 

毕业设计(论文)

开题报告

**院 系 计算机科学与工程学院**

**专 业 计算机科学与技术**

**姓 名 贺建安**

**一卡通号 213150728**

**学 号 09015322**

**指导教师 唐慧**

**开题报告日期 2019年3月18日**

|  |  |
| --- | --- |
| 论文题目 | **基于VTK的三维可视化平台开发** |
| 1. 选题背景和意义：   在临床医疗诊断中，计算机断层扫描(CT)和磁共振成像(MRI)是很常用的诊疗手段。通过CT或者MRI，医生可以得到病人身体内部的图像，从而更有效地进行诊断。但是，CT或MRI拍摄得到的原始图像是二维切片图像序列，通过浏览这些二维图像序列，医生很难对病人的病灶部位的位置、大小及形状进行精确的判断。  通过三维可视化方法，我们可以将二维图像序列重建为三维结构，立体地展现出病人组织结构的空间细节，并通过一些交互手段，实现立体结构的旋转、放缩等。通过设定合适的颜色域透明度，可以对不同的组织结构加以区分，或是只对某一感兴趣的组织结构进行绘制；这些三维可视化手段，可以帮助医生更加直观地观察病人的身体内部结构，从而帮助医生做出更加精确有效的诊断。  目前已有的一些三维可视化工具，在开源性，响应时间，绘制效果，交互复杂度等方面，或多或少都存在一些问题。本课题预期开发出一个可以进行实时可视化，绘制效果逼真，且交互方式友好的开源三维可视化平台，用于临床医疗诊断，帮助医生精准医疗。其可视化方法，也可用于地质勘探、气象分析、分子结构等科学可视化邻域，构造三维模型，强化视觉效果。 | |
| 1. 课题关键问题及难点： 2. **体绘制算法：**本课题所需设计的系统的输入是二维DICOM图像序列，通过读取这些图像序列，我们可以获取待绘制结构的体数据。体绘制算法是将这些体数据在屏幕上以二维形式展现，算法需要计算体数据中的每个元素(体素)对显示图像的影响，从而以合理的颜色与透明度展现在屏幕上，凸显出立体感与真实感。如何设计实现一个鲁棒且高效的算法，对绘制效果与速率至关重要，这也是本课题的核心； 3. **传递函数(transfer function)：**传递函数在体绘制中的角色是将体素信息映射成可绘制的属性，例如透明度与颜色。只有用一个好的传递函数，体绘制时才能清晰地将感兴趣的特征进行可视化。然而，设计一个好的传递函数，通常做法是需要用户不断试验与试错，是一个反复调参的枯燥过程。如何用更加友好智能的方法让用户快速设计出一个符合预期绘制效果的传递函数，是本课题需要攻克的一个难点； 4. **交互效果与速率：**对于体绘制形成的模型，我们还需要提供旋转，放缩等交互手段。为了能够在绘制时展现模型的内部结构，我们通常使用的是直接体绘制算法，即直接从原始数据到观测数据，不存在数据的预处理或预存。因此，当交互发生时，需要重新进行计算，而体数据通常十分庞大，交互时计算量较大，耗时较久。这对本课题所要求的实时性交互而言，是一个挑战。 | |

