



西安邮电大学

毕业设计（论文）

题目：基于 WEB 的 CT 医学图像三维重建系统的设计与实现

学院：计算机学院

专业：网络工程

班级：网络 2101

学号：04212023

学生姓名：秦勤

导师姓名：王亚刚 职称：副教授

起止时间：2024 年 11 月 20 日 至 2025 年 6 月 6 日

毕业设计（论文）承诺书

本人所提交的毕业设计（论文）《基于WEB的CT医学图像三维重建系统的设计与实现》是本人在指导教师指导下独立研究、写作的成果，毕业设计（论文）中所引用他人的文献、数据、图件、资料均已明确标注；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注明并表示感谢。

本人深知本承诺书的法律责任，违规后果由本人承担。

论文作者签名：_____秦勤_____

日期：_____2025年1月5日_____

关于毕业设计（论文）使用授权的声明

本人在导师指导下所完成的论文及相关的职务作品，知识产权归属西安邮电大学。本人完全了解西安邮电大学有关保存、使用毕业设计（论文）的规定，同意学校保存或向国家有关部门或机构送交论文的纸质版和电子版，允许论文被查阅和借阅；本人授权西安邮电大学可以将本毕业设计（论文）的全部或部分内容编入有关数据库进行检索，可以采用任何复制手段保存和汇编本毕业设计（论文）。本人离校后发表、使用毕业设计（论文）或与该毕业设计（论文）直接相关的学术论文或成果时，第一署名单位仍然为西安邮电大学。

本毕业设计（论文）研究内容：

☒可以公开

☐不宜公开，已办理保密申请，解密后适用本授权书。

（请在以上选项内选择其中一项打“√”）

论文作者签名：_____秦勤_____

导师签名：_____王亚刚_____

日期：_____2025年1月3日_____

日期：_____2025年1月3日_____

西安邮电大学本科毕业设计(论文)选题审批表

申报人	王亚刚	职 称	副教授	学 院	计算机学院
题目名称	基于 WEB 的 CT 医学图像三维重建系统的设计与实现				
题目来源	<input checked="" type="checkbox"/> 教师科研课题 <input type="checkbox"/> 教师专业实践 <input type="checkbox"/> 其他				
题目类型	<input type="checkbox"/> 艺术作品 <input type="checkbox"/> 硬件设计 <input checked="" type="checkbox"/> 软件设计 <input type="checkbox"/> 论文				
题目分类	<input checked="" type="checkbox"/> 工程实践 <input type="checkbox"/> 社会调查 <input type="checkbox"/> 实习 <input type="checkbox"/> 实验 <input type="checkbox"/> 其他				
题目简述	<p>（为什么申报该课题）</p> <p>在与中国人民解放军总医院的科研项目中，已经掌握了基于 CT 医学图像的三维重建系统的一些关键技术，并已经实现了单机版。由于单机版在系统部署、系统升级以及远程访问等均有一定的缺陷，因此，基于 WEB 前后台技术，使用微服务等方式，将该系统核心移植到服务端上，从而便于从 Internet 上方便地进行访问，为推广该软件产品提供了便利。本课题就是要在已有核心系统的基础上，完成 web 开发环境的搭建、对原有核心内容进行设计和服务封装，实现前后台各种交互，部署 web 服务，实现完整的基于 web 的 CT 医学图像三维重建系统。</p>				
对学生知识与能力要求	<p>1.熟悉 web 开发，例如 Django 框架、VUE 框架等；</p> <p>2.熟悉数据库开发</p> <p>3.熟悉 Python 程序开发，了解 VTK、ITK 等库</p>				
具体任务以及预期目标	<p>（应完成的具体工作，预期目标和成果形式）</p> <p>本课题的具体任务：</p> <p>在已有核心系统的基础上，完成 web 开发环境的搭建、对原有核心内容进行服务分解和服务封装，提取系统交互接口并实现前后台各种交互，最后将整个 WEB 系统部署在 Linux 系统上，实现完整的基于 web 的 CT 医学图像三维重建系统。</p> <p>预期目标和成果形式：</p> <p>1.原有核心系统的分析文档，实现功能的服务分解、功能封装，服务之间的接口定义，前后台接口定义等，形成文档 1 份</p> <p>2.基于 Django、VUE 等框架实现前后台业务逻辑，提供完整的可运行的代码</p>				
时间进度	<p>2024 年 11 月 25 日-11 月 24 日：完成毕业设计选题</p> <p>2024 年 11 月 25 日-2025 年 1 月 10 日：提交开题报告，前期检查</p> <p>2025 年 1 月 11 日-3 月 29 日：完成环境搭建，并完成前后台接口设计，中期检查</p> <p>2025 年 3 月 30 日-5 月 17 日：完成设计实现代码，进行代码验收</p> <p>2025 年 4 月 1 日-5 月 25 日：撰写毕业论文</p> <p>2025 年 5 月 26 日 6 月 1 日：完善毕业论文，进行论文答辩。</p>				

专业负责人审核意见	经审核，题目符合专业培养目标和教学要求，同意该毕业设计项目申报		
	签字： 金小敏 2024 年 11 月 14 日		
系（教研室）主任 签字	2024 年 11 月 15 日	主管院长 签字	2024 年 11 月 15 日

西安邮电大学本科毕业设计（论文）开题报告

学生姓名	秦勤	学号	04212023	专业班级	网络 2101
指导教师	王亚刚	题目	基于 WEB 的 CT 医学图像三维重建系统的设计与实现		
<p>选题目的（为什么选该课题）</p> <p>基于 WEB 的 CT 医学图像三维重建系统选题目的在于提升医学诊断的精准性与效率。传统二维 CT 图像难以直观呈现器官及病变的空间结构，医生判断依赖经验。本系统借助 WEB 技术，可便捷地远程访问与操作。通过将 CT 原图像进行三维重建，能生成直观的三维模型，清晰展示人体组织、器官及病变部位的空间位置、形态与毗邻关系。这不仅有助于医生更准确地进行病情诊断、制定治疗方案，还能为医学教学与研究提供更直观的资料，推动医学领域数字化、可视化发展。</p>					
<p>前期基础（已学课程、掌握的工具，资料积累、软硬件条件等）</p> <p>已学课程涵盖医学图像处理、计算机图形学、WEB 开发技术等，为系统设计奠定理论基础。掌握 Python、JavaScript 等编程语言，以及 vue Three.js 前端工具，Django 等后台工具。能熟练进行图像算法实现与可视化。资料积累方面，收集了大量 CT 医学图像数据集及相关研究文献。软硬件上，拥有高性能计算机，配备专业显卡以满足三维重建计算需求；使用 Visual Studio Code 等开发环境，确保代码编写与调试高效进行，为系统开发提供有力支撑。</p>					
<p>要研究和解决的问题（做什么）</p> <p>本次系统项目中，需开展全面研究与工作。三维重建算法上，探索体绘制、面绘制算法并优化，以精准生成人体组织器官三维模型。工具使用上，前端借助 Vue 搭建用户交互界面，实现图像上传、参数设置等功能，提供友好操作体验；后端利用 Django 构建系统框架，处理图像数据传输、存储及业务逻辑。同时，引入 Three.js 实现三维模型的可视化展示与交互操作，让用户能多角度观察模型。完成开发后，开展系统测试，用相关 CT 图像数据验证性能，针对重建速度、精度等问题，通过优化算法与代码，结合 Vue、Django、Three.js 的特性不断改进，提升系统稳定性与实用性。</p>					
<p>工作思路和方案（怎么做）</p> <p>前期开展需求调研，明确系统功能与性能要求。图像处理阶段，用 Python 结合 OpenCV 对 CT 图像预处理，提升质量。借助 Three.js 实现 WEB 端可视化。后端以 Django 搭建，处理数据交互；前端用 Vue 开发界面，实现图像上传、交互操作。开发中定期测试，修复漏洞。最后进行系统评估，收集用户反馈，根据实际使用情况，对算法、界面等进行优化完善，确保系统满足临床需求。</p>					
<p>指导教师意见</p> <div>签字：2025 年 1 月 2 日</div>					

西安邮电大学毕业设计（论文）成绩评定表（理工）

学生姓名		性别		学号		专业 班级			
课题名称									
指导教师意见	支撑指标点/赋分	3-2/20	4-2/20	5-3/10	7-2/10	8-2/10	11-2/10	12-2/20	合计
	得分								
	(评价与指标点的对应)								
	指导教师(签字): 年 月 日								
评阅教师意见	支撑指标点/赋分								合计
	得分								
	(评价与指标点的对应)								
	评阅教师(签字): 年 月 日								
验收小组意见	支撑指标点/赋分								合计
	得分								
	(评价与指标点的对应)								
	验收小组组长(签字): 年 月 日								
答辩小组意见	支撑指标点/赋分								合计
	得分								
	(评价与指标点的对应)								
	答辩小组组长(签字): 年 月 日								
学生总评成绩	评分比例	指导教师(20%)	评阅教师(30%)	验收小组(20%)	答辩小组(30%)	合计			
	评分								
	毕业论文(设计)最终等级制成绩(优秀、良好、中等、及格、不及格)								
答辩委员会意见	学院答辩委员会主任(签字、学院盖章): 年 月 日								

