

单位代码： 10293 密 级： 公开

南京邮电大学

# 专业学位硕士学位论文



论文题目： 基于激光雷达与视觉融合的 SLAM 系统  
设计与实现

学 号	1218012213
姓 名	陈鹏
导 师	李晓飞
专业学位类别	工程硕士
类 型	全日制
专业（领域）	电子与通信工程
论文提交日期	二〇二一年四月

# **Design and Implementation of SLAM System Based on Lidar and Vision Fusion**

Thesis Submitted to Nanjing University of Posts and  
Telecommunications for the Degree of  
Master of Engineering



By

Peng Chen

Supervisor: Prof. Xiaofei Li

April 2021

## 摘要

随着移动服务机器人产业的快速发展，其应用场景也变得更加复杂多样，凭借单一传感器进行同时定位与地图构建难以满足应用需求，采用多个传感器进行准确定位及建图已成为该领域的研究热点之一。为了克服二维激光雷达和深度相机传感器在同时定位与地图构建（SLAM）领域存在的技术问题，本文结合上述两种传感器的优劣情况，研究并实现基于激光雷达与视觉信息融合的 SLAM 算法。

本文首先针对传统 Gmapping 算法中采样粒子数固定不变导致系统资源浪费和构建地图精度较低的问题，提出一种基于 Gmapping 的可变粒子数算法。该方法能够根据环境的复杂程度动态调整系统中采样粒子数，通过大量实验找到环境复杂度和采样粒子数之间的对应关系。在 MATLAB 中仿真得到改进后的 Gmapping 算法中移动机器人运行时间更短，运动轨迹更接近真实值，提升了定位精准度和效率。其次，以开源 ORB-SLAM2 算法为基础提出一种实时栅格地图构建方法，该算法通过构建逆传感器模型结合占据栅格地图模型得到用于导航的地图，通过实验移动机器人能在现实会议室场景中构建出清晰准确的栅格地图。最后，提出了激光雷达与深度相机融合的 SLAM 方案。该方法的数据融合采用贝叶斯估计，将基于 ORB-SLAM2 算法与改进 Gmapping 算法构建的局部地图进行融合。在 KITTI 数据集上得到该融合方案的运动总累计误差最大值为 25.518332、均值为 12.312251，定位效果显著优于主流的单一传感器方法；在实际室内环境下进行单一传感器和融合 SLAM 之间的对比实验，融合后的 SLAM 方案得到更加全面准确的栅格地图，验证了融合方案的有效性和实用性，达到了预期设计要求。

**关键词：** 同时定位与地图构建，数据融合，激光雷达，深度相机

# Abstract

With the rapid development of the mobile service robot industry, its application scenarios have become more complex and diverse. Simultaneous positioning and map construction with a single sensor is difficult to meet application requirements. The use of multiple sensors for accurate positioning and mapping has become a research in this field. In order to overcome the technical problems of the two-dimensional lidar and depth camera sensors in the SLAM field, this thesis combines the advantages and disadvantages of the above two sensors to study and implement the SLAM algorithm based on the fusion of lidar and visual information.

Firstly, aiming at the problem that the fixed number of sampling particles in the traditional Gmapping algorithm leads to the waste of system resources and the low accuracy of map construction, this thesis proposes a variable particle number algorithm based on Gmapping. This method can dynamically adjust the number of sampled particles in the system according to the complexity of the environment. Through a large number of experiments to find the correspondence between the complexity of the environment and the number of sampled particles. The simulation in MATLAB shows that the motion trajectory of the mobile robot in the improved Gmapping algorithm is closer to the real value, which improves the positioning accuracy. Secondly, based on the open source ORB-SLAM2 algorithm, a real-time grid map construction method is proposed. The algorithm obtains a map for navigation by constructing an inverse sensor model combined with an occupancy grid map model, so that the mobile robot constructs a clear and accurate grid map in the real meeting room scene. Finally, a SLAM scheme combining lidar and depth camera is proposed. The data fusion of this method adopts Bayesian estimation, and the local map constructed by fusion of ORB-SLAM2 algorithm and improved Gmapping algorithm. On the KITTI data set, the maximum total cumulative error of motion max: 25.518332, mean: 12.312251, the positioning effect is significantly better than the mainstream single sensor method; the comparison experiment between single sensor and fusion SLAM is carried out in the actual indoor environment, and the fusion SLAM scheme gets a more accurate grid map, which verifies the effectiveness and practicability of the fusion scheme, and meets the expected design requirements.

