

件转换为指令专递给相应的模块。键盘事件主要针对渲染模块,不同的按键对应不同的指令,用于控制仿真场景中的物体,如地板、场景盒等,是否进行渲染,或者控制肝脏模型的渲染属性,如是否显示包围盒,网格线等。

GUI 参数设置子模块:主要针对形变仿真的参数,负责提供设置接口,使用户能够对这些参数进行交互式地调整。

2.3 渲染模块

渲染模块包括样本体纹理生成子模块、肝脏体纹理合成与映射子模块。

样本体纹理生成子模块:在基于样图的二维纹理合成中,首先需要获取二维的样图作为合成的参考。在三维空间中,合成体纹理也需要获取能够反映全局特征的纹理信息的三维样本块,即样本体纹理。在本文的研究中,需要以肝脏的轮廓中提取出的 ROI 区域为二维纹理样本,来合成包含丰富纹理信息的三维纹理块。

肝脏体纹理合成与映射子模块:为了将体纹理样本映射到肝脏体模型上形成纹理细节丰富且真实感强的肝脏体纹理,需要建立精细的肝脏体纹理空间。本文通过合成一个能够包含肝脏模型大小的体纹理空间,将体纹理从纹理空间映射到模型空间中,来生成具有真实感的肝脏模型。

2.4 切割模块

切割模块的主要功能是完成切割操作。切割操作是虚拟肝脏手术中最基本的交互方式之一,通过切割不仅需要模型分割成两个部分,还要将纹理从肝脏体纹理空间映射到切割面上,形成真实的切割面纹理。本系统采用鼠标来模拟虚拟手术器械进行切割操作,切割模块包括碰撞检测子模块和模型网络重构模块。

碰撞检测子模块在切割仿真的不同阶段起着不同的作用:在切割未执行阶段,碰撞检测主要用于确定切割操作的起始时间和起始位置;而在切割操作完成之后,碰撞检测则用于检测仿真物体与其他场景物体的接触情况,进而做出相应的处理,产生必要的碰撞反应。

模型网络重构子模块将用户通过鼠标在屏幕上划动形成连续的切割线量化为若干切割线段,然后将每一条线段投影到肝脏模型上与模型的表面产生交点,连接交点形成一个有界平面。接着通过径向基函数来拟合线段集所产生的有界平面来生成一个新的平面,即切割面。

3 关键技术

3.1 贝塞尔曲线

贝塞尔曲线^[12]由起始点、锚点(即终止点)和控制点组成,通过调整控制点,可以使其形状发生变化。贝塞尔曲线有以下几种形式,用于绘制各类二维平面上的曲线形状。

(1) 一阶贝塞尔曲线:

$$B(t) = (1-t)p_0 + tp_1, t \in [0, 1] \quad (1)$$

其中, $B(t)$ 为 t 时刻下的点的坐标值, p_0 为起始点, p_1 为终止点。

(2) 二阶贝塞尔曲线:

$$B(t) = (1-t)^2 p_0 + 2t(1-t)p_1 + t^2 p_2, t \in [0, 1] \quad (2)$$

二阶贝塞尔曲线由 p_0 、 p_1 和 p_2 三个定点确定,在二维平面上表现为一条平滑的抛物线。

(3) n 阶贝塞尔曲线:

通过低阶贝塞尔曲线的参数形式,可以推导出基于给定点 p_0 、 p_1 、 p_2 直至 p_n 的 n 阶贝塞尔曲线通式为:

$$B(t) = \sum_{i=0}^n \binom{n}{i} p_i (1-t)^{n-i} t^i, t \in [0, 1] \quad (3)$$

本文采用人工确定关键点结合贝塞尔曲线拟合的方式来提取肝脏的轮廓,为了获得较好的实验效果,需要找到合适的控制点。首先沿着肝脏的轮廓间隔地指定拟合的关键点,关键点的位置尽量贴合肝脏轮廓的边缘,依次连接轮廓上各关键点组成一个多边形。连接该多边形各边上的中点,得到中点的连线段。在改中点的连线段上找出一关键点,使得该点分线段的比与相邻两边的长度比相等,最后平移中点的连线段,使该线段上的关键点与对应的顶点重合,从而完成曲线拟合。曲线拟合示意图如图 2 所示。