支撑指标点/赋分,请按照最新版培养方案中对应的“毕业要求”进行指标点对应,“指标点”的数量可自行设定,标蓝处为

摘 要

随着医学图像处理技术向多模态融合与智能分析方向持续演进，其在临床诊断中的应用已从单一维度扩展至立体化解析，但传统医疗设备输出的二维序列图像仍受限于单视角（冠状位/矢状位/横断位）的成像特性，难以满足医生对复杂解剖结构的全景化评估需求。为此，基于 WEB 的 CT 医学图像三维重建系统的设计与实现，显著提高了医学图像分割的自动化程度与形态学分析精度，为构建科学化诊疗决策体系提供了重要技术支撑。

论文的主要内容包括

1. 完成了前后端技术栈整合与三维可视化框架搭建。在设计过程中，成功完成了前后端技术栈的深度整合工作，确保了数据的高效传输与交互的流畅性，为系统稳定运行奠定坚实基础。同时，搭建了基于 web 的三维可视化框架，凭借其强大的图形处理与渲染能力，能够将复杂数据以直观、生动的三维形式呈现。这两项关键任务的完成，极大地提升了项目的技术水准与可视化效果，为后续功能开发创造了有利条件。

2. 实现基于 Django 实现了三维重建后台处理模块开发。在设计进程中，我们充分发挥 Python 语言的优势，依托 Django 框架成功实现了三维重建后台处理模块的开发。Python 以其简洁易读的语法、丰富的库资源和高效的开发效率，为整个开发过程提供了极大便利。Django 框架奠定了稳固的底层支撑，而 Python 的灵活性使得我们能够更自由地设计算法与流程。通过 Python 编写的代码，该模块能够对输入的数据进行高效处理，精准完成三维重建任务。它的成功开发，为整个三维重建系统提供了核心的后台处理能力，有力推动了项目进展。

3. 完成了基于 VUE 及 ECharts 的 WEB 用户交互界面。VUE 凭借其高效的组件化开发模式与响应式数据绑定特性，让界面开发更加灵活高效；ECharts 则以丰富的图表类型和强大的可视化能力，为数据呈现提供了有力支持。Three.js 为 WebCT 医学图像三维重建系统提供强大渲染能力，实现高效可视化，提升交互体验与诊断精准度。二者结合，打造出了一个界面美观、操作便捷且能直观展示数据的 WEB 交互界面，极大提升了用户的使用体验。

关键词: Vue; Echart; 三维可视化; Three.js; 二维序列图像

ABSTRACT

As medical image processing technology advances towards multimodal fusion and intelligent analysis, its clinical diagnostic application has shifted from a one dimensional approach to a stereoscopic analysis. Traditional medical devices output two dimensional sequential images constrained by the single view imaging characteristics (coronal, sagittal, or axial plane), which fall short of doctors' requirements for a comprehensive assessment of complex anatomical structures. Consequently, the design and implementation of a WEB - based CT medical image three dimensional reconstruction system have significantly elevated the automation level of medical image segmentation and the precision of morphological analysis, offering crucial technical support for the establishment of a scientific diagnosis and treatment decision making system. The thesis mainly encompasses the following aspects: firstly, achieving the integration of front end and back end technology stacks and the construction of a three - dimensional visualization framework. During the design, I successfully accomplished the deep integration of front end and back end technologies, ensuring efficient data transmission and seamless interaction, thus establishing a robust foundation for the system's stable operation. Simultaneously, a WEB based three dimensional visualization framework was built, capable of presenting complex data in an intuitive and vivid three dimensional form with its powerful graphics processing and rendering capabilities, greatly enhancing the project's technical level and visualization effects and paving the way for subsequent functional development. Secondly, implementing the three dimensional reconstruction back end processing module based on the Django framework. Leveraging the advantages of Python, with its concise and readable syntax, abundant library resources, and high development efficiency, we developed the three dimensional reconstruction back end processing module within the Django framework. The Django framework provided a solid underlying foundation, while Python's flexibility allowed for more free algorithm and process design. Through Python written code, this module efficiently processes input data and accurately accomplishes three dimensional reconstruction tasks, serving as the core back end processing capability for the entire three dimensional reconstruction system. Thirdly, completing the WEB based user interaction interface based on VUE and ECharts. VUE, with its efficient component based development model and responsive data binding features, and ECharts, with its rich chart types and powerful visualization capabilities, were combined to create a WEB based user interaction interface that is visually appealing, easy to operate, and capable of visually displaying data, thereby greatly enhancing the user experience.

Key word: Vue; Echart; Three dimensional visualization; 2D sequential images

目 录

第 1 章 绪论.....	1
1.1 研究背景以及研究意义.....	1
1.2 相关研究.....	7
1.3 主要内容与章节简述.....	7
1.3.1 主要内容.....	7
1.3.2 章节简述.....	7
第 2 章 相关技术.....	8
2.1 Dicom 图像标准.....	8
2.2 ECharts 简介.....	8
2.3 Three.js 简介.....	9
2.4 Vue 简介.....	10
2.5 Django 简介.....	10
第 3 章 CT 医学图像三维重建的分析与设计.....	11
3.1 系统总体框架.....	11
3.2 后台功能模块.....	13
3.2.1 图像预处理.....	13
3.2.2 图像分割.....	14
3.3.3 三维重建表面.....	14
3.3 前台功能模块.....	15
3.3.1 用户交互页面.....	16
3.3.2 数据展示.....	17
3.3.3 三维模型简述.....	18
3.4 主要内容.....	18
第四章 测试与结果.....	20
4.1 实现环境.....	20
4.2 实现内容概述.....	20
4.3 具体实现内容.....	21
4.4 数据测试.....	23
第五章 总结与展望.....	26
结论.....	27

参考文献.....	28
致谢.....	30

第一章 绪论

医学图像的三维重建与可视化，无疑是一个兼具趣味性与挑战性的前沿领域。在该项目中，我们能够深入探索多种医学图像技术，并运用这些技术所获取的图像数据开展三维重建工作，最终将其以直观的形式可视化呈现，为医生与科研人员提供有力支持。这一过程涉及图像处理算法、计算机图形学技术以及医学影像学知识等多学科领域的融合应用。

当下，随着科技水平不断攀升以及人民生活方式日益优化，我国各行业均呈现出蓬勃发展的态势，医疗行业亦不例外。与此同时，民众对于医疗信息可视化的需求愈发强烈。就医之事无小事，轻则关乎病情判断，重则涉及生命安危，人们渴望更充分地掌握医疗信息，以此获得更多安全感。鉴于此，一项基于 WEB 的 CT 医学图像三维重建系统的设计与实现，不仅顺应了社会发展的潮流，更能让居民对医疗信息有更清晰的认识，增强其安全感。此外，这一项目的推进也将有效锻炼我们自身的专业能力。

近年来，医学影像技术发展迅猛，医学图像在临床诊断中的应用愈发广泛且关键，有力地推动了临床医学的发展进程。

通过对二维图像进行三维重建与可视化处理，能够将复杂的三维器官结构清晰直观地展现出来，这无疑有助于医生进行全面且精准的分析，进而提升医疗诊断水平。由此可见，医学图像的三维重建与可视化技术在医疗领域具备极为重要的实用价值。

1.1 研究背景以及研究意义

科学与技术进步推动微电子与计算机技术发展，催生了超声、CT、MRI 等医学成像技术，构建起全面的医学影像体系并应用于临床。借助这些技术，医生可获取二维断层序列图像进行医学分析，提升了诊断水平与手段。

但现有设备仅提供二维图像，医生需据此推断三维信息并凭经验诊断，此方法依赖主观判断，对医生要求高且效率低、准确性差。

因此，利用计算机与医学图像处理技术，准确提取感兴趣区域并构建三维可视化模型，对辅助医疗诊断意义重大。1973 年 Idesawa 出三维重建技术后，引发全球研究热潮，1992 年 Hartley 的完整重建理论推动其走向成熟。

1.2 相关内容研究

医学图像的三维重建与可视化，这是一个非常有趣且具有挑战性的设计！在这样的项目中，可以探索不同的医学图像技术，然后利用这些图像数据进行三维重建，并将其可视化以供医生或研究人员使用。可能会涉及到图像处理算法、计算机图形学技术、医学影像学知识等方面。

