



天津中德应用技术大学
Tianjin Sino-German University of Applied Sciences

本科生毕业设计

轮式机器人小车激光雷达的信号采集和处理
Signal Acquisition and Processing of Lidar
for Wheeled Robot Vehicle

姓 名	朱明浩
学 院	汽车与轨道交通学院
专 业	汽车服务工程
指导教师	汪磊
职 称	副教授
完成时间	2023 年 5 月 20 日



天津中德应用技术大学
Tianjin Sino-German University of Applied Sciences

本科生毕业设计

轮式机器人小车激光雷达的信号采集和处理
**Signal Acquisition and Processing of Lidar
for Wheeled Robot Vehicle**

姓 名	朱明浩
学 院	汽车与轨道交通学院
专 业	汽车服务工程
指导教师	汪磊
职 称	副教授
完成时间	2023 年 5 月 20 日

附件一：

天津中德应用技术大学 本科生毕业设计（论文）选题申报表

学 院	汽轨学院		申 报 人	姓 名	汪磊		
专 业	汽车服务工程			技术职务	正高	副高	中级
					√		
题目名称	轮式机器人小车激光雷达的信号采集和处理						
题目类型	科学技术	题目来源	创新竞赛				
课题来源、背景及意义	<p>激光雷达采用 TOF 方案，能够对周围环境进行 360° 二维扫描探测，主要应用于室内服务机器人、AGV、清扫消杀机器人、无人机等精确定位和避障的应用场合。激光雷达信号感知是自动驾驶传感器感知环节的重要内容，采用 ROS 系统对轮式机器人小车的多线激光雷达信号进行采集、处理及优化配准，对于学生进一步理解多线雷达工作原理、点云生成、数据提取和处理，具有良好的理论研究意义和实用价值。本课题来自于创新竞赛项目，企业提供部分技术指导。</p>						
任务及要求	<p>本课题拟基于 ROS 系统采集装备有激光雷达的轮式小车的周围环境信息，然后提取相关障碍物的点云信息，再通过数据预处理（坐标转换，去噪声等）、聚类（根据点云距离或反射强度）以提取信息特征，期望该课题可实现基于特征点信息提出改进的 ICP 配准算法，提高激光雷达点云信息的配准精度。</p>						
工作条件	UBUNTU 操作系统、ROS 系统、轮式机器人智能车（装配激光雷达）、UAV 实车环境感知模块。						
知识与能力要求	学生需掌握车用激光雷达的基本理论知识，具备检索和分析相关科学文献的能力，熟练掌握 ROS 系统基本使用方法和基本数据分析方法。						
系（教研室）审查意见：							
负责人(签名)： <u>吕冬慧</u> 2022 年 12 月 1 日							

附件四：



天津中德应用技术大学

Tianjin Sino-German University of Applied Sciences

毕业设计（论文）任务书

题 目：轮式机器人小车激光雷达的信号采集和处理

学 院：汽车与轨道交通学院

专 业：汽车服务工程

学生姓名：朱明浩

学 号：19424040214

起止日期：2022年12月1日 - 2023年5月20日

指导教师：汪磊 副教授

任务书下达日期： 2022年12月15日

任务书填写要求

1. 毕业设计（论文）任务书由指导教师根据各课题的具体情况填写，经专业负责人审查签字后生效。此任务书应在毕业设计（论文）开始一周内填好并发给学生；
2. 任务书内填写的内容，必须和学生毕业设计（论文）完成的情况相一致；
3. 任务书内有关“学院”、“专业”等名称的填写，应写中文全称，不能写数字代码。学生的“学号”要写全号（如：16014010101）；
4. 有关年月日等日期的填写，应当按照国标 GB/T 7408—94《数据元和交换格式、信息交换、日期和时间表示法》规定的要求，一律用阿拉伯数字书写。如“2004年3月21日”或“2004-03-21”。
5. 本毕业设计（论文）课题成果的要求，内容要具体化和数量化。如“毕业设计（论文）一套；A0号装配图纸1张；A2号电气控制原理图纸2张；实物样机1台；产品2件”等。

毕业设计(论文)任务书

1. 毕业设计(论文)课题背景及意义

激光雷达采用 TOF 方案,能够对周围环境进行 360° 二维扫描探测,主要应用于室内服务机器人、AGV、清扫消杀机器人、无人机等精确定位和避障的应用场合。激光雷达信号感知是自动驾驶传感器感知环节的重要内容,采用 ROS 系统对轮式机器人小车的多线激光雷达信号进行采集、处理及优化配准,对于学生进一步理解多线雷达工作原理、点云生成、数据提取和处理,具有良好的理论研究意义和实用价值。本课题来自于创新竞赛项目,企业提供部分技术指导。

2. 毕业设计(论文)课题任务的内容和要求

本课题拟基于 ROS 系统采集装备有激光雷达的轮式小车的周围环境信息,然后提取相关障碍物的点云信息,再通过数据预处理(坐标转换,去噪声等)、聚类(根据点云距离或反射强度)以提取信息特征,期望该课题可实现基于特征点信息提出改进的 ICP 配准算法,提高激光雷达点云信息的配准精度。

同时,要求论文逻辑清晰、格式规范、语言流畅、查阅文献数量充足且较具有广泛性和先进性、工作量饱满。

3. 毕业设计(论文)课题成果(包括毕业设计论文、图表、实物样品等):

毕业论文 1 篇。

4. 推荐参考资料:

- [1] 党亚南,田照星,郭利强. 车载激光雷达点云数据处理关键技术[J]. 太原: 中北大学 2022.
- [2] 余莹洁. 车载激光雷达的主要技术分支及发展趋势[J]. 科研信息化技术与应用. 2018.
- [3] 张小梅. 车载激光雷达测距方法及精度分析[D]. 成都: 电子科技大学 2019.
- [4] 刘尚富,胡辉. 国外激光雷达的发展趋势[J]舰船电子工程, 2017
- [5] 罗骞,金琦珺. 智能汽车激光雷达感知技术现状与发展分析[J]. 科技与创新. 2020.

- [6]B Davis. Innovative Technology Workshop on 3D LIDAR[R]. University of Minnesota: Center for Transportation Studies, 2016.
- [7]吴超. 光波导相控阵激光雷达接收系统设计与实验[D]. 西安: 西安电子科技大学, 2016.
- [8]张大印. 自动驾驶车载激光雷达关键技术研究[D]. 成都: 电子科技大学, 2019.
- [9]J Zeisler, H G Maas. Analysis of The Performance of A Laser Scanner For Predictive Automotive Application[J]. Isprs Annals of Photogrammetry Remote Sensing & Spatial Informa, 2015.
- [10]应欢. 低成本车载激光雷达系统设计[D]. 合肥: 中国科学技术大学, 2014.
- [11]董士军. 车载激光雷达系统设计与实验研究[D]. 南京: 南京理工大学, 2017.
- [12]戴永江. 激光雷达原理[M]. 国防工业出版社, 2002.
- [13]张大印. 自动驾驶车载激光雷达关键技术研究[D]. 成都: 电子科技大学, 2019.
- [14]钟敏. 基于视觉与激光雷达融合的移动机器人自主导航研究[D]. 哈尔滨工业大学, 2021
- [15]李若鹏. 基于车载激光雷达的点云匹配定位[D]. 北京: 北京建筑大学, 2020.
- [16]杨曦. 基于 ROS 的自主移动机器人系统设计[D]. 西安: 长安大学, 2019.
- [17]郑为奇, 汪地. ROS 室内移动机器人避障研究[J]. 上海大学上海市智能制造及机器人重点实验室, 2022.

所在专业审查意见:

同意

负责人: 胡顺堂

2023 年 4 月 1 日

附件五：



天津中德应用技术大学
Tianjin Sino-German University of Applied Sciences

本科生毕业设计（论文）开题报告

题 目：轮式机器人小车激光雷达的信号优化采集和处理

学 院：汽车与轨道交通学院

专 业：汽车服务工程

学生姓名：朱明浩

学 号：19424040214

起止日期：2022年12月1日~2023年6月4日

指导教师：汪磊

开题日期：2023年1月8日

一、开题报告内容（课题的目的意义、与本课题有关的国内外研究（应用）情况及发展趋势、课题主要研究内容、参考文献等）

（一）课题的目的意义

1.研究目的

激光雷达收集的每个点云数据都代表被测物体实际位置生成的三维点云坐标，点云数据的误差相对较小，保证了后续点云数据处理的准确性和可靠性^[1]。作为迅速获得三维点云数据的高效装备，激光雷达能准确迅速地显示被测物体的形状和变化，但三维激光扫描仪可以生成大量测量物体的表面数据信息，这些点云数据可能会有一些噪音，从而导致点云数据混杂，信息数据重复等问题^[2]。为了保证重建目标的高精度和高保真度，点云数据必须在表面重建之前进行处理，点云滤波和点云数据配准是激光雷达点云数据处理过程中最关键的步骤，它直接影响到后续模型重建的精度。因此，研究和探索激光雷达的点云数据处理技术势在必行^[3]。

2.研究意义

用于用于自动驾驶、车路协同、AGV、无人机等的激光雷达系统通常被归类为测距激光雷达，实现最佳的测距精度取决于信号检测方案和随后采用的信号处理方法的有效性。因此，在检查激光雷达回波时，利用有效的信号处理技术是至关重要的^[5]。测距激光雷达采用 TOF 方案，能够对周围环境进行 360° 二维扫描探测，主要应精确定位和避障等具体场景。激光雷达信号感知系统是自动驾驶传感器感知融合系统的关键组成部分，论文采用 ROS 系统对轮式机器人小车的激光雷达信号进行优化采集、处理及仿真实验，具有较好的理论研究意义，相关研究成果也必将具备良好的行业（自动驾驶、智能网联汽车、车路协同等）推广应用价值于进一步理解多线雷达工作原理、点云生成、数据提取和处理，具有重要的研究意义。

（二）国内外研究情况及发展趋势

1.国外研究概况

当前，国外发展车载激光雷达的企业主要有三类：第一类是激光雷达初创企业，以 Quanergy、Velodyne、IBEO、Innoviz Technologies 等为代表；第二类是车企，以 Waymo、福特、宝马、奔驰等为代表；第三类是汽车零部件企业，以博世、采埃孚、法雷奥等为代

表。为了更快地实现车载激光雷达的量产，车企、汽车零部件厂商与激光雷达企业之间进行跨界合作^[4]。

在自动驾驶车载激光雷达的商用方面，美国 Velodyne 公司生产的高性能激光雷达 HDL-64E 最远探测距离为 120m，测距精度高达 2cm，垂直方向和水平方向的角分辨率分别可以达到为 0.4° 和 0.08° ，在双接收模式下可以实现 220 万点/秒的数据点采集^[5]，可以满足 L3 自动驾驶等级的要求，实现有条件的自动驾驶。但是该激光雷达缺点是机械式扫描，体积大，而且价格昂贵，高达 7.5 万美元，他们最新推出的 128 线激光雷达可以最远探测到 300m 的目标，以及更高测量精度，但价格也更加昂贵，机械扫描结构同时具有稳定性和耐久度不足的缺点^[6]。Quanergy 公司提出的固态激光雷达 S3 能有效解决机械扫描结构带来的问题，利用光学相控阵原理来进行激光束扫描，该扫描方式与相控阵雷达类似，当我们改变发射阵列中发射单元的相位差时，激光的发射方向便发生变化，这样就可以通过调制的电信号来实现光束的指向扫描，在 150 米的探测距离下，可以对 120° 的水平角度和 10° ^[7]。2022 年，作为全球汽车激光雷达软硬件技术的领先者，Luminar 将在传感器领域取得突破，其最大探测距离将达到 600 米，在反射率低于 10% 时，对目标障碍物的检测和识别将达到 250 米^[8]。

虽然市场上推出了很多自动驾驶车载激光雷达，但是能通过严格的车规级验证并投入实用的很少^[9]。由法国汽车配件巨头法雷奥与德国激光雷达供应商 Ibeo 生产的 4 线 ScaLa 激光雷达便是其中之一，可以实现水平 145° 视场的距离探测，最远的探测距离为 150m，测距精度为 10cm^[10]。最新款的奥迪 A8 自动驾驶汽车搭载了该款激光雷达，是全球首个实现量产的 L3 级别自动驾驶的汽车，在某些环境条件下，可以完全交给自动化系统进行操控，目前可以实现最高车速 60km/h 的自动驾驶^[11]。

2.国内研究概况

我国在激光雷达技术方面的研究也逐步发展并取得了一定的成果。速腾聚创作为世界顶尖的激光雷达技术研发公司，在 2023 年 CES2023 展会上，速腾聚创推出了其第一款全固态激光雷达（RS-LiDAR-E1，简称为 E1），并在展会上正式亮相^[12]。禾赛科技在自主研发高线性数激光雷达方面已积累了足够的技术储备，我们认为，高性能、低成本、高可靠性的激光雷达将成为未来工业机器人及量产汽车 ADAS 领域的核心竞争力，而实现高性能、低成本、高可靠性的关键路径就是芯片化^[13]。华为的激光雷达将为长安和北汽提供服务，在 2020 广州车展之前，长安集团的总裁公开表示，将与华为和宁德时代联手，共同打造一

个高端的、智能化的汽车品牌。在长安“方舟”发布的新车型上，36个感应器将被配备，包括5个激光雷达^[14]。北汽ARCFOX BU负责人于2020年11月在“网络汽车乌镇夜谈”上表示，北汽与华为携手研发出了ARCFOX中最新的一款车型—HBT。这款新车将搭载96线激光雷达3个，毫米波雷达6个，摄像头12个，超声波13个，华为提供了352兆次/秒的计算能力^[15]。目前，车载激光雷达系统在多个领域具有显著优势，包括城市场景的三维重建和工业建模等^[16]。我国高度重视激光雷达系统的发展，许多研究机构都投入了大量的资源和人力。经过十多年的发展，中国的激光雷达技术已经取得了显著的进步，中国与国外的技术水平差距逐渐缩小^[17]。

（三）课题主要研究内容

本文主要内容安排如下：

第一章 绪论。介绍本课题的研究背景及意义，阐述激光雷达和点云数据处理的研究现状。主要叙述了激光雷达点云数据处理技术的研究目的与意义。

第二章 激光雷达和点云数据分析。详细的对激光雷达工作原理及点云数据处理进行说明，并介绍了激光雷达的工作原理原理，自此基础上介绍了激光雷达点云数据处理的基本流程。

第三章 激光雷达点云数据获取技术。先叙述如何进行点云数据采集，接着在此基础上描述了激光雷达点云数据的采集及数据特点，最后总结出一般情况下的激光雷达数据处理步骤。

第四章 点云数据获取。通过ROS系统中的Gazebo软件进行仿真实验，获取仿真建模的轮式机器人小车的激光雷达的点云数据。

第五章 点云信息处理。研究了点云的统计滤波方法，进行了离群点的去除；然后对点云配准算法进行了概述，对传统ICP点云配准算法进行了分析，提出传统点云配准的改进方法，并加以实验验证。

（四）参考文献

- [1]党亚南,田照星,郭利强.车载激光雷达点云数据处理关键技术[J].中北大学 2022.
- [2]余莹洁.车载激光雷达的主要技术分支及发展趋势[J].科研信息化技术与应用.2018.
- [3]张小梅.车载激光雷达测距方法及精度分析[D].电子科技大学 2019.
- [4]刘尚富,胡辉.国外激光雷达的发展趋势[J]舰船电子工程,2017

- [5] 罗骞, 金琦珺. 智能汽车激光雷达感知技术现状与发展分析[J]. 科技与创新. 2020.
- [6] B Davis. Innovative Technology Workshop on 3D LIDAR[R]. University of Minnesota: Center for Transportation Studies, 2016.
- [7] 吴超. 光波导相控阵激光雷达接收系统设计与实验[D]. 西安电子科技大学, 2016.
- [8] 张大印. 自动驾驶车载激光雷达关键技术研究[D]. 电子科技大学, 2019.
- [9] J Zeisler, H G Maas. Analysis of The Performance of A Laser Scanner For Predictive Automotive Application[J]. Isprs Annals of Photogrammetry Remote Sensing & Spatial Informa, 2015.
- [10] 应欢. 低成本车载激光雷达系统设计[D]. 中国科学技术大学, 2014.
- [11] 董士军. 车载激光雷达系统设计与实验研究[D]. 南京理工大学, 2017.
- [12] 戴永江. 激光雷达原理[M]. 国防工业出版社, 2002.
- [13] 张大印. 自动驾驶车载激光雷达关键技术研究[D]. 电子科技大学, 2019.
- [14] 钟敏. 基于视觉与激光雷达融合的移动机器人自主导航研究[D]. 哈尔滨工业大学, 2021
- [15] 李若鹏. 基于车载激光雷达的点云匹配定位[D]. 北京建筑大学, 2020.
- [16] 杨曦. 基于 ROS 的自主移动机器人系统设计[D]. 长安大学, 2019.
- [17] 郑为奇, 汪地. ROS 室内移动机器人避障研究[J]. 上海大学上海市智能制造及机器人重点实验室, 2022.

