西安郵電大學

毕业设计（论文）

题　　目：基于 WEB 的点云配准过程模拟及性能评测系统的设计与实现

学　　院：计算机学院

专　　业：网络工程

班　　级：2102

学　　号：04202051

学生姓名：吴柘兴

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

年　　月　　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 的点云配准算法过程模拟及性能评测系统的设计与实现》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：_____ 日 期：_____

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：_____ 导师签名：_____

日 期：_____ 日 期：_____

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 的点云配准算法过程模拟及性能评测系统的设计与实现				
题目来源	☑教师科研课题 ☐教师专业实践 ☐其他				
题目类型	☐艺术作品 ☐硬件设计 ☑软件设计 ☐论文				
题目分类	☑工程实践 ☐社会调查 ☐实习 ☐实验 ☐其他				
题目简述	（为什么申报该课题） 在与中国人民解放军总医院第四医学中心的合作项目中，需要开展骨科形态参数的各种自动测量任务。这这些任务中点云的配准算法是自动测量过程中的重要环节，配准的优劣程度及运算性能等直接影响了整个自动测量系统的执行效率和测量的准确性。本课题就是要对众多的配准算法进行性能比较，并能直观展示配准算法的执行关键步骤，对于整个系统中的配准算法选择和算法错误排查都有非常重要的意义。因此，需要开发一个基于 WEB 的点云配准算法过程模拟及性能评测系统，更适合基于 Internet 的访问，便于远程与合作方交流。				
对学生知识与能力要求	1.熟悉 web 开发的一些框架，例如 Django 框架、VUE 框架等； 2.了解一些三维处理的 js 库，例如 Three.js 等； 3.熟悉 Python 程序开发，了解 VTK、ITK 等库 4.了解 Linux 系统上 web 服务部署的基本方法。				
具体任务以及预期目标	（应完成的具体工作，预期目标和成果形式） 本课题的具体任务： 在已有核心系统的基础上，完成 web 开发环境的搭建、要对众多的配准算法进行性能比较，并能通过 web 界面，直观展示配准算法的执行关键步骤，实现中间结果的获取及展示，然后提取不同配准算法的运行信息和性能评价指标信息，进行可视化对比，最后将整个 WEB 系统部署在 Linux 系统上，实现完整的基于 WEB 的点云配准算法过程模拟及性能评测系统。 预期目标和成果形式： 1.系统设计文档 1 份 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				
时间进度	2024 年 11 月 25 日-11 月 24 日：完成毕业设计选题 2024 年 11 月 25 日-2025 年 1 月 10 日：提交开题报告，前期检查 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 2025 年 3 月 30 日-5 月 17 日：完成设计实现代码，进行代码验收 2025 年 4 月 1 日-5 月 25 日：撰写毕业论文 2025 年 5 月 26 日 6 月 1 日：完善毕业论文，进行论文答辩。				

专业负责人审核意见	签字：　　年　月　日		
系（教研室）主任签字	年　月　日	主管院长签字	年　月　日

西安邮电大学本科毕业设计（论文）开题报告

学生姓名	吴柘兴	学号	04202051	专业班级	网络 2102
指导教师	王亚刚	题目	基于 WEB 的点云配准算法过程模拟及性能评测系统的设计与实现		

选题目的（为什么选该课题）

随着医学影像诸如CT，MRI，Ultrasound技术的发展，医学图像在医疗诊断和治疗中发挥着不可替代的作用[1]。在医学领域中，往往需要开展骨科形态参数的各种测量任务。近年来,随着传感器技术的快速发展,三维点云数据的获取变得越来越容易,点云配准和图像-点云配准这两个计算机视觉中的关键研究领域也受到越来越多学者的关注。它们涉及将来自不同传感器、不同时间或不同视角获取的数据进行精确对齐的技术[2]。点云配准是指将两个或多个点云数据准确的对齐到同一坐标系中[2]。在这个配准的过程中，点云的配准算法是自动测量过程中的重要环节,配准的优劣程度及运算性能等直接影响了整个自动测量系统的执行效率和测量的准确性。因此，对众多的配准算法进行性能比较，并能直观展示配准算法的执行关键步骤，对于整个系统中的配准算法选择和算法错误排查都有非常重要的意义。

随着Web技术的不断发展和普及，基于Web的三维模型展示可以很好的对该配准过程进行模拟。这为医学研究和临床实践提供了更加灵活和便捷的展示方式。因此，本毕业设计选择设计一个基于Web的点云配准算法过程模拟及性能评测系统,对众多的点云配准算法进行过程模拟和性能评估，通过Web界面，直观的展示众多配准算法的执行步骤，实现中间结果的获取及展示，然后提取不同配准算法的运行信息和性能评价指标信息，进行可视化对比。以方便进行性能评估和错误排查。

该系统采用前后端分离的方式进行设计，在前端拟采用HTML标签+CSS样式+JavaScript+Three. js库，以此完成对系统页面的搭建；后端采用Python + VTK、ITK、open3d库+Django框架进行开发，实现算法模拟和评测功能。本系统使用Django作为后端框架，可以构建一个灵活、可扩展的Web应用程序。而Three. js作为前端库，可以提供高性能的三维模型渲染和交互功能。

Python是一种面向对象的程序设计语言，它作为通用且功能强大的编程语言广受好评。Python语言的语法简洁而清晰，具有丰富和强大的类库[4]。目前已经有大量的用于医学图像处理和分析的开发应用平台，其中ITK（Insight Segmentation and Registration Toolkit）主要提供了医学图像分割和配准等方面的功能，VTK（Visualization Toolkit）则提供了可视化方面的功能，用于观察结果以及进行交互显示[5]。

Django是一个功能丰富的快速Web开发框架,结合轻量级数据库应用SQLite,可以进行快速的Web应用开发[6]。在Web开发方面具有简洁、清晰、高效、安全的优点,特别适合快速构建各类Web应用[7]。Django的优点之一是其开发效率和易用性。Django框架有丰富的用于开发Web应用的组件，这些组件都是用Python开发的，并为开源界所修改和使用。设计Django框架的组件的目的是实现重用性，并使之具有易于性。Django框架中URL系统设计非常强大且灵活，可以在Web应用中为URL设计匹配模式，并用Python函数处理。这种设计使

Web应用开发者可以创建友好的URL，使之更适合于搜索引擎的搜索。Django框架遵循Web开发的MVC（Model模型、View视图、Controller控制器）架构。MTV框架也可以是一种MVC框架，而Django更关注的是模型（Model）、模板（Template）和视图（View）[4]。

Three. JS是一款优秀的WebGL框架，被广泛应用于各种项目之中。Three. JS采用了许多图形引擎的高级技巧来提升性能，包括阴影映射、光照模拟及纹理映射等技术，使得3D图形的渲染更加逼真和美观。同时，Three. JS还支持各种交互操作，如鼠标点击、拖拽等，使得用户可以更好地与3D图形进行交互[8]。我们可以通过Three. JS轻松实现多样化的3D图形效果，为网页带来更加丰富和生动的内容[9]。

计算机辅助治疗的核心是医学图像处理技术。医学图像的处理技术主要包含医学图像配准，医学图像分割，医学图像分析，以及医学图像的三维可视化成像技术等等[4]。在这些技术中，医学图像配准技术往往起着至关重要作用。

总而言之，基于Web的点云配准算法过程模拟及性能评测系统可以直观的展示众多配准算法的执行步骤，实现中间结果的获取及展示，更好的帮助计算机辅助医学治疗。

参考资料

[1] James C. Duncan and Nicholas Ayache. 2000. Medical Image Analysis: Progress over Two Decades and the Challenges Ahead. IEEE Trans. Pattern Anal. Mach. Intell. 22, 1 (January 2000), 85 - 106.

[2] 艾阳. 三维点云及图像-点云配准方法研究[D]. 吉林大学,2024. DOI: 10. 27162/d. cnki. gjlin. 2024. 006761.

[3] Burch C . Django, a web framework using Python: tutorial presentation[J]. Journal of Computing Sciences in Colleges, 2010.

[4] 白昌盛. 基于Django的Python Web开发[J]. 信息与电脑(理论版),2019,31(24):37-40.

[5] 李浩文,胡方旭,白亚南,等. 基于ITK与VTK的医学图像配准软件的开发[J]. 电子制作,2019,(17):46-47+30. DOI: 10. 16589/j. cnki. cn11-3571/tn. 2019. 17. 016.

[6] 龚新定,余艳梅,吴小强,等. 基于Django的实验室信息管理系统设计[J]. 微型机与应用,2016,35(22):108-111. DOI: 10. 19358/j. issn. 1674-7720. 2016. 22. 029.

[7] 刘班. 基于Django快速开发Web应用[J]. 电脑知识与技术,2009,5(07):1616-1618.

[8] 王鹏飞. 面向虚拟实验的WebGL开发框架的研究[D]. 北京邮电大学,2019.

[9] 马文妙. 基于BIM的装配式建筑构件建模及可视化研究与实现[D]. 河北工程大学,2020. DOI: 10. 27104/d. cnki. ghbjy. 2020. 000736.

[10] 孔琪. 面向髌骨脱位诊断的骨科CT图像自动测量[D]. 山东大学,2016.

前期基础（已学课程、掌握的工具，资料积累、软硬件条件等）

1. 已学课程

Python程序设计，数据结构与算法，web开发技术，web应用设计，Linux网络操作系统，数

据库原理。

2. 掌握的工具

PyCharm，MySQL数据库，python程序编程，Django框架、VUE框架，Redis数据缓存，了解了用于三维处理的Three. js库，了解了用于图像处理的VTK、ITK库。

3. 资料积累

《Python程序设计应用与案例》、《Web前端开发简明教程》、《深入理解Django》

4. 软硬件条件

笔记本电脑一台（处理器Intel i7，内存16GB），Windows11，PyCharm，Vscode，MySQL8. 0，Ubuntu操作系统。

要研究和解决的问题（做什么）

1. 用户界面的设计：应设计一个直观、易用的用户界面，直观的展示点云配准过程的中间结果和算法指标，让用户使用系统进行点云配准的模拟。

2. 后端配准的实现并记录中间结果：应在后端对源点云数据和目标点云数据进行配准并记录中间结果并记录在文件或数据库中。

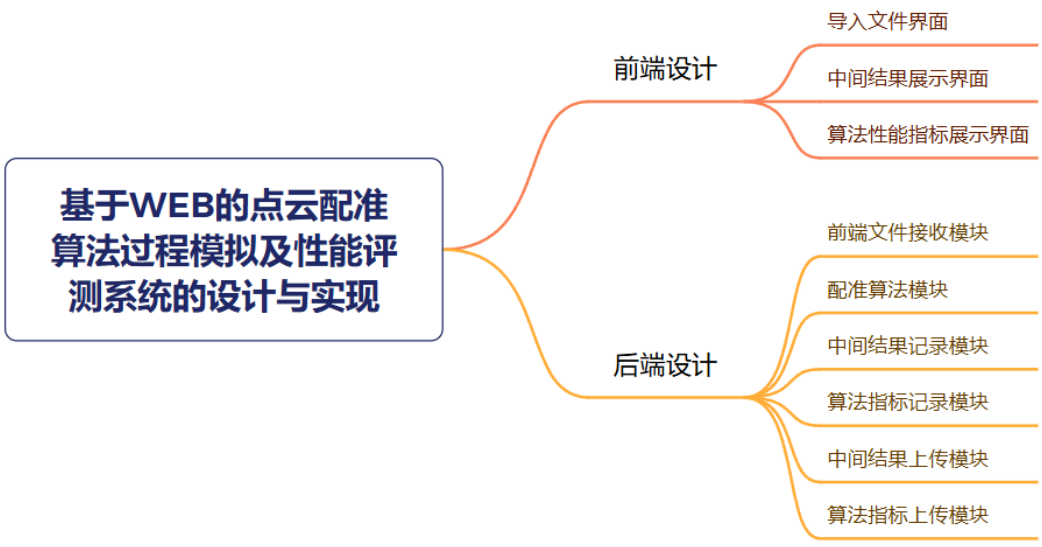
3. 点云数据前后端的交互和处理：应从前端获取源点云数据和目标点云，然后传给后端进行配准。后端也应把执行过程记录的中间结果和算法指标传给前端界面进行展示。

4. 点云数据转化为三维模型展示和交互性：在Web应用程序中应实现三维模型的实时展示和交互操作。这包括实现模型的灯光，材质，纹理，动画等得实现。

5. 配准过程可能时间过长导致TCP连接断开：应在配准过程中保证后端可以及时将中间结果传给前端。

工作思路和方案（怎么做）

1. 工作思路和方案



该系统采用前后端分离的方式进行设计，在前端拟采用HTML标签+CSS样式+JavaScript+Three. js库，以此完成对系统页面的搭建，前端计划三个界面：导入需要配准的文件和目标文件并传给后端进行配准、配准中间结果的展示界面、算法性能指标的

展示界面。后端采用python+VTK、ITK库+Django框架进行开发，实现算法模拟和评测功能。计划实现六个模块：前端文件接收模块、配准算法模块、中间结果记录模块、算法指标记录模块、中间结果上传模块和算法指标上传模块。

（1）前端界面的拟采用HTML标签+CSS样式+JavaScript+Three. js库实现可视化三维点云数据。Three. js是一个基于WebGL的JavaScript库，它使得在网页上创建和显示3D图形变得简单。使用Three. js实现前端点云数据的可视化。前端发送HTTP请求到Django后端来获取中间结果或算法指标。

