成的四面体，将其点、边、面加入链表，并删除已经分裂的四面体。

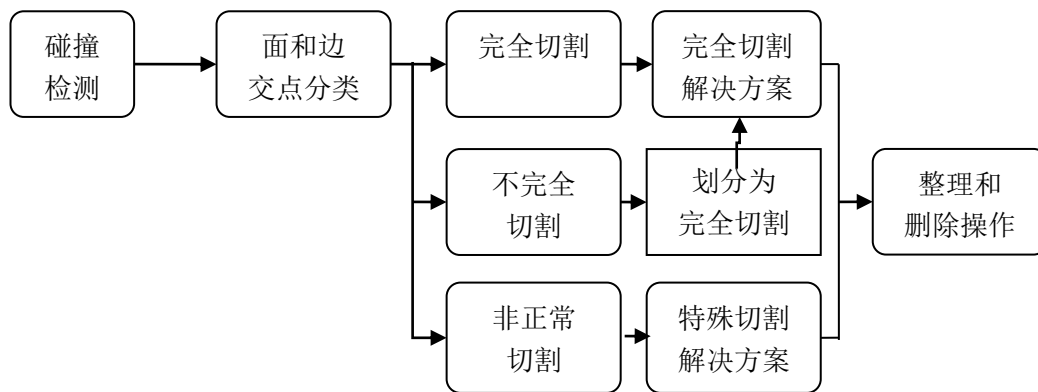


图 10 切割算法示意图

## 5.3 物理模型

软组织的物理模型放到这里介绍，主要是因为切割分裂算法中，分裂部分需要考虑到模型的形变效果。基于模型的物理特性，目前应用较为广泛的有“质点—弹簧模型”和“有限元模型”两类。

### 5.3.1 有限元方法

有限元模型的基本思想是，把连续体划分成许多在体内离散节点上相连的基本单元。在每个单元内用所假设的近似函数(插值函数)来分片地表示求解域内待定的变量，利用最小势

能原理建立起求解变量的平衡方程。通过单元间的连续性约束，在整个连续体上建立起关于所有节点变量的平衡方程，从而计算形变。

有限元方法能够体现形变中更多的细节部分，较真实地对软组织形变过程进行仿真，适合结合体模型进行计算。但是这个方法需要付出相当的计算代价和存储空间的开销，会影响系统的实时性。

### 5.3.2 质点弹簧模型

物体被离散成一些质点的集合，质点之间通过弹簧相连，构成一组网格结构，当整个系统的势能趋向于最小时，认为整个结构达到平衡状态。用这样的弹性网格结构，可近似模拟软组织的形变。如果在切割过程中破坏了固有的弹簧联结，则通过力学公式重新计算质点的位置，以达到模拟软组织被刀具切开后的效果<sup>[16][19]</sup>。

对一个单独的质点，其运动状态取决于其所受合力，主要考虑下面几个力的作用：

$$\text{弹簧力 } F_s = k_s(l - l_0)l_{i,j}$$

$$\text{阻尼力 } F_d = k_d(v_j - v_i)l_v$$

$$\text{重力 } F_g = k_g m_g l_g$$

$$\text{空气阻力 } F_a = -m_a k_a v_i l_v$$

系统的这一部分采用了经典的质点—弹簧模型，并忽略了重力和空气阻力的作用，假设手术中的器官是固定在平面之上的。为了运算中简单可行，又一定程度上保证切口的真实性，本设计系统不对质点弹簧重新构造模型，应用模型的点和边的数据结构直接执行。如果切割断掉某条边，则该边上的力消失，仅对该边上的两点的位置重新运算，不去叠加到更多的点上计算，只需考虑切割掉的这条边所蕴含的弹簧力和阻尼力即可，以保持切割过程的实时性。