随着科技水平和人民生活方式的提高，我国的各行各业都在蓬勃发展，这其中也包括医疗行业。并且，当下人们对医疗信息可视化的要求愈发迫切。要知道，就医从来都不是能掉以轻心的事，病情诊断稍有偏差就可能贻误治疗时机，严重起来更是直接关乎患者的生命健康，所以人们都希望对医疗信息的掌握更加充分，从而更具有安全感，所以一项基于 WEB 的 CT 医学图像三维重建系统的设计与实现不仅顺应社会发展进程，更能让居民对医疗信息更了解，更具安全感。同时也能锻炼我们本身的专业能力。

对二维图像实施三维重建与可视化操作，能够把复杂的三维器官结构以直观的形式呈现出来。这一技术手段为医生开展全面且精准的分析提供了有力支撑，有助于提升医疗诊断的准确性与可靠性。由此可见，医学图像的三维重建与可视化技术具备显著的实用价值。

在传统模式下，三维重建技术往往高度依赖特定的软件平台与硬件设备。这一特性使得不同医院或科室之间在资源共享方面面临诸多阻碍，难以实现资源的有效流通与整合。而基于 WEB 的三维重建系统则巧妙地突破了这一局限，它能够在任何具备网络连接条件的设备上顺畅运行，无需额外配置软件或硬件。这不仅显著降低了医疗成本，还极大地提高了设备的利用率和使用的灵活性，让医疗资源能够得到更高效的利用。

基于 WEB 的 CT 医学图像三维重建系统还具备个性化设置的功能。医生可以依据患者的具体病情和实际需求，灵活调整三维重建的参数与视角，从而获取更为清晰、精准的图像信息，为精准治疗提供坚实依据。此外，该系统能够与其他医疗信息系统实现无缝对接，整合各类医疗数据，为患者提供涵盖诊断、治疗、康复等全方位的医疗服务。

不仅如此，基于 WEB 的 CT 医学图像三维重建系统在医学研究和教育领域同样发挥着重要作用。医生、学者和研究人员可借助网络平台，便捷地共享和交流医学图像数据，促进跨地域、跨机构的合作，共同推动医学研究的深入发展。同时，该系统还可作为医学教育的得力工具，以直观的方式帮助学生深入理解人体解剖结构和疾病特征，提升医学教育的质量和效果。

第 2 章 相关技术

2.1 Dicom 图像标准

在医学影像技术不断演进的当下，不同医疗设备生产厂商采用的医学图像传输方式与存储格式差异显著，阻碍了图像及相关信息在各系统和应用间的交换，而历经十余年发展与优化的 DICOM 标准凭借诸多优势赢得医疗界及大多数设备生产商广泛认可，如今几乎所有类型医疗设备如 CT、MRI、超声成像设备等都已支持该标准，在推动医疗信息网络化、系统化、数字化进程中作用关键。DICOM 标准全面覆盖数字医学成像各环节，是医学影像领域唯一标准，规范并完善了医学图像的格式、信息内容、存储与传输方式、归档策略及交互流程等。DICOM 标准下的医学图像由图像数据和头文件信息构成，头文件记录与病人、检查序列及过程相关的元数据，对医学诊断与研究至关重要；图像数据由一系列数据元素组成，每个完整数据元素包含 UID、Tag 标签、VR、VL 和 VF 等部分，其中 UID 采用“<根>·<后缀>”结构确保唯一性，“根”由国家相关部门分配，“后缀”在机构内部唯一；Tag 的组号与元素号用两个十六进制数表示，组号奇偶性有特定含义，系统预定义标签组号为偶数，奇数组号供用户自定义且通常无需自行定义组号；VR 用于规范具有属性值的元素的取值范围与编码方式；VL 以字节数衡量数据元素值域存储空间大小，规定数值必须为偶数字节，不足时用空白字节补齐；VF 存储数据元素实际值，其存储长度严格遵循偶数字节要求。

2.2 Echarts

ECharts 是一款由百度开源的 JavaScript 数据可视化库，基于 Canvas 渲染技术构建，专为高效展示复杂数据而设计。ECharts 提供超过 50 种图表类型，涵盖基础图表和高级图表，可满足数据分析、报表展示、业务监控等多场景需求。支持实时数据更新、缩放拖拽、悬浮提示、多图表联动等交互功能，用户可通过动态操作深入探索数据细节，提升数据洞察力。通过配置项可自定义样式、动画效果及交互逻辑，适配不同业务场景的个性化需求，同时支持插件扩展以增强功能。兼容主流浏览器及移动端设备，支持响应式布局，可轻松集成至 Web 应用、小程序等项目中，确保多终端一致体验。ECharts 以轻量级、高性能著称，广泛应用于数据分析、物联网监控、智慧城市等领域，助力用户快速实现数据可视化目标。

ECharts 在基于 WEB 的 CT 医学图像三维重建系统设计与实现中发挥着重要作用。其丰富的图表类型与交互功能，可辅助系统实现直观的数据可视化。例如，在展示 CT 图像的三维重建结果时，ECharts 能通过 3D 图表功能，以三维空间形式呈现数据，利用图

形、颜色等视觉元素清晰表达病灶位置、形态及与周围组织关系。同时，其交互功能支持用户缩放、旋转三维模型，从不同角度观察细节。此外，ECharts 基于 WebGL 渲染，能在网页上实现高性能 3D 图形渲染，确保系统流畅运行。借助 ECharts，设计者可快速构建可视化界面，将复杂的 CT 医学图像三维重建数据转化为直观、易懂的图表，提升系统的可用性与诊断效率。

2.3 Three.js

Three.js 是一种用于实现可视化的工具，但严格来说，它是一个基于 WebGL 的开源 JavaScript 3D 图形库。Three.js 的核心功能是为开发者提供创建和处理 3D 图形的工具和 API，它封装了 WebGL 的底层细节，使得开发者无需深入了解复杂的图形学知识，就能在网页浏览器中快速创建 3D 场景和动画。它提供了丰富的 3D 对象、材质、光照和相机等组件，支持创建复杂的 3D 模型、动画效果以及交互功能。在可视化领域，Three.js 发挥着重要作用。例如，在可视化大屏上，它能够展示各种复杂的三维模型，如建筑模型、机械模型等，并支持对这些模型进行旋转、缩放、平移等操作，方便用户从不同角度观察细节。它还可以将数据转化为生动的三维动画效果，使数据更加直观易懂，例如用三维柱状图展示销售数据的变化趋势。此外，Three.js 的粒子系统可以创建烟花、烟雾等炫酷效果，增强可视化大屏的视觉冲击力。

Three.js 对基于 WEB 的 CT 医学图像三维重建系统的设计与实现帮助显著。它封装了 WebGL 底层操作，让开发者无需深入掌握复杂图形学知识，就能在网页浏览器中轻松创建和展示三维图形。在 CT 医学图像处理中，它可将二维切片数据转化为直观的三维模型，帮助医生更清晰观察人体内部结构。Three.js 支持添加光源、材质等，能增强三维模型的视觉效果，提升诊断准确性。同时，它还提供交互功能，允许医生通过鼠标等设备旋转、缩放模型，从不同角度观察细节。此外，Three.js 可结合行进立方体算法等实现体积渲染，处理整个三维体数据集，查看数据集内部结构，为医学诊断和治疗提供更强大的工具，也为医学研究和教育带来便利。

Three.js 的关键组件如图 2.1 所示，主要包括：

- （1）场景：3D 世界的容器，存放所有物体（模型、光源、相机）。
- （2）摄像机：定义观察视角（如 PerspectiveCamera 模拟人眼透视）。
- （3）渲染器：将场景渲染到 <canvas>。
- （4）光源：影响物体明暗。
- （5）动画系统：通过 requestAnimationFrame 实现物体变换（移动、旋转）或骨骼动画。