（2）对于前后台数据的交互，前端选择源点云文件和目标点云文件如. ply格式的文件，在前端经过three. js库实现可视化处理，传送给后端，后端可以采用Open3D这一数据处理库使用希望的配准算法如ICP对数据进行配准。后台进行配准对中间结果的记录可以在每个关键步骤之后保存当前状态下的点云数据或变换矩阵。例如，在迭代最近点（ICP）算法的每次或若干次迭代后，可以保存当前的最佳匹配结果到一个文件中。

（3）针对后端结果传给前端，计划使用Django框架来构建一个API服务如RESTful API，这个API将会接收配准请求，并返回中间结果。迭代计算完成后，根据前端请求，将记录文件的点云数据或变换矩阵转化适合传输的格式（如PLY文件），并通过HTTP响应发送给前端。前端从后端API拉取点云数据后，将接收到的文件又通过three. js库实现可视化处理。

（4）针对算法指标的记录，可以记录每次或若干次迭代后的关键指标如迭代次数、均方根误差、消耗时间等指标到一个列表或字典当中，通过API响应返回给前端。前端收到后使用图表库如Chart. js来绘制随时间变化的指标曲线。

（5）针对配准过程中可能导致的连接断开问题，可以采用定时向服务器发送GET请求来保持连接。

2. 毕业论文进度计划

2024年11月25日-11月24日：完成毕业设计选题，查阅相关资料和文献，撰写开题报告。

2024年11月25日-2025年1月10日：提交开题报告，学习并熟悉Django框架开发，了解熟悉three. js库，VTK、ITK库的使用。

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，准备中期检查。

2025年3月30日-5月17日：完成前后端设计实现代码，进行代码验收。

2025年4月1日-5月25日：撰写毕业论文。

2025年5月26日-6月1日：完善毕业论文，进行论文答辩。

指导教师意见

签字：　　年　月　日

西安邮电大学毕业设计（论文）成绩评定表

<table>
<tr><td>学生姓名</td><td>吴柘兴</td><td>性别</td><td>男</td><td>学号</td><td>04202051</td><td>专业班级</td><td>网络工程
2102</td></tr>
<tr><td>课题名称</td><td colspan="7">基于 WEB 的点云配准算法过程模拟及性能评测系统的设计与实现</td></tr>
<tr><td rowspan="4">指导教师意见</td><td>支撑指标点/赋分</td><td>3-2/20</td><td>4-2/20</td><td>5-3/10</td><td>7-2/10</td><td>8-2/10</td><td>11-2/10</td><td>12-2/20</td><td>合计</td></tr>
<tr><td>得分</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="8"></td></tr>
<tr><td colspan="8">指导教师(签字)：　　年　月　日</td></tr>
<tr><td rowspan="4">评阅教师意见</td><td>支撑指标点/赋分</td><td></td><td></td><td></td><td></td><td></td><td></td><td>合计</td></tr>
<tr><td>得分</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="8"></td></tr>
<tr><td colspan="8">评阅教师(签字)：　　年　月　日</td></tr>
<tr><td rowspan="4">验收小组意见</td><td>支撑指标点/赋分</td><td></td><td></td><td></td><td></td><td>合计</td></tr>
<tr><td>得分</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="6"></td></tr>
<tr><td colspan="6">验收小组组长(签字)：　　年　月　日</td></tr>
<tr><td rowspan="4">答辩小组意见</td><td>支撑指标点/赋分</td><td></td><td></td><td></td><td></td><td>合计</td></tr>
<tr><td>得分</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="6"></td></tr>
<tr><td colspan="6">答辩小组组长(签字)：　　年　月　日</td></tr>
<tr><td rowspan="3">学生总评成绩</td><td>评分比例</td><td>指导教师(20%)</td><td>评阅教师(30%)</td><td>验收小组(20%)</td><td>答辩小组(30%)</td><td>合计</td></tr>
<tr><td>评分</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="4">毕业论文(设计)最终等级制成绩（优秀、良好、中等、及格、不及格）</td><td colspan="2"></td></tr>
<tr><td>答辩委员会意见</td><td colspan="7">

学院答辩委员会主任(签字、学院盖章)：　　年　月　日</td></tr>
</table>

摘　要

点云配准技术是三维计算机视觉和相关应用领域的关键技术，其旨在将不同来源下或不同视角下的点云数据对齐到统一坐标系下。然而，不同的配准算法原理复杂难懂，迭代过程抽象，且算法性能的评估往往缺乏直观的观察手段。为了解决这一问题，本系统设计并实现了一个基于 Web 的点云配准算法过程模拟及性能评测系统。

本系统主要实现了以下功能：

1. 配准过程的可视化模拟与性能展示

本系统前端使用 Three. js 库实现点云数据的动态加载与配准过程的可视化模拟，并通过 Chart. js 库将性能指标数据图表化展示。

2. 配准算法的后端实现

本系统后端采用 Python 语言和 Django 框架构建，通过集成 Open3D 和 pycpd 等主流点云处理库，实现了迭代最近点（ICP，Iterative Closest Point）算法和相干点漂移（CPD，Coherent Point Drift）算法。

通过本系统，用户可以通过 Web 页面直观地观察到选定的配准算法的迭代过程，观察源点云如何逐步地与目标点云对齐。同时，用户可通过本系统提供的性能指标图表，帮助用户分析算法的性能指标，方便用户学习算法、选择算法以及初步的错误排查。经过测试表明，该系统能准确模拟算法过程并展示算法性能，对用户提升点云配准算法的理解和应用具有积极的意义。

关键词：点云配准；过程模拟；性能评测；Three. js；Django 框架

ABSTRACT

Point cloud registration technology is a key technology in 3D computer vision and related applications, which aims to align point cloud data from different sources or different viewing angles into a unified coordinate system. However, the principles of different registration algorithms are complex and difficult to understand, the iterative process is abstract, and the evaluation of algorithm performance often lacks intuitive observation methods. In order to solve this problem, this system designs and implements a web-based point cloud registration algorithm process simulation and performance evaluation system.

The system mainly realizes the following functions:

1. Visual simulation and performance display of the registration process

The front-end of the system uses the Three.js library to realize the visual simulation of the dynamic loading and registration process of point cloud data, and displays the performance index data graphically through the Chart.js library.

2. Back-end implementation of the registration algorithm

The backend of the system is built using Python language and Django framework, and the Iterative Closest Point (ICP) algorithm and Coherent Point Drift (CPD) algorithm are realized by integrating mainstream point cloud processing libraries such as Open3D and pycpd.

Through this system, users can visually observe the iterative process of the selected registration algorithm through the web page, and observe how the source point cloud is aligned with the target point cloud step by step. At the same time, users can help users analyze the performance indicators of algorithms through the performance index charts provided by the system, which is convenient for users to learn algorithms, select algorithms and preliminary error troubleshooting. The test results show that the system can accurately simulate the algorithm process and demonstrate the algorithm performance, which is of positive significance for users to improve the understanding and application of point cloud registration algorithm.

Key words： Point cloud registration；Process simulation；Performance Evaluation；Three.js；Django framework

目　录

第 1 章　绪论

1.1　研究背景与意义

近些年来，三维扫描技术和传感技术的发展迅速，点云数据作为一种重要的三维数据表现形式，具有能够直接获取物体表面三维几何信息和纹理信息的特点。因此，在医学影像、三维重建、自动驾驶、机器人导航等众多领域表现出了巨大的应用潜力。例如，在医学影像领域，通过对 CT 或 MRI 等医学图像进行三维重建，医生可以得到病灶或器官的点云模型，这些点云模型可以帮助医生进行病情诊断和手术规划；在三维重建领域，使用摄影测量或激光扫描技术获取的点云数据，能够真实地还原场景状况或物体的三维结构。

点云配准技术的目的是将不同视角、不同时间或来自不同传感器的点云数据转换到统一的坐标系下，从而获得完整、精确的三维模型或场景信息[1][2]。点云配准的精度和效率直接影响着后续三维重建、目标识别、场景理解等任务的性能。因此，其在相关应用中占据着核心地位。

现在，已有多种不同的点云配准算法，这些配准算法在原理、适用场景和算法性能上具有各自的特点。然而，面对当今日益复杂的应用需求和不断增长的数据量，怎样深入地理解不同配准算法的过程、准确评估和比较它们的性能，以及针对特定场景去选择合适的配准算法，仍然面临着重大的挑战。这不仅要求研究者对各类算法的理论基础有深入的理解，还需要具备有效的实验方法和评价指标用于算法评测。

同时，Web 技术凭借其跨平台、便捷的部署方式、交互能力强等优势，在科学可视化和交互式工具开发领域表现出了巨大的潜力。利用 Web 技术构建点云配准算法的可视化分析和算法比较工具，可以有效地帮助研究人员和工程师们更加直观地理解算法原理、调试算法的关键参数、比较算法的性能，并为算法的优化和应用提供便利。因此，本系统希望能够设计一个辅助用户直观理解算法配准过程，查看算法配准指标，并为算法的选择与优化提供数据支持的 Web 系统。

综上所述，深入研究点云配准算法，理解算法过程和比较算法性能具有重要的学术价值和研究意义。利用 Web 技术构建的点云配准工具，将有助于提升研究效率，促进点云配准技术在不同领域更广泛的应用与发展。

1.2　国内外研究现状

点云配准技术一直是计算机视觉和三维数据处理领域的研究热点。近些年来，出现了大量的配准算法和用于处理点云数据可视化的软件和库。对于“基于 WEB 的点云配准算法过程模拟及性能评测系统的设计与实现”，国内外的研究主要围绕以下几个方面展开。

1.2.1 点云配准算法的研究进展

点云配准一直是计算机视觉和三维数据处理领域的研究热点。近年来，涌现出大量的配准算法，其中迭代最近点（Iterative Closest Point, ICP）及其变体是最经典和广泛应用的一类算法。ICP 算法通过迭代地寻找两个点云之间的最近对应点，并计算最优的刚性变换（旋转和平移）来最小化对应点之间的距离，直至收敛。针对 ICP 算法收敛速度慢、易陷入局部最优等问题，研究人员提出了诸如 Chen 和 Medioni[3]及 Bergevin 等人[4]提出的 Point-to-Plane ICP、Segal 等人[5]提出的 Generalized ICP (GICP) 等改进算法。

除了基于点特征匹配的 ICP 系列算法，还有一种主流的配准方法是基于概率模型的配准，例如高斯混合模型（Gaussian Mixture Model, GMM）配准及其代表性算法相干点漂移（Coherent Point Drift, CPD）。CPD 算法将一个点集建模为 GMM 的质心，并通过概率方法寻找两个点集之间的最优对齐[6]。CPD 算法在处理噪声和离群点方面表现出一定的鲁棒性，但计算复杂度相对较高。

近几年来，深度学习技术发展迅速，基于深度学习的点云配准方法[14][15]也逐渐成为研究的热点。这些方法通常利用神经网络直接从点云数据中学习特征表示，并预测变换参数，从而实现快速且鲁棒的配准。然而，深度学习方法在训练阶段对大规模、高质量的数据集有较强的依赖性，并且在未见场景或不同类型数据上的泛化能力仍有提升的空间。

1.2.2 点云处理与可视化软件和库的现状

目前存在许多成熟的点云处理与可视化软件和库，为点云数据的分析和应用提供了强大的工具。例如：

Open3D 是一个大型的开源库[7]，支持 Python、C++等，该库提供了点云数据的处理、三维重建、可视化等功能。Open3D 以其易用性和可视化方面的优点为很多用户所用，但主要定位于本地环境的开发和应用。

Point Cloud Library (PCL) [8]是另一个大型的开源 C++库，包含了各种点云处理算法，涵盖了滤波、特征提取、配准、分割、识别、可视化等多个方面。PCL 功能全面且性能高效，但其 API 较为复杂，且主要面向开发者进行算法实现和应用开发。

CloudCompare 是一款开源的点云处理软件，提供丰富的点云处理功能，例如点云配准、滤波、分割、测量和可视化等功能。该软件功能强大，但主要是一个桌面应用程序，侧重于专业用户对点云数据的处理和分析。

本毕业设计的研究侧重点在于使用 Web 技术构建点云配准过程的模拟展示。我们的目标是开发一个基于 Web 平台的交互式工具，通过本系统，用户可以直观地观察不同配准算法的执行过程、配准过程中参数的变化，以及不同算法之间的性能差异。使用 Web 系统，用户可以在无需安装额外软件的情况下，方便地学习和理解点云配准的原理和配准过程。

1.2.3 基于 Web 的 3D 应用和科学计算平台的研究

近些年来，基于 Web 的 3D 应用技术和科学计算平台的研究日益受到关注。同时，Web Graphics Library (WebGL)技术的成熟，在浏览器中渲染和交互复杂的三维模型成为很多用户的选择。出现了许多基于 Web 的 3D 可视化库（如 Three.js、Babylon.js）和科学计算库（如 NumPy.js、TensorFlow.js），这些库的出现为开发基于 Web 端的应用系统提供了强大的技术支持。

当前，已经有一些研究探索了将科学计算和可视化工具迁移到 Web 平台的实验。例如，一些在线平台提供了分子可视化、地理信息系统（GIS）数据展示的功能。然而，在点云配准领域，基于 Web 端的过程模拟工具相对较少。