二、进度及预期结果

起止日期	主要内容	预期结果
2022. 12. 1- 2023. 1. 8	根据选题, 查阅文献资料, 研究激光雷达的原理和数据采集; 研究 ROS 编程基础。撰写开题报告, 完成答辩 PPT。	熟练使用软件。完成答辩 PPT, 通过导师审批并完成开题答辩。
2023. 1. 9- 2023. 3. 23	完成激光雷达 ROS 驱动和采集并进行数据预处理。	得到数据并完成预处理。
2023. 3. 24- 2023. 3. 31	进行毕业设计论文中期检查。	提交中期检查表。

2023. 4. 1- 2023. 5. 1	进行论文的撰写并查重。	完成毕业设计并通过查重。
2023. 5. 2- 2023. 5. 4	申请毕业答辩资格。	通过学院审核取得答辩资格。
2023. 5. 5- 2023. 5. 15	制作毕业答辩 PPT, 准备毕业答辩。	完成 PPT 并通过毕业答辩。
2023. 5. 16- 2023. 6. 4	进行毕业设计归档。	完成毕业设计归档。
完成课题的 现有条件	<p>(1) 通过多年的汽车服务工程专业的学习, 已经掌握了相关的理论知识, 并有了较为系统、全面的学习。</p> <p>(2) 掌握车用激光雷达的基本理论知识、ROS 基本使用方法和点云数据分析方法。</p> <p>(3) 为了完成本课题的研究, 本人利用图书馆、知网、百度等资源, 已经查阅了丰富的文献资料, 为本文的研究奠定了理论基础。</p>	
指导教师 意见	<p>同意开题</p> <p>指导教师: <u>汪磊</u> 2023 年 01 月 08 日</p>	
开题答辩 小组意见	<p>同意开题</p> <p>组 长: <u>汪磊</u> 2023 年 01 月 08 日</p>	

天津中德应用技术大学
本科生毕业设计（论文）的声明

本人郑重声明：所呈交的毕业设计（论文），是本人在指导教师指导下，进行研究工作所取得的成果。除文中已经注明引用的内容外，本毕业设计（论文）的研究成果不包含任何他人创作的、已公开发表或没有公开发表的作品内容。对本设计（论文）所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。本毕业设计（论文）原创性声明的法律责任由本人承担。

毕业设计（论文）作者签名：朱明浩

2023年5月20日

本人声明：该毕业设计（论文）是本人指导学生完成的研究成果，已经审阅过设计（论文）的全部内容，并能够保证题目、关键词、摘要部分中英文内容的一致性和准确性。

毕业设计（论文）指导教师签名：王磊

2023年5月20日

摘 要

激光雷达信号可使自动驾驶汽车在非接触条件下获取车外障碍物的高密度三维信息，以达到实时避障的目的。激光雷达检测信号以点云形式分布在空间中，研究并优化点云信号的采集与处理方法，对于提高自动驾驶过程中的传感器感知融合质量、避障精确性及行车安全性意义重大。本文以激光雷达采集到的点云数据为主要研究对象，面向点云数据采集和处理的关键技术开展了如下工作：

(1) 本文研究了激光雷达点云数据的获取方法，采用 ROS 系统建立一个带有 VLP-16 激光雷达的轮式机器人小车模型，使用 Gazebo 软件搭建仿真地图，通过指令使上述小车在地图中随机运动，并对激光雷达点云数据进行采集。

(2) 针对激光雷达扫描数据时所产生的噪声，本文基于统计滤波点云算法，使用 Statistical Outlier Removal 滤波器进行去除离群点操作，以便更清晰、全面的获取点云数据集。

(3) 当原始点云和目标点云之间的重叠区域较小时，迭代最近点 ICP 算法将出现配准精度减弱的现象。为解决上述问题，本文提出了基于特征点的 ICP 点云配准算法，以有效减少错误配准的发生几率。仿真实验证明，该算法可有效提升激光点云信息的配准精度。

关键词：激光雷达；点云数据；数据采集；点云滤波；ICP 配准

ABSTRACT

LIDAR signals enable autonomous vehicles to acquire high-density three-dimensional information of obstacles outside the vehicle under non-contact conditions for the purpose of real-time obstacle avoidance. LIDAR detection signals are distributed in space in the form of point clouds, and the study and optimisation of point cloud signal acquisition and processing methods are of great significance to improve the quality of sensor sensing and fusion, obstacle avoidance accuracy and driving safety in the process of autonomous driving. In this paper, we focus on the point cloud data collected by LiDAR and carry out the following work on the key technologies of point cloud data acquisition and processing:

(1) In this paper, the method of acquiring LIDAR point cloud data is studied. A wheeled robot cart model with VLP-16 LIDAR is built using the ROS system, a simulation map is constructed using Gazebo software, and the above cart is made to move randomly in the map through commands, and LIDAR point cloud data is collected.

(2) To address the noise generated when the LIDAR scans the data, this paper uses a Statistical Outlier Removal filter based on a statistical filtering point cloud algorithm to remove outliers in order to obtain a clearer and more comprehensive point cloud dataset.

(3) When the overlap area between the original point cloud and the target point cloud is small, the iterative nearest point ICP algorithm will have a weakened alignment accuracy. To solve the above problems, this paper proposes an ICP point cloud alignment algorithm based on feature points to effectively reduce the occurrence of misalignment. Simulation experiments demonstrate that the algorithm can effectively improve the alignment accuracy of laser point cloud information.

Key words: Laser radar; Point cloud data; Date acquisition; Point cloud filtering; ICP alignment

目 录

第一章 绪论	1
1.1 选题背景及研究的目的和意义	1
1.1.1 研究背景	1
1.1.2 研究意义	1
1.2 国内外研究现状	1
1.2.1 车载激光雷达系统的发展	2
1.2.2 点云滤波的研究现状	3
1.2.3 点云配准研究现状	4
1.3 本文研究内容及技术路线	5
第二章 激光雷达和点云数据分析	7
2.1 激光雷达的工作原理	7
2.2 点云数据处理	7
2.2.1 点云统计滤波	7
2.2.2 PCL 点云配准	8
2.3 PCL 点云库	9
2.4 本章小结	10
第三章 激光雷达点云数据获取	11
3.1 激光雷达点云数据的采集	11
3.2 激光雷达点云的相关参数和特点	11
3.3 激光雷达数据处理步骤	13
3.4 本章小结	13
第四章 轮式小车多线激光雷达点云数据采集	14
4.1 ROS 小车仿真模型搭建	14
4.2 试运行小车并收集数据	15
4.3 本章小结	18
第五章 轮式小车激光雷达点云数据处理	19
5.1 点云统计滤波	19
5.1.1 激光雷达扫描数据噪声来源	19

5.1.2 统计滤波算法	19
5.1.3 Statistical Outlier Removal 滤波器移除离群点	21
5.2 ICP 点云配准	22
5.2.1 点云配准的阶段	23
5.2.2 ICP 配准算法	23
5.3 基于特征点的 ICP 点云配准	26
5.4 本章小结	28
第六章 总结和展望	29
6.1 全文总结	29
6.2 未来展望	29
参考文献	30
致 谢	32
附 录	33

第一章 绪论

1.1 选题背景及研究的目的和意义

1.1.1 研究背景

激光雷达技术研究始于 20 世纪 60 年代,但直到 20 世纪 80 年代,研究人员才开始使用激光雷达获取三维点云数据^[1]。激光雷达技术是雷达技术与激光技术相结合发展的结果,与传统的雷达发射器发射微波不同,激光雷达主要是用探测器发射的脉冲信号照射所测目标,通过所测目标物的表面接收反射信号,可计算出目标物体与激光雷达的距离,通过结合方位角和俯仰角,可计算出被测目标的三维空间坐标。由于激光探头安装在一个均匀旋转的平面上进行不间断的扫描,可以获得 360° 范围的空间数据,即点云数据。根据激光雷达固定平台的不同,可以将激光雷达分为车载激光雷达、机载激光雷达和星载激光雷达。其中,机载激光雷达系统则用于防灾和地理地形测绘,而星载激光雷达用于捕捉全球气候变化现象。车载激光雷达和地面激光雷达系统可以获得高精度、高层次的目标三维数据,为地形测绘、古建筑和文物的修复以及物体的重建提供必要的信息。

与其他光学系统相比,激光雷达检测方法不易受到外部环境变化和其他因素的影响,因此从被测物体获得的信息更加准确。此外,激光雷达的特点^[2]是体积小、重量轻、功耗低,因此,全球众多学者开始研究与激光雷达点云数据处理相关的热点课题,激光雷达的数据处理技术在军事和民用领域都得到了应用。

1.1.2 研究意义

激光雷达收集的每个点云数据都代表被测物体实际位置生成的三维点云坐标,点云数据的误差相对较小,保证了后续点云数据处理的准确性和可靠性^[3]。作为迅速获得三维点云数据的高效装备,激光雷达能准确迅速地显示被测物体的形状和变化,但三维激光扫描仪可以生成大量测量物体的表面数据信息,这些点云数据可能会有一些噪音,从而导致点云数据混杂,信息数据重复等问题。为了保证重建目标的高精度和高保真度,点云数据必须在表面重建之前进行处理,点云滤波和点云数据配准是激光雷达点云数据处理过程中最关键的步骤,它直接影响到后续模型重建的精度。因此,研究和探索激光雷达的点云数据处理技术势在必行^[4]。

用于用于自动驾驶、车路协同、AGV、无人机等的激光雷达系统通常被归类为测距激光雷达,实现最佳的测距精度取决于信号检测方案和随后采用的信号处理方法的有效性。因此,在检查激光雷达回波时,利用有效的信号处理技术是至关重要的^[5]。测距激光雷达采用 TOF 方案,能够对周围环境进行 360° 二维扫描探测,主要应精确定位和避障等具体场景。激光雷达信号感知系统是自动驾驶传感器感知融合系统的关键组成部分,本文采

用 ROS 系统对轮式机器人小车的激光雷达信号进行优化采集、处理及仿真实验，具有较好的理论研究意义，相关研究成果也必将具备良好的行业（自动驾驶、智能网联汽车、车路协同等）推广应用价值^[6]。

1.2 国内外研究现状

1.2.1 车载激光雷达系统的发展

激光雷达技术最初是由美国宇航局在 1970 年为阿波罗登月计划开发的，现在已经成为一种重要的测量工具，获得了学者的极大关注。这项技术可分为空对空探测和地对地探测，取决于被探测的目标。到 20 世纪 80 年代末，地对地探测技术进展迅速，在实时获取三维空间信息方面取得了显著的进步。发达国家，包括美国和德国，开始探索激光雷达技术在商业应用中的可行性。这项研究对这个领域的发展产生了实质性的影响。

美国 *Velodyne* 公司研制的高性能激光雷达 (*HDL-64E*)，最大探测距离 120 米，测距精度 2 厘米，纵向 0.4° ，横向 0.08° ，双路接收方式下，数据点数可达 220 万点/秒，可满足 L3 级自动驾驶需求，实现有条件无人驾驶。在 2021 年，*Quanergy* 公司推出的固体激光雷达 *S3*，能够有效地解决机械扫描结构所带来的问题，它采用了光学相控阵原理来进行激光束扫描，这种扫描方式与相控阵雷达相似，当改变发射阵列中发射单元的相位差时，激光的发射方向便会发生改变，从而可以通过调制的电号来实现光束的指向扫描，在 150 米的探测距离下，可以对 120° 的水平角和 10° 进行扫描。2022 年，作为全球汽车激光雷达软硬件技术的领先者，*Luminar* 将在传感器领域取得突破，其最大探测距离将达到 600 米，在反射率低于 10% 时，对目标障碍物的检测和识别将达到 250 米。

我国在激光雷达技术方面的研究也逐步发展并取得了一定的成果。速腾聚创作为世界顶尖的激光雷达技术研发公司，在 2023 年 CES2023 展会上，速腾聚创推出了其第一款全固态激光雷达 (*RS-LiDAR-E1*，简称为 *E1*)，并在展会上正式亮相。禾赛科技在自主研发高线性数激光雷达方面已积累了足够的技术储备，我们认为，高性能、低成本、高可靠性的激光雷达将成为未来工业机器人及量产汽车 ADAS 领域的核心竞争力，而实现高性能、低成本、高可靠性的关键路径就是芯片化。华为的激光雷达将为长安和北汽提供服务，在 2020 广州车展之前，长安集团的总裁公开表示，将与华为和宁德时代联手，共同打造一个高端的、智能化的汽车品牌。在长安“方舟”发布的新车型上，36 个感应器将被配备，包括 5 个激光雷达。北汽 *ARCFOX BU* 负责人于 2020 年 11 月在“网络汽车乌镇夜谈”上表示，北汽与华为携手研发出了 *ARCFOX* 中最新的一款车型—*HBT*。这款新车将搭载 96 线激光雷达 3 个，毫米波雷达 6 个，摄像头 12 个，超声波 13 个，华为提供了 352 兆次/秒的计算能力。目前，车载激光雷达系统在多个领域具有显著优势，包括城市场景的三维重建和工业建模等。我国高度重视激光雷达系统的发展，许多

研究机构都投入了大量的资源和人力。经过十多年的发展，中国的激光雷达技术已经取得了显著的进步，中国与国外的技术水平差距逐渐缩小。

1.2.2 点云滤波的研究现状

由于不可避免的测量误差和仪器采集或操作误差，采样的点云往往包含大量的无用信息，并不完全与期望的数据一致。这些噪声数据直接影响了点云成像的质量。因此，对采集的点云数据进行滤波处理，是减少或消除测量过程中人为或随机因素造成的误差的关键步骤。为了解决这个问题，国外研究人员开始研究点云滤波算法。*Fleishman* 提出的一种方法是建立一个点云网格模型并使用双边滤波来消除噪声。该技术包括在每个方向上应用不同的高斯核函数，高斯函数的标准差不同，以保持点云的特征并去除噪声的各向异性。*Weyrich* 等人提出了一种去除点云中离群点的方法，该方法为离群点定义了两个特征，并据此将其去除，从而更好地去除此类点。*Chen* 等人提出了一种消除平面对去噪效果影响的方法，将点云从主点云中分离出来，对剩余的点云进行平均统计处理。然而，这种方法对内部噪声的去除效果并不好^[7]。*Gu* 等人提出了一种利用高斯曲率对点云模型进行分类的方法，该方法可以与改进的中值滤波和双边滤波算法一起使用。*Schall* 等人介绍了一种计算含有噪声的点云和整体点云的贝叶斯分布的技术，根据概率最大的可能空间分布，通过移动点云内的所有点来进行平滑处理。*Zhang* 等人介绍了一种对光滑表面点云进行重采样的新技术，该技术通过将主体像素单元约束到下表面，实现了对 CVT 的有效计算^[8]。*W. Huang* 将二维图像超域平均法应用于三维点数据，用于去除散乱点云中的异常值。然而，这种方法的有效性在很大程度上取决于过程中使用的网格的大小。*Liu* 提出了一种双向数据的点云过滤方法，基于点云的三维重构数据，使算法更加复杂。最后，*Alex* 应用移动最小二乘法来拟合任何一个点的 k 邻域，通过移动点到其相应的拟合点来实现表面去噪。然而，这种方法需要解决一个非线性优化问题，导致大量计算和高复杂度。虽然研究人员提出了各种点云过滤算法，但每种算法都有其优点和局限性。