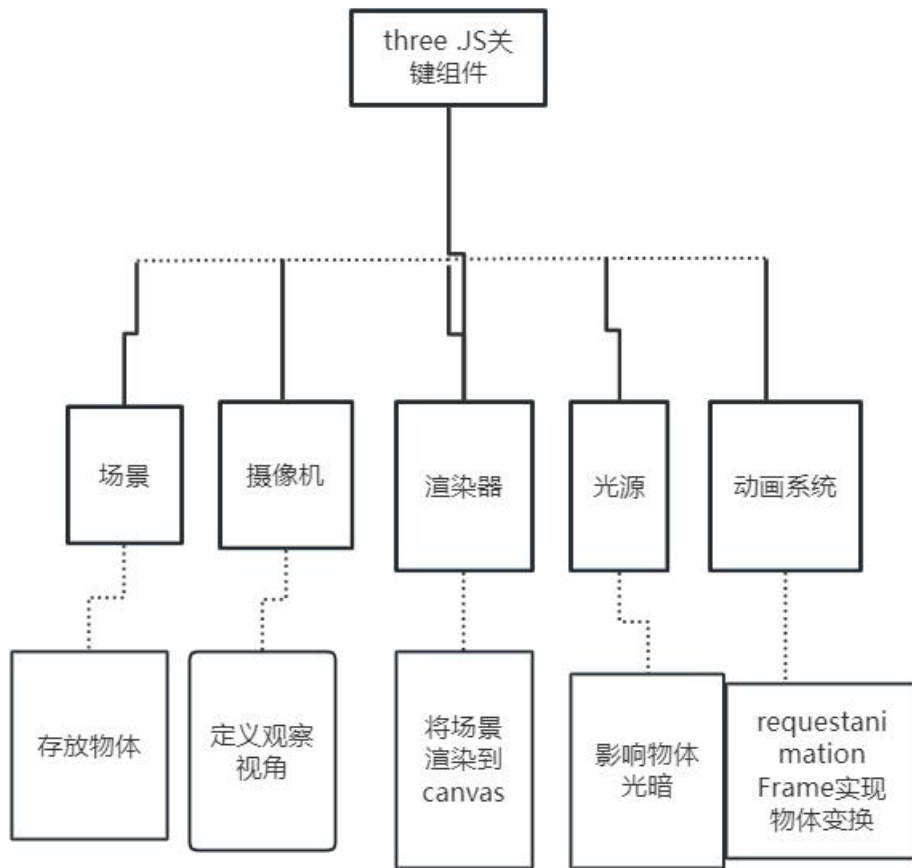


图 2.1 Three.js 的关键组件

2.4 VUE

Vue 是一个轻量级、渐进式 JavaScript 框架，专为构建用户界面设计。Vue 通过数据劫持实现数据与视图的自动同步。这种模块化设计提升了代码复用性，便于团队协作和长期维护，尤其适合大型项目的开发。Vue 在内存中构建虚拟 DOM 树，通过对比新旧虚拟 DOM 的差异，最小化实际 DOM 操作。这一机制显著提升了渲染性能，确保复杂交互和大量数据更新时的流畅体验。Vue 的响应式数据绑定特性，能让数据与视图自动同步。在系统中，当 CT 图像数据或三维重建参数变化时，视图会实时更新，医生无需手动刷新页面即可看到最新图像，极大提升诊断效率。可将系统拆分为多个独立组件，如图像加载组件、三维渲染组件、交互控制组件等。各组件可独立开发、测试和维护，提高代码复用性，降低开发复杂度，方便团队协作。

Vue 拥有庞大的插件生态系统，可快速集成各种图像处理工具和库。例如集成 DICOM 图像解析插件，方便处理 CT 医学图像数据；还能集成三维渲染库，增强三维重建效果展示。

借助 Vue 的渐进式框架设计，可逐步增强应用功能。通过优化界面交互，如添加平滑的动画效果、友好的操作提示等，让医生更便捷地操作三维重建系统，提升使用体验。

2.5 Django

Django 作为一款强大的全栈 Python Web 开发框架，其核心功能亮点突出：它提供涵盖数据库交互到前端渲染的完整工具链，实现全流程高效开发；内置的 ORM 功能将数据库表映射为 Python 类，开发者可便捷操作数据，还支持多种数据库并简化复杂查询；其自动生成的 Admin 后台管理系统，无需额外编码就能快速获得功能完备的数据管理界面；URL 路由通过灵活规则将请求精准导向对应视图函数，视图函数则高效处理业务逻辑并返回响应；模板引擎支持模板继承、标签和过滤器，让动态页面渲染轻松且代码更易维护；表单处理模块能自动生成表单、完成数据验证并处理错误，还能与模型无缝关联；用户认证与权限管理系统功能完备，涵盖用户注册登录、密码重置及细粒度权限控制；国际化与本地化功能则让应用轻松适配多语言环境，通过专业文件管理翻译文本并正确处理日期数字格式，全方位助力开发者构建高效、安全且功能丰富的 Web 应用。

第 3 章 设计与实现

本章将详细阐述基于 WEB 的 CT 医学图像三维重建系统的设计与实现方案。该系统聚焦于为用户打造一个直观、交互性强的平台，医生等用户可借此更便捷地操作、查看与分析 CT 医学图像的三维重建结果，提升诊疗效率与准确性。

3.1 系统总体框架

图 3.1 为本系统的设计框架图，主要分为前台功能模块和后台功能模块。



图 3.1 基于 web 的 CT 医学图像三维重建系统框架

前台功能模块包含两个子模块，分别是用户交互界面和数据展示模块，旨在为用户提供便捷、直观的操作和查看体验。

(1) **用户交互界面**：此模块注重用户与系统之间的互动，涵盖登录/注册界面以及主控制面板。登录/注册界面是用户进入系统的入口，保障用户身份的验证和系统的安全性；主控制面板则为用户提供系统主要功能的操作入口，方便用户进行各种设置和控制。

(2) **数据展示模块**：专注于数据的可视化呈现，包含三维可视化窗口、二维切片浏览以及三维模型旋转/缩放功能。三维可视化窗口能够让用户以立体的方式查看数据，更直观

地理解数据的空间结构；二维切片浏览则允许用户从不同切面查看数据细节；三维模型旋转/缩放功能进一步增强了用户对数据的观察和分析能力，用户可以从多个角度、不同尺度对模型进行查看。

后台功能模块涉及系统对数据的处理过程，包含图像预处理、图像分割和表面重建算法三个关键步骤，确保系统能够准确、高效地处理图像数据。

(1) 图像预处理：作为数据处理的第一步，主要对输入的二维图片进行显示处理，为后续的分割和重建工作提供高质量的图像数据。

(2) 图像分割：该步骤采用阈值分割和边缘平滑优化技术。阈值分割算法基于图像中目标区域与背景区域在灰度值等特征上存在的差异，通过设定一个恰当的阈值，将图像像素划分为满足阈值条件的目标像素集合与不满足条件的背景像素集合，从而实现目标区域与背景区域的有效分离。边缘平滑优化则进一步处理分割后的图像边缘，使其更加平滑、准确，提高分割结果的质量。

(3) 表面重建算法：运用 Marching Cubes 实现和三角网格简化方法。Marching Cubes 算法是一种常用的三维表面重建算法，能够从体数据中提取出等值面，构建出物体的三维表面模型；三角网格简化则是在保证模型基本特征的前提下，减少三角网格的数量，降低模型的复杂度，提高后续处理和渲染的效率。

3.2 后台功能模块

3.2.1 图像预处理

医学图像预处理旨在改善图像质量，为后续诊断、分析等任务提供更可靠的基础。常见操作包括去噪，以消除图像中的随机噪声，提升清晰度；图像增强，通过调整对比度、亮度等突出关键结构与细节；像配准技术旨在针对在不同时间点、采用不同成像模态所获取的图像，通过特定的空间变换算法，使这些图像在空间坐标上达到精确对齐，从而为后续的综合分析提供可靠的图像数据基础。；还有归一化处理，统一图像的灰度范围等。经过预处理，能有效减少图像中的干扰因素，突出病变特征，帮助医生更准确地解读图像，提高疾病诊断的准确性与效率。

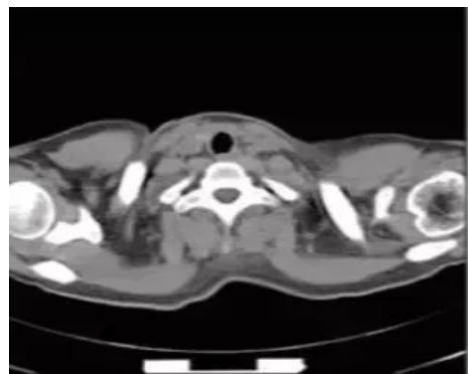
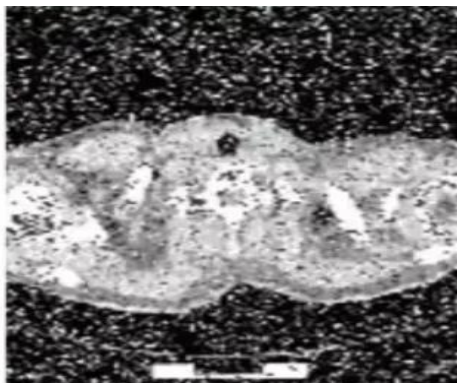


图 3.2 原图像

图 3.3 去噪波后的图像

傅里叶变换：傅里叶变换将信号从时域或空间域转换到频域，能揭示信号的频率成分。在三维重建里，它可对三维投影数据进行处理，将投影数据转换到频域，利用频域特性去除噪声，提取有效信息，再通过逆傅里叶变换重建出物体内部的三维结构图像。

图像增强：在三维重建中，图像增强能改善图像质量。它通过锐化边缘、增强对比度等操作，让图像细节更突出，使三维重建算法能更精准地提取特征，提升重建模型的清晰度与准确性。

直方图均衡化：在三维重建中，图像可能存在对比度低的问题，影响重建精度。直方图均衡化通过拉伸像素强度分布，使图像灰度级分布更均匀。比如一张灰暗的医学 CT 图像，原直方图集中在低灰度区，均衡化后，像素分布扩展到整个灰度范围，增强了图像对比度，让组织边界更清晰，有助于三维重建时更精准地提取特征，提升重建模型的质量。