本研究目的是填补这一空白，利用 Web 技术的优势，开发出一个易于使用、交互性强的点云配准学习平台，为研究人员和初学者提供一个直观有效的工具。

1.3 主要研究目标与内容

本研究目的设计并实现一个基于 Web 的点云配准算法过程模拟和性能评测系统。该系统的核心目标是通过 Web 页面实现可视化和交互操作，帮助用户深入理解不同的点云配准算法执行的关键步骤、获取和展示配准过程中的中间结果，并进行算法性能的评估和基础地错误排查。

1.3.1 研究目标

具体而言，本系统将实现以下几点核心功能：

1. 直观展示配准算法的执行关键步骤功能：

本功能通过 3D 可视化技术，清晰地展示配准过程中点云的变换过程，能够使用户直观地理解算法的迭代流程。

2. 中间结果的获取及展示功能：

本功能用户可以查看在配准过程中关键的中间结果，例如初始点云、变换后的点云、对应点云位置的对比、每次迭代计后得到的变换矩阵等，使用户更好地理解算法内部的运作机制。

3. 性能评估和错误排查功能：

本功能提供点云配准算的性能评估和指标展示，以可视化图表和表格的方式展示。

1.3.2 研究内容

为了实现上述研究目标，本系统将主要研究以下内容：

用户界面的设计与开发：需要设计一个简洁直观的 Web 用户界面，方便用户实现点云数据的上传、配准算法的选择、配准过程的控制，并实现可视化结果的展示。

后端配准算法的实现与中间结果的记录：使用 Python 和 django 后端技术，系统至少实现两种主流的点云配准算法（如 ICP、CPD 等），并在算法执行的过程中记录关键的中

间结果数据，用于前端的可视化展示。

前后端数据的交互与通信：需要研究和实现高效可靠的前后端数据交互机制，确保点云数据、算法参数、中间结果和性能指标能够在浏览器端和服务器端之间正确地传输。

基于 Web 的 3D 可视化实现：使用 WebGL 相关的 JavaScript 库（如 Three.js），实现点云数据的三维渲染和交互功能，主要包括点云的颜色、大小和观察视角的控制等，使 Web 界面能够动态展示配准过程和中间结果。

性能指标的可视化展示：在后端配准过程中计算配准性能指标并记录，然后将结果传递到前端进行可视化展示，前端以图表或数值的形式呈现配准误差、迭代次数和运行时间等性能指标信息。

通过以上研究内容，本系统旨在为点云配准的学习、研究和应用提供一个方便、直观且高效的点云配准工具。

1.4 本文结构安排

本文主要内容介绍如下：

第一章为绪论部分，介绍了本课题的研究背景与意义、国内外研究现状和主要研究与内容。

第二章为关键技术及理论介绍部分，介绍了本系统所用的关键技术和理论介绍。

第三章为系统分析与设计部分，介绍了本系统总体框架和功能模块设计。

第四章为系统实现部分，介绍了本系统的前端实现和后端实现。

第五章为系统测试部分，介绍了本系统测试结果并对测试结果进行分析。

第六章为总结与展望部分，总结了本系统功能并分析了系统不足和局限性，最后对系统未来做出展望。

第 2 章　关键技术及理论介绍

2.1　点云数据及其处理

点云是在同一空间参考系下表达目标空间分布和目标表面特性的海量点集合。每个点通常包含三维坐标（X, Y, Z），并可能附带其他信息，如颜色（R, G, B）、法向量、反射强度等。点云数据具有不规则、无序、数据量大等特点，能够真实、直接地反映物体的三维形状和表面细节。

本系统中主要处理的点云数据格式包括：

PLY 格式：该格式由斯坦福大学开发，设计用于存储三维扫描仪产生的图形数据。它可以存储顶点、面片、颜色、法向量等多种信息，支持 ASCII 和二进制两种存储方式，具有较好的可扩展性。

STL 格式：该格式最初为快速原型制造技术设计的文件格式，主要通过描述物体表面的三角面片来表示三维模型。本系统通过对 STL 模型的顶点进行采样，将其转换为点云数据进行处理（如 views.py 中的 stl_to_pointcloud 函数）。

原始的点云数据往往包含噪声、冗余信息、密度不均等问题，这些问题将直接影响后续配准的精度和效率。针对这些问题，常见的预处理步骤包括：

去噪：移除离群点和测量误差引起的噪声点。

下采样：在保持点云主要特征的前提下，减少点云数量，提高处理效率。

法向量估计：为点云中的每个点计算法向量，有助于后续的特征提取和配准。

尽管本系统核心在于配准过程模拟，并未深度集成复杂的预处理模块，但理解预处理的重要性对于点云数据的处理非常重要。

2.2　Web 开发技术

为了实现系统的跨平台访问、交互式可视化和便捷操作，本系统采用了前后端分离的 Web 架构。

2.2.1　前端技术

前端负责用户界面的展示、用户交互以及三维点云和性能指标的可视化。

本系统使用的前端技术如下：

1. HTML:用于构建前端网页的基本结构和内容骨架。
2. CSS: 用于定义前端网页的样式和布局，美化用户界面。

3. JavaScript：是一种Web的脚本语言，用于实现网页的动态交互行为、DOM操作、异步数据请求以及控制前端逻辑。
4. Three.js：一个基于WebGL的JavaScript 3D图形库，它封装了底层的WebGL接口，可以在浏览器中创建和展示三维动画并实现场景交互。本系统使用Three.js来实现点云数据的加载、渲染、用户交互控制等操作以及配准过程的动态可视化。
5. Chart.js：一个简单的JavaScript图表库，本系统中用于将性能评测数据（如本系统的RMSE、Fitness、迭代耗时数据）以折线图的形式直观展示给用户。

2.2.2 后端技术

后端负责处理业务逻辑、执行点云配准算法、管理数据以及提供 API 接口供前端调用。本系统使用的后端技术如下：

1. Python：一种解释型、面向对象的高级编程语言，以其语法简洁、易学易用以及其丰富的第三方库生态而闻名。Python在科学计算、数据分析和Web开发领域都有广泛应用，是本系统后端开发的首选语言。
2. Django：一个基于Python的高级Web框架，遵循MVT(Model-View-Template)设计模式。它鼓励快速开发和简洁实用的设计，提供了包括ORM、URL路由、模板引擎、表单处理、安全防护等在内的一整套Web开发组件[10]。本系统使用Django搭建后端服务，处理前端的文件上传请求，调用点云处理库执行配准算法，并将结果以JSON格式通过API接口返回给前端。

2.3 三维数据处理及可视化库

为了高效地处理点云数据和执行配准算法，本系统集成了多个强大的第三方 Python 库。

2.3.1 0pen3d

Open3D[11]是一个开源的、现代化的三维数据处理库，支持快速开发处理三维数据的软件。它提供了丰富的功能，包括：

1. 三维数据结构：如点云、网格。
2. 三维数据处理算法：如滤波、特征提取、配准、重建、场景分割等算法。
3. 可视化：提供便捷的三维数据可视化工具。
4. 提供便捷的Python和C++接口。

本系统主要使用 Open3D 库进行点云文件的读取（如.ply）、ICP 配准算法的调用（registration_icp）、点云的基本操作以及开发阶段的快速可视化验证。

2.3.2 Pycpd

Pycpd 库是一个使用纯 Python 实现的 Coherent Point Drift (CPD)算法的库，该库可以实现 CPD 的刚性、仿射和非刚性配准。它为用户提供了易于使用的 API，方便用户将 CPD 算法集成到 Python 项目中。本系统使用 pycpd 库实现 CPD 算法的模拟，并通过回调机制在每次迭代后获取中间结果，用于过程展示和性能分析。

2.3.3 VTK 与 ITK

VTK 是一个开源的、用于三维计算机图形学、图像处理和可视化的软件系统。它包含了大量的算法和数据结构，在科学数据可视化方面功能强大。ITK 是一个开源的、跨平台的系统，为开发者提供了广泛的图像分析（特别是医学图像）软件工具，其在图像分割和配准领域具有深厚积累[12][13]。

虽然 VTK 和 ITK 在医学图像处理和三维可视化领域非常重要，并且在本项目的初期调研中被提及，但考虑到 Web 端集成的便捷性、轻量化需求以及 Open3D 和 pycpd 在点云处理和特定算法实现上的直接性，本系统最终主要依赖 Open3D 和 pycpd 进行核心算法的实现和数据处理。VTK/ITK 的强大功能为未来系统扩展（如更复杂的医学场景[17]或体数据可视化）提供了潜在的技术储备。

2.4 点云配准算法原理

点云配准是将两个或多个在不同坐标系下获取的点云数据，通过寻找它们之间的空间变换关系（通常为刚性变换，即旋转和平移），将其统一到同一坐标系下的过程。本系统重点模拟和评测以下几种经典的配准算法：

2.4.1 ICP 算法

ICP 算法是一种应用广泛的点对点迭代配准方法，由 Besl 和 McKay[9]于 1992 年提出。其基本思想是：给定源点云 P 和目标点云 Q，通过迭代寻找 P 中点在 Q 中的最近对应点，然后计算一个最优刚性变换矩阵，使得这些对应点之间的平均距离最小。迭代过程不断优化变换矩阵，直至满足收敛条件后停止迭代。

ICP 算法的主要步骤[16]如下：

1. 选择对应点：对于源点云P中的每个点或其子集，在目标点云Q中找到其欧氏距离最近的点作为对应点。
2. 计算变换矩阵：基于找到的对应点对，计算使它们之间误差平方和最小的刚性变换矩阵（旋转R和平移t）。常用的方法是基于奇异值分解（SVD）。
3. 应用变换：将计算得到的变换矩阵应用于源点云P，得到变换后的点云P'。
4. 评估误差：计算P'与Q之间对应点的均方根误差（RMSE）或平均距离。
5. 迭代：若误差小于预设阈值或达到最大迭代次数，则停止迭代；否则，以P'为新的源点云，返回步骤1。

本系统中 perform_icp_with_log 函数封装了 Open3D 库中的 ICP 实现方法，并在每次迭代后记录每次迭代的 RMSE、匹配度（Fitness）和变换矩阵等信息。ICP 算法的优点是原理简单、精度较高（特别是在初始位姿较好时），缺点是对初始位姿敏感，容易陷入局部最优，并且对噪声和离群点较为敏感。

2.4.2 CPD 算法

CPD 算法是一种概率性的点云配准方法，由 Myronenko 和 Song 于 2010 年提出[6]。它将点云配准问题建模为两个点集之间的概率密度估计问题。其中一个点集（通常是源点云）被视为高斯混合模型（GMM）的质心，另一个点集（目标点云）被视为从该 GMM 中生成的数据点。CPD 算法通过最大化似然函数来迭代地优化 GMM 质心的位置以及它们之间的变换关系（刚性或非刚性）。

对于刚性配准，CPD[6]算法的主要步骤如下：

1. 初始化：将源点云Y的每个点视为GMM的一个质心，初始化变换参数（旋转R、平移t）和GMM的方差。
2. E步 (Expectation)：给定当前的GMM参数和变换，计算目标点云X中的每个点属于源点云Y中每个GMM分量（质心）的后验概率。
3. M步 (Maximization)：基于E步计算得到的后验概率，更新GMM的质心位置（即对源点云Y应用新的变换）、变换参数（R和t）以及方差，以最大化期望对数似然函数。
4. 迭代：重复E步和M步，直到算法收敛（例如，参数变化小于阈值或达到最大迭代次数）。

本系统中 perform_CPD_with_log 函数利用 pycpd 库实现 CPD 的刚性配准，并在回调函数 visualize_open3d 中记录每次迭代的误差和变换矩阵。CPD 算法的优点是对噪声、离群点和点云缺失具有较好的鲁棒性，且不需要显式寻找点对对应关系。缺点是相对于 ICP，其计算复杂度可能更高。

在许多实际应用中，点云之间可能存在非刚性形变，例如医学图像中的器官形变。为了处理这种情况，CPD 算法可以扩展到非刚性配准。

非刚性 CPD[6]算法的主要步骤如下：

1. 概率模型的建立：与刚性CPD相似，非刚性CPD也是将源点云Y的每个点视为GMM的质心，目标点云X视为从该GMM中生成的数据点。不同之处在于，非刚性CPD不再求解一个全局的旋转和平移，而是求解一个位移场（displacement field）v，该位移场描述了源点云中每个点如何移动以匹配目标点云。
2. 参数化：位移场 v(Y) 可以表示为控制点（通常是源点云Y自身，或其子集）的加权线性组合，权重系数 W 是需要优化的参数。v(Y) = G * W。其中，G 是一个核

矩阵（Kernel Matrix），其元素 G_ij 表示源点云中第 i 个点和第 j 个点之间的核函数值（如高斯核）。

3. 目标函数：非刚性CPD的目标函数通常包含两项：

   数据拟合项 (Likelihood Term)：与刚性CPD类似，衡量变换后的源点云与目标点云之间的匹配程度。

   正则化项 (Regularization Term)：基于运动模型的平滑性约束，惩罚过于剧烈的变形，防止点云在配准过程中发生不希望的扭曲或撕裂。正则化项的强度由一个参数 λ (lambda) 控制。