**Key words: SLAM, Data Fusion, Lidar, Depth camera**

# 目录

第一章 绪论 .....	1
1.1 课题研究背景及意义 .....	1
1.2 国内外研究现状 .....	2
1.2.1 机器人研究现状 .....	2
1.2.2 SLAM 研究现状 .....	4
1.2.3 多传感器融合研究现状 .....	6
1.3 本文研究内容及章节安排 .....	7
第二章 系统建模与传感器数据预处理 .....	9
2.1 机器人运动模型及里程计标定 .....	9
2.1.1 机器人运动模型 .....	9
2.1.2 里程计标定 .....	11
2.2 激光雷达数学模型及运动畸变去除 .....	12
2.2.1 激光雷达数学模型 .....	14
2.2.2 运动畸变去除 .....	15
2.3 相机数学模型及内参标定 .....	16
2.3.1 相机数学模型 .....	17
2.3.2 相机内参标定 .....	19
2.4 激光雷达与相机外参标定 .....	20
2.5 本章小结 .....	22
第三章 基于激光雷达的 SLAM 算法研究 .....	23
3.1 同时定位与地图构建理论基础 .....	23
3.1.1 SLAM 数学模型 .....	23
3.1.2 SLAM 分类 .....	24
3.1.3 地图表示 .....	26
3.2 基于滤波器的激光 SLAM 算法研究 .....	27
3.2.1 贝叶斯滤波算法研究 .....	27
3.2.2 粒子滤波算法研究 .....	29
3.2.3 RBPF-SLAM 算法研究 .....	30
3.2.4 Gmapping 算法研究 .....	31
3.3 改进的 Gmapping 算法与实验分析 .....	34
3.3.1 基于可变粒子数的 Gmapping 算法 .....	34
3.3.2 算法仿真实验 .....	35
3.4 本章小结 .....	37
第四章 基于 Kinect 的视觉 SLAM 算法研究 .....	38
4.1 视觉 SLAM 框架 .....	38
4.1.1 视觉里程计 .....	38
4.1.2 后端优化 .....	39
4.1.3 回环检测 .....	39
4.1.4 地图构建 .....	39
4.2 ORB-SLAM2 算法研究 .....	39
4.2.1 特征追踪 .....	40
4.2.2 局部优化 .....	40
4.2.3 回环检测 .....	41

4.3 基于 ORB-SLAM2 的栅格地图构建 .....	41
4.3.1 ORB-SLAM2 建图总体方案 .....	42
4.3.2 逆传感器建模 .....	42
4.3.3 栅格地图模型 .....	44
4.3.4 实验与结果分析 .....	45
4.4 本章小结 .....	45
第五章 基于激光雷达和视觉融合的 SLAM 算法研究 .....	47
5.1 多传感器信息融合 .....	47
5.1.1 信息融合层次 .....	47
5.1.2 融合方法分析 .....	48
5.2 激光雷达和视觉融合方案 .....	50
5.2.1 贝叶斯估计融合法 .....	50
5.2.2 激光雷达与深度相机融合方案 .....	50
5.2.3 融合算法仿真实验 .....	52
5.3 实验平台搭建 .....	53
5.3.1 硬件环境 .....	54
5.3.2 软件环境 .....	54
5.4 实验和结果分析 .....	55
5.5 本章小结 .....	58
第六章 总结与展望 .....	59
6.1 全文总结 .....	59
6.2 研究展望 .....	59
参考文献 .....	61
附录 攻读硕士学位期间申请的专利 .....	63
致谢 .....	64

# 第一章 绪论

## 1.1 课题研究背景及意义

近年来,随着物联网、大数据、5G、人工智能和计算机、传感器、仿生学、新材料等领域的快速推动,智能服务机器人进入了迅猛发展的黄金时期<sup>[1]</sup>。机器人领域正从之前的工业生产和制造方向朝着家庭服务、医学手术辅助、教育培训、探索搜救、军事国防等智能服务机器人领域不断发展<sup>[2-3]</sup>。机器人作为人工智能的一个重要载体,是一个国家科研、创新以及工业生产是否强大的重要衡量标准,现如今已成为世界各国科技创新和工业生产的着眼点<sup>[4]</sup>。

随着人们生活节奏加快、生活品质提升,智能服务机器人已逐步走进人们的生活中<sup>[5]</sup>。智能陪护机器人如图 1.1 所示,可用于社区服务中心、老年公寓以及敬老院等场所,能够与其它生理采集传感器结合,进行特定生命体征信号的检测与监控,如心率血压检测、摔倒检测等,通过智能语音交互,社区诊断、远程医疗等形式,居家智能服务机器人将为老年人的居家养老生活带来了极大的便利,有效地解决了人口老龄化带来的一系列社会问题<sup>[6-7]</sup>;消防机器人如图 1.2 所示,能在特定场景中,帮助消防救援人员深入环境恶劣甚至危险的事故现场进行灭火,同时通过数据获取、分析、上传,快速高效地处理救火队员在事故现场遇到的安全隐患以及数据缺失不全等问题。消防指挥中心对上传的结果加以分析,及时对火情作出科学判断和正确合理的决策<sup>[8]</sup>;扫地机器人、教育机器人以及迎宾机器人必将在不久的将来为人们提供有效的高质量服务,为人们生活增添乐趣。