图像归一化：图像归一化是把图像数据调整到统一范围，消除量纲影响。在三维重建里，不同来源图像可能尺度、亮度等有差异。归一化后，数据具有可比性。

3.2.2 图像分割

图像分割是图像处理领域的关键操作，它把图像精准划分成多个具有独特性质且彼此界限特定的区域。借助此操作，能高效提取出我们感兴趣的目标。这一过程是图像处理迈向图像分析的重要过渡，在医学影像领域意义重大。其核心在于，通过一系列技术手段，精准定位并分离出图像中我们关注的对象。从数学视角而言，图像分割实质上是将数字图像分割成互不重叠的区域，同时对每个区域内的像素赋予相同标识，以此完成区域划分与标记

图像分割的原理其实不难理解，就好比给图像里的像素点“分门别类”，通过设定不同的特征阈值来实现。常用的特征呢，一部分是直接从原始图像里“扒”下来的灰度或彩色特征，还有一部分是对原始灰度、彩色值“加工”后得到的特征。以原始图像 $f(x, y)$ 为例，依据一定准则从中确定一个特征值 T ，以此将图像分割为两个部分。当取 $b_0 = 0$ （代表黑色）、 $b_1 = 1$ （代表白色）时，就实现了我们常见的图像二值化操作。这种基于特征阈值的图像分割方法在诸多领域都有广泛应用，比如文档处理、医学影像分析等，能有效提取关键信息，简化后续处理流程。

3.2.3 三维表面重建

1. Marching cube 算法

Marching Cubes 堪称计算机图形学领域具有里程碑意义的创新成果。该算法专注于从三维离散数据场中提取等值面，在医学可视化领域，尤其是 CT 扫描等三维重建工作中，

扮演着至关重要的角色，发挥着难以替代的作用。从本质上讲，**Marching Cubes** 算法既属于体渲染方法，也是一种用于提取等值面的技术。体渲染，简而言之，就是依据体积数据进行图像渲染。像点云这类数据，其渲染过程本质上也是基于体积数据来完成的。

该算法的核心贡献在于，它能够将复杂的体数据转换为清晰直观的三维表面模型，为医生和研究人员理解复杂的解剖结构提供了极为直观的途径。

Marching Cubes 算法的基本原理是，将三维数据场分割成一系列微小立方体（即体素），随后对每个体素进行逐一处理，以确定等值面与该体素的相交情况。

从直观层面理解，**Marching Cubes** 算法的运行过程，就是利用大量微小立方体对空间进行离散化采样。通过在这些小立方体内生成三角面片，来近似重建等值面。立方体的尺寸越小（意味着采样密度越高），重建得到的表面就越精确，但与此同时，计算成本也会随之增加。

步骤如下：

首先将图像的四个连续切片数据读取至内存空间以形成局部三维数据块，随后依据体元扫描规则对其中两个相邻切片展开逐层遍历操作；针对每个体元（由两个切片中的对应像素对构成的三维立方体），提取其八个顶点的灰度值并与预设的等值面阈值进行对比，通过比较结果计算得到立方体的索引值；利用该索引值从预先构建的查找表中检索对应的边列表信息，明确等值面与体元棱边的交点分布；针对边列表中的每条棱边顶点，基于其灰度值与阈值的线性关系，运用线性插值方法精确计算出三角面片顶点的三维坐标位置；同时，分别计算每个体元顶点的单位法线向量，并通过插值算法将法线信息传递至对应三角形面片的各个顶点，以保留表面光照细节；最终，通过遍历数据场中的所有体元，依次连接各体元生成的三角面片，并合并所有立方体的等值面片段，从而构建出完整的三维表面模型，实现从离散体数据到连续几何表面的重建。

2. 泊松重建算法

泊松重建算法是一种基于隐式曲面的三维表面重建算法。泊松重建算法的核心思想是通过隐函数表面重建方法，将点云数据转换为指示函数（点云内部为 1，外部为 0），然后求解泊松方程得到曲面的隐式表示，最后提取等值面生成三维模型。它能够处理带有噪声的点云数据，生成光滑的三维表面。

使用八叉树结构存储点集，根据采样点集的位置定义八叉树，然后细分八叉树使每个采样点都落在深度为 D 的叶节点。对八叉树的每个节点设置空间函数 F ，所有节点函数 F 的线性和可以表示向量场 V ，基函数 F 采用了盒滤波的 n 维卷积。均匀采样的情况下，假设划分的块是常量，通过向量场 V 逼近指示函数的梯度，采用三次条样插值（三线插值）。

求解泊松方程：方程的解采用拉普拉斯矩阵迭代求出。为得到重构表面，需要选择阈值获得等值面；先估计采样点的位置，然后用其平均值进行等值面提取，然后用移动立方体算法得到等值面。

该算法优势一次性把所有的点都考虑在内，对噪声点有很好的弹性。利用泊松方程

在边界处没有误差的特点，得到的模型不存在假的表面框。泊松方程允许的层次结构支持局部的基函数，对稀疏线性系统的情况有很好的支持。

CT 医学图像数据可能存在噪声和不完整情况，泊松重建算法能通过隐函数处理方式，综合考量所有点云数据，有效降低噪声影响，生成更贴合真实解剖结构的三维表面模型，为医生提供精准诊断依据。该算法可生成光滑的三维表面，避免模型出现尖锐突起或凹陷。在 Web 系统中展示时，光滑表面更利于医生观察和分析，提升诊断效率和准确性。基于 Web 的系统面向不同用户，泊松重建算法的强大处理能力，可适应不同 CT 数据质量，保证系统在多种环境下稳定运行，提高系统实用性和用户满意度。

3.3 前台功能模块

前台功能致力于为用户打造便捷、高效且友好的操作体验。用户能够轻松上传 CT 医学图像，上传后即可预览图像，快速确认图像准确性。同时，提供基础参数调整选项，如对比度、亮度等，满足用户个性化查看需求。用户发起三维重建请求后，系统会实时展示重建进度，完成后以直观方式呈现三维重建结果，方便用户多角度观察、分析医学影像，助力医疗诊断与科研工作。

3.3.1 用户交互页面

该登录页面的用户交互界面注重便捷与友好。输入用户名和密码时，有清晰的占位符提示，避免输入错误。点击“登录”按钮，响应迅速，若信息有误会及时弹出提示框告知。同时，“忘记密码”链接易被发现，点击后按指引可重置密码。“注册新用户”入口也清晰，引导新用户完成注册。整体交互流畅自然，降低用户操作门槛，提升登录体验。

1. 登录页面

该登录页面整体设计遵循简洁专业、安全可靠原则，契合医疗行业特质。页面布局清晰，分为登录表单、辅助功能与安全提示四大模块。登录表单区提供用户名、密码输入框，设计简洁，方便用户操作。辅助功能区设有“忘记密码”“注册新用户”等链接，满足用户不同需求。安全提示区则展示隐私政策、版权声明及安全认证标识，让用户放心使用，保障登录过程安全可信。



图 3.4 登录页面

2. 主控制面板

主控制页面是用户进入系统后的核心操作界面，设计兼顾功能集成与操作便捷性。页面布局清晰，分为菜单导航区、功能操作区与状态显示区。菜单导航区提供、三维重建、三维图像查看等核心功能入口，分类明确，方便快速定位。功能操作区支持参数调整、重建任务启动等交互，采用直观的表单与按钮设计。状态显示区实时反馈任务进度、系统通知或错误提示，确保用户掌握操作动态。整体界面简洁高效，满足医疗人员对 CT 图像三维重建的全流程需求。



图 3.5 主控制面板

3. 三维可视化窗口

三维可视化窗口是系统核心交互模块，支持 CT 医学图像的三维展示与操作。窗口采用 WebGL 或 Three.js 技术渲染，呈现高精度立体影像，用户可直观观察解剖结构。提供多视角旋转、缩放、平移功能，通过鼠标拖拽或触控手势自由调整视角。支持透明度调节、伪彩渲染等参数设置，辅助医生识别病灶。内置测量工具（如距离、角度测量）与标注功能，便于临床分析。界面响应迅速，兼容主流浏览器，为远程医疗协作与教学提供高效可视化支持。