我国学者在点云数据滤波方面做了大量的工作，并提出了许多不同的滤波算法。苏志勋对中值滤波算法进行了改进，将点云法的矢量应用于二维影像的对比，而不是像元的排序选取。但是，在点云中，因为需要对矢量进行计算和分类，这种方法很耗时，特别是对于分布式的散点云。杨永涛等人提出了一种分散场景的离群点去除算法，首先对小范围的点云数据进行直接过滤，然后利用 k 邻域内的密度来约束检测的控制阈值^[9]。该算法对于具有高密度的散点云，可以得到很好的去除离群点的结果，但是会有部分噪音未被完全剔除。胡国辉通过对点云曲率特征值的计算，对点云曲率特征值进行了加权处理，并提出了三边滤波算法。王立辉提出了针对不同噪声的单独处理方法，采用模糊平均聚类算法进行聚类，双边滤波进行平滑，取得了较好的噪声去除效果。然而，聚类算法和双边滤波的组合在计算上是很密集的。戴士杰等人提出了一种在有效点云的邻域内通过点云的张量投票来处

理散乱的三维点数据的算法。这种方法获得了采样点的特征方向，随后沿着特征方向调整点云，并应用扩散张量的特征值来实现自适应速率调整^[10]。这些提出的方法显示了点云过滤的巨大潜力，可以为未来的研究提供有价值的见解。

1.2.3 点云配准研究现状

迭代最接近点 (*ICP*) 算法是由 *Besl* 和 *Mckay* 在 1992 年提出的，是三维点云对齐研究中的一个重要里程碑。*ICP* 算法本质上是在两个集合中寻找最近的点对，并估计出一个刚性变换来对齐它们。然后，这个变换被应用到一个集合的点上，这个过程被迭代直到收敛。同年，*Chen* 和 *Medioni* 提出了一种基于与 *ICP* 算法相同的算法框架的对齐算法，但使用点到最近点的切平面的距离作为目标函数，而不是 *ICP* 算法中的点间距离。

尽管 *ICP* 算法很有效，但它有几个局限性。例如，它假设一个点集是另一个点集的子集，当这个假设不成立时，就会产生一个错误的匹配，导致 *ICP* 收敛到正确的解决方案上的负面影响。此外，*ICP* 算法必须准确估计迭代开始时的初始位置，否则，它可能会产生一个局部最优解。此外，*ICP* 算法需要在每次迭代时对点集中的所有点进行相应的点搜索。

为了应对这些挑战，许多研究人员提出了对传统 *ICP* 算法的各种改进。例如，在 *ICP* 算法的基础上，*Sharp* 提出了一种引入欧几里得不变量特征的方法，如曲率、矩不变量和球面谐波函数不变量（球面谐波）来确定相应的点对，从而加速了对准，减少了算法落入局部最优的可能性。*Chen* 等人介绍了一种新的重采样方法^[11]，用于生成光滑的表面点云。该技术主要通过将主体像素单元限制在下表面来有效计算点云上的 *Centroidal Voronoi Tessellation* (*CVT*)，而下表面是由最佳拟合平面的集合代表的。*Torabi* 等人将点云配准问题转化为优化问题，并建立了覆盖点云的最小误差函数模型，与其他配准方法相比，有效地提高了可靠性^[12]。

此外，这些改进的配准算法可以适应不同的场景，不需要一个点集完全被另一个点集覆盖。例如，韩宝昌等人提出了一种分层划分区域的点云全局配准方法，对于重叠区域小、目标特征不明显的点云数据取得了满意的配准结果。王岩提出了一种自适应阈值最近点迭代算法，设置了收敛阈值和相应的点距，使迭代过程更加合理，在时间复杂度和配准精度方面都优于 *ICP* 算法。

此外，胡章芳等人在 *ICP* 配准算法中引入了动态迭代角度因子，使配准精度有了明显的提高^[13]；李强等人提出了一种基于多特征匹配的配准算法，克服了 *ICP* 算法搜索对应点对精度差的问题^[14]。杨清科等人利用随机抽样一致性 (*RANSAC*) 平面检测算法，通过提取外轮廓线作为相应的特征来识别建筑物的主要立面^[15]。随后，他们将管理立面的语义规则和立面之间的拓扑关系纳入其中，构建了建筑立面的三维线框模型。具体来说，该方法首先采用移动最小二乘法对分割后的点云进行局部拟合，然后以满足多个特征相似

性约束的点作为对应点对，寻找坐标变换矩阵，完成点云配准。

1.3 本文研究内容及技术路线

本文主要内容安排如下：

第一章，绪论，主要叙述了激光雷达和激光雷达点云数据处理技术的研究目的与意义。并对当前激光雷达、点云滤波与点云配准算法的研究现状进行了归纳和总结。

第二章，激光雷达和点云数据分析。详细的对激光雷达工作原理，并对激光雷达点云处理所用到的方法进行了理论分析，最后介绍了点云数据处理的 *PCL* 点云库。

第三章，激光雷达点云数据获取。首先对激光雷达点云数据采集的原理和工作过程进行了分析，再针对激光雷达点云的相关参数与数据特点进行了分析，最后总结处理点云数据的工作流程。

第四章，轮式小车点云数据的采集。首先采用 *Gazebo* 软件，对点云数据采集所需的轮式小车和环境仿真地图进行建模。然后通过相关指令采集到所需的激光雷达数据信息。

第五章，轮式小车点云数据的处理。首先研究点云噪声的来源，然后研究点云的滤波方法；最后对传统 *ICP* 点云配准算法进行分析，提出点云配准的改进方法，并加以实验验证。

图 1-1 为本文的技术路线图。

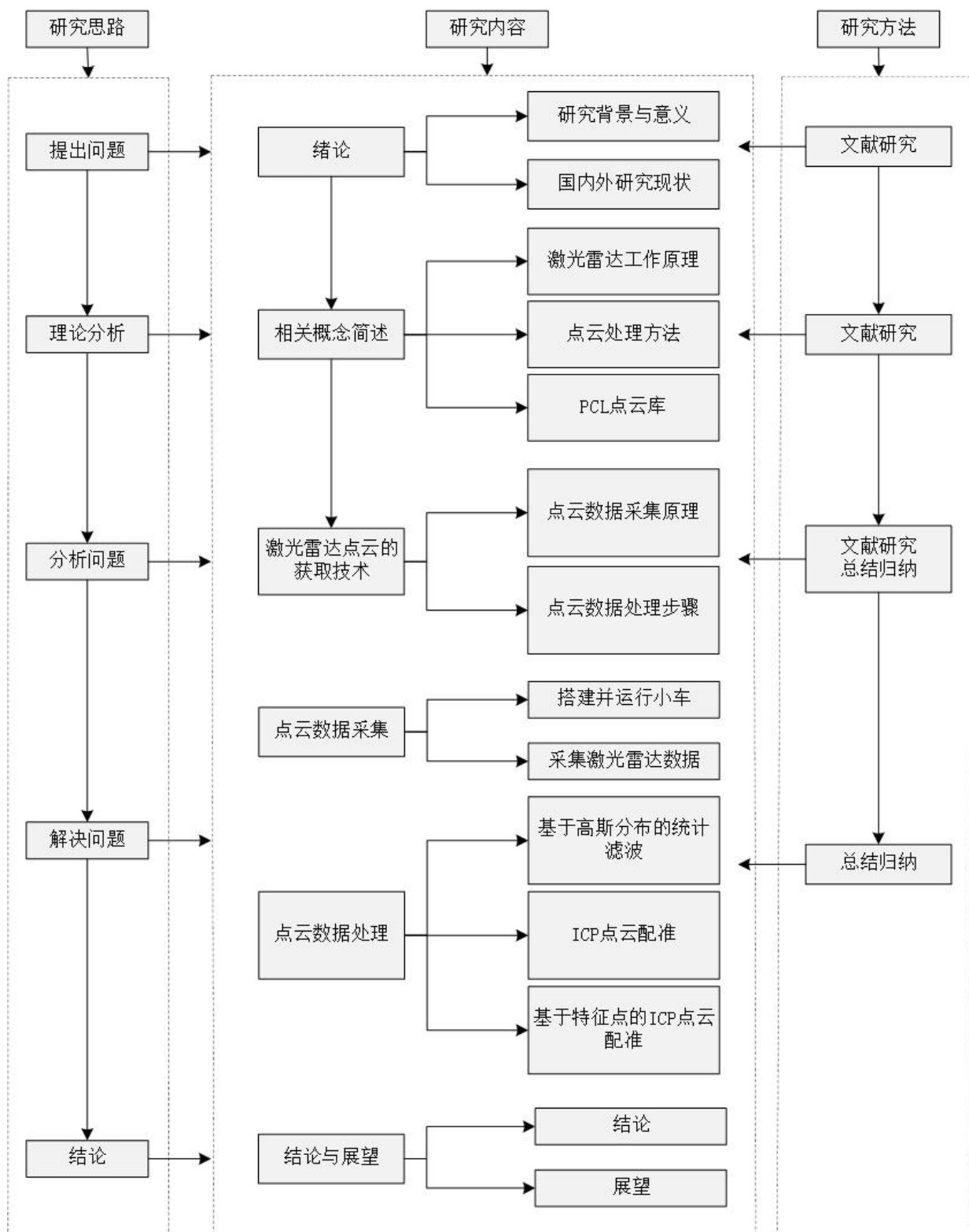


图 1-1 论文技术路线图

第二章 激光雷达和点云数据分析

2.1 激光雷达的工作原理

激光雷达系统由以下几个部分组成,包括控制系统、信息处理系统、激光接收器、激光发射器和显示设备。与传统的微波雷达系统类似,激光雷达也是利用发射探针向外发射光脉冲。接着,所述激光接收设备采集所述激光对所述目标对象进行扫描所生成的所述回波信号。在这以后,信息系统对信号进行处理以获得所需的目标信息。最后,显示设备展示图形,并保存相关的点云数据^[3]。当激光雷达接通电源时,由扫描仪及光学传送系统将一系列的激光脉冲讯号送至扫描区,从而使扫描区开始工作。再利用控制系统对扫描装置进行控制,使激光脉冲信号在三维空间中进行扫描。当激光到达障碍物后,物体表面反射激光束,产生反射回波信号。回波检测器和光电信号转换单元接收并将反射信号转换为电信号,然后传输到上层计算机系统进行处理。最后,在三维直角坐标系中将转换后的电信号被转换为坐标格式^[16]。工作原理示意图见图 2-1。

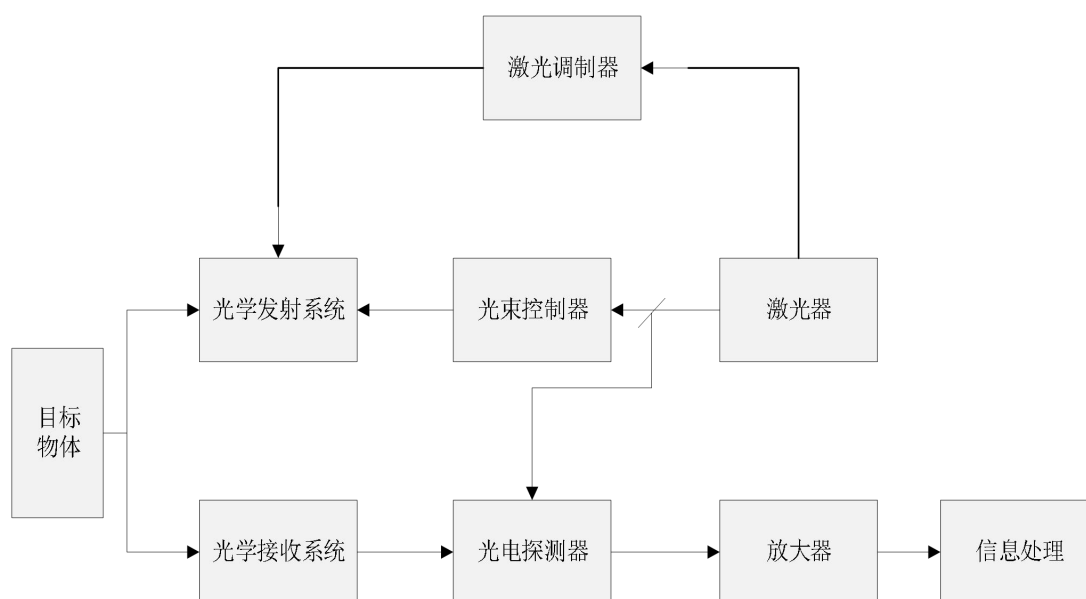


图 2-1 激光雷达工作原理

激光雷达只能提供最原始的点云数据,并不能直接输出环境中目标物的分类、检测和跟踪等信息,所以研究人员就在点云数据的基础上,基于 *PCL* 库研究算法,最终完成了对环境中目标物的识别、跟踪、检测和可视化。

2.2 点云数据处理

2.2.1 点云统计滤波

激光雷达生成的点云数据集经常会出现错误的回波信号,因为这个原因很可能会导致数值的不准确^[17]。此外,由于受到扫描结果和设备精度、周围环境等其他因素造成的测

量误差的影响，点云数据中不可避免地存在噪声。为了更好地处理后续的点云数据，如分割、配准和重建，在滤波过程中去除离群点和噪声是至关重要的。统计滤波通常用于处理激光点云数据，即对每个点的邻域进行统计分析，超出正常范围的点被定义为离群噪声点并随后被去除。离群点云通常表现为点云中大部分点以外的分散点，并会严重影响点云的成像、分割和后续的目标识别。一个区域内的点云数量表明了成像对象所占据的空间，区域内点云数量越多，越密集，则可以说明这块区域内所包含信息越多，越有可能含有物体，反之，较少的点可能表明存在噪声而不是目标。通过计算点云内的平均距离，超出一定范围的点可以被识别，并作为离群点被移除。

2.2.2 PCL 点云配准

为了获取被测物体的完整数据模型，需要确定合适的坐标系，并将各个视角获取的点集合并到统一坐标系下，形成完整的点云，以便进行可视化操作。然而，点云存在旋转、平移等不完整问题，因此需要对局部点云进行配准处理。点云数据的配准可以通过手动配准、依赖仪器的配准或自动配准等方式实现。自动配准技术利用计算机算法或统计学规律，计算两个点云之间的错位程度，以实现点云的自动配准。本质上，自动配准是通过在不同坐标系中获取的数据点云进行坐标系变换，以获取完整数据模型。关键问题是如何确定旋转矩阵 R 和平移向量 T 的参数，使两个角度测量的三维数据经过坐标变换后的距离最小。目前的配准算法根据过程可以分为整体配准和局部配准两种方法。*PCL* 提供了一个单独的注册模块，该模块实现了与注册相关的基本数据结构和经典注册算法，例如 *ICP* 算法。

PCL (Point Cloud Library) 是一个用于点云处理的开源库，点云配准是其中最为重要的部分之一。*PCL* 点云配准的基本原理是将多个点云数据集中的点匹配到一个共同的参考坐标系中，以便进行后续的分析 and 处理。通过配准，可以实现点云数据的精准对齐，进而获取更为全面、准确的点云数据信息。这是点云处理中不可或缺的一步，为各种基于点云的应用提供了坚实的基础。*PCL* 提供了丰富的点云配准算法和工具，可以满足不同应用场景下的需求，为点云处理提供了便利和高效。