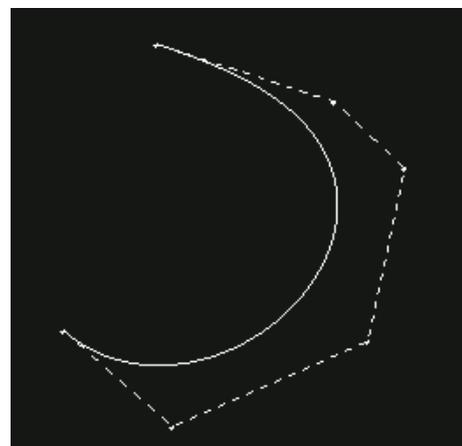


图 2 曲线拟合示意图

3.2 区域生长法

区域生长法的基本思想是根据像素的相似性生长的原则来划分集合构成分割的区域. 首先在预分割的区域内确定一个种子点, 该点必须是待分割区域内具有代表性的点, 种子点可以是单个像素点, 也可以是相似的像素点区域. 在种子点的邻域内寻找与该点相似的像素, 相似性的判定的原则是计算周边像素点与种子点彩色、梯度、灰度等量间的距离是否在差值范围内, 将符合条件的像素点包含到种子点的区域内, 然后以新区域内的像素点作为新的种子点继续生长, 直到找不到满足条件的像素点为止^[13].

3.3 层间插值法

假设相邻的断层图像为第 $k-1$ 层和第 $k+1$ 层, 需要通过插值法计算出第 k 层的图像轮廓数据. 但由于肝脏的轮廓包含了大量的点, 直接对其进行形状插值将引入大量的计算, 为了简化计算, 通常采用的方法是在原始的轮廓中寻找一个点, 使剩下的点构成的多边形的面积与原来最接近. 以此类推, 分别找到面积差值最小的 m 个点. 本文采用了等分圆周的方法来简化轮廓点数, 从而获得最逼近原始轮廓的多边形. 首先假设原始轮廓上的点数为 N , 经过简化后轮廓上的最终点数为 M ($M < N$), 计算相邻轮廓 C_{k-1} 和 C_{k+1} 的质心 \bar{m}_{k-1} 、 \bar{m}_{k+1} :

$$\bar{m}_{k-1} = \begin{cases} \bar{x}_{k-1} = \frac{1}{N_{k-1}} \sum_{i=1}^{N_{k-1}} x_i \\ \bar{y}_{k-1} = \frac{1}{N_{k-1}} \sum_{i=1}^{N_{k-1}} y_i \end{cases} \quad (4)$$

$$\bar{m}_{k+1} = \begin{cases} \bar{x}_{k+1} = \frac{1}{N_{k+1}} \sum_{i=1}^{N_{k+1}} x_i \\ \bar{y}_{k+1} = \frac{1}{N_{k+1}} \sum_{i=1}^{N_{k+1}} y_i \end{cases} \quad (5)$$

过 C_{k+1} 的质点 \bar{m}_{k+1} 作一条平行于 Z 轴的直线, 平移 C_{k-1} 使该直线经过其质点 \bar{m}_{k-1} , 然后将 C_{k-1} 投影到 C_{k+1} 所在的平面上, 记其投影轮廓为 C'_{k-1} . 接下来以质点为起点, 任意角度为起始角度, 间隔角度为 $360^\circ/M$ 向外作射线, 射线与 C'_{k-1} 和 C_{k+1} 的交点即为新的轮廓点, 只要 M 值足够大, 就能够使新的轮廓点连接成的多边形尽可能地逼近原始轮廓线, 分别记简化后的轮廓为 C''_{k-1} 和 C'_{k+1} , 假设第 $k-1, k, k+1$ 层 Z 方向上的坐标为 z_{k-1}, z_k 和 z_{k+1} , 则计算 C_k 的插值公式为:

$$C_k = \frac{z_{k+1} - z_k}{z_{k+1} - z_{k-1}} \cdot C''_{k-1} + \frac{z_k - z_{k-1}}{z_{k+1} - z_{k-1}} \cdot C'_{k+1} \quad (6)$$