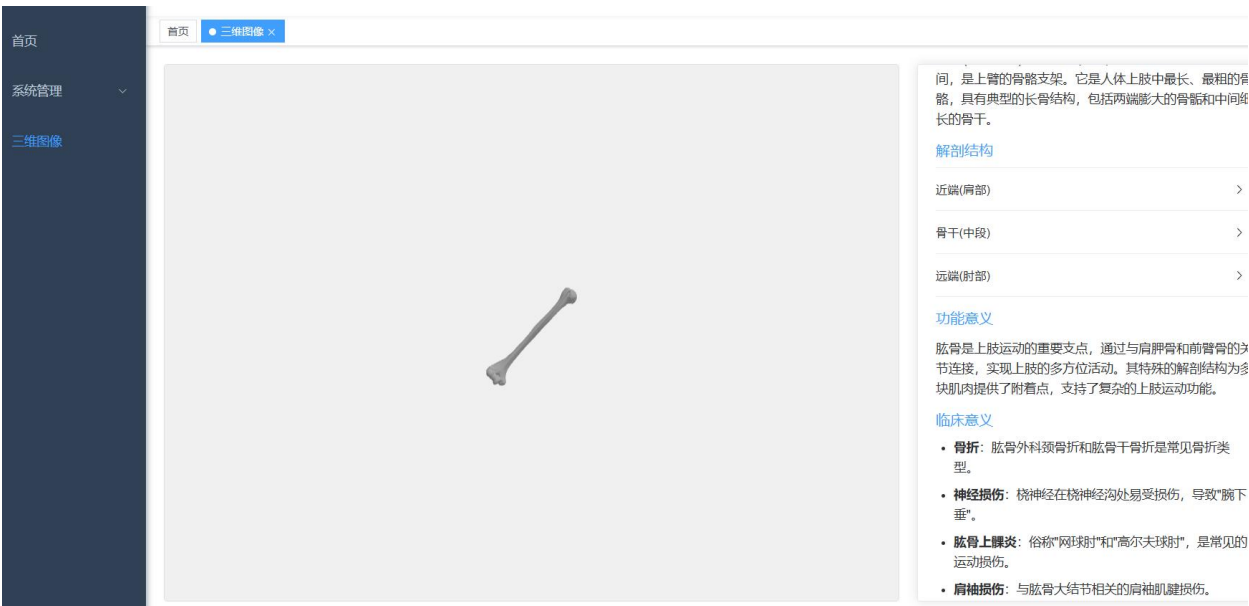


图 3.6 三维可视化窗口

3.3.2 三维展示核心代码模块

三维展示的核心实现逻辑可整合为以下关键环节：首先需搭建三维空间基础框架，通过创建场景容器承载所有可视化对象，并配置相机以确定观察视角，其中相机类型（如透视相机模拟人眼视觉或正交相机保持比例恒定）的选择直接影响空间呈现效果，需根据具体需求调整视野角度（FOV）、宽高比及近远裁剪面参数；随后需定义三维模型的几何形状与材质属性，几何体通过顶点、边和面构成基础结构，材质则控制模型外观表现，二者结合形成可渲染的网格对象；最终通过渲染器将场景与相机视角转换为二维图像输出，并配合动画循环持续更新模型状态，同时需处理窗口尺寸变化等交互事件以优化用户体验，复杂场景还需集成光照系统、模型加载器及用户交互控件以增强真实感与功能性。

```
<script>
// 导入 Three.js 但不让它成为 Vue 响应式系统的一部分
import * as THREE_ORIGINAL from 'three';
import { GLTFLoader as GLTFLoaderOriginal } from 'three/examples/jsm/loaders/GLTFLoader.js';
import { OrbitControls as OrbitControlsOriginal } from 'three/examples/jsm/controls/OrbitControls.js';

// 创建一个非响应式的对象来存储 Three.js 相关的内容
// 使用一个简单的技巧来避免 Vue 的响应式系统
const THREE = window.THREE = THREE_ORIGINAL;
const GLTFLoader = GLTFLoaderOriginal;
const OrbitControls = OrbitControlsOriginal;

// 使用非响应式对象存储 Three.js 状态
const threeState = {
  scene: null, // Three.js 场景对象
  camera: null, // Three.js 相机对象
  renderer: null, // Three.js 渲染器对象
  controls: null, // Three.js 控制器（如 OrbitControls）
  model: null, // 加载的 3D 模型
  isAutoRotating: false, // 是否自动旋转模型
  animationId: null // 动画循环的 ID（用于取消动画）
};
```

图 3.7 导入 Three.js 库及其相关组件

这段代码首先将 Three.js 核心库及 GLTFLoader、OrbitControls 等扩展组件独立导入后，显式挂载到 window.THREE 对象，既确保全局可访问性又避免 Vue 的响应式系统介入；随后创建 threeState 对象集中管理三维状态，包括场景、相机、渲染器等核心组件及动画控制参数，这种模式通过物理隔离将 Three.js 相关逻辑与 Vue 的响应式数据体系完全解耦，既防止了 Vue 对大型三维对象的性能损耗，又通过单例化设计简化了跨组件共享 Three.js 实例的复杂度，特别适合需要高性能渲染的三维可视化场景，开发者可在 Vue 生命周期中按需初始化或销毁 threeState 对象，既保持了 Three.js 对 Canvas 的直接操作能力，又维护了 Vue 组件的响应式特性边界。


```

...initThreeJS() {
...  const container = this.$refs.humerusContainer;
...  const width = container.clientWidth;
...  const height = container.clientHeight || 600;

...  // 创建场景
...  threeState.scene = new THREE.Scene();
...  threeState.scene.background = new THREE.Color(0xf0f0f0);

...  // 创建相机
...  threeState.camera = new THREE.PerspectiveCamera(75, width / height, 0.1, 1000);
...  threeState.camera.position.set(0, 0, 10);

...  // 创建渲染器
...  threeState.renderer = new THREE.WebGLRenderer({ antialias: true });
...  threeState.renderer.setSize(width, height);
...  threeState.renderer.setPixelRatio(window.devicePixelRatio);
...  threeState.renderer.shadowMap.enabled = true;
...  container.appendChild(threeState.renderer.domElement);

...  // 添加轨道控制器
...  threeState.controls = new OrbitControls(threeState.camera, threeState.renderer.domElement);
...  threeState.controls.enableDamping = true;
...  threeState.controls.dampingFactor = 0.05;

...  // 添加光源
...  const ambientLight = new THREE.AmbientLight(0xffffff, 0.5);
...  threeState.scene.add(ambientLight);

```

表 3.8 初始化 Three.js 场景

```

// 使用正确的路径，注意这里使用public目录下的路径
loader.load(
  '/models/human_humerus.glb',
  (gltf) => {
    threeState.model = gltf.scene;

    // 调整模型位置和大小
    threeState.model.position.set(0, 0, 0);
    threeState.model.scale.set(this.scaleValue, this.scaleValue, this.scaleValue);

    // 遍历模型中的所有网格，启用阴影
    threeState.model.traverse((node) => {
      if (node.isMesh) {
        node.castShadow = true;
        node.receiveShadow = true;
      }
    });

    threeState.scene.add(threeState.model);

    // 调整相机位置以适应模型
    this.fitCameraToModel();
  },
  (xhr) => {
    // 加载进度
    this.loadingProgress = Math.round((xhr.loaded / xhr.total) * 100);
  },
  (error) => {
    // 加载模型失败
  }
);

```

图 3.9 三维重建模块

这部分代码实现了一个基于 Three.js 的医学图像三维重建模块，主要包含初始化场景和加载模型两大核心功能。initThreeJS 方法负责三维环境的基础搭建：首先获取 DOM 容

器尺寸，创建 Three.js 场景并设置浅灰色背景；接着初始化透视相机（75 度视野），配置 WebGL 渲染器（开启抗锯齿和阴影映射），并将渲染器 DOM 元素挂载到容器中。通过 OrbitControls 添加带阻尼效果的轨道控制器，支持用户交互式旋转/缩放视图。场景照明系统包含环境光（0.5 强度）和方向光（0.8 强度），后者启用阴影投射以增强三维真实感。loadModel 方法通过 GLTFLoader 加载位于 public 目录的肱骨 GLB 模型，集成加载管理器实现进度监控（更新 loadingProgress 状态）。模型加载完成后，通过 traverse 遍历所有网格启用阴影投射/接收，调整模型位置和缩放比例（受 scaleValue 控制），最后调用 fitCameraToModel 自动调整相机视角以适配模型尺寸。错误处理机制捕获并显示加载异常，确保用户体验的健壮性。该设计实现了医学 CT 数据的三维可视化基础框架，为后续的交互操作和临床分析提供技术支撑。

```
fitCameraToModel() {  
  if (!threeState.model) return;  
  
  // 创建一个边界框  
  const boundingBox = new THREE.Box3().setFromObject(threeState.model);  
  const center = boundingBox.getCenter(new THREE.Vector3());  
  const size = boundingBox.getSize(new THREE.Vector3());  
  
  // 计算合适的相机位置  
  const maxDim = Math.max(size.x, size.y, size.z);  
  const fov = threeState.camera.fov * (Math.PI / 180);  
  let cameraZ = Math.abs(maxDim / Math.sin(fov / 2));  
  
  // 设置相机位置  
  threeState.camera.position.set(center.x, center.y, center.z + cameraZ * 1.5);  
  
  // 更新控制器  
  threeState.controls.target.set(center.x, center.y, center.z);  
  threeState.controls.update();  
}
```