PCL 中的点云配准可以大致分为两个阶段：特征提取和匹配。特征提取是指从点云中提取代表性的特征，如角、平面和其他突出的特征。匹配包括比较不同点云的特征以计算它们的相对位置和方向，最终将它们注册到一个共同的坐标系中。

第一阶段是利用 *Harris* 角点检测、*SIFT* 和 *FPFH* 等算法从点云中提取代表性特征，如角点和平面。*FPFH (Fast Point Feature Histograms)* 是一种快速有效的特征描述算法，可以计算出点云的局部特征，使其很适合大规模的点云数据处理。该算法根据每个点及其最近的邻居产生一个具有 33 个元素的局部特征向量，代表点云中的局部几何结构。这个向量在匹配阶段作为描述点云之间相似性的基础。

第二阶段，匹配，包括通过特征比较和注册到一个共同的坐标系中，计算不同点云之

间的相对位置和姿势。*PCL* 提供了一系列的配准算法，包括经典的 *ICP*、*NDT* 和 *GICP*，仅举几例。*ICP* (*Iterative Closest Point*) 使用一个迭代过程来寻找点云之间的最佳匹配，首先选择一个点云作为参考，然后将其他点云与之匹配，得到它们之间的转换矩阵。变换后的点云成为新的参考，这个过程不断重复，直到达到收敛效果。

除了 *ICP* 之外，*PCL* 还提供了一系列改进的点云配准算法。正态分布转换 (*NDT*) 算法尤其值得一提，它能够有效地处理点云中的噪声和不完整性，将其转换为概率分布进行匹配。另一方面，广义 *ICP* (*GICP*) 算法在 *ICP* 的基础上引入了一个局部翻译模型，在解决非刚性物体对齐的挑战方面表现更好。

总的来说，*PCL* 的点云对准提供了一个全面的解决方案，将多个点云数据集匹配到一个共同的参考坐标系。凭借其一系列的特征描述和对准算法，*PCL* 具有通用性，可以应用于各种点云场景，使其成为许多应用中的重要工具。

2.3 PCL 点云库

从算法层面上看，*PCL* 点云库主要包括点云降噪、点云压缩、点云配准、点云特征表达等。通过使用基类的方法，将各个算法之间的差异化，归纳出一些共同的函数，这样可以提高算法在实现过程中的严密性和清晰性。

在激光雷达点云数据处理过程中，点云库 (*point cloud Library, PCL*) 是一个多功能模块化 C++ 模板库，它集成了 *Boost*、*Eigen*、*FLANN*、*VTK*、*CUDA*、*OpenNI*、*QHull* 等多种第三方库^[18]。它主要依靠 *Eigen* 库来执行大多数数学运算，同时利用 *TBB* 和 *OpenMP* 库来实现多核并行性。*FLANN* 主要用于支持 *k* 近邻搜索，*PCL* 点云库利用 *Boost* 共享指针进行数据传输，从而避免了点云数据的重复。自 0.6 版本以来，*PCL* 库已经可以在多个操作系统上使用，包括 *Windows*、*MacOS*、*Linux* 和 *Android*^[18]。

从算法角度来看，*PCL* 库主要集成了点云降噪、点云压缩、点云配准、特征表达等点云数据处理算法。利用基类来区分每组算法并总结通用函数，可以提高算法实现过程的严谨性和清晰度。由于 *PCL* 库的稳健性、灵活性和易用性，它被广泛应用于各个领域，如机器人、计算机视觉和三维成像等。

利用 *PCL* 库接口的点云数据处理工作流程可以概括为以下几点：

- 实例化一个算法类对象；
- 将点云数据加载到处理函数中；
- 设置算法的关键参数；
- 调用处理函数以获得期望的结果。

(1) 输入/输出模块

加载获取的点云数据对后续处理至关重要。*PCL* 库有其专有的 *PCD* 文件格式，它还提供了将其他点云文件格式转换成 *PCD* 文件格式的接口。该库还集成了大多数市售点云

采集设备的接口。举例来说，为了兼容微软的 *Kinect* 和 *XtionPro* 这样的设备，*PCL* 程序库整合了一个叫做 *OpenNI* 的第三程序库，以便更好的从那些设备中获取数据。此外，在最新的 1.8.0 版本的 *PCL* 库中，为 *Velodyne LiDAR* 设计了专门的功能接口，以实现兼容，使数据访问速度更快。该库为 *VLP-16* 雷达采集的激光雷达数据提供了专门的 *API* 函数，将 *pcap* 格式转换为 *pcd* 格式。

(2) 可视化模块

VTK 和 *CL* 都有其可视化数据库，*VTK* 提供了多平台的三维点云数据表面渲染技术，包括支持纹理、张量和体积法的可视化。

PCL 可视化库结合了 *PCL* 库和 *VTK* 库，为 n 维点云数据结构提供了可视化层，实现了处理后多维数据的快速渲染和显示。可视化库提供了几种方法。

(1) 对任意维度空间的点云数据进行实时渲染和视觉属性设置，如点的颜色、大小和透明度通道；

(2) 基本的三维图形绘制方法（如圆柱体、直线、多边形等），以及来自用户定义的点集的函数接口；

(3) 二维平面的直方图可视化模块（*PCL* 直方图可视化器）；

(4) 多个绘图手柄。在三维矩形坐标系中，用户可以选择用于定位点的尺寸或用于渲染的颜色；

(5) 深度图像可视化模块。

2.4 本章小结

本章首先介绍了激光雷达的工作原理，再对点云数据处理中的滤波和配准方法进行了理论分析，并阐明了点云数据滤波和配准算法的原理，最后介绍了点云数据处理所依赖的 *PCL* 点云库及基于 *PCL* 接口的点云数据处理工作流程，为后续的三维点云数据处理工作提供了方法依据。

第三章 激光雷达点云数据获取

3.1 激光雷达点云数据的采集

激光雷达点云数据采集是指通过激光雷达设备对目标场景进行扫描，记录每个点的位置、距离、反射强度等信息，生成大量的三维点云数据的过程。具体的采集过程可以归纳为以下几点：

- (1) 激光发射器向目标场景发射一束激光。
- (2) 激光束打到目标物体的表面并被反射回来。
- (3) 接收器接收反射的激光束并记录激光的时间戳。
- (4) 根据激光的发射和接收时间，计算出激光束的传播时间和距离。
- (5) 根据激光束的传播时间和距离，确定激光束打到目标物体表面的位置。
- (6) 这个过程不断重复，以扫描整个目标场景，并记录每一束激光击中目标物体表面的位置和反射强度信息。

综上，将所有记录的激光束打到目标物体表面的数据结合起来，生成点云数据，其流程图如 3-1 所示。

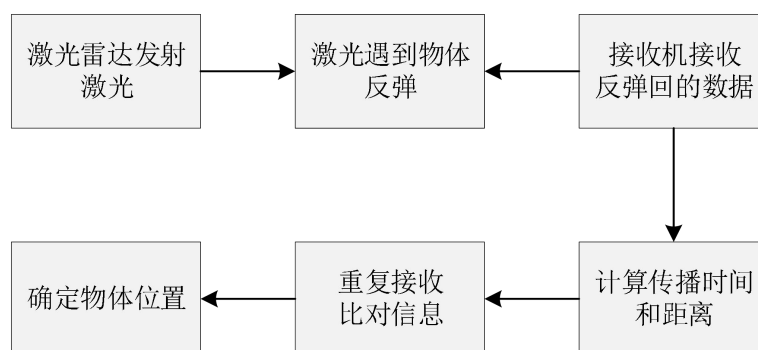


图 3-1 激光雷达数据采集过程

在获取激光雷达点云数据的过程中，目标场景的复杂性和不同的扫描角度是至关重要的考虑因素。通常同时使用多个激光雷达设备，或者使用机器人自动驾驶仪进行扫描，以提高采集效率和质量。此外，点云数据采集的后期处理是必要的，如数据过滤、对齐、重建、分类、聚类和其他操作，以实现更精细的分析和处理。

3.2 激光雷达点云的相关参数和特点

激光点云是指由三维激光雷达设备生成的空间点的数据集，点云中的每个点包括三维坐标（XYZ）和激光反射强度（Intensity），后者取决于各种因素，如目标表面材料和粗糙度、激光入射角、激光波长和激光雷达的能量密度。准确记录三维坐标和反射强度对于成功利用激光点云数据的一系列应用至关重要，包括地形建模、植被分析和物体识别。

为了全面了解点云，表 3-1 概述了其相关参数和特点。

表 3-1 激光雷达点云的相关参数

参数名	参数介绍	与点云的关系
角分辨率	激光雷达的角度分辨率是指两个相邻的点云之间的角度间隔,它可以进一步分为水平和垂直分辨率。	较小和较高的角度分辨率会导致更密集的点云和提高目标物体的分辨率。
视场角	类似于角度分辨率,与激光雷达的可探测范围有关,可以进一步细分为水平视场角和垂直视场角。	较大的视场角扩大了可探测到的点云的范围。
帧率	为激光雷达每秒进行的扫描次数,在这些系统的功效和效率中起着关键作用。	更高的帧率会导致更快地获取点云信息,从而增强激光雷达系统的实时性能,提高帧率对于激光雷达系统的无缝和快速运行是必不可少的。
点频	指的是激光雷达每秒发射的点云数量,是一个直接影响系统传感能力的关键参数。	较高的点频率会导致发射更多的点云,从而提高所获数据的整体质量和分辨率。因此,在高精度点云信息的情况下提高点频率是必不可少的。
反射率	激光雷达系统扫描目标后可返回的点云的百分比,对系统的性能和准确性起着根本的作用。	它决定了返回到激光雷达系统的能量,进而影响到系统的探测距离。反射率越高,返回的能量就越多,从而激光雷达系统的精度和灵敏度越高

点云数据是一种由无数个点组成的三维数据结构,每个点通常包含其在三维坐标系中的位置,以及可选的颜色、法向量和属性。点云数据表现出几个固有的特性,其中旋转不变性和置换不变性是值得特别关注的两个关键特性。

旋转不变性是指点云数据在三维空间中的属性,其中翻译和旋转不会修改数据的基本属性。这一特性对于处理和分析点云数据至关重要,因为在实际应用中,获取具有固定方向和位置的点云数据具有挑战性。通过保持旋转不变性,点云的特征提取和识别仍然不受翻译和旋转的影响。为了实现旋转不变性,通常的做法是利用旋转不变的特征来描述点云数据,如法向量和曲率。

置换不变性指的是点云数据的特性,尽管点的顺序发生了变化,但仍然保持不变。换句话说,点云数据中任何点的排列都应该产生对它所代表的物体的相同描述。这一特点在

点云数据处理和分析中也是至关重要的，因为不同的点云数据可能有不同的点排列，但仍然描述同一个物体。为了实现包络不变性，点的局部坐标系统，如基于旋转的局部参考系统（ RF ），通常被用来描述点云数据。这些技术使点云数据中的每个点都能被转化为一个局部坐标系，从而使点的排列与点云数据的描述不相关。

3.3 激光雷达数据处理步骤

图 3-2 描述了使用激光雷达技术处理点云数据的典型工作流程，此过程需要几个基本步骤，对有效的数据处理至关重要。

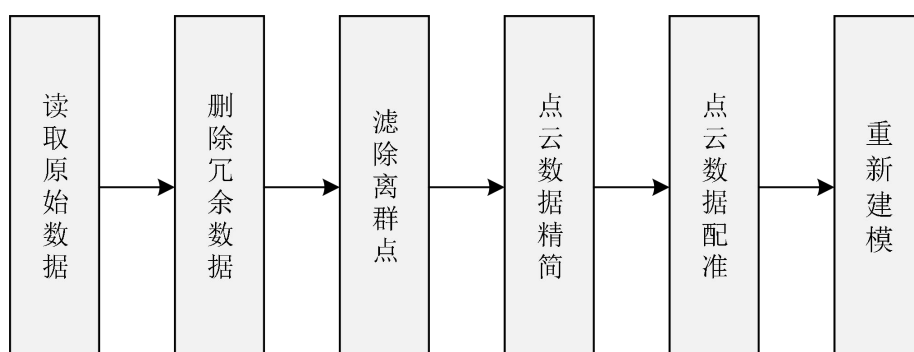


图 3-2 点云数据处理流程

图 3-2 描述了点云数据处理中涉及的一般步骤，根据激光点云数据的具体要求和特点，可能需要对这些步骤的顺序和次序进行某些调整，或者完全省略某些步骤。需要注意的是，点云数据的处理是一个高度细微和动态的过程，需要灵活性和适应性。

- (1) 去除冗余数据，以消除设备采集过程中可能引入的噪声点。
- (2) 识别并去除离群的噪声点，以提高数据的质量。
- (3) 进行数据精简，以减少大而密集的三维点云数据集的大小。
- (4) 进行点云数据的配准，即在不同的坐标系和视点之间转换点云坐标。
- (5) 通过拟合配准的点云数据来进行模型重建，以获得物理模型。

消除离群点和噪声点以及配准点云数据是处理三维点云数据的必要步骤，影响着处理后的点云数据的质量，同时也是后续三维场景重建任务的基础。

3.4 本章小结

本章首先对激光雷达点云数据采集的原理和工作过程进行了分析，再针对激光雷达点云的角分辨率、视场角、帧率、点频、反射率等相关参数与数据特点进行了列表分析，最后基于消除离群点、噪声点及配准的一般要求，给出了处理点云数据的典型工作流程。

第四章 轮式小车多线激光雷达点云数据采集

4.1 ROS 小车仿真模型搭建

第一步：创建功能包。

工作空间可以简单理解为工程目录，通过 `mkdir` 创建一个工作空间文件夹，这里我们命名为 `jubot_demo`，创建后进入该目录，如图 4-1 所示。

```
cd catkin_ws
catkin_create_pkg jubot_demo urdf xacro
cd jubot_demo/
mkdir urdf
mkdir launch
mkdir meshes
mkdir config
```

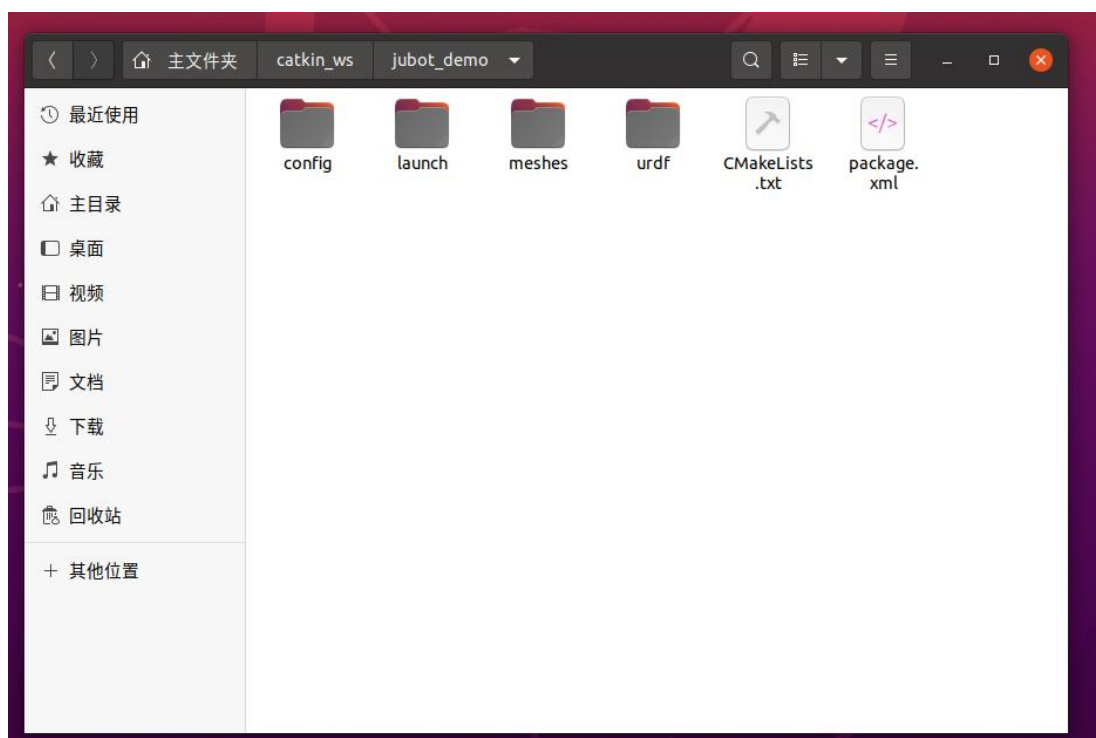


图 4-1 工作空间目录

第二步：编写 `urdf` 和 `launch` 文件。

在 `launch` 目录下，新建一个 `launch` 文件，该 `launch` 文件需要启动 `RViz`，并导入 `urdf` 文件，`RViz` 启动后可以自动载入解 `urdf` 文件，并显示机器人模型。