质量弹簧模型相比于有限元模型，弱点是很多的。采用质量弹簧模型实际上是牺牲一部分精度，保证运算的实时性。

## 5.4 切割效果

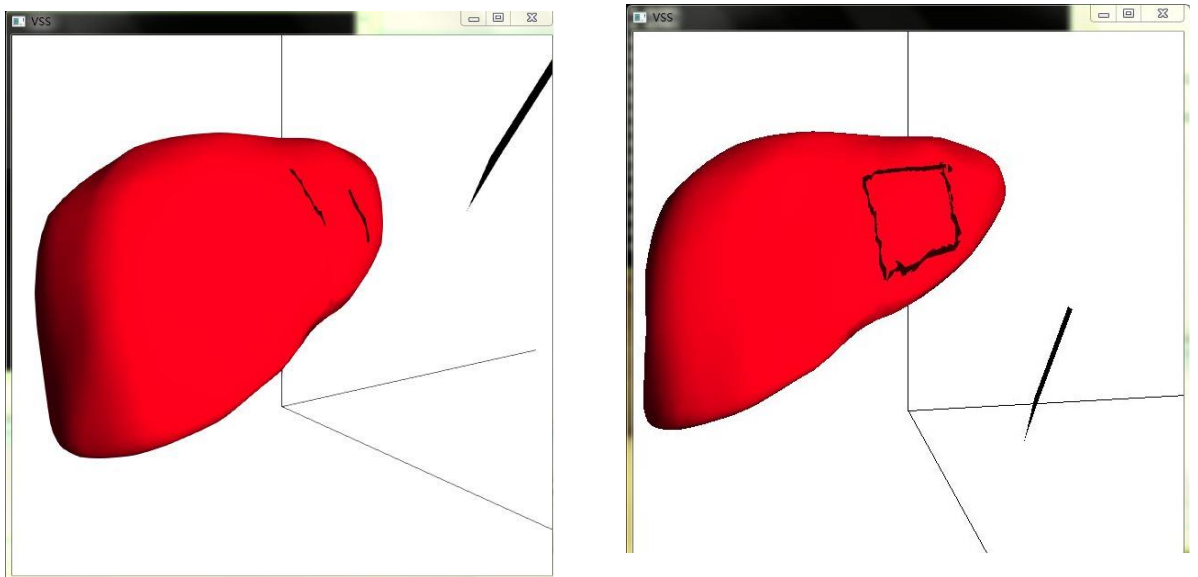


图 11 肝脏模型切割

## 6 系统的实现

### 6.1 数据结构

系统中，软组织模型采用面模型和体模型构成。因此，我们定义了点、线、三角形面元、四面体体元等类型，进行相关数据和切割分裂过程中相关状态的存储。以点类为例，点类中的成员包括点的三维坐标、点在 obj 文件中的序号（便于查询）、点是否在表面、是否处于切割状态等，其成员函数包括对点的构建、查询等操作。下面列出了点 point 类的成员。

```
class point
{
private:
    double px,py,pz;    //坐标
    int index;          //序号
    bool onsurface;     //是否在表面
    bool cross;         //是否被切割
    bool onsplit;       //是否分裂
    int split;          //分裂标识
    double normal[3];   //法线坐标
    double mass;        //质量
}
```

系统中对点、线、面元、体元的管理主要由单链表结构实现，方便进行插入节点、删除节点和查询节点等基本操作。

### 6.2 OpenGL 的应用

OpenGL 是实现 3D 图像显示的重要库，通过 OpenGL 库中的光照渲染等函数，可以使模型更具真实感。运用 OpenGL 库，我们实现了几种不同的渲染方式，包括线框模式、贴图模式等。