图 3.9 调整相机视角

实现了自动调整相机视角以适配三维模型尺寸的功能。首先通过 Box3.setFromObject 计算模型的包围盒，获取其几何中心（center）和尺寸（size）。接着确定模型的最大维度（maxDim），基于相机视场角（fov）计算理论上的相机距离（cameraZ），公式 $\text{maxDim} / \sin(\text{fov}/2)$ 确保模型完整纳入视野。为留出适当观察空间，最终相机位置在中心点基础上沿 Z 轴偏移 1.5 倍计算距离。随后更新轨道控制器的目标点（target）为模型中心，并调用 controls.update 同步控制器状态。这种设计实现了模型加载后的自动视角适配，避免手动调整的繁琐操作，特别适用于医学图像三维重建场景中不同尺寸解剖结构的可视化需求，确保模型始终以最佳比例呈现于视口中央。

```

.app-container {
  padding: 20px;
}

#humerus-container {
  width: 100%;
  height: 600px;
  margin-bottom: 20px;
  border: 1px solid #dcdfe6;
  border-radius: 4px;
  overflow: hidden;
  position: relative;
}

.humerus-info {
  height: 600px;
  overflow-y: auto;
}

.info-content {
  h3 {
    margin-top: 15px;
    margin-bottom: 10px;
    color: #409EFF;
    font-size: 16px;
  }
  p {

```

图 3.10 定义 humerus-container 的容器

```

disease-content {
  display: flex;
  flex-wrap: wrap;
  gap: 15px;

  .disease-item {
    flex: 1 1 300px;

    h4 {
      color: #409EFF;
      margin-bottom: 8px;
    }

    p {
      font-size: 14px;
      line-height: 1.6;
      text-align: justify;
    }
  }
}

.el-button-group {
  margin-right: 10px;
}

.el-card {
  margin-bottom: 15px;

```

图 3.11 定义了多个 CSS 类，用于设置不同元素的样式

上图内容定义了一个医学三维重建系统的 Vue 组件样式，通过精细的 CSS 规则实现了可视化容器与信息面板的响应式布局。

`app-container` 设置全局内边距，`#humerus-container` 定义三维渲染区（全宽+固定高度 600px）添加 1px 边框和圆角 `overflow:hidden` 确保内容裁剪，`position:relative` 为子元素定位提供基准。`.humerus-info` 同样设置 600px 高度，`overflow-y:auto` 实现内容溢出时垂直滚动，内部文字采用 14px 基础字号、1.6 倍行距的易读排版。`.el-collapse-item__content` 调整折叠项内边距，优化文字段落间距。`.disease-content` 使用 flex 布局+gap 间距实现卡片式排列，`.disease-item` 设置弹性基准宽度 300px，标题采用主题色（#409EFF）强化视觉层级。`.el-button-group` 和 `.el-card` 分别调整按钮组间距和卡片下边距，保持 UI 元素的一致性。该样式方案特别针对医学影像系统的可视化交互场景，通过尺寸约束与响应式设计平衡了

三维渲染区的视觉冲击力和信息面板的可读性，蓝色系标题配色符合医疗类应用的行业视觉规范。

3.4 本章小结

基于 WEB 的 CT 医学图像三维重建系统的三维可视化具有重大意义。在临床层面，它打破二维影像的平面局限，以直观三维模型呈现解剖结构与病灶，助力医生精准定位、评估病情，提升诊断准确率与手术规划效率；于教学而言，交互式三维模型可替代传统标本，支持多角度观察与操作，降低教学成本，提升学习效果；从行业趋势看，其跨平台特性推动医疗资源远程共享与协作，促进分级诊疗落地，推动医疗数字化与智能化发展

第四章 测试与结果

在基于 WEB 的 CT 医学图像三维重建系统的设计与实现过程中，经过一系列算法优化、界面交互打磨及多轮测试验证，成功构建了一套功能完备、操作便捷的可视化平台。本部分将重点呈现系统核心功能的实现成果，涵盖三维重建模型的精度、交互操作的流畅性以及临床适用性验证结果。通过真实 CT 数据案例，直观展示系统在病灶定位、结构解析及多视角观察等方面的优势，为后续临床应用提供可靠依据。

4.1 实现环境

服务器端：Windows Server 等稳定的服务器操作系统。

客户端：Microsoft，无需特定操作系统限制。处理器为 Intel(R)Core(TM)i3，CPU M3902.67Hz，内存 2.0G，硬盘 500G。

开发环境：

编程语言：Python（结合 Django），用于后端逻辑的开发。

前端技术：VUE、CSS3、JavaScript 等，用于构建用户界面和实现交互功能。

数据库：如 MySQL，用于存储患者信息、CT 图像数据及重建结果等。

网络环境

高速网络连接：确保服务器与客户端之间的数据传输速度足够快，以支持实时或近实时的三维重建和图像显示。

网络安全：采用 HTTPS 协议进行数据传输加密，确保患者信息和图像数据的安全性。

开发工具与平台

集成开发环境：Visual Studio Code 等，提供代码编辑、调试和版本控制等功能。

4.2 实现内容

在系统架构上，采用客户端、服务器端和存储端三部分。客户端负责用户交互，服务器端处理图像，存储端负责存储。功能模块包括图像上传，将 CT 图像传至服务器；图像预处理，对图像去噪、增强；图像分析，进行特征提取和分类；结果展示，将处理结果呈现给用户。

技术实现上，前端基于 VUE、Echarts、JavaScript 等，使用 WebGL 进行三维渲染，实现响应式界面和交互。后端用 Node.js 处理数据和请求，数据库用 MySQL 等存储数据，Redis 等提高访问速度。还需解决数据量大、内存管理、渲染效率、交互响应等技术难点，确保系统稳定性和可扩展性，提供高质量可视化效果。

4.3 具体内容结果展示

本系统为用户提供了便捷且专业的医学影像处理功能：可将 CT 图像上传至系统，系统随即对图像进行自动化预处理，先通过去噪算法去除背景噪声与伪影，提升图像纯净度，再利用图像增强技术强化组织边界与细节，使图像更清晰鲜明；随后，系统采用面绘制技术，将二维医学图像数据转化为三维医学图像，精准呈现肱骨，包括其位置分布、生成一些关键信息；最终，系统以多视图窗口形式同时展示图像各个角度及三维图像，支持用户进行三维旋转 缩放 平移等交互操作。



图 4.3 旋转功能



图 4.4 缩放功能展示

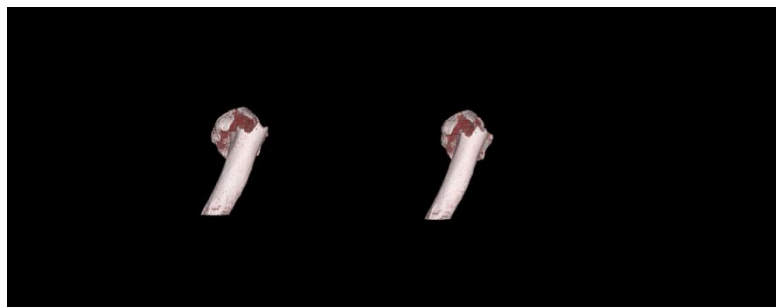


图 4.6 三维模型平移

三维重建系统中的肱骨三维模型交互式功能展示，为医学实践带来革新体验。通过鼠标拖拽、手势触控等操作，用户可自由旋转、缩放肱骨模型，从任意角度细致观察其结构。在医学教育场景，学生能自主探索肱骨的形态、毗邻关系，如清晰查看肱骨小头与桡骨小头的关节对应，加深对解剖知识的理解。临床诊断时，医生可交互式调整视角，精准评估肱骨病变范围，如对肱骨骨肿瘤边界进行多角度确认，提高诊断准确性。手术规划阶段，交互式功能助力医生模拟不同手术入路，观察肱骨及其周围血管、神经的相对位置，优化

手术方案，降低风险。这种交互式展示打破了传统观察的局限，让用户更直观、深入地理解肱骨的解剖与病变特征，为医学教学、诊断与治疗提供了强大支持。

4.4 测试数据

1 登录模块测试

在基于 WebCT 的医学图像三维重建系统研发进程中，登录页面作为用户与系统交互的起始点，其稳定性和易用性对用户体验及系统整体安全性有着关键影响。为确保该页面在各种环境下都能正常运行，为后续功能操作提供坚实基础。

表 1 登录页面测试

测试编号	测试项	输入	预期输出	实际输出	结论
1-1	正确输入信息	用户名 admin 密码 admin 123456	提示注册成功， 跳转登录页	提示注册成功， 跳转登录页	符合预期
1-2	忘记密码输入错误	密码输入错误	登录失败	登录失败	符合预期



图 4.2 登录页面

本 web 登录页面设计简洁直观。用户需在相应输入框准确填写用户名与密码，确保信息无误。点击“登录”按钮提交，若信息正确则进入系统；若错误会有提示。

2. 三维图像显示模块测试

三维图像显示模块是三维重建系统中的核心组件，负责将重建后的三维数据以直观方式呈现给用户。该模块支持点云的渲染交互。使用者可自行旋转、缩放和平移，从不同角度观察细节。