第三步：优化 `xacro` 模型。

在 `urdf` 模型和 `RViz` 可视化的基础上，我以方形全向轮车为基础，进行功能实现，`xacro` 模型其实就是调用全局变量并结合函数。

第四步：搭建 *Gazebo* 仿真环境。

1. 打开 *Gazebo*（终端直接输入 *Gazebo*）

2. 左上角 *edit->building editor*

3. 拖拉左边的组件到白色区域，开始创建

4. 创建好后 *file->save world as* 以 *.world* 格式保存在功能包下，最好新建个 *world* 文件夹，再在 *model* 中打开文件，添加障碍物模型

5. 添加 *gazebo_robot.launch* 文件中的地图路径，改为自己保存地图位置的路径：

```
<include file="$ (find jubot_demo)/launch/gazebo_world.launch"/>
```

小车在 *Gazebo* 中的显示模型如图 4-2 所示。

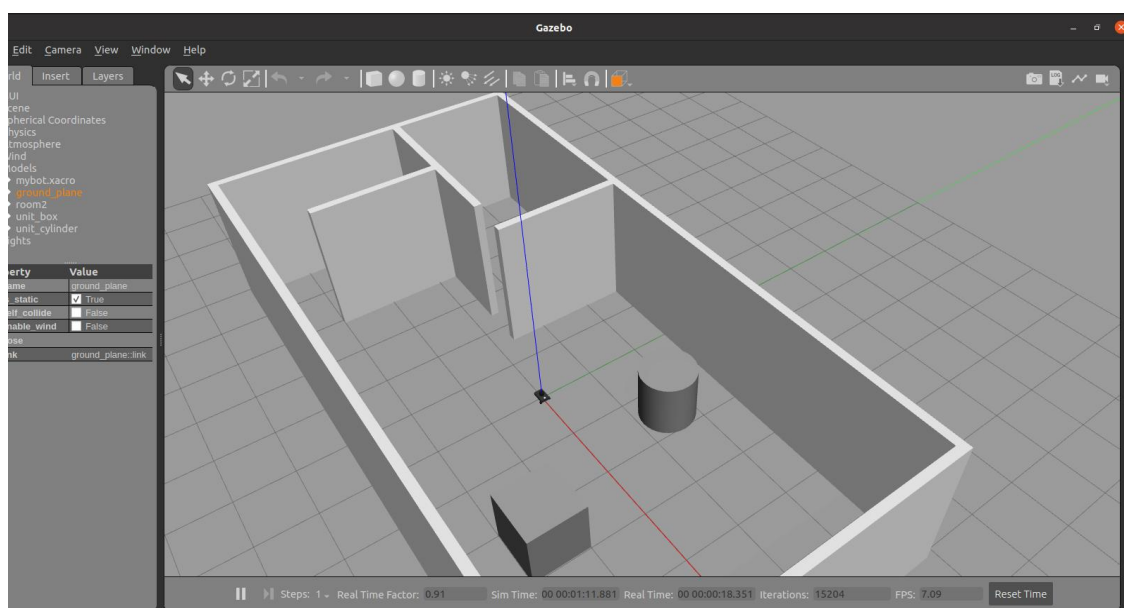


图 4-2 Gazebo 仿真图像

4.2 试运行小车并收集数据

在终端中输入 `roslaunch jubot_demo simulation_robot.launch`，启动 *Gazebo*，可以看到机器人在图中显示出来；新打开一个终端，输入 `rqt_image_view`，打开相机显示；再打开一个新的终端，输入 `roslaunch teleop_twist_keyboard teleop_twist_keyboard.py` 启动键盘控制节点，可以控制仿真小车运动，并且可以在 *Image_view* 中看到相机视野的变化，如图 4-3 所示。

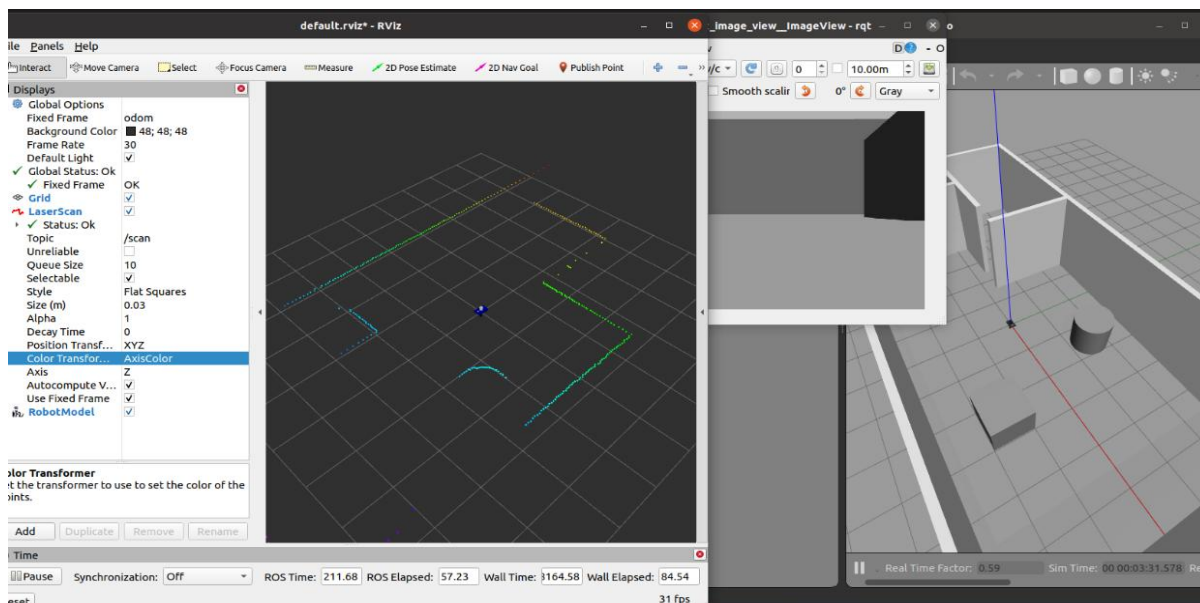
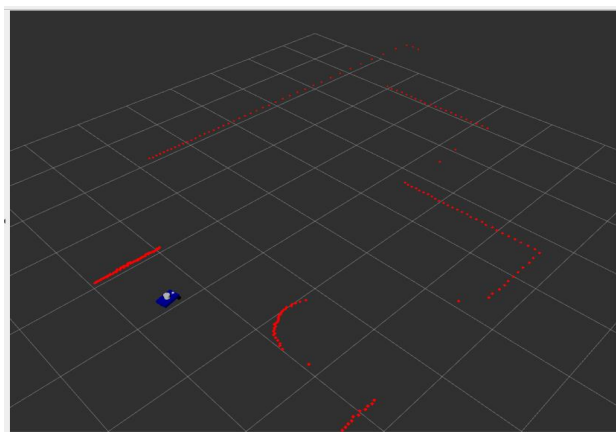
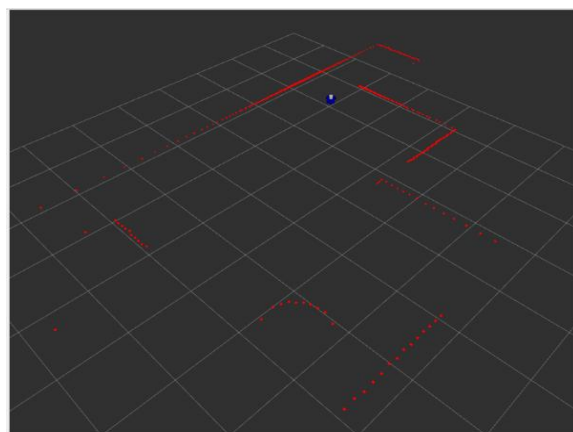


图 4-3 RViz 点云显示

打开 *RViz*，增加 *RobotModel* 和 *LaserScan* 显示，调节话题路径，可以看到机器人小车模型和激光雷达数据点云。运行键盘控制节点，可以看到激光雷达数据也会随着机器人小车的运动而改变，如图 4-4 a-b 所示。



a 柱体旁的点云



b 拐角处的点云

图 4-4 不同位置下的点云显示

打开 *mybot.xacro* 文件，更换激光雷达为 *VLP-16*，获得点云图如图 4-5 所示。

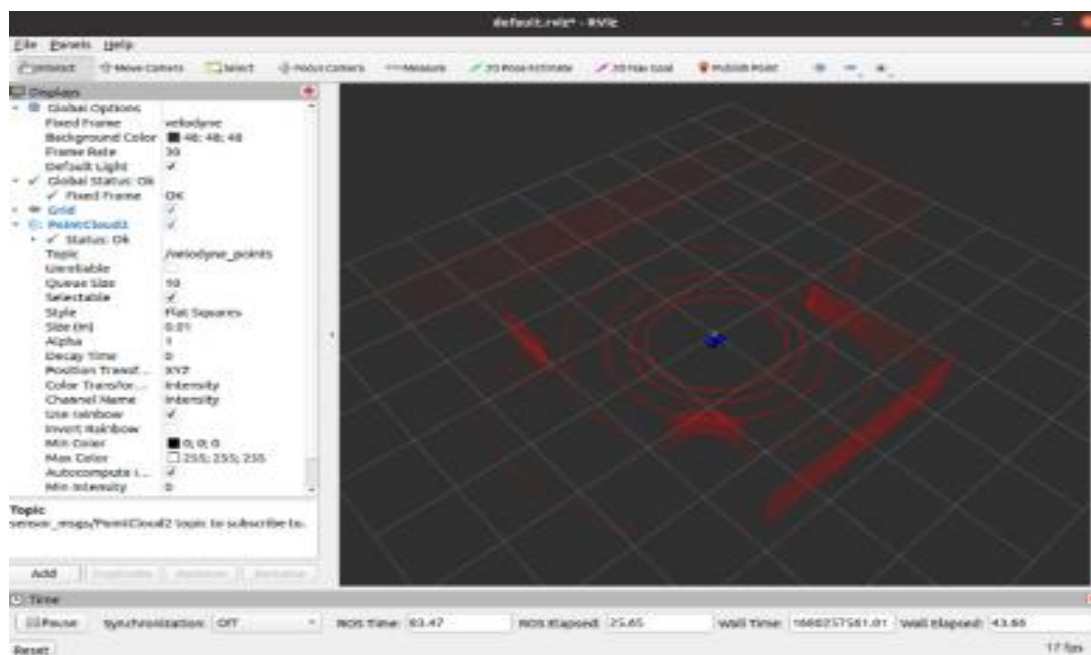


图 4-5 VLP-16 点云图

通过 *roscat* 指令输出点云 *bag* 文件，从 *bag* 数据包中提取雷达点云数据，并将雷达点云数据转换为 *pcd* 格式，然后会得到按照时间戳排序的雷达点云数据，如图 4-6 所示。

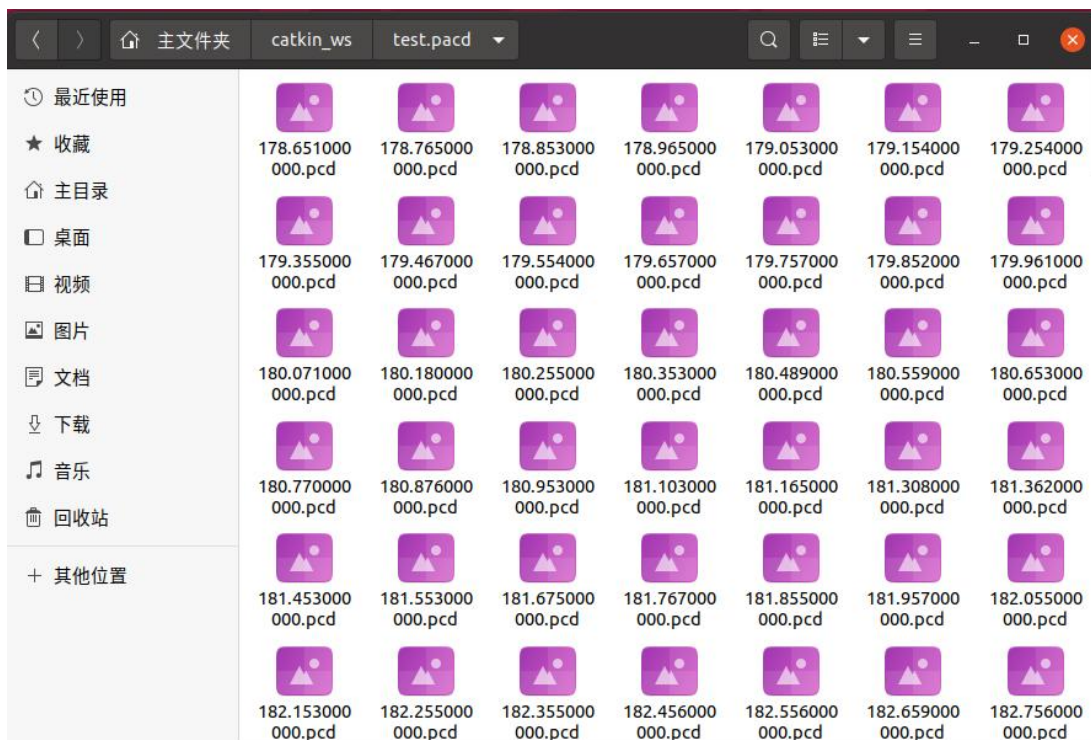
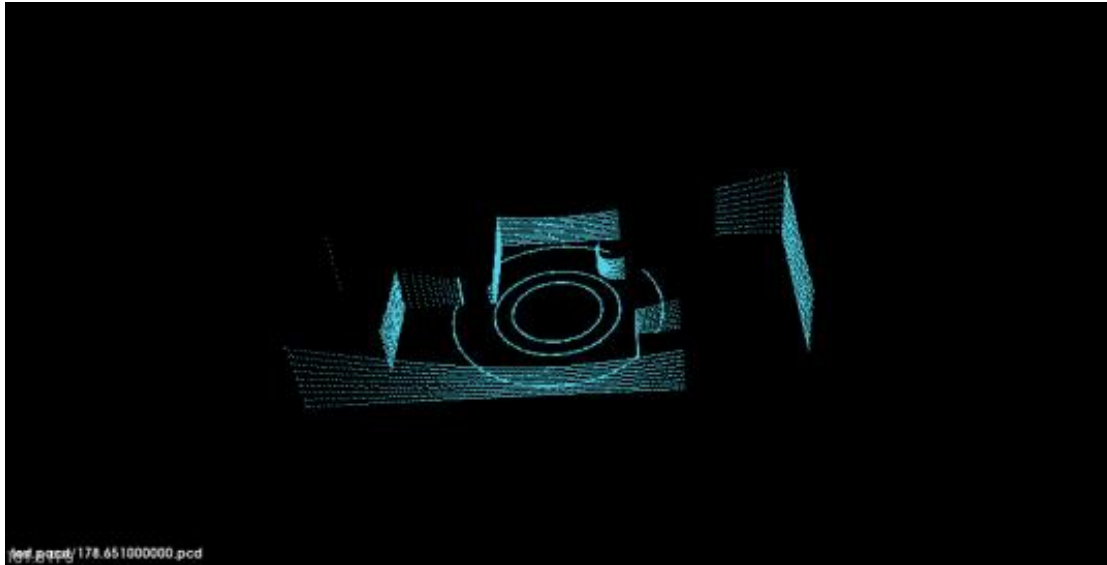
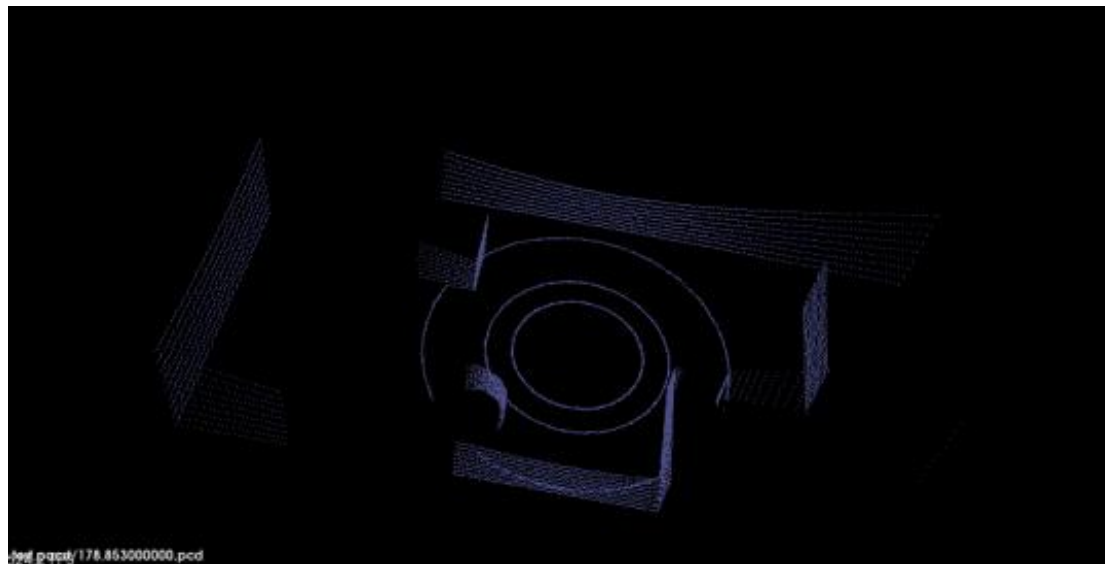