4. 迭代优化（EM算法）：与刚性CPD一样，非刚性CPD也采用EM算法进行迭代优化：

   E步 (Expectation)：给定当前的GMM参数和变换，计算目标点云X中的每个点属于源点云Y中每个GMM分量（质心）的后验概率。

   M步 (Maximization)：基于E步计算得到的后验概率，更新GMM的质心位置（即对源点云Y应用新的变换）、变换参数（R和t）以及方差，以最大化期望对数似然函数。

5. 收敛：迭代过程持续进行，直到算法收敛（例如，目标函数值的变化小于阈值，或达到最大迭代次数）。

非刚性 CPD 可以使用 pycpd 库通过 DeformableRegistration 类来实现。用户可以指定核函数类型、正则化参数 alpha 和 beta 来执行非刚性 CPD 算法。

2.4.3 BCPD 算法

BCPD 算法[20]是 CPD 算法的一种贝叶斯扩展，它在 CPD 的基础上引入了贝叶斯框架，通过对模型参数施加先验分布，能够更好地处理不确定性、噪声，并能自动估计一些超参数。BCPD 在鲁棒性和精度方面通常优于 CPD，尤其是在具有挑战性的场景下。由于项目时间限制和实现复杂度，本系统暂未集成 BCPD 算法的模拟，但其作为一种重要的鲁棒配准算法，具有进一步研究和实现的价值，并在前端界面预留了选项。

第 3 章　系统分析与设计

本章将详细介绍基于 WEB 的点云配准算法过程模拟及性能评测系统的整体架构、功能模块划分以及关键的接口设计。系统旨在为用户提供一个直观、交互式的平台，以模拟和理解不同点云配准算法的执行过程，并对算法性能进行评估。

3.1　系统总体架构

本系统采用经典的 B/S (Browser/Server) 架构，并遵循前后端分离的设计思想，以提高开发效率、可维护性和可扩展性。

前端：用户直接交互的界面，负责点云数据的上传、算法选择、配准过程的可视化展示、中间结果的动态呈现以及性能指标的图表化显示。主要技术包括 HTML、CSS、JavaScript，并借助 Three.js 库进行三维渲染，Chart.js 库进行图表绘制。

后端：负责处理核心业务逻辑，包括接收前端上传的文件、调用相应的点云配准算法（ICP、CPD 等）、记录算法执行过程中的中间状态和性能数据，并将处理结果通过 API 接口返回给前端。主要技术为 Python 语言和 Django Web 框架，并使用 Open3D、pycpd 等库执行点云处理和配准任务。

本系统的整体数据流和交互流程如图 3.1 所示：

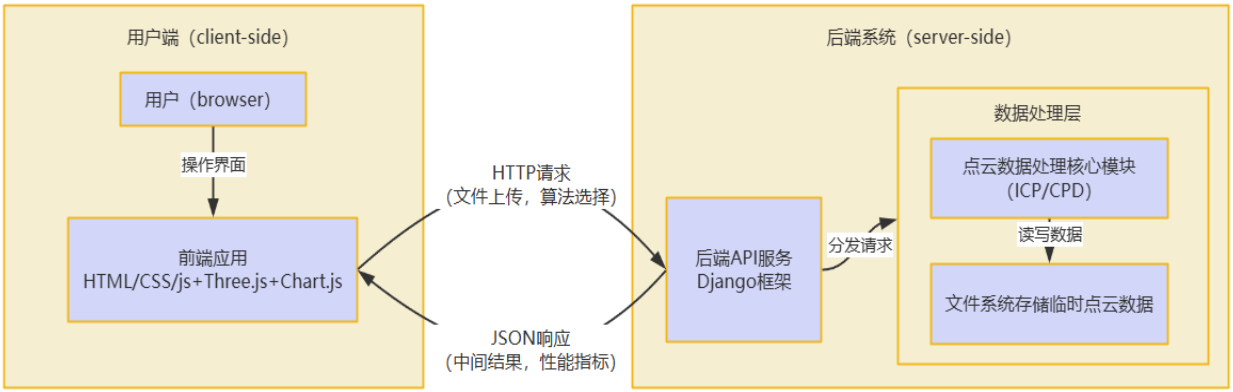


图 3.1　系统数据流及交互图

3.2　功能模块设计

根据系统的核心目标和用户需求，系统主要划分为以下几个功能模块，这些模块协同工作，共同完成点云配准的模拟和性能评测任务。

3.2.1　用户交互与数据上传模块

本模块是用户与系统交互的入口，负责接收用户的操作指令和数据输入。

文件选择与上传功能：设计两个文件输入框，允许用户分别选择源点云文件和目标点云文件。支持的文件格式包括.ply, .stl。用户选择文件后前端对文件类型进行初步校验，之后通过 HTTP POST 请求将选定的文件发送到后端。

算法选择功能：设计一个下拉选择框，列出系统中支持的点云配准算法（如 ICP, CPD）供用户选择。用户选择的算法类型将作为参数和文件一同发送到后端。

配准启动控制：提供一个“开始配准”按钮，用户点击后触发文件上传和后端配准流程。

3.2.2 点云配准核心处理模块

该模块是系统的核心处理模块，负责实际的点云配准计算和过程记录。

文件接收与解析：接收前端上传的点云文件。根据文件扩展名（.ply, .stl）调用相应的库和函数进行文件读取和解析。对于 STL 文件，实现 stl_to_pointcloud 函数，将其顶点采样转换为 Open3D 点云对象。点云解析成功后将解析后的点云数据传递给相应的算法处理单元。

ICP 算法实现与日志记录 (perform_icp_with_log)：通过调用 Open3D 库的 registration_icp 函数执行 ICP 配准，并设计为手动迭代模式，以便在每次（或每 N 次）迭代后捕获中间结果。记录的关键信息包括：当前迭代次数、均方根误差（RMSE）、匹配度（Fitness）、单次迭代耗时、当前累积变换矩阵。所有日志信息存储在结构化数据中，供后续返回给前端。

CPD 算法实现与日志记录 (perform_CPD_with_log)：使用 pycpd 库的 RigidRegistration 类执行刚性 CPD 配准。通过 pycpd 的回调函数机制（在 visualize_open3d 中封装的逻辑），在每次迭代后捕获中间结果。记录的关键信息包括：当前迭代次数、CPD 算法的误差值（Error）、单次迭代耗时、当前累积变换矩阵（从旋转矩阵 R 和平移向量 t 构建）。日志信息同样结构化存储。

中间结果管理：算法在迭代过程中产生的变换矩阵序列是实现过程模拟的重要数据信息。这些矩阵被收集起来，以便前端能够逐帧展示源点云的变换过程。所有记录的中间结果和最终性能指标被整合，准备通过 API 响应发送。

该模块具体流程图如图 3.2 所示。

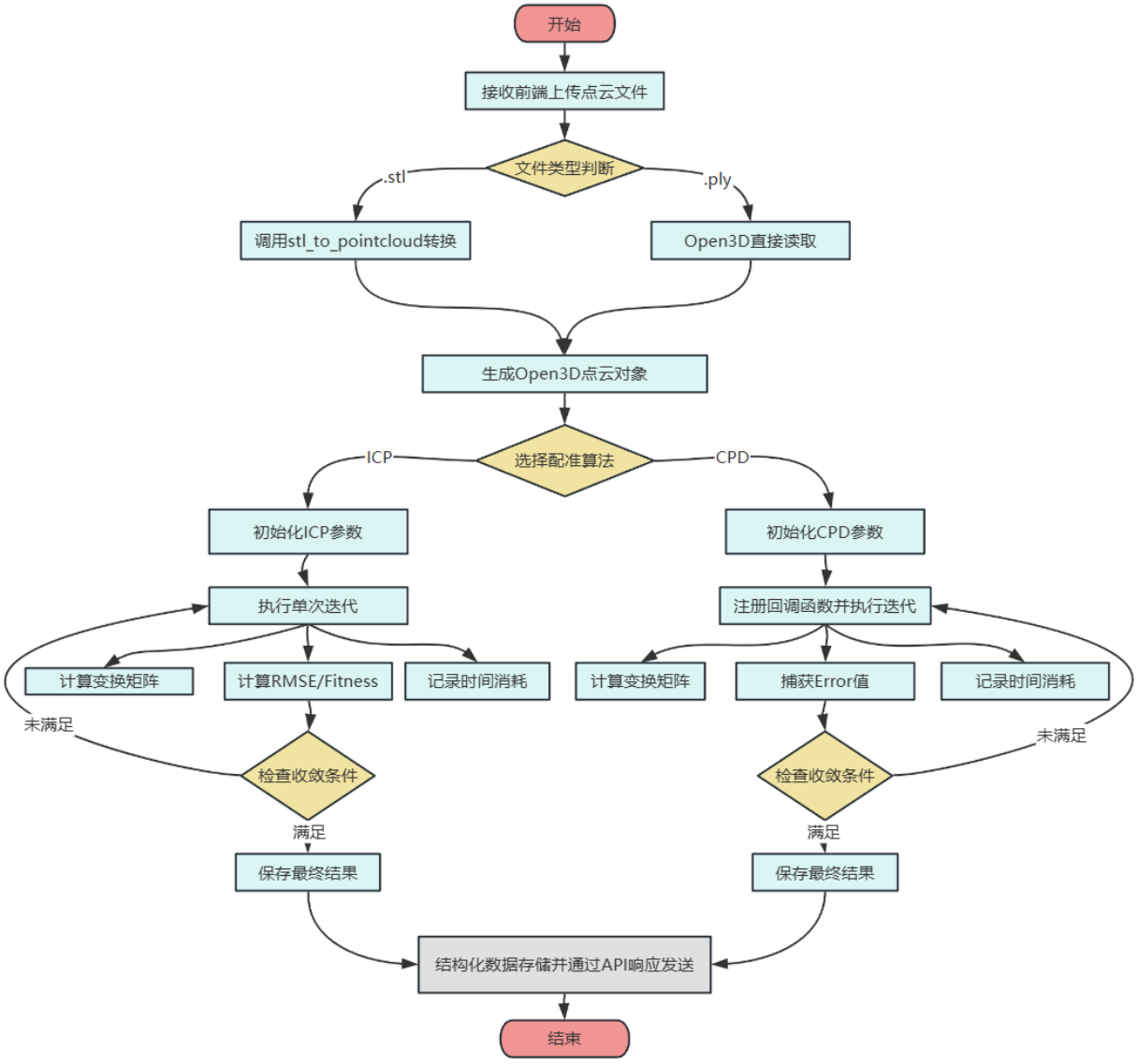


图 3.2　点云配准核心处理模块流程图

3.2.3　三维可视化与过程模拟模块

该模块负责将抽象的点云数据和配准过程以直观的三维形式展现给用户。

初始点云显示：前端接收后端返回的原始源点云和目标点云数据（坐标列表）。根据返回的点云数据使用 Three.js 创建点云几何体（THREE.BufferGeometry）和材质（THREE.PointsMaterial），并在场景中渲染。源点云和目标点云使用不同颜色（红色和绿色）进行区分。提供相机控制（如 OrbitControls）允许用户旋转、缩放、平移观察场景。

配准中间过程的动态可视化：前端接收后端返回的变换矩阵序列（transformations）。并提供一个滑块（iterationSlider），其范围对应迭代次数（即变换矩阵的数量）。当用户拖动滑块时，updateTransformation 函数被调用。该函数根据滑块选择的迭代索引，获取对应的变换矩阵。将选定的变换矩阵应用于原始源点云的顶点坐标，动态更新源点云在场景中的位置和姿态，从而模拟配准的迭代过程。

最终配准结果展示：当滑块在最后一次迭代位置，或配准完成后，展示最终的配准对

齐效果。

3.2.4 性能指标展示模块

该模块负责提取、传递和展示算法的性能评估指标。

后端指标提取与汇总：在配准算法执行完毕后，后端整理出最终的性能指标，如总迭代次数、总耗时、最终 RMSE/Fitness (ICP)、最终 Error (CPD)。同时，将每次迭代记录的性能指标（如 process_log 中的 rmse, fitness, time_once, error 列表）一并准备好。

前端指标表格展示 (.table-result)：将后端返回的最终性能指标以清晰的文本或表格形式展示在控制面板区域。

前端指标图表化展示 (renderProcessChart)：利用 Chart.js 库，将后端返回的迭代过程中的性能指标（如 RMSE、Fitness、Error、单次耗时随迭代次数的变化）绘制成折线图。这使得用户可以直观地观察算法的收敛行为和性能变化趋势。

3.2.5 文件存储模块

考虑到本系统的核心功能是实时模拟和评测，主要数据（如上传的点云、中间变换矩阵、性能日志）在单次会话中处理和展示，所以本系统使用临时文件存储所需处理的点云文件。

临时文件存储：上传的点云文件会在后端临时存储于服务器的文件系统中（temp 目录），处理完毕后即被删除，以避免占用过多磁盘空间。

3.3 接口设计

前后端分离架构的核心在于清晰定义的 API 接口。本系统主要通过一个核心 API 接口进行数据交互。

API 接口属性如表 1 所示：

表 1 系统接口属性表

接口属性	备注
接口端点	/uploadfile/
HTTP 方法	POST
请求参数（FormData）	algorithm（String）：用户选择的配准算法名称 file1（File）：源点云文件 file2（File）：目标点云文件
请求头部	X-CSRFToken：Django CSRF 保护机制所需的令牌