表 2 三维页面测试

测试编号	测试项	输入	预期输出	实际输出	结论
1-1	进入三维页面	查看	显示三维肱骨页面	显示三维肱骨页面	符合预期
1-2	肱骨缩放	滑动鼠标	肱骨放大缩小	肱骨放大缩小	符合预期
1-3	肱骨旋转	鼠标拖动	肱骨旋转	肱骨旋转成功	符合预期
1.4	肱骨平移	鼠标拖动	肱骨左右平移	平移成功	符合预期

第五章 优化与展望

在当前 WebCT 医学图像三维重建系统的设计与实现过程中，尽管已经取得了一定的成果，但仍存在诸多可优化的空间。首先，从性能角度出发，系统在处理大规模医学图像数据时，加载速度和渲染效率仍有待提升。通过采用更高效的图像压缩算法和数据传输协议，可以显著减少数据传输时间，提高系统的响应速度。同时，优化渲染管线，利用现代图形处理单元的并行计算能力，可以加速三维模型的渲染过程，提升用户体验。

系统在用户交互方面也有进一步优化的必要。当前的用户界面虽然已经具备了一定的易用性，但在复杂操作和高级功能的使用上，仍可能给用户带来一定的学习成本。因此，通过设计更加直观、简洁的用户界面，以及提供详细的操作指南和在线帮助，可以降低用户的使用门槛，提高系统的普及率。

随着技术的不断发展，基于 WebCT 的医学图像三维重建系统有望融入更多的创新元素。一方面，可以探索将深度学习技术应用于医学图像的三维重建过程中。通过训练深度学习模型，使其能够自动识别并提取医学图像中的关键特征，从而辅助医生进行更准确的诊断和治疗规划。另一方面，虚拟现实和增强现实技术的兴起，为医学图像的三维可视化提供了新的可能性。通过将三维重建结果与 VR/AR 技术相结合，医生可以在虚拟环境中进行更直观、更真实的手术模拟和训练，提高手术技能和安全性。

随着 5G 等高速网络技术的普及，远程医疗和协作诊断成为可能。基于 WebCT 的医学图像三维重建系统可以进一步拓展其应用场景，支持多用户在不同地点进行实时的三维图像查看、标注和讨论，为远程医疗协作提供有力的技术支持。

在医学图像的处理和传输过程中，安全性和隐私保护是至关重要的。未来，系统需要进一步加强数据加密和访问控制机制，确保医学图像数据在传输和存储过程中的安全性。同时，通过采用匿名化处理和差分隐私等技术手段，可以在保护患者隐私的前提下，实现医学图像数据的共享和利用，为医学研究和教学提供更多的数据支持。

展望未来，基于 WebCT 的医学图像三维重建系统将在医学领域发挥越来越重要的作用。随着技术的不断进步和应用场景的不断拓展，系统将变得更加智能、高效和易用。医生可以借助这一系统，更快速、更准确地获取患者的解剖信息，制定个性化的治疗方案。同时，患者也可以通过这一系统，更直观地了解自己的病情和治疗进展，增强治疗的信心和依从性。此外，随着远程医疗和协作诊断的普及，基于 WebCT 的医学图像三维重建系统将成为连接不同医疗机构、促进医学交流与合作的重要桥梁。

结 论

在医疗信息化与可视化技术快速发展的背景下，在医疗技术不断革新、临床需求日益精细化的今天，我致力于开发一套“医生友好、患者受益”的基于 Web 的 CT 医学图像三维重建系统。从需求调研到系统落地，始终以“让医学影像更直观、让诊断决策更高效”为初心，将技术温度融入每一个功能细节。本设计以提升 CT 影像诊断效率与精准度为目标，设计并实现了一套基于 Web 的 CT 医学图像三维重建系统。通过整合前端交互、后端计算与医学影像处理技术，系统实现了从 DICOM 数据导入、三维重建到可视化分析的全流程功能，为临床诊断与医学研究提供了高效工具。

系统采用前后端分离架构：前端基于 Vue.js 框架构建用户界面，实现三维模型的交互式展示与操作（如旋转、缩放、剖面切割）；后端以 Django 为核心，负责 DICOM 数据解析、三维重建算法调用及用户权限管理。

通过 pydicom 库解析影像元数据，完成图像归一化与降噪；集成 Marching Cubes 等经典算法，在保留专业重建算法（如 Marching Cubes）的基础上，开发智能预处理模块，自动优化图像噪声与伪影，让技术门槛不再是精准诊断的阻碍。结合 WebGL 加速渲染，实现高精度三维模型生成。

这套系统的诞生，不仅是代码与算法的结晶，更是我们对“医疗无界、生命有情”的深刻理解。未来，我将继续以医生的手、患者的眼、技术的脑，与大家共同书写医学影像的“温度革命”。

参考文献

- [1]关天民.基于 CT 数据的医学图像处理系统设计[J].中国医学物理学杂志. 2019.
- [2]虢伟.基于 Web 的医学 CT 三维重建系统关键技术研究.[D]. 华中科技大学,2007.
- [3]王斐 基于医用 CT 图像的三维重建精度及方法研究.[D].山东大学.2010.
- [4]韩成虎.CT 图像三维重建系统的研究与应用.[D].华南理工大学.2010.
- [5]魏星.杨枢.基于 CT 的医学图像三维重建[J].福建电脑.2011.
- [6]陈雷霆 房春兰 基于 Web 模式的医学图像三维重建与切割.[J].计算机应用研究.2009.
- [7]王从军 莫健华 基于体数据的 CT 图像三维重建.[C] 2004 年中国机械工程学会年会.2004
- [8]董玉莹 医学 CT 图像的三维可视化研究.[J].山东科技大学.2019
- [9]马鑫.基于 web 的医学影像三维可视化系统的设计与实现. [C].电脑知识与技术.2024.
- [10]Soferman Z.Advanced graphics behind medical virtual reality: Evolution of algorithms, hardware, and software interfaces 1998 .
- [11]王晓.医学 CT 三维重建技术的研究与实现 .[D].华中科技大学.2007
- [12]张鑫 基于医学 CT 图像的三维重建研究 [J] 中国医药导报, 2009.
- [13]卢朝峰 CT 图像三维重建的研究与实现. [D]复旦大学, 2013.
- [14]Zhou Y. Volume Partitioning: A Coherent Subvolume Projection Approach for Volume Rendering 1994.
- [15]Cohen D. Proximity Clouds: An Acceleration Technique for 3D Grid Traversal 1994.
- [16]周宴 医学 CT 图像的三维可视化技术研究 .[D].重庆大学, 2005.
- [17]高关心 谢伟 基于 MATLAB 的脑 CT 图像三维重建研究 .[J]. 中国数字医学.2015.
- [18]高辉 邱兆文 三维重建技术在临床医学中的应用.[J].软件导刊.2018.
- [19]胡莎莉 医学 CT 图像三维重建系统的实现.[C]电工理论与新技术 2004 年学术研讨会, 2004.
- [20]刘光孟 基于医学 CT 序列图像的分割与重建技术研究. [D].大连交通大学.2018.
- [21]李金 胡战利 基于 MC 算法的 CT 图像三维重建[J]应用科技, 2008.

致 谢

夏天，迈着轻盈的步伐悄然降临，我的四年西安邮电大学生活也即将画上句号。时光宛如潺潺溪流，从我们的指缝间无声无息地淌过；岁月恰似无形的鞭子，不停地催促着我们奋勇前行。

此刻，我站在人生的十字路口，心中满是对过往校园生活的深深眷恋。我怀念课堂上老师激情澎湃的讲解，怀念与同学们在校园小径上的欢声笑语，更怀念老师、同学们给予我的悉心关怀、耐心指导与热情鼓励。

在这里，我要特别感谢我的导师王亚刚教授。在王老师细致入微的指导与无微不至的关怀下，我在 Linux 学习的道路上取得了显著的进步。王老师对待科学研究的态度严谨认真、一丝不苟，对待我们学生更是关怀备至、悉心教导。他不仅严格要求我们开展科研工作，还鼓励我们积极拓展其他兴趣爱好，希望我们成为热爱生活、热爱学习的人。

在西安邮电大学的这段时光里，我收获的远不止科研学术知识。老师们的谆谆教诲和同学们的鼓励陪伴，让我变得更加勤奋好学，为我未来的职业生涯奠定了坚实的基础。这些积极的能量，如同璀璨的星光，将照亮我生活的每一个角落，给予我向上的力量，让我能够从容地迎接未来的挑战。

与同学们共度的日子，总是充满了欢声笑语，那些欢乐的时光总是那么短暂。如果可以，我真希望时间能走得慢一些，让我能在这美丽的校园、这纯粹的氛围中多停留片刻。