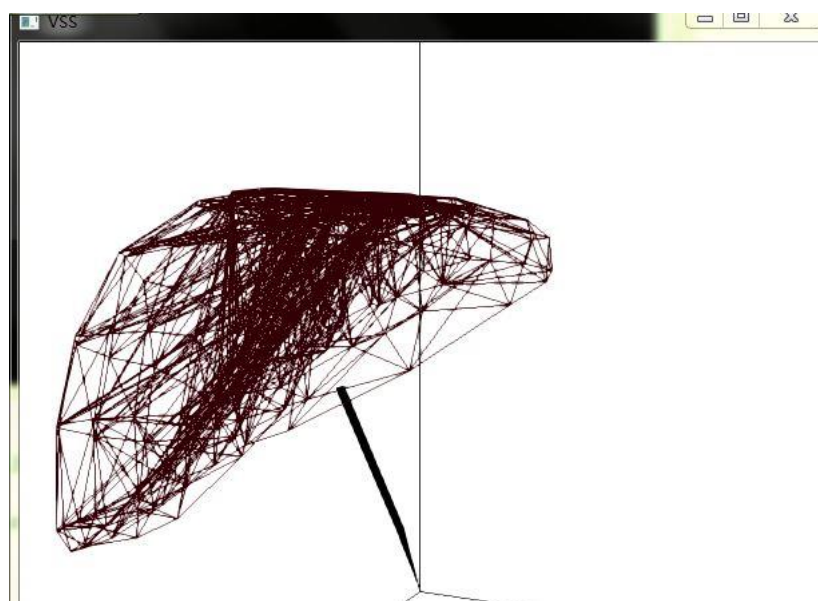


图 12 肝脏体模型

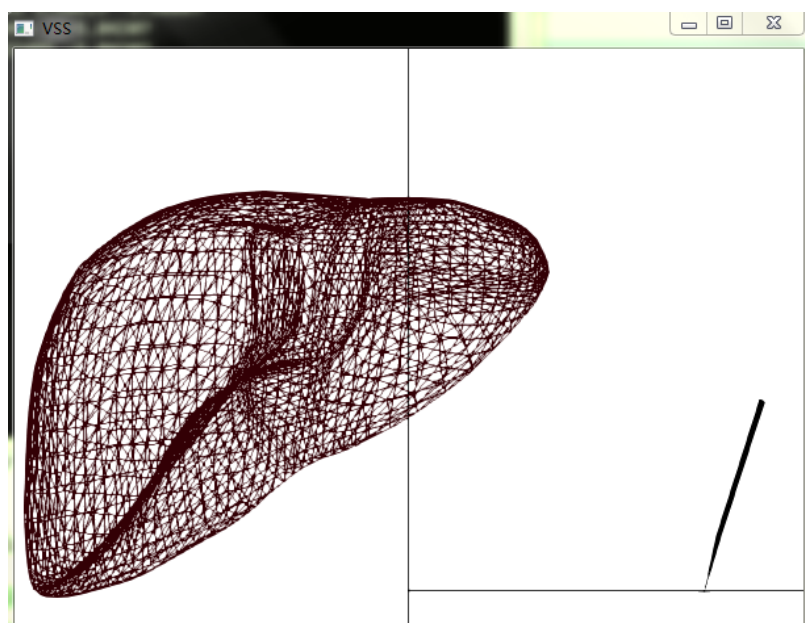


图 13 肝脏线框模型

光照与纹理对一个模型的真实性的来说至关重要。为一个模型建立光照效果是非常复杂的过程。计算机图形学上的光照模型，是根据对象表面的光照效果物理公式推导出来的，OpenGL 中自带丰富的光照处理函数。下图显示了模型加光照前后的效果。

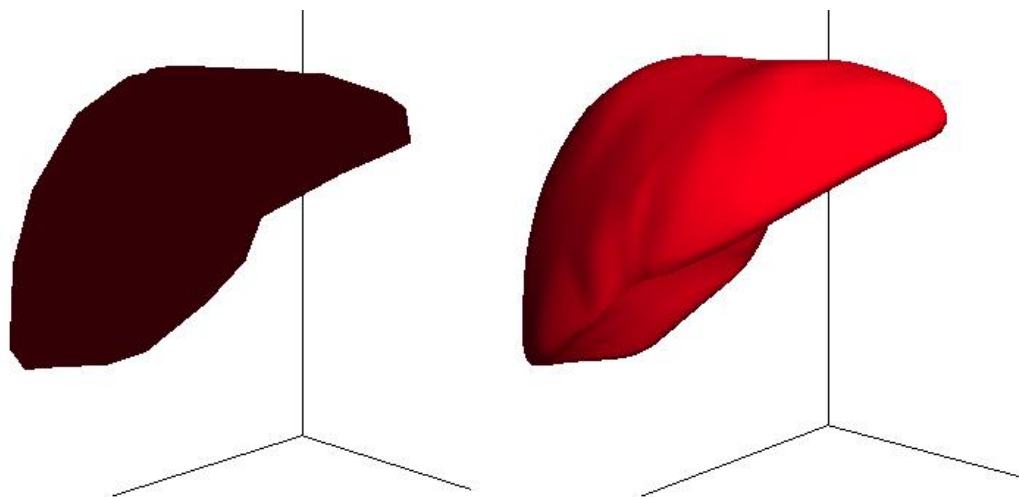


图 14 肝脏模型加光照前后

### 6.3 图形应用界面

一个软件程序的图形应用界面和用户交互是很重要的环节。用户可能会想要在屏幕上点击某一位置以获得该点的坐标或颜色等数据，又或者，用户需要用手术刀操作，切除软组织，还要得到力的反馈。软件程序中的菜单项应该具备一些功能，比如说改变显示的种类、颜色、背景等等。这些都要应用大量的交互式图形技术和控制设备。