响应数据格式图 3.3 所示：

```
{
    "final": { // 最终配准结果总结
        "iterations": Number,        // 总迭代次数
        "fitness": Number,           // (ICP) 最终匹配度 (0-1)
        "rmse": Number,              // (ICP) 最终内点均方根误差
        "time_count": Number,        // 总耗时 (秒)
        // (CPD specific fields like final error could be here if applicable)
    },
    "transformations": [ // 每次迭代后的累积变换矩阵列表 (4x4矩阵的列表表示)
        [[r11,r12,r13,tx], [r21,r22,r23,ty], [r31,r32,r33,tz], [0,0,0,1]], // Iteration 1
        ... // Iteration N
    ],
    "process_log": { // 每次迭代的详细日志
        "iteration": [1, 2, ..., N], // 迭代次数列表
        "rmse": [val1, val2, ...],       // (ICP) 每次迭代的RMSE列表
        "fitness": [val1, val2, ...],    // (ICP) 每次迭代的Fitness列表
        "error": [val1, val2, ...],      // (CPD) 每次迭代的Error列表
        "time_once": [t1, t2, ...]       // 每次迭代的耗时列表 (秒或毫秒)
    },
    "pcd1_points": [ // 原始源点云坐标列表 (用于前端Three.js渲染)
        [x1,y1,z1], [x2,y2,z2], ...
    ],
    "pcd2_points": [ // 原始目标点云坐标列表
        [x1,y1,z1], [x2,y2,z2], ...
    ]
}
```

图 3.3　响应数据格式

3.4　本章小结

本章详细描述了系统的总体架构、各功能模块的设计以及核心的 API 接口。这些设计为下一章的系统实现奠定了基础。

第 4 章　系统实现

本章在前三章系统分析与设计的基础上，详细介绍基于 WEB 的点云配准算法过程模拟及性能评测系统的具体实现过程。内容包括前端用户界面与交互逻辑的实现、后端核心算法处理与 API 服务的实现，以及在实现过程中遇到的一些关键问题的解决方案。

4.1　前端实现

前端主要负责用户界面的构建、用户交互逻辑的处理、三维点云数据的可视化以及性能指标的图表化展示。所有前端代码均在 index.html 文件中实现，利用 HTML 构建页面结构，CSS 美化样式，JavaScript 执行动态逻辑和与后端 API 的交互。

4.1.1　页面布局与样式

前端页面的整体布局采用 CSS Grid 实现，将页面划分为左侧控制面板区、右侧三维可视化区和下方的配准指标区，如 index.html 中 及其子元素的 CSS 样式所示。

控制面板 (.control-panel)：包含文件上传输入框、算法选择下拉菜单、开始配准按钮以及最终配准结果的文本展示区域。样式通过 CSS 定义，使其具有清晰的视觉分隔和良好的用户体验。

可视化区域 (.vis-area)：用于嵌入 Three.js 渲染的 Canvas 画布，并包含一个 HUD（Heads-Up Display）层，显示迭代次数和控制迭代过程的滑块。

指标区域 (.metric-panel)：用于嵌入 Chart.js 生成的性能指标图表。

页面布局如图 4.1 所示。

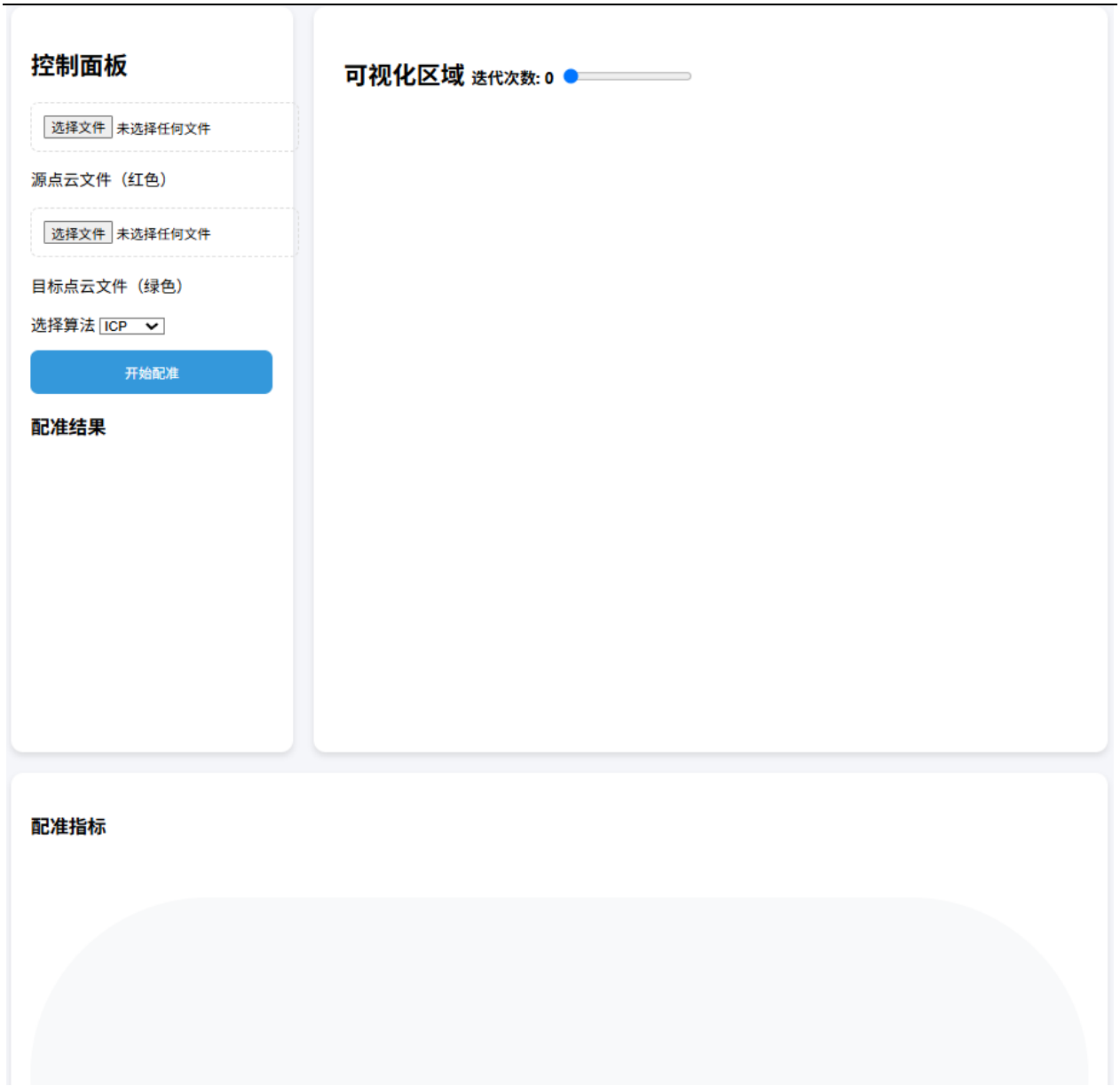


图 4.1　系统页面布局

4.1.2　Three.js 场景搭建与点云可视化

三维可视化是本系统的核心功能之一，通过使用 Three.js 库实现。

初始化场景：在 loadPointClouds 函数中，首先判断 isThreeInitialized 标志，如果是首次加载，则创建 THREE.Scene、THREE.PerspectiveCamera 和 THREE.WebGLRenderer。场景背景色设置为浅灰色（0xcccccc）。渲染器尺寸根据其容器动态设置。

点云对象创建：createPointCloud(points, color)函数负责将后端传来的点云坐标数据转换为 Three.js 可用的格式。具体步骤如下：

1. 将点坐标数组points.flat()转换为Float32Array。
2. 创建THREE.BufferGeometry对象，并将顶点数据设置为其position属性。
3. 创建THREE.PointsMaterial，设置点云颜色和点的大小。

4. 使用几何体和材质创建THREE.Points对象，并将其添加到场景中。

点云对象创建流程图如图 4.2 所示：

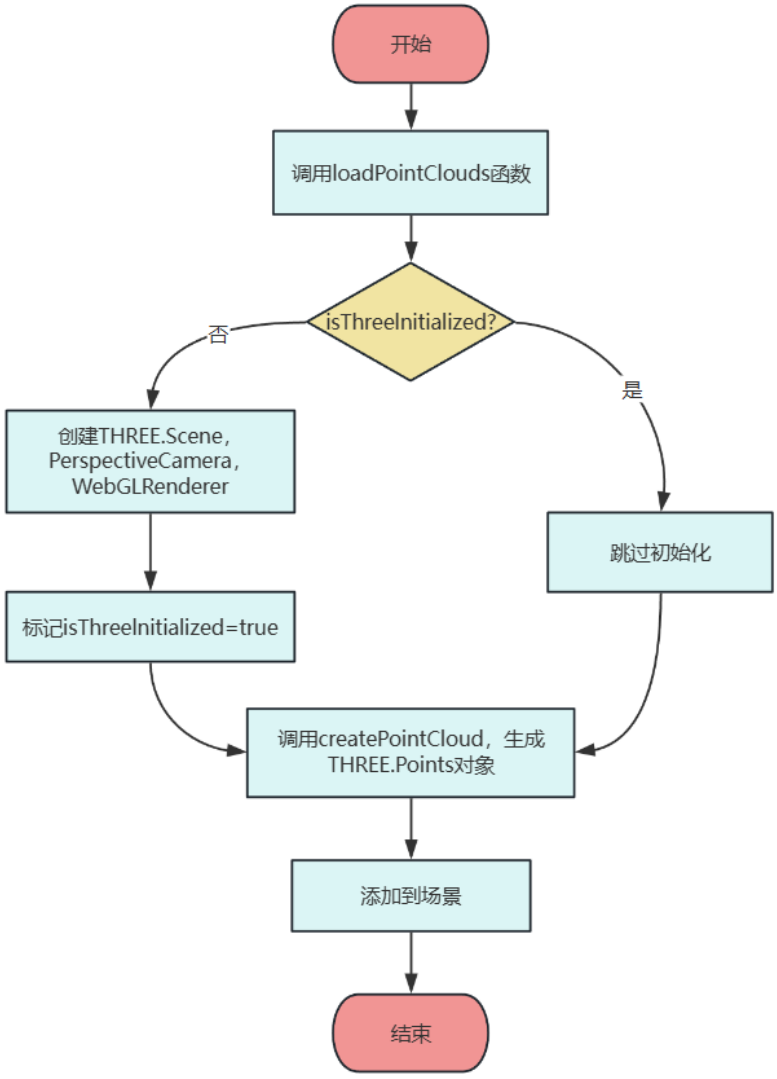


图 4.2　点云对象创建流程图

相机自适应调整：adjustCameraToFit()函数在点云加载后被调用，它计算源点云和目标点云的合并包围盒（THREE.Box3），然后根据包围盒的尺寸和中心动态调整相机的位置、朝向以及近远裁剪面，确保两个点云都能完整清晰地显示在视口中。

交互控制：使用 THREE.OrbitControls 控制器，允许用户通过鼠标拖拽、滚轮缩放等方式自由地观察三维场景。控制器在相机或场景发生变化时触发 change 事件，进而调用renderer.render(scene, camera)重新渲染。

动画循环：animate()函数通过 requestAnimationFrame 创建渲染循环，确保场景在需要时（如控制器操作后）能够平滑更新。

4.1.3　用户交互逻辑

文件上传与算法选择：HTML 中的<input type="file" id="file1">和<input type="file"

id="file2">用于选择源和目标点云文件。<select id="algorithms">用于选择配准算法。“开始配准”按钮的 onclick 事件绑定到 uploadFiles()函数。

配准过程启动：uploadFiles()函数首先获取用户选择的文件和算法。然后创建一个 FormData 对象，将文件和算法名称附加到其中。调用 fetch('/uploadfile/', ...)向后端 API 发送 POST 请求。

迭代过程控制：initSlider(maxIterations)函数根据后端返回的迭代总次数初始化滑块(<input type="range" id="iterationSlider">)的最大值。滑块的 input 事件会实时更新迭代次数标签 (iterationLabel)，并调用 updateTransformation(index)来显示对应迭代步骤的点云状态。

4.1.4 点云数据加载与变换更新

接收后端数据：在 uploadFiles()的.then(data => { ... })回调中，接收后端返回的 JSON 数据。

初始点云加载：调用 loadPointClouds(data)，该函数从 data.pcd1_points 和 data.pcd2_points 中获取原始点云坐标，并创建 Three.js 点云对象。同时，将 data.transformations（变换矩阵序列）存储在全局变量 transformations 中，并将源点云的原始顶点坐标扁平化后存储在 originalSourceVertices 中，作为后续变换的基准。

动态变换应用：updateTransformation(index)函数是过程模拟的核心。具体步骤如下：

1. 根据传入的index从transformations数组中获取对应的4x4变换矩阵。
2. 将该矩阵（NumPy数组在后端转为列表，前端再转为THREE.Matrix4）应用到originalSourceVertices中的每个原始顶点。通过调用vec. applyMatrix4(matrix)应用变换。
3. 设置positions.needsUpdate = true告知Three.js顶点数据已更改，需要在下次渲染时更新GPU缓冲。
4. 调用renderer.render(scene, camera)重新渲染场景。

4.1.5 性能指标可视化

表格展示：在 uploadFiles()的成功回调中，根据算法类型从 data.final 中提取最终性能指标（迭代次数、Fitness、RMSE、总耗时等），动态创建 HTML 内容并插入到.table-result 区域。