图 4-6 不同时间戳的点云数据

安装 *pcl_tools* 通过使用 *pcl_viewer* 指令查看 *pcd* 文件,不同时间戳的点云图,如图 4-7 a-b 所示。



a 178.651000000



b 178.853000000

图 4-7 不同时间戳的点云图

4.3 本章小结

本章首先采用 *Gazebo* 软件，对点云数据采集所需的轮式小车和环境仿真地图进行了建模；考虑到模型的泛化性，对轮式机器人小车分别搭载单线激光雷达和 16 线激光雷达进行仿真实验，最后通过 *RViz* 软件确认话题路径和点云情况，环境信息确认后以受控指令方式采集点云数据。

第五章 轮式小车激光雷达点云数据处理

5.1 点云统计滤波

点云滤波是处理和平滑三维点云数据的一个关键步骤。它的主要目的是消除噪声和异常点，以提高点云数据的准确性和质量。点云数据本质上是一个庞大的三维点的集合，代表了物体表面或环境中的特定位置。然而，在数据采集过程中，固有的传感器限制或环境因素会在点云数据中引入各种类型的噪声、异常值或冗余信息。这些问题会严重影响点云数据的后续分析和处理。因此，点云滤波在确保最终结果的可靠性和准确性方面起着关键作用。

5.1.1 激光雷达扫描数据噪声来源

在利用激光雷达技术获取三维点云数据的过程中，由于各种内部或外部因素，噪声点的存在是不可避免的^[19]。扫描数据中产生噪声的原因有多种，大致可分为以下几种^[20]：

(1) 被扫描物体的体积不对称、材料的低激光吸收率和轮廓缺陷都会导致生成的点云数据出现误差^[21]。

(2) 由于激光雷达系统本身固有的缺点，如：振动、刷新频率、扫描装置精度，GPS 同步精度、光斑能量强度等，都会造成获取的点云数据中的噪声。

(3) 在扫描过程中，由于外部因素，如场景中的物体的突然阻挡，或操作仪器时摇晃造成的人为错误，也会产生意外误差。

因此，在进一步处理之前，对点云数据进行去噪是一个必不可少的步骤，因为它可以提高后续三维场景重建的准确性。虽然激光雷达技术不能完全消除噪声，但适当的去噪技术可以帮助减轻噪声的影响，提高最终结果的可靠性。

5.1.2 统计滤波算法

图 5-1 显示了该算法的流程图。首先，对点云数据集中的所有点进行 k 邻域搜索，以计算每个点与其 k 邻域之间的距离集合。接下来，计算所有点与它们的 k 个邻居之间的距离的平均值 μ 和标准偏差 σ 。使用一个比例因子 α ，然后确定距离阈值 d_{max} ，即 $d_{max} = \mu + \alpha \times \sigma$ 。最后，对数据集中的所有点进行处理，那些与 k 个邻居的平均距离超过阈值 d_{max} 的点被识别并随后被删除。该算法的具体步骤可以总结为以下几点：

(1) 首先，必须确定点云的总数 n 。对于点云数据集中的每个点 p ，通过 k 邻里搜索确定其邻里。然后计算点 p 和其邻域内每个点之间的平均距离；

(2) 为了获得一组 n 个距离 $\{d_1, d_2, d_3, \dots, d_n\}$ 的平均值 μ 和标准偏差 σ ，可以用式 (5.1) 和式 (5.2) 计算这些统计参数；

$$\mu = \frac{1}{n} \sum_{i=1}^n d_i \quad (5.1)$$

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (d_i - \mu)^2} \quad (5.2)$$

(3) 基于高斯分布的方法使用平均值 μ 和标准偏差 σ 来确定最大阈值 $d_{max} = \mu + \alpha \times \sigma$ ，其中系数 α 决定了阈值的范围。 α 的适当值取决于具体的点云分布和数据中存在的噪声程度。选择一个太小的系数可能会导致有用的点云数据作为异常值被删除，而选择一个太大的系数则可能导致无效的去噪结果。因此，为每个点云数据集仔细选择适当的系数是至关重要的，以确保在保留有用数据和有效去除噪声之间取得最佳平衡；

(4) 对每个点的平均距离 μ 与阈值 d_{max} 进行比较，超过这个阈值的点被确定为离群的噪声点，随后从数据集中删除。

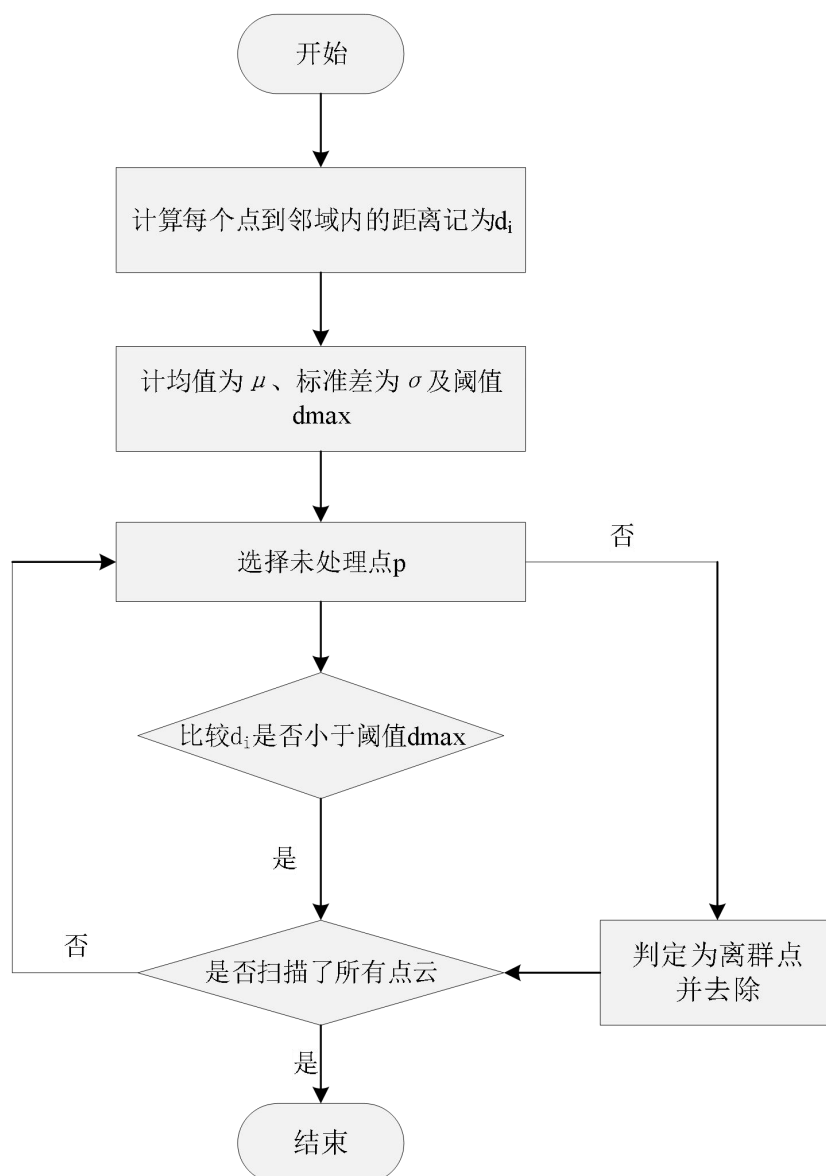


图 5-1 统计滤波流程图

5.1.3 Statistical Outlier Removal 滤波器移除离群点

Statistical Outlier Removal (SOR) 过滤器是一种广泛使用的点云过滤算法，它采用统计分析来消除数据中的离群点。该算法假设点云数据遵循高斯分布，并根据数据的平均值和标准差检测异常值。

SOR 过滤器的主要概念包括计算点云数据集中每个点周围的平均值和标准差。基于这些统计数据，计算出每个点的 *Z-score* 值。然后，通过比较每个点的 *Z-score* 值和预定的阈值，该算法识别并删除 *Z-score* 值超过阈值的点，这些点被认为是离群值。

统计离群点去除 (*SOR*) 滤波器往往用于去除点云数据中的稀疏离群点噪声，这些噪声可能是由于点云采集过程中的测量噪声引起的。这些噪声点稀疏地分布在点云空间中，在估计点云的局部特征（如法向量和曲率变化率）时，可能会导致不正确的计算结果，这反过来又会影响到后处理任务，如点云对齐。统计过滤器假设点云中所有的点和它们的 k 个最近的邻居之间的平均距离遵循高斯分布，并且可以根据这些距离的平均值和方差来确定距离阈值。当一个点和它的 k 个最近的邻居之间的平均距离超过这个阈值时，这个点就被认为是一个离群点并被移除。该过滤器的实现是通过首先遍历点云来计算每个点与其 k 个最近邻居之间的平均距离。接下来，计算所有平均距离的平均值 μ 和标准差 σ ，以确定距离阈值 d_{max} ，即 $d_{max} = \mu + \alpha \times \sigma$ ，其中 α 是一个取决于邻居数量的比例因子。最后，再次遍历点云，并剔除具有 k 个邻居点且平均距离大于 d_{max} 的点。

统计滤波器被广泛用于去除点云中的显著异常值，这些异常值通常由测量噪声引入，并在空间中呈现稀疏分布。每个点所表达的信息量可以理解为点密度的函数，其中密集的区域信息量更大。然而，离群点往往包含无用的、信息量较小的噪声信息。因此，可以定义点云在密度低于某个阈值的区域是无效的。为了去除离群点，要计算每个点到其最近的 k 个邻居的平均距离。所得的距离应该形成高斯分布，从而可以确定一个距离阈值，超过这个阈值的点可以被剔除。激光扫描通常会生成密度不均匀的点云数据集，而测量误差会引入稀疏的离群点。在这种情况下，估计局部点云特征（如取样点的法向量或曲率变化率）在计算上可能很复杂，并导致不正确的数值，最终导致处理失败，如点云配准。

为了解决点云中的离群点问题，对每个点的邻居进行了统计分析，不符合某些标准的点被修剪掉。具体来说，这是通过计算输入数据中每个点到邻近点的距离分布来实现的。对于每个点，计算到所有相邻点的平均距离，假设结果分布遵循由其平均值和标准差决定的高斯形状。那些平均距离超过标准范围的点可以被识别为离群点，并从数据中删除。这种方法有效地剔除了噪声点，确保后续处理任务，如点云配准，是在一个更干净、更可靠的数据集上进行的。

如图 5-2 滤波前的点云图和 5-3 滤波后的点云图为离群点去除前后的对比图。

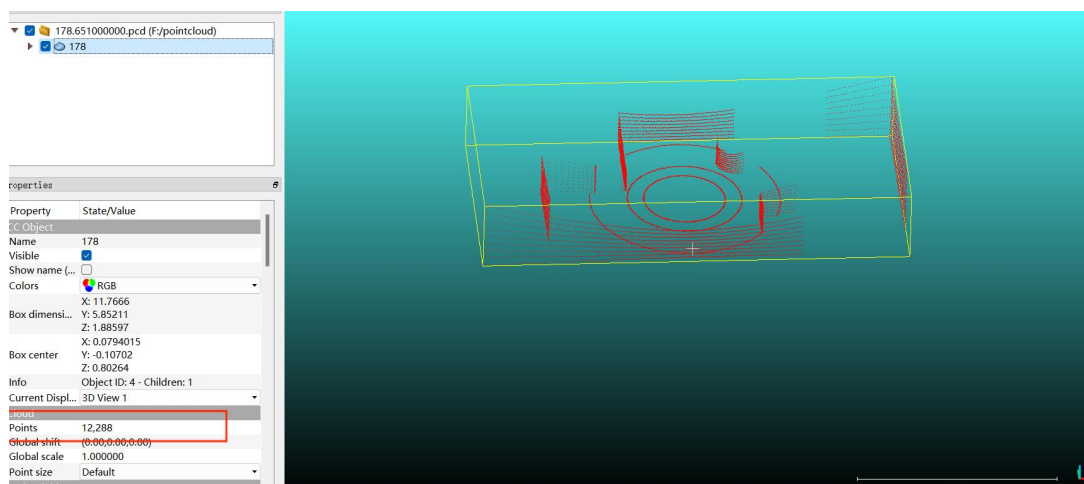


图 5-2 滤波前的点云

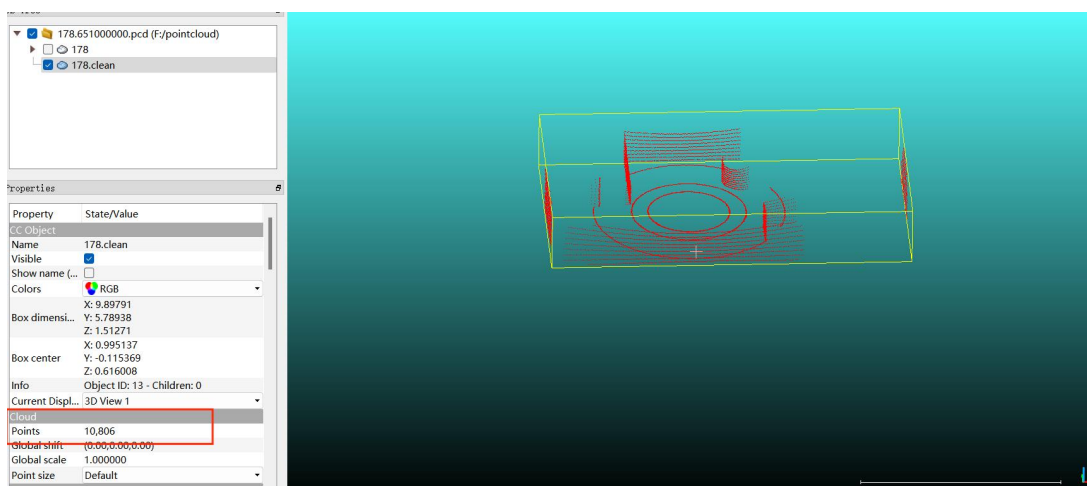


图 5-3 滤波后的点云

5.2 ICP 点云配准

在获取三维点云数据时，经常会遇到某些不可控的因素，如环境中的障碍物或不均匀的光照。这些因素会导致被测目标的点云信息不完整。为了准确捕捉目标的完整信息，又必需从不同角度进行多次扫描。从所得到的三维点云数据中，可以计算出一个最佳的变换矩阵，利用刚体变换将点云配准到一个共同的坐标系中。