本论文实现的系统中，除了切割外，提供了一些辅助功能，包括通过鼠标获取空间点坐标、对模型进行旋转、模型表现形式的切换等。

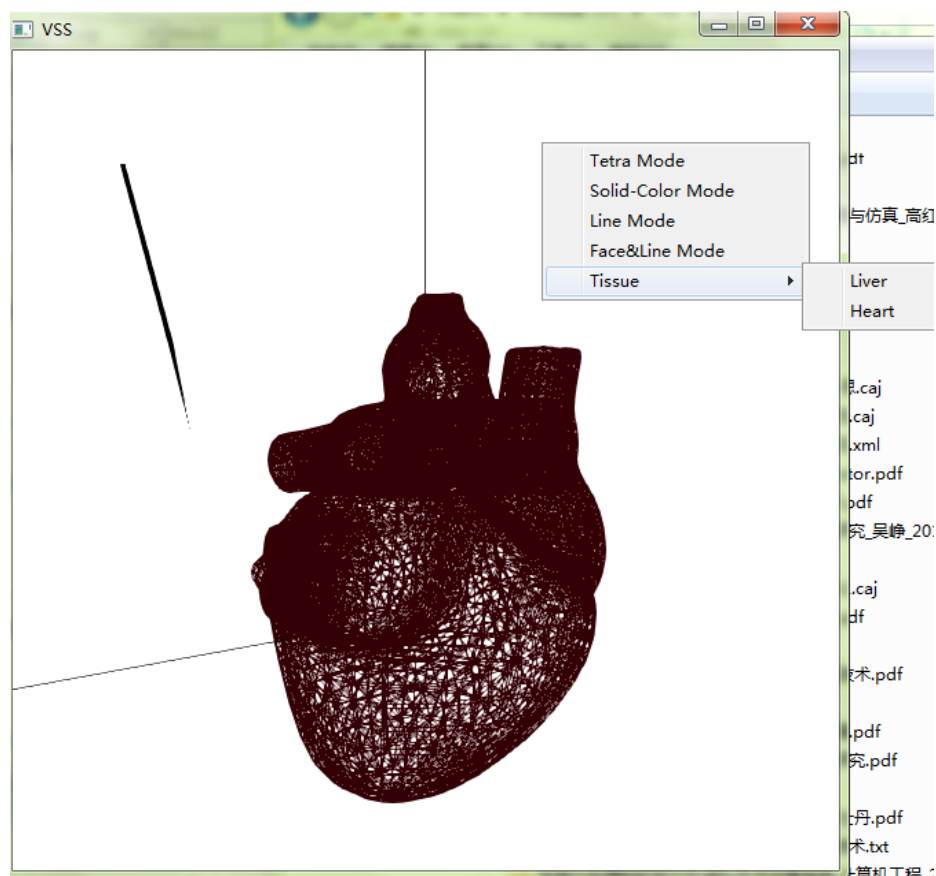


图 15 辅助功能示意图

## 6.1 力反馈设备的应用

系统的实现运用到了多线程技术。其中力反馈设备的交互就单独由一个线程来管理，实时监视力反馈设备的状态，并更新其传达给用户的力。

系统对力反馈设备的应用主要基于 CHAI3D 函数库实现。对力反馈进行识别、绑定和初始化之后，定义更新函数，监视设备状态，获取用户动作。在具体实现上，涉及到对库函数的学习和运用，这里不赘述。

碰撞检测模块检测到切割后，对用户给予触觉上的反馈，反馈力的更新算法如下：

```
If collisionDetected
    getLinearVelocity(linearVelocity);
    cVector3d force = cMul(-10, linearVelocity); //对力的大小进行换算
    hapticDevice->setForce(force); //为力反馈设备传送数据
```

## 7 总结与展望

### 7.1 工作总结

本文研究了软组织手术三维虚拟仿真系统中的一些关键技术，包括几何模型的构建、碰撞检测算法、切割分裂算法和 GPU 加速技术等，涉及到计算机图形学、建模方法和切割算法等多方面知识；实现了具有一定真实性和实时性的虚拟手术仿真系统，力反馈设备的应用提高了系统的交互性。

### 7.2 工作展望

系统中仍存在一定的不足之处，可以从以下几个方面进行完善：

- (1) 软组织弹性力学模型的优化。好的弹性力学模型能使系统具有更好的真实性。
- (2) 使用具有更多自由度的力反馈设备。本系统使用的设备具有三个自由度，因此只能实现简单的切割操作，增加设备的自由度，甚至是增加设备的数量，则能在系统上进行拓展，实现更多功能。
- (3) 渲染过程的优化。若能增加虚拟场景的渲染，加入血液流出等动作，应该能更真实地模拟手术场景；但是这也需要一定的医学知识，来保证场景与所模拟过程的一致性。

## 8 致谢

感谢李政道先生和秦惠箬女士的箬政基金资助，感谢箬政项目提供的这个平台，让我们在本科阶段能探索自己感兴趣的科研领域，感谢箬政项目管理委员会多次举办模拟国际学术会议、君政周活动等，让我们有机会和更多学者进行交流，认识自己的不足。