3.4 基于 CUDA 的并行计算

基于 CUDA 的并行计算主要分为以下 6 个步骤, 计算流程如图 3 所示.

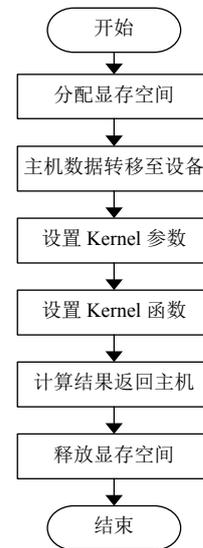


图 3 基于 CUDA 的并行计算流程图

步骤 1. 在利用 CUDA 进行并行计算之前, 需要在 GPU 上为需要进行运算的数据及运算结果申请相应的显存空间.

步骤 2. 在 GPU 中分配好相应的显存空间后, 将内存地址中需要在 GPU 上进行运算的数据拷贝至显存空间中.

步骤 3. 在利用 CUDA 进行 GPU 计算时, 需要启动合适的线程数来覆盖并行计算的规模. 例如进行网格节点着色计算时, 定义 Kernel 函数名为 `facePointKernel`, 其线程表达式为 `facePointKernel<<<(faceNum+1023)/1024,1024>>>(…)`, 其中 `faceNum` 为表面网格节点数, `<<<(faceNum+1023)/1024,1024>>>` 表示启动的线程块数为 $(\text{faceNum}+1023)/1024$, 每个线程块启动 1024 条线程, (\dots) 中为需要转移到 GPU 上计算的数据.

步骤 4. 设置 Kernel 函数即 CUDA 的核函数. 所有的并行计算都在 Kernel 中进行, 为了保证计算的并行性, 需要建立线程索引, 使索引的每一个线程都运行一次 Kernel 函数中的计算.

步骤 5. 在 Kernel 中完成计算后, 需要将计算的结果传回主机, 将数据从 GPU 的显存拷贝到 CPU 的内存中.

步骤 6. 在 CUDA 上完成计算后, 也需要释放动态分配的显存空间, 以免造成泄漏.

4 实现

在虚拟肝脏手术中, 需要利用人体连续横断面中肝脏轮廓的数据来生成肝脏模型以及肝脏的样本纹理块. 本文采用了美国可视化人体项目的 VHP 图片集来作为研究的数据来源, 因此对 VHP 数据集进行肝脏轮廓的提取就成为了研究工作的基础.

对 VHP 数据集进行肝脏轮廓提取是对一系列图像进行处理的过程, 因此在进行轮廓提取的操作之前, 需要先将数据集导入到系统中. 为了确保数据集能够顺序地在系统中进行处理, 在导入前对数据集采用依顺序编号命名的原则, 使系统在导入数据集的过程中能够按照各图片的文件名进行先后次序的排列. 系统利用 OpenCV 的 cvLoadImage() 函数来实现图像数据的读取, 由于 VHP 数据集的图像数据较大, 包含了许多无用的信息, 为了避免进行过多的处理, 需要先对读取的图像进行裁剪处理, 根据肝脏所在区域大小将图像裁剪成分辨率为 600×630 的图像集. 在如图 4 所示的系统界面中, 能够查看导入到系统后经过裁剪的每一张肝脏横断面图像.