图表展示：renderProcessChart(logData)函数负责生成性能曲线图。根据当前选择的算法 (document.getElementById('algorithms').value)，从 logData (即后端返回的 data.process_log)中提取相应的指标序列。使用 new Chart(ctx, { ... })创建 Chart.js 图表实例，配置图表类型为 line，并设置 labels（X 轴数据，指代迭代次数）和 datasets（Y 轴数据，RMSE 曲线、Fitness 曲线等）。

4.1.6 与后端 API 的通信

与后端的通信使用浏览器原生的 fetch API 向后端/uploadfile/端点发送异步 POST 请求。请求体使用 FormData 对象封装文件和算法参数。为了处理 Django 的 CSRF 保护，在请求头中添加了 X-CSRFToken。这个 Token 通过 getCookie()函数从浏览器 cookie 中获取。通过.then(response => response.json())处理响应，将 JSON 字符串解析为 JavaScript 对象。通过.catch(error => { ... })处理网络错误或请求失败的情况。

4.2 后端实现

后端基于 Django 框架，使用 Python 编写，负责处理文件上传、执行点云配准算法、记录中间过程并返回结果等功能。主要的实现逻辑在 views.py 文件中实现。

4.2.1 Django 项目配置与视图函数

URL 路由配置：在 Django 项目的 urls.py 文件中配置 URL 模式，将/uploadfile/路径映射到 views.upload_file_select 视图函数。

视图函数 upload_file_select(request)：使用@csrf_exempt 装饰器以确保前端正确发送 CSRF token 并由 Django 中间件处理。处理 POST 请求。从 request.POST.get('algorithm')获取算法选择。从 request.FILES.get('file1')和 request.FILES.get('file2')获取上传的文件对象。

4.2.2 文件处理与点云加载

临时文件存储模块：上传的文件首先被保存在服务器的 temp 目录下，用于后续处理时，库能够通过文件路径读取所要处理的文件。文件名保持与上传时一致。

本系统支持两种点云格式，文件解析的方法分别如下：

STL 文件：调用自定义的 stl_to_pointcloud(stl_path, num_samples)函数。该函数使用 stl 库的 mesh.Mesh.from_file()读取 STL 文件，提取所有三角面片顶点，然后通过随机采样得到指定数量的点，最后将这些点转换为 o3d.geometry.PointCloud 对象。

PLY 文件：使用 Open3d 库的 o3d.io.read_point_cloud(path)直接读取 PLY 文件为 Open3D 点云对象。

数据提取：将加载的 Open3D 点云对象的点坐标转换为 NumPy 数组，再转换为 Python 列表，用来方便后续序列化为 JSON 并发送给前端。

4.2.3 ICP 算法封装与日志记录

参数初始化：设置 ICP 算法的初始变换矩阵为单位矩阵 (np.identity(4))，最大迭代次数，收敛阈值等。

单步迭代：核心是设置 ICPConvergenceCriteria(max_iteration=1)，使得 registration_icp 函数在每次调用时仅执行一次迭代。

迭代循环：在一个 for 循环中手动控制迭代过程：

1. 记录当前时间，调用registration_icp()执行单次迭代。
2. 从返回的reg_result中获取当前的RMSE (inlier_rmse)、Fitness (fitness) 和本次迭代计算得到的变换矩阵 (reg_result.transformation)。
3. 更新累积变换：将本次迭代的变换矩阵与之前的累积变换矩阵相乘，得到新的总变换矩阵，并保存。
4. 计算单次迭代耗时。
5. 将迭代次数、RMSE、Fitness、变换矩阵、耗时存入log_data字典的对应列表中。
6. 检查收敛条件（如变换矩阵变化量delta小于阈值），若满足则跳出循环。

关键代码如图 4.3 所示：

```
for iter in range(max_iterations):
    # 执行单次ICP迭代
    current_time = time.time()
    reg_result = registration_icp(
        pcd1, pcd2, threshold, transformation,
        *args: estimation, criteria=single_step_criteria)
    # 计算变换矩阵变化量（Frobenius范数）
    delta = np.linalg.norm(reg_result.transformation - prev_transformation)
    log_data['transformation_changes'].append(delta)
    # 更新变换矩阵
    transformation = reg_result.transformation
    prev_transformation = transformation.copy()
    # 记录数据
    once_time = time.time() - current_time
    log_data['iteration'].append(iter + 1)
    log_data['rmse'].append(reg_result.inlier_rmse if reg_result.inlier_rmse is not None else float('inf'))
    log_data['fitness'].append(reg_result.fitness)
    log_data['time_once'].append(once_time)
    log_data['transformations'].append(transformation.T.copy())  # 保存当前变换矩阵
    # 检查收敛
    if delta < delta_transformation_threshold:
        break
```

图 4. 3　ICP 算法手动控制迭代

最终结果：记录总耗时，并使用最终的变换矩阵对源点云进行变换，通过o3d.visualization.draw_geometries()在后端展示最终配准结果。

4. 2. 4　CPD 算法封装与日志记录

数据准备：将 Open3D 点云对象的点坐标转换为 NumPy 数组 (np.asarray(pcd.points))，因为 pycpd 库接收 NumPy 数组作为输入。

CPD 对象初始化：创建 RigidRegistration 对象，传入目标点云 X (通常是固定不动的) 和源点云 Y (需要被变换的)，以及最大迭代次数等参数。

回调函数 visualize_open3d：这是记录 CPD 中间过程的关键。该函数使用 functools.partial 将 log_data 和 reg 对象（CPD 注册器实例）预先绑定到回调函数中。该回调函数在 pycpd 的每次迭代结束时被调用，接收 iteration, error, X, Y_transformed (当前变换后的源点云)作为参数。在回调函数内部：

1. 计算单次迭代耗时。
2. 记录迭代次数、误差值、单次迭代耗时到log_data。
3. 从reg对象中获取当前的旋转矩阵reg.R和平移向量reg.t。
4. 构建4x4的齐次变换矩阵。
5. 将该变换矩阵存入log_data['transformations']。

回调函数关键代码如图 4.4 所示：

```
def visualize_open3d(iteration, error, X, Y, log_data, reg):
    current_time = time.time()
    # 计算本次迭代耗时（当前时间 - 上次记录时间）
    if iteration == 0:
        iter_duration = current_time - log_data['_last_time']
    else:
        iter_duration = current_time - log_data['_last_time']
    # 更新源点云
    source_pcd = o3d.geometry.PointCloud()
    source_pcd.points = o3d.utility.Vector3dVector(Y)

    log_data['iteration'].append(iteration)
    log_data['error'].append(error/100)
    log_data['time_once'].append(iter_duration*1000)
    # 更新最后记录时间
    log_data['_last_time'] = current_time
    # 获取当前变换矩阵（刚性变换）
    R = reg.R  # 旋转矩阵 (3x3)
    t = reg.t.reshape(3, 1)  # 平移向量 (3x1)

    # 构建4x4齐次变换矩阵
    transform_matrix = np.eye(4)
    transform_matrix[:3, :3] = R
    transform_matrix[:3, [3]] = t
    log_data['transformations'].append(transform_matrix.T.copy())
```

图 4.4　CPD 实现中回调函数核心代码

执行配准：调用 reg.register(callback)启动 CPD 配准过程。

最终结果：记录总耗时，获取最终变换后的源点云 reg.transform_point_cloud(Y)，并在后端通过 o3d.visualization.draw_geometries()展示最终配准结果。

4.2.5　API 响应数据构建

在 upload_file_select 视图函数中，根据选择的算法，将 final_result（包含最终性能指标）、process_log（包含每次迭代的详细日志）、transformations 列表（每次迭代的变换矩阵）以及原始点云坐标 pcd1_points 和 pcd2_points 封装到一个 Python 字典 response_data 中。

response_data 的数据结构如图 4.5 所示：

```
response_data = {
    'final': {
        'fitness': final_result.fitness,
        'rmse': final_result.inlier_rmse,
        'transformation': final_result.transformation.tolist(),
        'iterations': len(process_log['iteration']),
        'time_count': time_count
    },
    'transformations': [
        t.tolist() for t in process_log['transformations']  # 转换numpy矩阵为嵌套列表
    ],
    'process_log': {
        'iteration': process_log['iteration'],
        'rmse': process_log['rmse'],
        'time_once': process_log['time_once'],
        'fitness': process_log['fitness']
    },  # 过程日志
    'pcd1_points': pcd1_points,  # 原始点云1坐标
    'pcd2_points': pcd2_points,  # 原始点云2坐标
}
```

图 4.5　响应数据数据结构

使用 JsonResponse(response_data)将该数据结构序列化为 JSON 字符串，并作为 HTTP 响应返回给前端。

4.3　本章小结

本章详细介绍了系统的实现细节，前后端主要功能的代码实现逻辑。这些功能的实现确保了系统能够按照设计目标稳定运行，并为下一章的系统测试和结果分析提供了功能基础。

第 5 章　系统测试与结果分析

本章主要对基于 WEB 的点云配准算法过程模拟及性能评测系统进行系统测试。主要介绍展示测试用例在不同场景下的测试结果，包括点云配准的效果、中间过程的可视化模拟以及性能指标的分析。通过这些测试与分析，验证系统的可行性、有效性和实用价值。

5.1　测试用例设计

为了全面评估系统的功能和性能，本系统设计了以下测试用例。

5.1.1　功能测试用例

功能测试用例如表 2 所示：

表 2　测试用例设计表

测试模块	测试项	测试步骤	预期结果
文件上传与算法选择	PLY 格式文件上传	1. 选择源/目标点云为. ply 格式文件。2. 点击“开始配准”。	文件成功上传，后端能正确解析，前端能正确显示初始点云。
	STL 格式文件上传	1. 选择源/目标点云为. stl 格式文件。2. 点击“开始配准”。	文件成功上传，后端能将 STL 转换为点云并正确解析，前端能正确显示初始点云。
	算法选择	1. 在下拉菜单中选择算法。2. 上传有效点云并配准	系统能根据选择调用对应的后端算法，并正确展示性能指标。
ICP 算法模拟	ICP 算法执行与过程模拟	1. 选择 ICP 算法。2. 上传一对点云。3. 点击“开始配准”。 4. 观察三维可视化区。	源点云随着滑块拖动逐步向目标点云对齐，迭代次数标签正确显示。
	ICP 性能指标展示	完成 ICP 配准后查看“配准结果”表格和“配准指标”图表。	表格和图表正确显示配准结果和配准指标。
CPD 算法模拟	CPD 算法执行与过程模拟	1. 选择 CPD 算法。 2. 上传一对点云。3. 点击“开始配准”。4. 观察三维可视化区。	源点云随着滑块拖动逐步向目标点云对齐，迭代次数标签正确显示。
	CPD 性能指标展示	完成 CPD 配准后查看“配准结果”表格和“配准指标”图表。	表格和图表正确显示配准结果和配准指标。
界面与交互测试	Three. js 场景交互	点云加载后，在可视化区域使用鼠标左键拖动、右键拖动、滚轮缩放。	场景能够流畅地进行旋转、平移、缩放操作。
	迭代滑块控制	配准完成后，拖动迭代滑块。	滑块响应灵敏，点云状态随滑块位置实时更新。

5.1.2 测试数据集

为了进行有效的测试，选用了以下几组点云数据：

Bunny 1(PLY 格式)： 经典的计算机图形学模型。用于测试 PLY 文件处理、ICP 算法在标准模型上的配准效果和性能。

Bunny 2(PLY 格式)：另一组 Bunny 点云数据，该组数据初始位置偏差较大。用于测试 ICP 和 CPD 算法在标准模型上的配准效果和性能差异。

Dragon(PLY 格式)：用于测试 PLY 文件处理、ICP 算法在标准模型上的配准效果和性能。

Monkey(PLY 格式)：用于测试 PLY 文件处理、CPD 算法在标准模型上的配准效果和性能。

胫骨数据(STL 格式)：用于测试 STL 文件处理、CPD 算法在标准模型上的配准效果和性能。

5.2 测试结果与分析

5.2.1 文件上传与算法选择基础功能测试

通过执行表 5-1 中的文件上传与选择相关的测试用例，系统均能按预期工作：

PLY, STL 格式的点云文件均能被后端成功解析并加载。stl_to_pointcloud 函数能有效地将 STL 网格转换为点云数据。算法选择功能正常，后端能根据选择调用 ICP 或 CPD 的相应处理逻辑。