本研究是在我的导师王贺升教授的亲切关怀和悉心指导下完成的，感谢导师一年多来的指导和各方面的帮助。

感谢研究生学姐蔡茂佼在毕业前对我项目中的指导。感谢我的父母给予我的支持和关怀。感谢所有给予我指关心和支持的人们。

## 参考文献

- [1] Mendoza C. , Laugier C. Simulating Cutting in Surgery Applications Using Haptics and Finite Element Models[C]. Proceedings of the IEEE Virtual Reality,2003
- [2] Zhang H. , Shahram P. On Cutting and Dissection of Virtual Deformable Objects[C]. International Conference on Robotics & Automation, 2004, 4: 3908-3913
- [3] Nienhuys H. W. , Frank A. Supporting Cuts and Finite Element Deformation in Interactive Surgery Simulation[C]. MICCAI, 2001: 148-152
- [4] Cofin.S., Delingae H., AyacheN. Efficient Linear Elastic Models of Soft Tissues for Real-time Surgery Simulation[R]. INRIA Tech. Report,N35 10,1998
- [5] 吴雯, 孙剑, 王平安. 用于三维软组织实时切割的混合凝聚有限元模型[J]. 计算机学报, 2002, 25(9): 953-958
- [6] 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, 2000,12(5): 553-556
- [7] P. Huang, L. Gu, and S. Zhang, "Real-Time Simulation for Global Deformation of Soft Tissue Using Deformable Centerline and Medial Representation," ISBMS, pp. 67-74, 2006.
- [8] S. Zhang, L. Gu, et al., "Real-time virtual surgery simulation employing MM-Model and Adaptive spatial hashing," ICAT, 2006.
- [9] I. Bitter, R. Van Uiter, I. Wolf, et al., "Comparison of Four Freely Available Frameworks for Image Processing and Visualization That Use ITK," IEEE Transactions on Visualization and Computer Graphics, vol. 13, no. 3, pp.483-493, 2007.
- [10] 阎丽霞 虚拟手术关键技术研究[D]. 杭州: 浙江大学, 2001
- [11] Andrew B. Mor. Takeo Kanade. Modifying soft tissue models: progressive cutting with minimal new element creation[J]. In: S.L.Delp, A.M. DiGioia, B.Jaramaz, eds. Medical Image Computing and Computer-Assisted Intervention. Heidelberg, Berlin. Springer-Verlag, 2000, 1935(2000):598~607
- [12] 魏迎梅. 虚拟环境中碰撞检测问题的研究[D]. 中国人民解放军国防科学技术大学, 2000:28~38
- [13] Van den Bergen G. Efficient collision detection of complex deformable models using AABB. [J] Journal of Graphics Tools: JGT, 1997, 2(4): 1-14
- [14] Wagner C, Sehll M. Manner Collision Detection and Tissue Modeling in a VR Simulator for Eye Surgery. [J] In: ACM International Conference Proceeding Series, Proceedings of the workshop on Virtual environments, PP. 27-36
- [15] Daniel Bielser, M H. Gross. Interactive simulation of surgical cuts[J]. In: Eighth Pacific Conference on Computer Graphics and Applications Hong Kong, China: IEEE Computer Society Press, 2000,116~125
- [16] Donald Hearn, M. Pauline Baker. 计算机图形学[M]. 北京:电子工业出版社,2011:511~531
- [17] 邓郑祥. OpenGL 编程指南[M]. 北京:人民邮电出版社,2005:245~305
- [18] 石琼. 基于 OpenGL 的三维建模实现方法[J]. 计算机工程与应用. 2004:18
- [19] 高永辉 虚拟手术系统关键技术研究[D] 上海交通大学 2011:76~89
- [20] 吴恩华, 柳有权 基于图形处理器(GPU)的通用计算. 计算机辅助设计与图形学学报 2004,16(5):601~612

- [21] Morten Bro-Nielsen. Finite element modeling in surgery simulation[J]. Proceedings of the IEEE, 1998, 86(3): 490~503
- [22] Daniel Bielser, M H. Gross. Interactive simulation of surgical cuts[J]. In: Eighth Pacific Conference on Computer Graphics and Applications Hong Kong, China: IEEE Computer Society Press, 2000,116~125
- [23] Daniel Bielser, M H. Gross. Open surgery simulation [C]. Proceedings of Medicine Meets Virtual Reality, 2002. Newport Beach, CA, USA: IOS Press, 2002: 57-62.
- [24] Andrew B. Mor. Progressive Cutting with Minimal New Element Creation of Soft Tissue Models for Interactive Surgical Simulation [D]. USA: The Robotics Institute, Carnegie Mellon University, 2001