图 4 图片导入与显示

肝脏轮廓提取的精度直接影响到三维重建后肝脏模型的还原程度, 为了提高图像分割的精度, 本系统采用人工参与肝脏轮廓关键点的指定, 再用贝塞尔曲线拟合曲线的方法来标记出肝脏的轮廓, 由于导入到系统中的图像为连续的肝脏横断面图像, 因此相邻两层之间的轮廓差异性较小, 可以采用当前图像指定的关键点作为下一张图像的起始点, 再进行细微的调整即

可. 随着肝脏轮廓的变化, 只需根据需要对关键点的数量进行增加或删除, 便可以方便地标记出 VHP 数据集中各横断面上的肝脏轮廓. 贝塞尔曲线拟合的结果如图 5 所示.

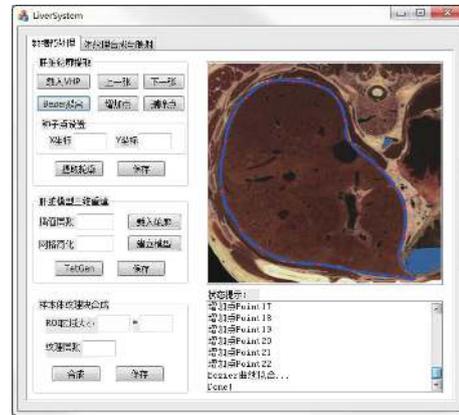


图 5 拟合结果

通过人工拟定关键点及贝塞尔曲线的拟合, 在连续的横断面图像中标记出肝脏区域的轮廓, 然后对肝脏的区域进行分割提取. 图像分割结果如图 6 所示.

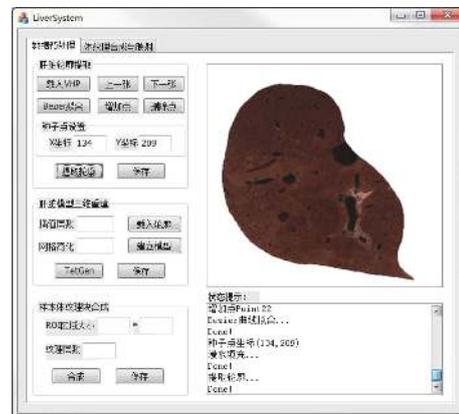


图 6 图像分割结果

利用分割出的肝脏轮廓数据, 采用层间插值及面绘制的方法对肝脏模型进行三维重建. 层间插值的目的是为了建立更精细的肝脏模型. 根据 VHP 数据集提供的数据, 其相邻的横断面之间的距离为 1 mm, 为了改善断层的过渡情况, 提高模型的精度, 在相邻两层切面之间插入额外的两层切面, 插值面的轮廓形状由相邻的两切面计算得到. 本系统采用基于多边形逼近的形状插值法来进行层间的插值, 首先利用等圆周的方法来简化原轮廓上点的数量, 以获得最大程度逼近原始轮廓面积的多边形, 再利用 3.3 中的插值公式进行插

值轮廓的计算. 在进行层间插值后, 构建模型的切面的总层数为 $188+187 \times 2=562$ 层.

本系统采用 VTK 工具包结合 Marching Cubes 面绘制的方法, 利用 562 层切面轮廓进行肝脏模型的三维重建. 在尚未进行网格数简化的情况下, 即网格简化率为 1 时, 未进行网格削减的肝脏表面网格模型如图 7 所示.

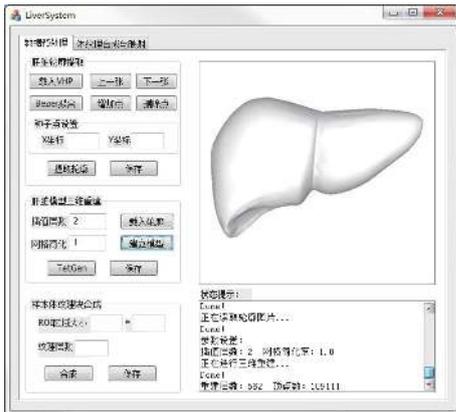


图 7 尚未进行网格削减的肝脏表面网格模型

未经过网格简化处理的模型包含了过多的顶点, 这将给后续的映射计算及切割带来大量的数据计算, 造成系统实时性的降低. 为了避免这种情况的发生, 需要对初始模型进行一定程度的网格削减工作. 本系统采用了 VTK 封装的 VTKDecimatePro 网格削减类, 通过设置不同的简化率来减少初始模型表面的网格数目, 网格简化率为 0.3 时所生成的肝脏体模型如图 8 所示.