前端 Three.js 场景能正确渲染不同来源的初始点云，并用不同颜色（源点云红色，目标点云绿色）加以区分，如图 5.1 和图 5.2 所示。

图 5.1　ply 格式文件正确处理

图 5.2　stl 格式文件正确处理

5.2.2　ICP 算法模拟与性能分析

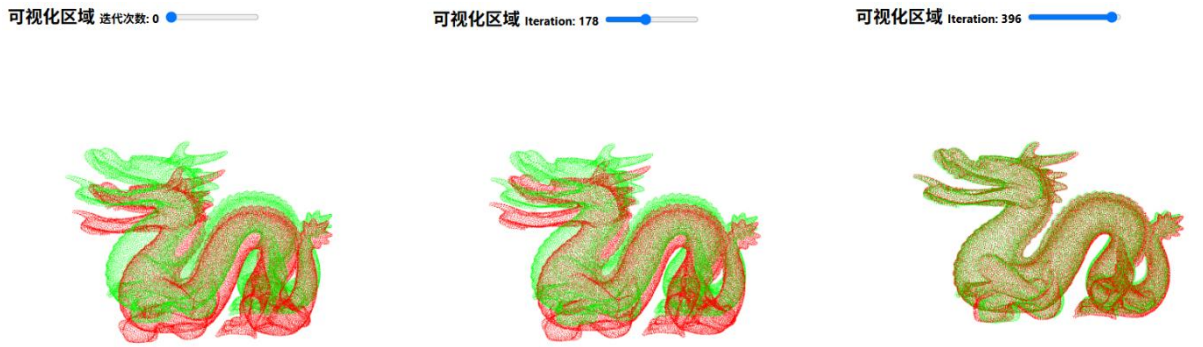
使用 Bunny 和 Dragon 两组点云对 ICP 算法进行模拟与性能分析。

配准过程模拟：启动 ICP 配准后，后端开始迭代计算。数据返回给前端后，用户可拖动迭代滑块，可以清晰地观察到红色的源点云如何一步步地旋转和平移，逐渐与绿色的目

标点云对齐。图 5.3 和图 5.4 分别展示了 Bunny 和 Dragon 两组数据在 ICP 配准过程中的几个关键迭代步骤的可视化效果。

a） 迭代初期　　b） 迭代中期　　c） 迭代后期

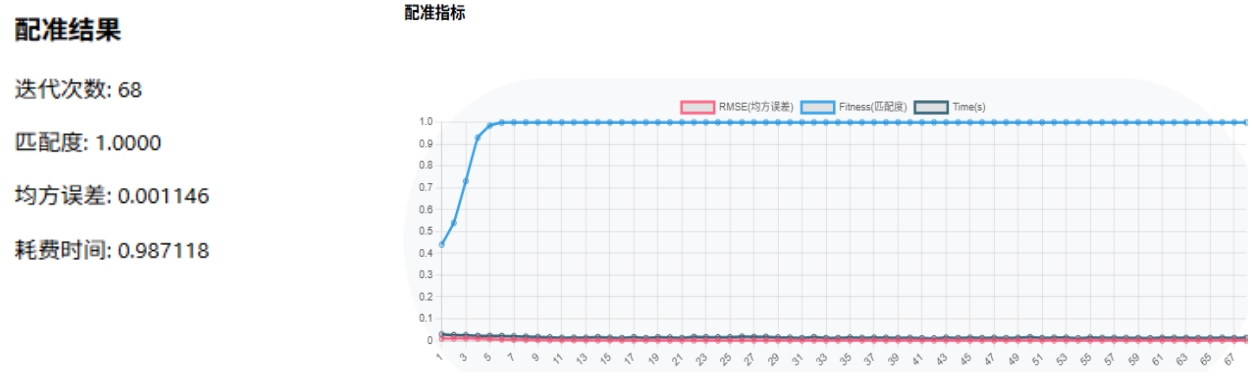
图 5. 3　bunny 迭代可视化



a） 迭代初期　　b） 迭代中期　　c） 迭代后期

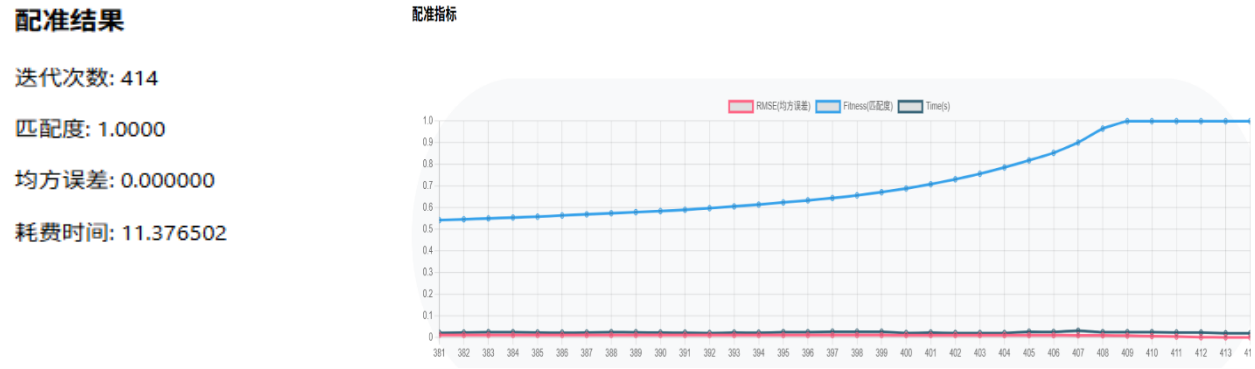
图 5. 4　dragon 迭代可视化

性能指标分析：配准完成后，控制面板左下角的“配准结果”表格显示了本次 ICP 配准的最终性能，如图 5.5(a)和 5.6（a）所示。下方的“配准指标”图表则展示了 RMSE、Fitness 和单次迭代耗时随迭代次数的变化曲线，如图 5-5(b)和 5.6（b）所示。



a） 配准结果　　b） 配准指标

图 5. 5　ICP 算法性能指标（bunny）



a） 配准结果　　b） 配准指标(最后 30 次)

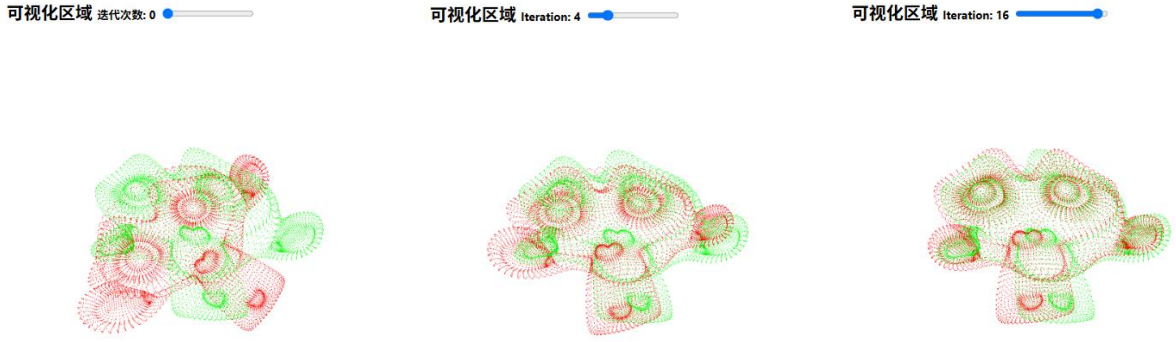
图 5.6　ICP 算法性能指标（dragon）

结果分析：从图 5.5(b)和 5.6(b)可以看出，ICP 算法在对数据的处理中，算法的 Fitness 上升迅速，表明算法收敛速度较快。最终的 Fitness 值接近 1，RMSE 值非常小，表明配准精度很高。单次迭代耗时相对稳定。这与 ICP 算法的特性相符。该模拟过程直观地展示了 ICP 的迭代优化特性。

5.2.3　CPD 算法模拟与性能分析

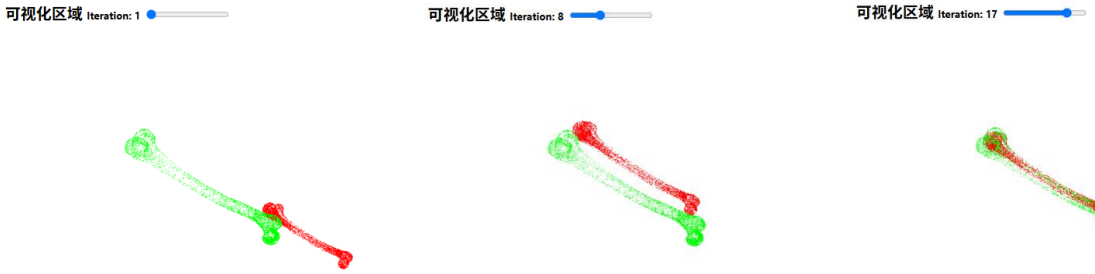
使用 Monkey 和胫骨两组点云对 CPD 算法进行模拟与性能分析。

配准过程模拟：选择 CPD 算法并启动配准。与 ICP 类似，用户可通过拖动迭代滑块，观察到源点云一步步向目标点云对齐的过程。图 5.7 和图 5.8 分别展示了 Monkey 和胫骨两组数据在 CPD 配准过程中的几个关键迭代步骤的可视化效果。



a） 迭代初期　　b） 迭代中期　　c） 迭代后期

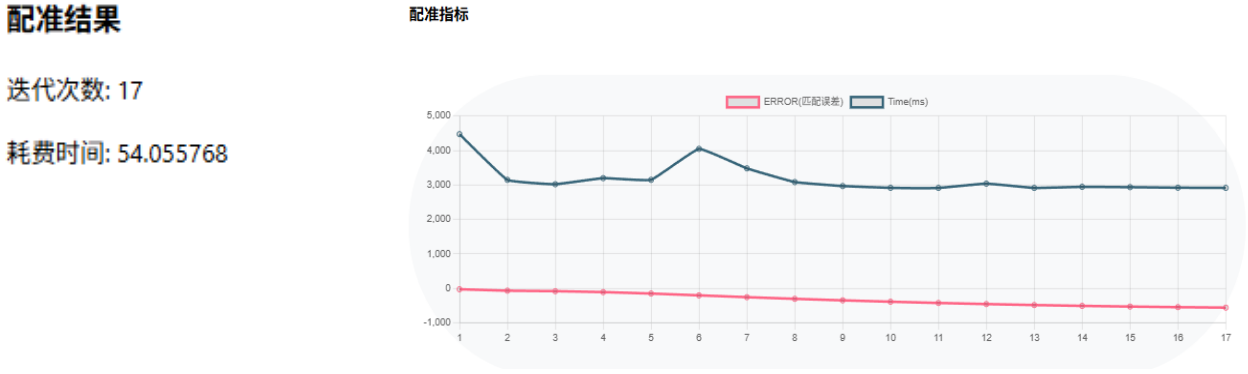
图 5.7　monkey 迭代可视化



a） 迭代初期　　b） 迭代中期　　c） 迭代后期

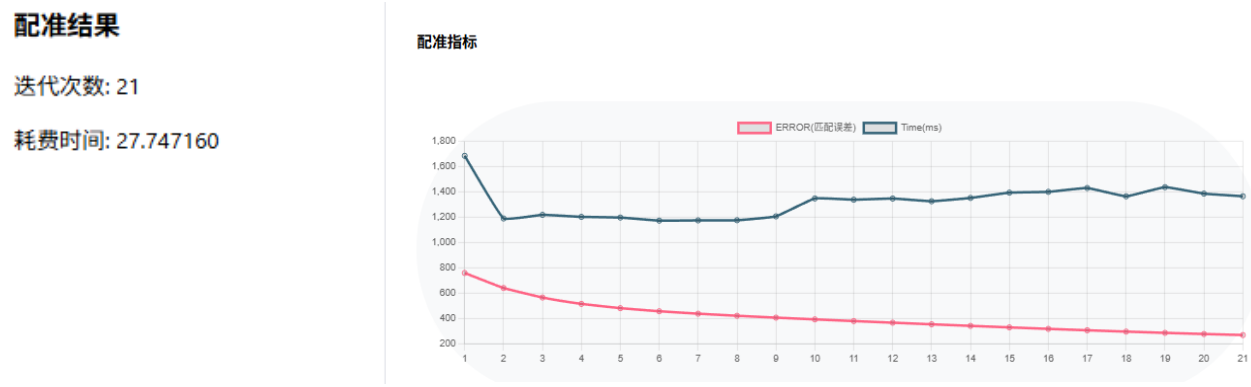
图 5. 8　胫骨迭代可视化

性能指标分析：CPD 配准完成后，“配准结果”表格如图 5.9(a)和 5.10（a）所示。“配准指标”图表如图 5.9(b)和 5.10(b)所示，展示了 Error 和单次迭代耗时随迭代次数的变化。



a） 配准结果　　b） 配准指标

图 5. 9　CPD 算法性能指标（monkey）



a） 配准结果　　b） 配准指标

图 5. 10　CPD 算法性能指标（胫骨）

结果分析：从图 5.9(b)和 5.10(b)可以看出，CPD 算法的 Error 值随着迭代次数的增加而逐渐减小，并最终趋于稳定，表明算法收敛。CPD 的单次迭代耗时可能稍高，总迭代次数可能不同，总耗时也可能有所差异。这反映了 CPD 算法基于概率模型进行点集匹配的计

算特性。

5.2.4 算法对比分析

使用 Bunny2 这组点云，分别使用 ICP 和 CPD 算法对该数据集进行配准，并观察其模拟过程和性能指标。

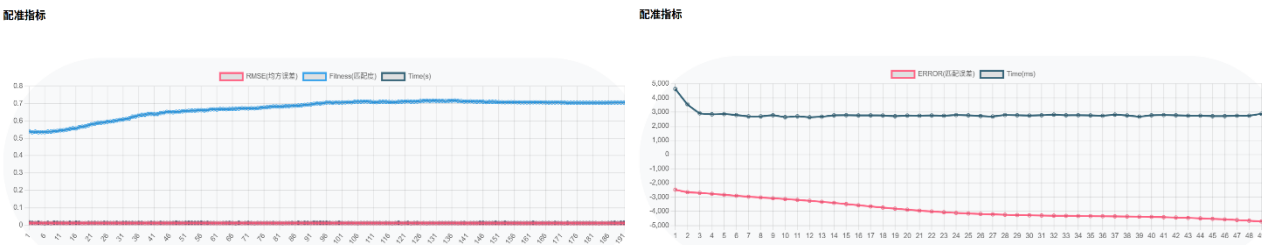
ICP 表现：对于初始偏差较大的情况，标准 ICP 可能容易陷入局部最优，导致配准失败或精度不高。过程模拟可以直观地看到 ICP 是如何“卡住”的。