点云配准是激光雷达点云数据处理的一个关键步骤。它涉及到将点云数据从多个局部坐标系转换到一个共同的坐标系，从而提高后续三维重建过程的精度。点云配准的质量直接影响到最终重建的三维模型的准确性。点云配准的主要目的是将从不同角度获得的点云数据无缝转换到一个标准化的坐标系中，同时有效地合并重叠的部分^[22]。因此，必须确保以高精度进行配准，以获得可靠和准确的三维重建结果。

5.2.1 点云配准的阶段

点云配准由两个阶段组成：粗配准 (*Coarse Registration*) 和精配准 (*Fine Registration*)。矩阵变换的过程就是点云配准的过程。

粗配准是精配准的前置步骤，当源点云和目标点云不知道它们的初始相对位置时使用。它的主要目的是在初始条件未知的情况下快速估计一个近似的点云配准矩阵^[23]。通过对两个点云的粗略配准，旋转矩阵 R 和平移矢量 T 的初始值可以被确定为两个点云的任何初始状态。计算过程需要相对较高的计算速度，而结果的准确性不需要过于精确。

常见的粗配准算法的思路包括了：通过统计学概率、基于全局搜索策略以及基于局部特征描述等方法。

精配准是一个更精确的配准过程，它建立在粗配准的结果之上。它涉及到迭代最近点算法 (*ICP* 算法) 和正态分布转换 (*NDT* 算法) 等其他方法，使用已知的初始变换矩阵来细化对齐。

5.2.2 ICP 配准算法

点云配准主要是应用 *ICP* 算法来完成的任务的^[24]，图 5-4 中的流程图总结了标准 *ICP* 算法的简化方案。

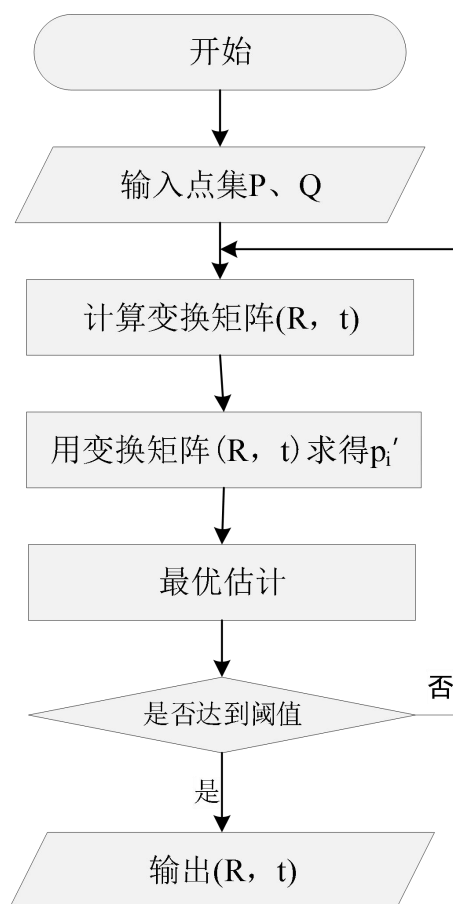


图 5-4 ICP 配准流程图

ICP 配准算法的目的是通过迭代选择相应的点对来确定最佳的转换矩阵，直到达到所需的收敛精度。使用 *ICP* 算法的配准过程包括在源点云 $Q=\{q_j|q_j \in R^3, j=1,2,\dots,m\}$ 和目标点云 $P=\{p_i|p_i \in R^3, i=1,2,\dots,n\}$ 之间找到一个满足最佳收敛条件的变换矩阵，其中 m 和 n 代表两个点云的大小。旋转矩阵和平移矩阵分别表示为 R 和 t ，而 $f(R,t)$ 代表误差函数，表征源点云和目标点云在变换矩阵 (R,t) 影响下的差异。求解变换矩阵 (R,t) 的过程成为寻找使 $f(R,t)$ 最小化的最优解 (R,t) 的过程。由于 *ICP* 算法能有效地将点云高精度地对齐，因此被广泛应用于各个领域，如计算机视觉、机器人学和三维扫描^[25]。

其中，

$$f(R,t)=\frac{1}{n}\sum_{i=1}^n \|q_i-(Rp_i+t)\|^2 \quad (5.3)$$

ICP 配准步骤：

- (1) 在目标点云 P 中取点集 $p_i \in P$;
- (2) 找出源点云 Q 中的对应点集 $q_i \in Q$ ，使得 $\|q_i-p_i\|=\min$;
- (3) 计算旋转矩阵 R 和平移矩阵 t ，使得误差函数最小;
- (4) 对 p_i 使用上一步求得旋转矩阵 R 和平移矩阵 t 进行旋转和平移变换，得到新的对应点集 $p'_i=\{p'_i=Rp_i+t, p_i \in P\}$;
- (5) 计算 p'_i 与对应点集 q_i 的平均距离：

$$d=\frac{1}{n}\sum_{i=1}^n \|p'_i-q_i\|^2 \quad (5.4)$$

如果源点云和目标点云之间的距离 d 低于指定的阈值或达到最大的迭代次数，迭代计算就终止了。否则，该过程将返回到第 2 步，继续完善排列，直到满足收敛条件。

ICP 算法涉及几个关键点，包括：

- (1) 获取原始点集：这可以通过各种方法实现，如均匀采样、法矢采样、随机采样等。
- (2) 对应点集的确定：源点云和目标点云之间的对应点可以通过点对点、点对投影或点对面的方法建立。
- (3) 变换矩阵的计算：变换矩阵可以用四元数法或 *SVD* 奇异值分解法等方法计算。本文使用采集的激光雷达点云数据进行了传统 *ICP* 算法实验，如图 5-5 和 5-6 所示：

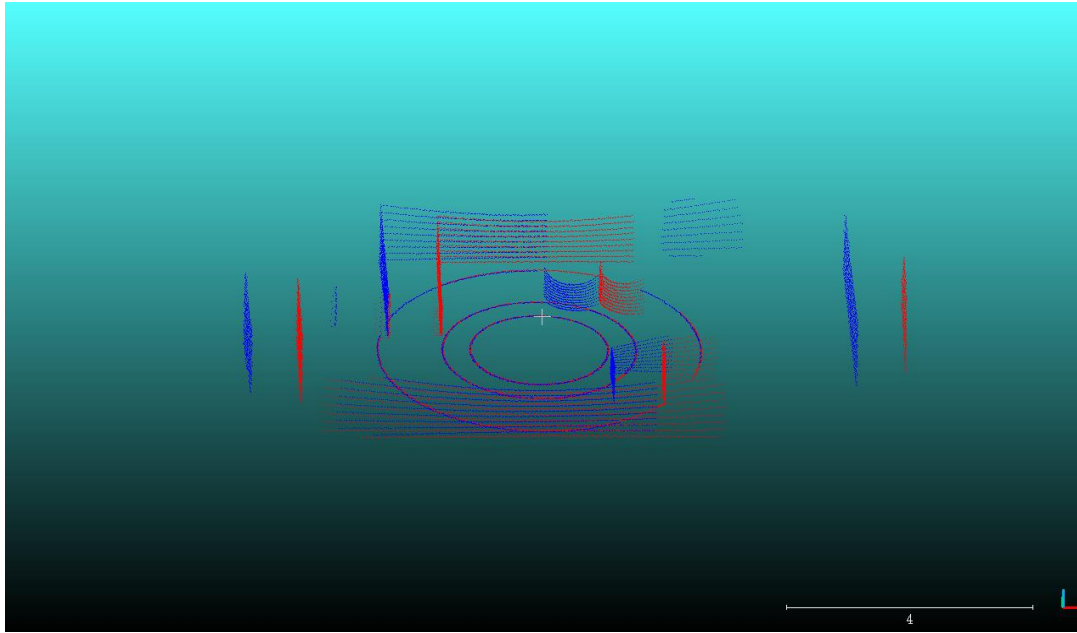


图 5-5 ICP 配准前

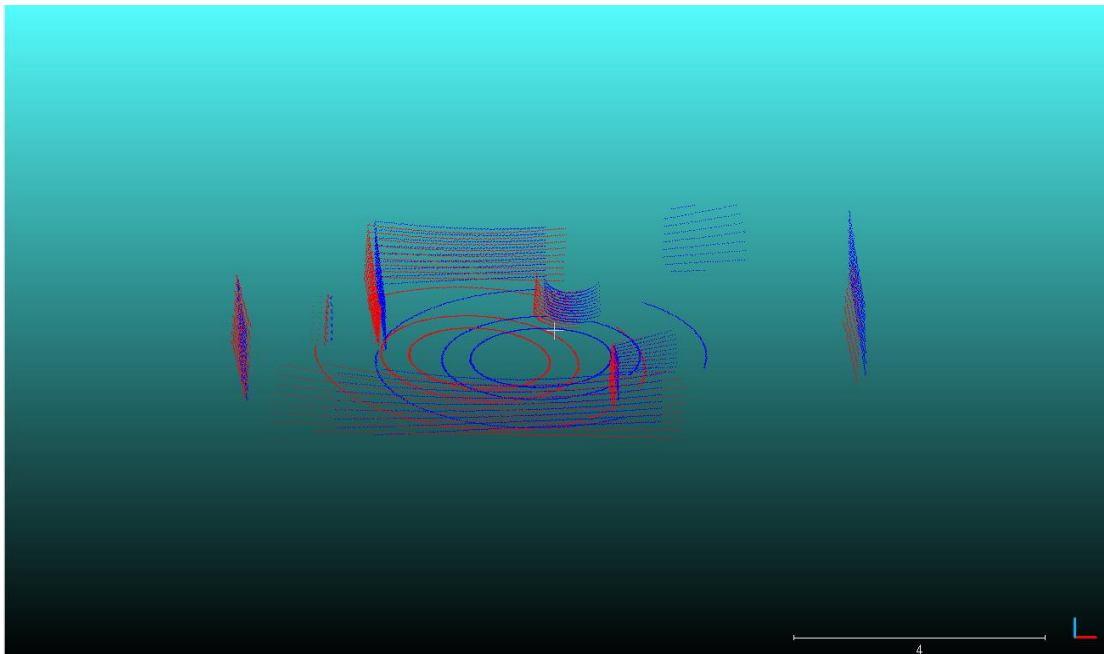


图 5-6 ICP 配准后

图 5-6 为传统 *ICP* 配准算法配准后的效果图，配准精度为 0.228912，变换矩阵为：

$$(R, t) = \begin{bmatrix} 0.995 & -0.069 & 0.068 & -0.885 \\ 0.070 & 0.997 & -0.020 & 0.074 \\ -0.066 & 0.025 & 0.997 & -0.031 \\ 0.000 & 0.000 & 0.000 & 1.000 \end{bmatrix} \quad (5.5)$$

从图中可以看出两点云基本已经重合，但是还存在一定差异。

点云配准的原始 *ICP* 算法存在一些不足之处。首先，它要求两个点集有重叠的区域，

这在某些情况下很难实现。此外，该算法的计算量大且速度慢。这是因为它采用了欧氏距离最近的点作为对应点，这可能导致不正确的对应。因此，获得的对应点可能包含一些错误。

5.3 基于特征点的 ICP 点云配准

从点云数据中有效地提取特征点对于实现快速、准确的点云对准至关重要。三维点云数据中的特征点可以被识别为一组在局部邻域表现出最大曲率变化的点，通常是在相邻表面的交点或角点。当点云数据量较小时，在配准前提取特征点可以极大地提高算法的速度。然而，对于大的点云数据或复杂的形状，特征点的提取会很耗时，导致配准算法的执行时间延长。为了解决这个问题，本文使用基于法向量的特征点提取方法，以准确有效地识别特征点。

本文提出了一种基于特征点的 *ICP* 配准算法，以实现精确的点云配准，并克服了传统 *ICP* 算法的低执行效率。在该算法中，目标点云中的特征点是通过确认该点邻域内法向量的变化幅度来提取的。其算法流程图如图 5-7 所示。

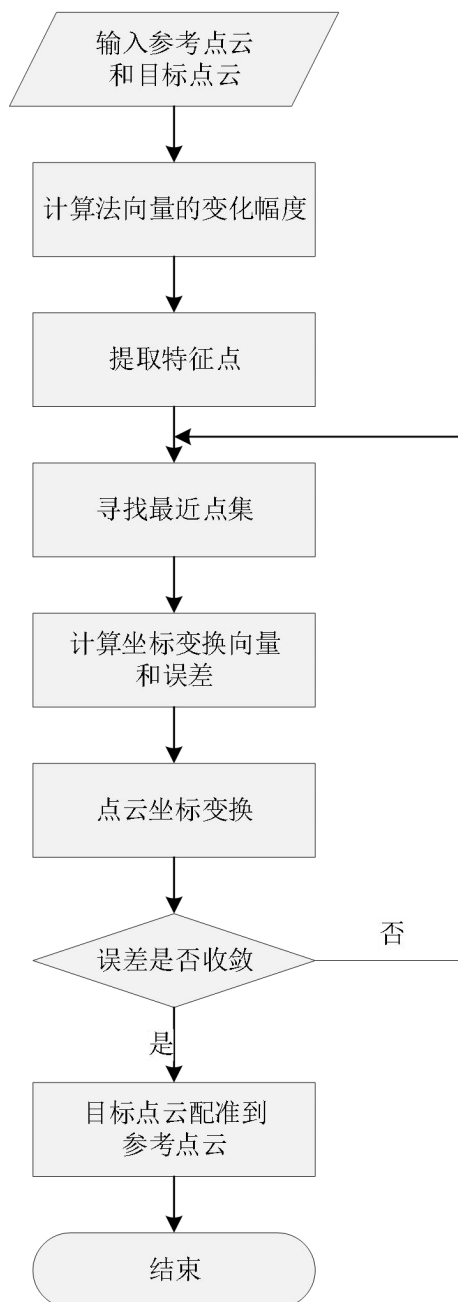


图 5-7 基于特征点的 ICP 点云配准流程图

实验结果分析：

为了验证本文提出的算法的适用性，选择上一节中的点云滤波过的数据作为实验数据集进行配准。首先，采用主成分分析法对滤波后的点云数据进行初步配准。然后对粗配准后的点云采用迭代最接近点（ICP）算法和本文提出的改进后的配准算法。最后，对实验结果进行分析，以评估所提出的算法的性能。