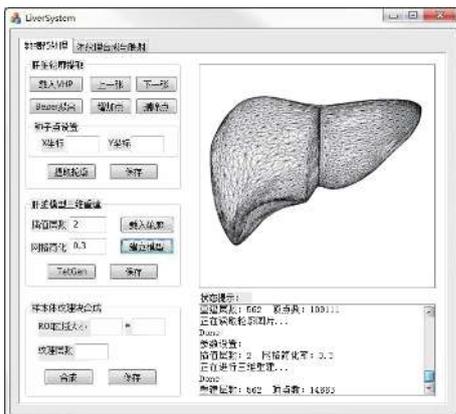


图 8 简化率为 0.3 的肝脏表面网格模型

5 切割效果与分析

本文实现了肝脏体模型的切割和切面体纹理的快

速绘制. 本文分别在 GT 540M 和 GTX 650 的 GPU 以及 Intel® Core™ i7-2670QM 2.2 GHz 的 CPU 上进行了实验, 并选用了顶点数 14 663, 三角面片数 18 676 的肝脏模型作为切割模型, 针对 3 组不同形状的切割进行了实验数据的记录, 并与 CPU 下的切面纹理的映射速度进行了对比, 切割效果如图 9 所示.

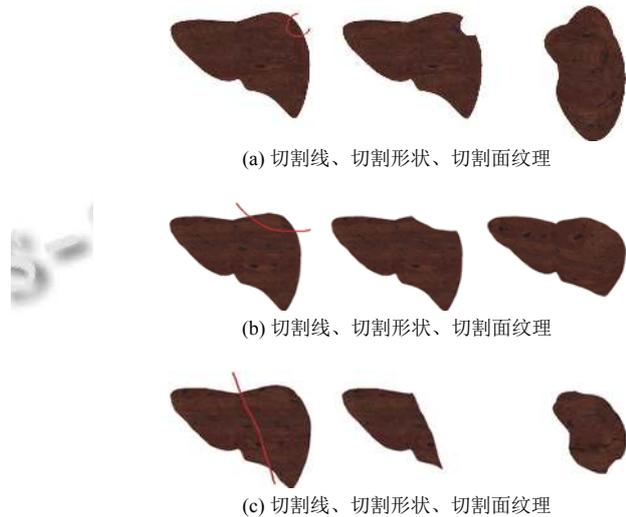


图 9 切割效果图

由图 9 可以看出, 利用鼠标模拟虚拟手术刀具在肝脏模型上画出的曲线, 可以将肝脏模型切割成任意形状, 且能够看到切割产生的新平面上的纹理. 表 1 所示为图 9 中 3 组不同切割形状的模型在 GPU 和 CPU 下的切割时间对比.

表 1 3 组不同切割形状的切割时间对比 (单位: s)

切割形状	CPU	GT 540M	GTX 650
(a)	26.42	1.294	0.773
(b)	34.17	1.359	0.841
(c)	42.72	1.465	0.978

6 结论

本文的主要工作包括肝脏模型的肝脏轮廓提取和三维重建. 在肝脏轮廓提取的工作中, 首先分析了各图像分割算法的利弊, 提出了一种人工干预的曲线拟合方法来标记肝脏区域的轮廓, 接着通过区域生长法提取出该轮廓区域. 然后采用面绘制的方法, 结合层间插值技术, 利用 VTK 工具包进行肝脏模型的三维重建工作, 建立了肝脏网格表面模型, 并引入 CUDA 技术, 提高运算效率, 为医学诊断和手术规划提供了辅助工具.