CPD 表现：CPD 由于其概率模型和对点云整体分布的考虑，在处理初始偏差较大或有部分缺失和噪声的情况下，通常表现出更好的鲁棒性。其 Error 曲线可能仍然能够收敛到一个较低的水平，视觉上的配准效果可能优于 ICP。

ICP 与 CPD 算法对比测试结果如图 5.11 和 5.12 所示：

a） ICP 算法配准前后对比　　b） CPD 算法配准前后对比

图 5.11 ICP 与 CPD 算法配准对比



a） ICP 算法配准前后对比　　b） CPD 算法配准前后对比

图 5.12 ICP 与 CPD 算法性能对比

结果分析：通过对比测试，系统能够清晰地展示不同算法在不同场景下的特性和优劣。例如，在初始位姿较好、点云质量较高时，ICP 可能更快更精确；而在具有挑战性的场景

下，CPD 可能更鲁棒。这种对比分析对于用户理解算法适用性、选择合适的配准策略具有重要参考价值。

5.2.5 系统测试结果总结

通过上述测试与分析，对本系统功能测试结果总结如下：

1. 直观展示算法过程：通过三维可视化和迭代滑块，用户不再仅仅看到最终结果，而是能够逐步地观察点云配准的每一步是如何进行的，这对于教学和初学者理解算法原理非常有帮助。
2. 辅助算法选择与参数调试：用户可以在同一平台上，使用相同的数据，快速比较不同算法（目前是ICP和CPD）的配准效果、收敛速度和计算开销。性能曲线图为分析算法行为提供了量化依据。虽然本系统未提供参数调整功能，但其展示的性能指标变化过程可以启发用户思考参数对结果的影响。
3. 错误排查与分析：当配准效果不佳时，通过观察中间过程，可以初步判断问题所在。
4. 便捷的Web访问方式：无需安装专业软件，用户通过浏览器即可使用，方便快捷。

5.3 本章小结

本章对所设计的点云配准算法过程模拟及性能评测系统进行了详细的测试。功能测试结果表明，系统各模块均能按设计要求正常工作，支持多种点云文件格式，能够正确执行 ICP 和 CPD 配准算法，并准确记录和展示中间过程及性能指标。通过对不同测试场景和数据集的分析，验证了系统在算法模拟、性能可视化和辅助用户理解算法特性方面的有效性和实用价值。测试结果也为下一章的总结与展望提供了依据。

第 6 章　总结与展望

本章对基于 WEB 的点云配准算法过程模拟及性能评测系统进行总结，概述了系统实现的主要功能，分析了当前系统存在的不足与局限性，并对未来系统的进一步完善提出了展望。

6.1　工作总结

本毕业设计设计并实现了一个基于 WEB 的点云配准算法过程模拟及性能评测系统。主要完成的工作成果如下：

1. 系统设计与实现：

系统采用前后端分离的 B/S 架构，前端利用 HTML, CSS, JavaScript 以及 Three.js 和 Chart.js 库构建了用户友好的交互界面、三维点云可视化模块和性能指标展示模块。

后端基于 Python 的 Django 框架，实现了文件上传与解析（支持 PLY, STL 格式）、点云配准算法的调用（集成了 Open3D 实现的 ICP 算法和 pycpd 库实现的 CPD 算法）、中间过程数据的记录与提取以及 API 服务。

实现了用户上传源点云和目标点云，选择配准算法，系统执行配准并返回详细的过程数据和最终性能指标的功能。

2. 核心功能——过程模拟与可视化：

通过前端 Three.js 技术，实现了对点云配准迭代过程的动态三维可视化。用户可以通过滑块控制，直观地观察源点云在每次迭代后如何逐步向目标点云对齐，增强了对算法行为的理解。

成功捕获并记录了 ICP 算法每次迭代的变换矩阵、RMSE、Fitness 和耗时，以及 CPD 算法每次迭代的变换参数、Error 和耗时。

3. 核心功能——性能评测与展示：

系统能够提取并展示配准算法的最终性能指标，如总迭代次数、最终精度（RMSE/Fitness/Error）、总耗时等，方便用户对算法的整体表现进行评估。

利用 Chart.js 库，将迭代过程中的关键性能指标（RMSE, Fitness, Error, 单次耗时）以折线图的形式进行可视化，清晰地展示了算法的收敛特性和效率变化趋势。

6.2　系统不足与局限性

尽管本系统基本实现了预期功能，但在实际应用和进一步发展中仍存在一些不足和有待改进之处：

1. 支持的算法有限：

目前系统仅集成了 ICP 和 CPD（刚性）两种较为基础的配准算法。点云配准领域还有许多其他重要的算法（如 NDT, FGR, Go-ICP, 以及 CPD 的非刚性变体、BCPD 等），当前系统尚未覆盖。

2. 缺乏参数调整功能：

用户无法在前端调整配准算法的关键参数（如 ICP 的最大迭代次数、收敛阈值，CPD 的正则化参数等），这限制了用户对算法行为进行更深入探索和优化的能力。

3. 大规模点云处理性能瓶颈：

对于非常大规模的点云数据，当前基于 HTTP 同步请求的处理方式以及前端 Three.js 的渲染性能可能会遇到瓶颈，导致处理时间过长或浏览器卡顿。

4. 数据持久化与用户管理缺乏：

系统目前不保存用户的操作历史或上传的数据，也无用户账户管理功能，每次使用都需要重新上传数据，不利于项目的持续性工作和协作。

6.3 未来展望

针对上述不足，结合点云技术和 Web 应用的发展趋势，对本系统未来的改进和扩展方向提出以下展望。

1. 扩展配准算法库：

逐步集成更多先进和经典的配准算法，如正态分布变换（NDT）、基于分支定界的全局最优 ICP（Go-ICP）、非刚性 CPD 以及 BCPD 等。

2. 增强用户交互与参数控制:

在前端界面为每种算法提供可配置的关键参数输入框，允许用户自定义参数并实时观察其对配准过程和结果的影响。

3. 优化系统性能：

对于大规模点云处理，后端可以引入异步处理机制等，避免 HTTP 请求长时间阻塞，并通过 WebSocket 向前端实时推送进度和中间结果，以优化系统性能。

4. 引入用户管理与数据持久化：

增加用户注册登录功能，允许用户保存上传的点云数据、配准项目、参数配置和历史结果，方便用户回看和分享。后端使用数据库系统来替代临时文件系统，以支持更复杂的数据管理需求。

结　论

本次毕业设计旨在设计与实现一个基于 WEB 的点云配准算法过程模拟及性能评测系统，以应对在点云处理领域，尤其是算法学习和算法选型过程中，对算法内部迭代机制缺乏直观理解以及对不同算法性能难以便捷比较的挑战。通过综合运用 Web 全栈开发技术、三维可视化技术以及主流点云处理库，本研究成功构建了一个具有创新性和实用价值的在线工具。

本系统采用前后端分离的 B/S 架构。前端基于 HTML、CSS 和 JavaScript，并核心利用 Three.js 库实现了三维点云的动态渲染和配准过程的交互式模拟，通过 Chart.js 库将复杂的性能指标数据转化为易于理解的图表。后端则依托 Python 语言和 Django 框架，集成了 Open3D 和 pycpd 等库，实现了对 ICP 和 CPD（刚性）配准算法的封装、中间迭代数据的精确捕获与记录，以及通过 API 向前端提供服务。系统支持 PLY、STL 等多种常见点云格式文件的上传与处理。测试结果表明，系统各功能模块运行稳定，能够准确模拟所选算法的配准过程，并有效展示其性能指标。该系统对于计算机视觉、医学影像处理等相关领域的学生和研究人员在理解、学习和选择点云配准算法方面具有积极的辅助作用。

尽管本系统已取得初步成果，但仍存在一些可完善之处，如支持更多算法类型、引入用户参数调整功能、优化大规模数据处理性能以及增加点云预处理模块等。这些不足也为未来的研究和系统迭代指明了方向。

综上所述，本论文完成的基于 WEB 的点云配准算法过程模拟及性能评测系统，成功地将复杂的点云算法处理与直观的 Web 可视化技术相结合，为点云配准领域提供了一个有益的工具，具有良好的应用前景和进一步发展的潜力。

参考文献

[1] 李建微, 占家旺. 三维点云配准方法研究进展[J]. 中国图象图形学报, 2022, 27(02):349-367.

[2] 艾阳. 三维点云及图像-点云配准方法研究[D]. 吉林大学, 2024. DOI:10. 27162/d. cnki. gjlin. 2024. 006761.

[3] Chen Y, Medioni G. Object modelling by registration of multiple range images[J]. Image and vision computing, 1992, 10(3): 145-155.

[4] Bergevin R, Soucy M, Gagnon H, et al. Towards a general multi-view registration technique[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1996, 18(5): 540-547.

[5] Segal A, Haehnel D, Thrun S. Generalized-icp[C]//Robotics: science and systems. 2009, 2(4): 435.

[6] Myronenko A, Song X. Point set registration: Coherent point drift[J]. IEEE transactions on pattern analysis and machine intelligence, 2010, 32(12): 2262-2275.

[7] 李文静.基于倾斜摄影和点云数据的区域地震反应可视化研究[J].四川建材,2024,50(03):35-36+44.

[8] 李朋超,王金涛,宋吉来.基于PCL的3D点云视觉数据预处理[J].计算机应用,2019,39(S2):227-230.

[9] Besl P J, McKay N D. Method for registration of 3-D shapes[C]//Sensor fusion IV: control paradigms and data structures. Spie, 1992, 1611: 586-606.

[10] 白昌盛.基于Django的Python Web开发[J].信息与电脑(理论版),2019,31(24):37-40.

[11] Zhou Q Y, Park J, Koltun V. Open3D: A modern library for 3D data processing[J]. arXiv preprint arXiv:1801.09847, 2018.

[12] 王镜宇,郭际香.基于ITK医学图像配准[J].现代计算机(专业版),2017,(05):64-66+76.

[13] 金钊.基于VTK的医学图像三维重建关键技术及交互的研究[D].山东大学,2018.

[14] 齐含,刘元盛,宋庆鹏,等.基于深度学习的点云配准方法综述[C]//中国计算机用户协会网络应用分会.中国计算机用户协会网络应用分会2022年第二十六届网络新技术与应用年会论文集.北京联合大学智慧城市学院;北京市信息服务工程重点实验室;,2022:63-67.DOI:10.26914/c.cnkihy.2022.049250.

[15] 贾庆元.面向协同感知的点云配准方法研究与设计[D].北京邮电大学,2024.DOI:10.26969/d.cnki.gbydu.2024.002424.

[16] 周春艳,李勇,邹峥嵘.三维点云ICP算法改进研究[J].计算机技术与发展,2011,21(08):75-77+81.

[17] Duncan J S, Ayache N. Medical image analysis: Progress over two decades and the challenges ahead[J]. IEEE transactions on pattern analysis and machine intelligence, 2002, 22(1): 85-106.

[18] 戴静兰,陈志杨,叶修梓.ICP算法在点云配准中的应用[J].中国图象图形学报,2007,(03):517-521.

[19] Dirksen J. Learn Three. js: Programming 3D animations and visualizations for the web with HTML5 and WebGL[M]. Packt Publishing Ltd, 2018.

[20] Hirose O. A Bayesian formulation of coherent point drift[J]. IEEE transactions on patter n analysis and machine intelligence, 2020, 43(7): 2269-2286.

致　　谢

时光荏苒，岁月如梭，此刻毕业设计也将就此落幕。回首这段充实而难忘的旅程，我的心中充满了感激。本毕业设计论文的顺利完成，离不开许多人的悉心指导、无私帮助和默默支持，在此，我谨向他们致以最诚挚的谢意。

首先，我要由衷感谢我的指导教师王亚刚老师。从论文选题、开题报告的撰写，到系统设计、系统实现，再到后面论文的反复修改，王老师都倾注了大量心血。他在我遇到困难时给予我精准的指导和点拨，他严谨的治学态度、深厚的专业素养和诲人不倦的师者风范，令我受益匪浅。

其次，感谢母校多年来提供的良好教育资源和学习环境，正是老师们的辛勤付出和专业教导，为我打下了坚实的理论基础和专业知识，使我能够顺利开展本次毕业设计的研究工作。老师们在课堂上的谆谆教导和课后的答疑解惑，都让我铭记在心。

此外，我要感谢我的家人和朋友们，他们是我最坚实的后盾，是他们的理解、支持和鼓励为我提供前行的动力和信心。

最后，再次向所有关心、帮助和支持我的人们表示最衷心的感谢！这段毕业设计的经历不仅提升了我的专业技能，更锻炼了我的独立思考和解决问题的能力，将是我人生中一笔宝贵的财富。我将带着这份收获和感恩，在未来的道路上继续努力前行。