如图 5-8 所示，为基于传统 ICP 算法改进算法配准实验图：

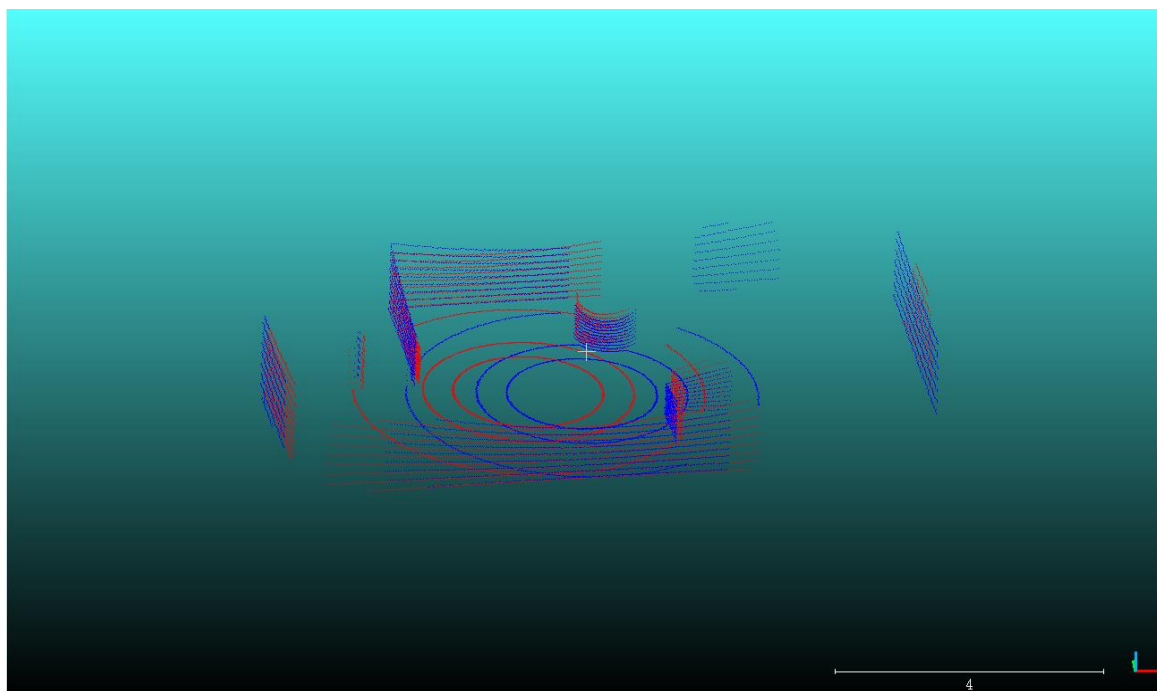


图 5-8 改进后的配准图

上图 5-8 为基于特征点的 *ICP* 算法配准效果图，配准精度为 0.191704，变换矩阵为：

$$(R,t)=\begin{bmatrix} 0.999 & -0.045 & 0.004 & -0.799 \\ 0.045 & 0.999 & -0.004 & 0.120 \\ -0.004 & 0.004 & 1.000 & -0.003 \\ 0.000 & 0.000 & 0.000 & 1.000 \end{bmatrix} \quad (5.6)$$

因为配准精度为迭代结束之后所有对应点间的距离平方和，配准效果越好，配准精度数值越小。由图中的配准效果和配准精度 $0.191704 < 0.228912$ ，得在配准效果上本算法优于 *ICP* 算法。

5.4 本章小结

本章首先分析了激光雷达扫描数据的噪声来源，介绍了统计滤波算法的关键滤波器 *SOR*，并采用 *SOR* 对轮式小车激光雷达点云进行了滤波处理。然后以传统 *ICP* 算法流程为分析对象，得出其缺点和不足之处。最后，提出了基于特征点的 *ICP* 配准算法，仿真实验结果表明 *ICP* 配准算法能够很好地实现点云的高精度配准。

第六章 总结和展望

6.1 全文总结

本文对雷达点云采集、点云滤波、点云配准等关键技术进行了较为深入的研究。从激光雷达点云数据特性及噪音建模入手，对其进行了理论分析。在此基础上，对基于 *ICP* 的点云数据融合方法进行了深入的研究，并对其优劣进行了分析。基于以上分析，本文给出了 *ICP* 算法的一些改进建议。最后，对已有的经典算法进行了仿真，并对仿真结果进行了分析。

现将本文所做的工作总结如下：

(1) 为了研究激光雷达和点云数据处理的研究现状，通过网络上查阅大量文献。根据研究中使用的相应算法，对所查阅的文献进行了分类和组织。

(2) 对激光雷达原理和点云处理进行研究，并对激光雷达的工作原理进行系统的介绍，接着介绍了统计滤波和点云配准的原理，最后对 *PCL* 点云库进行介绍。

(3) 介绍了激光雷达点云数据的采集方法，接着根据激光雷达点云的数据介绍了不同其不同的格式及各个格式的特点，最后对当前点云数据处理的步骤进行了总结。

(4) 通过 *ROS* 系统进行轮式机器人小车仿真，并采集 *VLP-16* 的激光雷达点云数据。

(5) 点云配准算法的研究比较和改进算法的提出。首先对点云配准算法进行了概述，对传统 *ICP* 点云配准算法进行了分析和比较，并总结了传统 *ICP* 点云配准算法的优缺点，最后提出了基于改进迭代最近点 (*ICP*) 算法的点云配准方法，并进行了实验进行比较分析。

6.2 未来展望

虽然本文对激光雷达点云的信号采集和数据处理技术进行了一定的研究，但是受本人的研究水平和时间的限制，本文仍然存在以下几方面需要改进之处：

(1) 滤波算法精度不高。本文采用的是 *SOR* 滤波器，主要是根据密度去除离群点，对密度差异较大的离群点去除效果较好，但对密度差异较小的离群点去除效果并不理想，所以还需对该算法进行改进。

(2) 配准算法的精度不高。虽然本文对配准算法的准确性进行了改进，但还需要进一步的研究来增强对准方法的普遍适用性，并提高起算数据的准确性。

(3) 配准算法的智能程度。在本文使用的算法中，点云数据滤波和配准过程的参数都是手动设置的。未来的研究可以探索结合机器学习技术来开发智能点云数据过滤和配准算法。

参考文献

- [1]党亚南,田照星,郭利强. 车载激光雷达点云数据处理关键技术[J]. 计算机测量与控制, 2022,30(01):234-238+245.
- [2]程川,宋春华,王鹏. 车载激光雷达发展研究综述[J]. 装备制造技术, 2022(05):247-251.
- [3]李宏宇. 激光雷达的点云数据处理研究[D]. 长春: 长春理工大学, 2019.
- [4]戴永江. 激光雷达原理[M]. 国防工业出版社, 2002.
- [5]余莹洁. 车载激光雷达的主要技术分支及发展趋势[J]. 科研信息化技术与应用, 2018, 9(06):16-24.
- [6]张小梅. 车载激光雷达测距方法及精度分析[D]. 电子科技大学, 2019.
- [7]Chen C S, Chen P C, Hsu C M. Three-dimensional object recognition and registration for robotic grasping systems using a modified viewpoint feature histogram[J]. Sensors, 2016,16(11): 1969-1971.
- [8]Chen Z, Zhang T, Cao J, et al. Point cloud resampling using centroidal Voronoi tessellation methods[J]. Computer-Aided Design, 2018, 102:12-21.
- [9]杨永涛,张坤,黄国言,吴培良. 邻域密度约束的动态标准差阈值三维点云数据离群点检测方法[J]. 计算机辅助设计与图形学学报, 2018, 30(06):1034-1045.
- [10]戴士杰,任永潮,张慧博. 各向异性扩散滤波的三维散乱点云平滑去噪算法[J]. 计算机辅助设计与图形学学报, 2018, 30(10):1843-1849.
- [11]Chen Z, Zhang T, Cao J, et al. Point cloud resampling using centroidal Voronoi tessellation methods[J]. Computer-Aided Design, 2018, 102:12-21.
- [12]Torabi M, Mousavi G S M, Younesian D. A new methodology in fast and accurate matching of the 2D and 3D point clouds extracted by laser scanner systems[J]. Optics & Laser Technology, 2015, 66: 28-34.
- [13]胡章芳,漆保凌 罗元,等. V-SLAM 中点云配准算法改进及移动机器人实验[J]. 哈尔滨工业大学学报, 2019.
- [14]李强,高保禄,窦明亮. 基于多重特征匹配的点云配准算法[J/OL]. 计算机应用研究: 1-7[2020-01-04].
- [15]杨清科,李永强,李立雪,孙渡,李鹏鹏,范辉龙. 车载联合机载点云数据的建筑物立面建模[J]. 测绘科学, 2019,44(02):94-101.
- [16]朱常兴. 激光雷达技术及其在自动驾驶领域的应用[J]. 自动化博览, 2019(12):54-59.
- [17]杨旭. 激光雷达点云数据的滤波和分割处理[D]. 哈尔滨: 哈尔滨工业大学, 2020.
- [18]王小清. 基于激光雷达三维点云数据的配准技术[D]. 南京: 东南大学, 2017.
- [19]李雨欣. 基于激光雷达的点云数据处理算法研究[D]. 长春: 长春理工大学, 2020.
- [20]艾卫涛. 三维激光扫描点云去噪及在建模中的应用[D]. 西安: 西安科技大学, 2017.
- [21]Wang X, Zhang X. Rigid 3D Point Cloud Registration Based on Point Feature Histograms[C]//2017 2nd International Conference on Machinery, Electronics and Control Simulation (MECS 2017). Atlantis

Press, 2016.

- [22]郭思猛. 三维点云的特征点提取与配准技术研究[D]. 绵阳: 西南科技大学, 2018.
- [23]程亚丽. 三维激光扫描点云数据配准算法研究[D]. 昆明: 昆明理工大学, 2017.
- [24]杜超. 基于激光雷达的点云数据处理研究[D]. 西安: 西安工业大学, 2022.
- [25]姜小磊. 车载激光雷达点云数据的分析与配准[D]. 长春: 吉林大学, 2019.

致 谢

行笔至此，我的本科论文撰写即将画上圆满的句号！在天津中德应用技术大学学习的快乐时光也步入尾声！时光飞逝岁月如梭，在不经意间，时间已经从指间悄悄飞过，还来不及慢慢品味研究生生活，就即将迈入下一段征程。在此，我真诚的向在本科学习生活中，对我指导、关心、帮助和支持的所有人表示衷心的感谢！

首先，我要感谢我的指导老师汪磊老师，汪老师理论功底深厚、治学态度严谨，为我的学习提供了很多的帮助。本文从选题、结构设计、初稿修改到最后成稿，都得到了汪老师的悉心指导，给予了我极大的关心和帮助，使我能够顺利的完成毕业论文的写作。他不辞辛劳的工作态度使我受益匪浅。

其次，我要感谢汽车与轨道交通学院的所有老师，同时，我也要感谢企业指导老师涂宁工程师给予的诸多技术指导。各位老师们在论文写作过程中给予了我很大的帮助，使我能够顺利的完成毕业论文。

然后，感谢四年来一起经历丰富多彩的大学室友，同班同学，志同道合的朋友们的陪伴和关心，日月如梭，最后敬我逝去的大学四年学习生活。

最后，我要感谢我的父母。正因为有家人们的鼓励和陪伴我才可以走到今天这一步。

感谢所有在我写作期间帮助我的老师、同学、父母，你们的帮助对我完成论文是不可或缺的，对此，我永远铭记在心！

附录

ROS 小车部分代码如下：

```

<launch>
  <arg name="x_pos" default="0.0"/>
  <arg name="y_pos" default="0.0"/>
  <arg name="z_pos" default="0.0"/>
  <param name="/use_sim_time" value="true" />
  <include file="$(find jubot_demo)/launch/gazebo_world.launch"/>
  <param name="robot_description" command="$(find xacro)/xacro --inorder $(find
jubot_demo)/urdf/mybot.xacro" />
  <node pkg="gazebo_ros" type="spawn_model" name="spawn_urdf" args="-urdf
-model mybot.xacro -x $(arg x_pos) -y $(arg y_pos) -z $(arg z_pos) -param robot_description"
/>
  <node name="robot_state_publisher" pkg="robot_state_publisher"
type="robot_state_publisher" />
</launch>

<?xml version="1.0" encoding="UTF-8"?>
<launch>
  <include file="$(find slam_bot)/launch/robot_description.xml"/>
  <arg name="world" default="empty"/>
  <arg name="paused" default="false"/>
  <arg name="use_sim_time" default="true"/>
  <arg name="gui" default="true"/>
  <arg name="headless" default="false"/>
  <arg name="debug" default="false"/>
  <include file="$(find gazebo_ros)/launch/empty_world.launch">
    <arg name="world_name" value="$(find jubot_demo)/world/room2.world"/>
    <arg name="paused" value="$(arg paused)"/>
    <arg name="use_sim_time" value="$(arg use_sim_time)"/>
    <arg name="gui" value="$(arg gui)"/>
    <arg name="headless" value="$(arg headless)"/>
    <arg name="debug" value="$(arg debug)"/>
  </include>

```

```
</include>
```

滤波部分代码:

```
pcl::StatisticalOutlierRemoval<pcl::PointXYZ> Static;
Static.setInputCloud(cloud);
Static.setMeanK(50);
Static.setStddevMulThresh(1.0);
Static.filter(*cloud_filtered);
std::cerr << "Cloud after filtering:" << std::endl;
std::cerr << *cloud_filtered << std::endl;
```

ICP 配准部分代码:

```
void
print4x4Matrix (const Eigen::Matrix4d & matrix)
{
    printf ("Rotation matrix :\n");
    printf ("    | %6.3f%6.3f%6.3f| \n", matrix (0, 0), matrix (0, 1), matrix (0, 2));
    printf ("R = | %6.3f%6.3f%6.3f| \n", matrix (1, 0), matrix (1, 1), matrix (1, 2));
    printf ("    | %6.3f%6.3f%6.3f| \n", matrix (2, 0), matrix (2, 1), matrix (2, 2));
    printf ("Translation vector :\n");
    printf ("t = < %6.3f, %6.3f, %6.3f>\n\n", matrix (0, 3), matrix (1, 3), matrix (2, 3));
}

void
keyboardEventOccurred (const pcl::visualization::KeyboardEvent& event,
                        void* nothing)
{
    if (event.getKeySym () == "space" && event.keyDown ())
        next_iteration = true;
}

int
main ()
{
    PointCloudT::Ptr cloud_in (new PointCloudT);
    PointCloudT::Ptr cloud_tr (new PointCloudT);
```

```

PointCloudT::Ptr cloud_icp (new PointCloudT);
int iterations = 1; // Default number of ICP iterations
pcl::console::TicToc time;
time.tic ();
if (pcl::io::loadPLYFile ("fish-2.ply", *cloud_in) < 0)
{
    PCL_ERROR ("Error loading cloud %s.\n");
    return (-1);
}
std::cout << "\nLoaded file " << "fish-2.ply" << " (" << cloud_in->size () << " points)
in " << time.toc () << " ms\n" << std::endl;
Eigen::Matrix4d transformation_matrix = Eigen::Matrix4d::Identity ();
double theta = M_PI / 8;
transformation_matrix (0, 0) = cos (theta);
transformation_matrix (0, 1) = -sin (theta);
transformation_matrix (1, 0) = sin (theta);
transformation_matrix (1, 1) = cos (theta);
transformation_matrix (2, 3) = 0.4;
std::cout << "Applying this rigid transformation to: cloud_in -> cloud_icp" <<
std::endl;
print4x4Matrix (transformation_matrix);
pcl::transformPointCloud (*cloud_in, *cloud_icp, transformation_matrix);
*cloud_tr = *cloud_icp;
time.tic ();
pcl::IterativeClosestPoint<PointT, PointT> icp;
icp.setMaximumIterations (iterations);
icp.setInputSource (cloud_icp);
icp.setInputTarget (cloud_in);
icp.align (*cloud_icp);
icp.setMaximumIterations (1);
std::cout << "Applied " << iterations << " ICP iteration(s) in " << time.toc () << "
ms" << std::endl;
if (icp.hasConverged ())
{

```

```
std::cout << "\nICP has converged, score is " << icp.getFitnessScore () <<
std::endl;
std::cout << "\nICP transformation " << iterations << " : cloud_icp -> cloud_in"
<< std::endl;
transformation_matrix = icp.getFinalTransformation ().cast<double>();
print4x4Matrix (transformation_matrix);
}
else
{
PCL_ERROR ("\nICP has not converged.\n");
return (-1);
}
```