参考文献

- 1 Lee SJ, Woo SY, Huh KH, *et al.* Virtual skeletal complex model- and landmark-guided orthognathic surgery system. *Journal of Cranio-Maxillofacial Surgery*, 2016, 44(5): 557–568. [doi: [10.1016/j.jcms.2016.02.009](https://doi.org/10.1016/j.jcms.2016.02.009)]
- 2 Maini L, Verma T, Sharma A, *et al.* Evaluation of accuracy of virtual surgical planning for patient-specific pre-contoured plate in acetabular fracture fixation. *Archives of Orthopaedic and Trauma Surgery*, 2018, 138(4): 495–504. [doi: [10.1007/s00402-018-2868-2](https://doi.org/10.1007/s00402-018-2868-2)]
- 3 Marinković D, Zehn M. Corotational finite element formulation for virtual-reality based surgery simulators. *Physical Mesomechanics*, 2018, 21(1): 15–23. [doi: [10.1134/S1029959918010034](https://doi.org/10.1134/S1029959918010034)]
- 4 Rahman F, Celebi AA, Louis PJ, *et al.* A comprehensive treatment approach for idiopathic condylar resorption and anterior open bite with 3D virtual surgical planning and self-ligated customized lingual appliance. *American Journal of Orthodontics and Dentofacial Orthopedics*, 2019, 155(4): 560–571. [doi: [10.1016/j.ajodo.2017.08.032](https://doi.org/10.1016/j.ajodo.2017.08.032)]
- 5 汪军, 刘冬. 一种胆囊切除虚拟手术仿真训练平台研究. *系统仿真学报*, 2016, 28(8): 1853–1862.
- 6 肖亮, 欧阳锡武, 毛林峰, 等. 计算机三维手术计划系统在肝脏外科教学中的应用研究. *中国医学工程*, 2017, 25(1): 1–4.
- 7 莫建清, 何汉武, 李晋芳. 膝关节镜手术规划及训练系统. *计算机应用与软件*, 2018, 35(6): 214–219, 278. [doi: [10.3969/j.issn.1000-386x.2018.06.039](https://doi.org/10.3969/j.issn.1000-386x.2018.06.039)]
- 8 高洪林, 张超, 程云为. 踝关节骨折数字化虚拟手术设计系统的建立及其临床应用. *江苏医药*, 2018, 44(10): 1156–1159.
- 9 黄铭明, 苏其瑜, 曾维, 等. 下颌骨骨折虚拟手术培训系统研究. *计算机工程与应用*, 2018, 54(23): 223–229. [doi: [10.3778/j.issn.1002-8331.1708-0130](https://doi.org/10.3778/j.issn.1002-8331.1708-0130)]
- 10 Dong Y, Lefebvre S, Tong X, *et al.* Lazy solid texture synthesis. *Computer Graphics Forum*, 2008, 27(4): 1165–1174. [doi: [10.1111/j.1467-8659.2008.01254.x](https://doi.org/10.1111/j.1467-8659.2008.01254.x)]
- 11 潘翔. 基于复用计算的肝脏软组织体纹理合成方法研究 [硕士学位论文]. 福州: 福州大学, 2014.
- 12 Eslahchi MR, Kavooosi M. The use of Jacobi wavelets for constrained approximation of rational Bézier curves. *Computational and Applied Mathematics*, 2018, 37(3): 3951–3966. [doi: [10.1007/s40314-017-0552-8](https://doi.org/10.1007/s40314-017-0552-8)]
- 13 彭英, 陈志新. 浅海水深遥感影像的水陆分割研究. *科技信息*, 2009, (29): 46–47. [doi: [10.3969/j.issn.1001-9960.2009.29.030](https://doi.org/10.3969/j.issn.1001-9960.2009.29.030)]