注：开题报告可单独装订，但在院（系）范围内，封面和装订格式必须统一。

|  |
| --- |
| 三、文献综述（或调研报告）：  将二维切片图像序列进行三维重建与可视化的算法可分为两大类，一类是基于中间几何图元的面绘制方法，另一类是直接基于三维体数据的体绘制方法。其中，面绘制中最经典的算法是Matching Cube(MC)算法，MC算法大致思想是计算每个体素的梯度值，与预先设定的表面阈值比较，进行等值面提取[1]。即是以表面的方式绘制三维模型。面绘制方法优点是计算量小，运行速度快[2]。但是，在医学图像三维重建中，我们的需求是不仅仅是展示人体组织与器官的表面轮廓，更需要可视化其内部结构，展示出空间立体细节，在一幅图像中显示多种物质的综合分布情况。这便要求使用体绘制的方法。  体绘制方法是直接将体数据按其颜色与透明度投射到绘制窗口，将所有立体细节同时展现到二维窗口。目前主流的体绘制算法有：光线投射算法(Ray-casting)，抛雪球算法(Splatting)和错切-形变算法(Shear-warp)[3]。  光线投射算法是基于物理光线模型，把每个体素看作是可以发射、反射和吸收光线的粒子，依据体素的介质特征得到它们的的颜色与透明度，并沿着视线观察方向积分，最后在绘制窗口形成具有半透明效果的图像[4]。可以形象地理解为一条从体素出发沿着视线方向发出的光线，穿越整个图像序列，过程中被其他体素反射或吸收，直至光线穿越整个图像序列，最后得到的颜色与透明度就是绘制图像的颜色与透明度[5]。  抛雪球算法是反复对体素进行计算[6]。算法使用一个称为足迹的函数计算每一个体素投影的影响范围，用高斯函数定义强度分布(中心强度大，四周强度小)，从而计算其对图像的总体贡献，并加以合成，形成最后的图像[7]。该算法可渐进地进行显示，由于模仿了雪球被抛到墙壁上所留下的一个扩散状痕迹地现象，故取名“抛雪球法”[6]。  错切-形变算法是将三维视觉变化分解为三维错切变化与二维形变变化。体数据按照错切变化矩阵进行错切形变，投影到错切空间形成一个中间图像，然后中间图像经过形变处理得到结果图像[8]。  这三种算法中，光线投影算法绘制出来地图像品质最高，因为算法遵从的是实际物理光线模型。绘制速度最快的是错切-形变算法，因为算法内的形变处理可并行计算。另外的体绘制算法还有投影成像算法、频域变化算法、基于纹理映射算法等[3]。目前，随着硬件计算的发展，各种体绘制算法也逐渐被移植到GPU上实现，计算速度可大大提升。  传递函数是直接体绘制的基础，它在体绘制中的作用是将体素信息映射为可绘制的属性，例如透明度与颜色[9]。良好的传递函数可以揭示数据中的重要结构，在体绘制时可以清晰地将感兴趣的特征进行可视化。然而，设定一个好的传递函数往往是一个挑战，原因有以下两点：第一，设置传递函数的接口不受原数据集的约束或指导，缺乏指导迫使用户进入一个不断实验的交互模式，在这种模式下，只有通过观察由于增量调整而导致的立体绘制中的变化，才能探索传递函数域[9]。反复试错的过程中，可能会让用户迷失其中，或者陷入局部最优；第二，传递函数本质上是非空间的，它们的颜色和透明度的赋值与空间位置并不存在某种映射关系[9]。因此，用户如果想把立体中的某个区域与其他区域分隔开，他并不能按照这样的空间信息来设置传递函数[10]。  想要传递函数域更易于用户去探索，可以考虑一下两种途径：一是设计更细粒度的传递函数，使得传递函数每一个维度的变化对最终的绘制图的影响单一。这种方式即可以提供更加精细的绘制效果，也能使用户在探索过程中更有条理。多维传递函数是常用的一种手法[9]；二是提供更加直观与便捷的直接操作控件，让用户可以直接对着绘制图进行交互式设置[10]。通过直接操作控件，用户直观地在绘制图上进行操作，操作的结果被间接转化为传递函数的某个变量值。  **参考文献：**   1. W.E. Lorenson and H.E. Cline, Marching cubes: a high resolution 3D surface construction algorithm. In: M.C. Stone, Editor, SIGGRAPH '87 Proceedings, ACM, New York (1987), pp. 163–169. 2. Levoy M. Display of Surfaces from Volume Data[J]. IEEE Computer Graphics & Application, 1988, 8(3):29-37. 3. Huang, Jian, Mueller, Klaus, Crawfis, Roger, Bartz, Dirk, Meissner, Michael. A Practical Evaluation of Popular Volume Rendering Algorithms[P]. Volume Visualization, 2000. VV 2000. IEEE Symposium on,2000. 4. Terry S. Yoo, Ulrich Neumann, Henry Fuchs , Stephen M. Pizer, Tim Cullip, John Rhoades, Ross Whitaker. Direct Visualization of Volume Data. IEEE Computer Graphics and Applications, v.12 n.4, p.63-71, July 1992 5. Levoy M. Hybrid ray tracer for rendering polygon and volume data[J]. IEEE Computer Graphics & Applications, 1990, 10(2):33-40. 6. Westenberg M A, Roerdink J B T M. X-Ray Volume Rendering by Hierarchical Wavelet Splatting[J]. Machine Graphics & Vision, 2000, 3:159 – 162. 7. Lee Westover. Footprint evaluation for volume rendering[J]. ACM SIGGRAPH Computer Graphics, v.24 n.4, p.367-376, Aug. 1990. 8. Lacroute P, Levoy M. Fast volume rendering using a shear-warp factorization of the viewing transformation[C] SIGGRAPH '94 Proceedings of the 21st annual conference on Computer graphics and interactive technique. 1995:451-458. 9. Kniss, J., Kindlmann, G., Hansen, C.. Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets[P]. Visualization, 2001. VIS '01. Proceedings,2001. 10. Li, Lu, Peng, Hu, Chen, Xun. Visualization of boundaries in volumetric data sets through a what material you pick is what boundary you see approach[J]. COMPUTER METHODS AND PROGRAMS IN BIOMEDICINE, 2016, 126: 76-88 |
| 四、方案（设计方案、或研究方案、研制方案）论证：  本项目开发的三维可视化平台，其绘制过程基于VTK实现，用户界面基于Qt5实现。平台的输入是一组DICOM图像序列，来自同一组织结构的同一方向的切片。输出的是重建后的三维立体图，展现该组织结构的立体细节。在该三维立体图的绘制窗口内，用户可通过鼠标或键盘对立体图进行旋转，放缩。此外，用户还可以通过一些选择或滑块控件，对绘制效果(如颜色或透明度)进行控制，或只绘制感兴趣的区域。  体绘制流程如下：    按照以上体绘制流程，本项目的技术路线如下：    用户在基于Qt5的界面上的操作通过Qt的信号槽机制调用相应的响应函数。上图所示的纵向流程为VTK的绘制管道，从Reader开始，读取用户打开的文件夹中的DICOM图像序列，构建出三维体数据场；最终在QVTKWidget上进行显示，我们可以为该控件的Interactor自定义交互方式；绘制过程中，需要通过传递函数为立体传递颜色、透明度、梯度等属性，这些属性的设定，我们可以通过Qt控件暴露给用户。  关于实时性交互，一方面，可以使用基于GPU的绘制函数，并行计算加快速度；另一方面，VTK绘制管道中使用了lazy evaluation机制，即只有当数据需要被计算时，绘制流程才会开始。因此，我们可以在用户进行交互时(如对绘制图进行旋转、放缩，或是修改传递函数)，才调用render()函数进行绘制，这样可以减少不必要的计算量。  关于传递函数的设置，我们的目标是以更加智能便捷的控件调控方式提供给用户。首先，我们可以从多个维度来进行调控，将三维体数据场的标量值、梯度幅值、高阶导数等作为传递函数的定义域，来调控颜色(RGB)与透明度(α)，同时对光照模型(ambient, diffuse, specular)也加以调控。此外，通过对三维体数据场的自身特征进行统计分析，我们也可以为用户预设传递函数。例如，根据灰度标量值与连通性来区分不同的物质，设置不同的颜色传递函数；通过梯度幅值或其二阶导数来区分材料边界，设置较高的不透明度。用户可以在这些预设传递函数的基础上加以微调，从而减少工作量。  其次，在控件的设计上，一方面，对于全局性属性的设定，我们可以让用户直接在这些属性图上面进行设置。例如，对于颜色的设置，我们可以让用户直接在颜色直方图上进行拖拽设置，设置的过程就是改变直方图，同时使绘制图也发生相应的改变；另一方面，我们可以使用直接操纵控件，例如对于感兴趣区域的选择，可以让用户直接在绘制图上拖拽三个维度的剪切平面，平面围住的区域就认为是感兴趣区域。也可以设计一些探针控件，使用户可以直接在绘制图或者剪切平面图上进行点击，并显示点击区域的属性值，辅助用户设置传递函数。  最后，我们可以完善一些细节方面。例如，当用户设置传递函数时，我们可以为用户保存设置记录，方便用户回退或者下一次的绘制；以及其他一些辅助功能，比如快速调整到指定视角、背景颜色设置、键盘交互、标尺、3D cube等。通过完善这些细节，我们可以进一步提高设置传递函数时的用户体验。 |
| 五、进度安排：   * 2019年3月17日前，完成翻译论文与开题报告的编辑、审核与提交； * 2019年4月14日前，完成软件初步数据读取，体绘制及交互功能；提交中期检查表； * 2019年4月30日前，进一步完善软件各功能模块，完全论文初稿； * 2019年5月10日前，完成软件的验收；完善论文内容与格式； * 2019年5月20日前，完成论文的提交； |