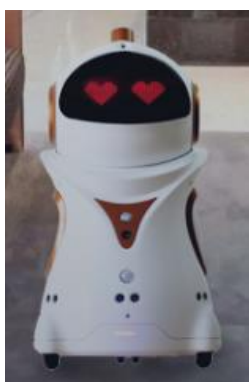


图 1.1 陪护机器人



图 1.2 消防机器人

国际机器人联合会(IFR)结合机器人不同的使用场景,将其主要划分成服务型机器人和生产型机器人两类<sup>[9]</sup>。其中,生产型机器人表示广泛应用在工业生产领域的辅助机器人和移动机器人。在《2020 年世界机器人报告》中表示,有两百多万台工业机器人服务于全球相关

企业，发展势头迅猛。新机器人的销量依然比较可观，2019 年各个国家总出货量为 37 万余台。机器人目前已经进入到 2.0 时代，移动机器人在其自主判断、运动控制以及人机交互等方面取得明显进展，已经融入我们生活的每个角落。

2020 年疫情以来，快递物流行业出现了一段迅猛增长。按照国家邮政局提供的信息表示，从 2020 年从 1 月到 11 月，全国快递业务量高达七百多亿件，同比增长三分之一。自动导引车（Automated Guided Vehicle, AGV）在智能化仓储配送中发挥了至关重要的作用，代替人类进行长时间的重复劳动，有效地缓解了快递存储压力<sup>[10]</sup>。实际应用环境中 AGV 小车大多提前设置路线，通过电磁、磁带以及视觉信息完成导航，但是任务结束后需要重新布置电磁和磁带，导致投资成本加大。随着应用的推广和技术的逐渐成熟，AGV 将开始向自主化、智能化、便于维修保养等方向发展。同时定位和构建地图（SLAM）技术成为当前许多公司以及专家学者们研究的重点问题之一。自主定位导航主要包括搭载激光雷达传感器的激光 SLAM 和搭载相机传感器的视觉 SLAM 两大类<sup>[11]</sup>。其中 2D 激光 SLAM 多用于扫地机器人、巡逻机器人以及迎宾机器人等室内场景，3D 激光 SLAM 常用于需要三维环境信息的无人驾驶领域。激光雷达相比视觉传感器扫描分辨率更高、处理速度更快、抗干扰能力更强，非常适合在移动机器人上完成自主定位导航。视觉 SLAM 中相机传感器因为小巧便携、结构简单、成本低等优点推动了近几年视觉 SLAM 的飞速发展<sup>[12]</sup>。

随着智能服务机器人应用场景的不断扩大以及环境更加复杂多样，只采用单一传感器进行同时定位与地图构建存在很大局限性，具体表现在缺失部分环境信息从而造成地图不完全准确。激光 SLAM 中使用 2D 激光雷达只能扫描到一个平面内的环境信息，且遇到玻璃墙面时无法探测距离，而 3D 激光雷达因其价格昂贵导致短时间内很难大规模应用；视觉 SLAM 中传感器受环境光的影响较大，在光线暗处或无纹理区域很难工作<sup>[13-14]</sup>。基于此，本课题在保证准确性和低成本的情况下融合 2D 激光雷达和视觉信息构建更加完整精准的地图。期待随着智能服务机器人多传感器信息的不断融合和发展，同时定位与地图构建技术将日趋成熟和完善。

## 1.2 国内外研究现状

### 1.2.1 机器人研究现状

自美国斯坦福研究所查理·罗森（Charlie Rosen）等人 1956 年开始经过六年研究制造出世界第一台移动机器人 Shakey 算起，移动机器人已经发展超过五十年。Shakey 如图 1.3 所示，



搭载了控制电机、视觉相机、防碰撞传感器等,运用人工智能相关技术对环境进行分析建模、设定动作并完成简单任务。虽然 Shakey 不如目前移动机器人性能优越,但在研究它的过程中收获了许多宝贵经验,为后续机器人的自主定位导航以及路径规划等领域奠定了基础<sup>[15]</sup>。美国 Cincinnati Milacron 公司于 1973 年制造出机器人 T3,完成了小型计算机和机器人的首次合作。1978 年美国 Unimation 公司制造出一款符合行业标准的工业机器人 PUMA,至今仍应用于工厂生产。1997 年美国宇航局研制出“探路者”号火星车,此后又不断研制了“机遇号”、“勇气号”、“好奇号”并送上了火星。

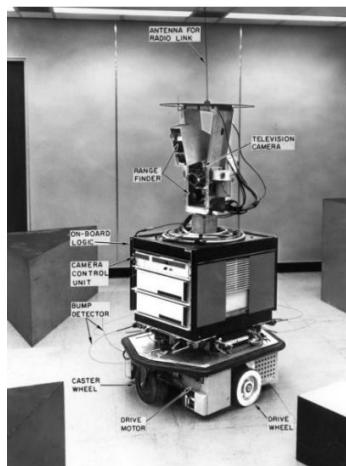


图 1.3 Shakey 移动机器人

日本对于仿人形机器人领域的研究处于世界领先水平<sup>[16]</sup>。1973 年,日本早稻田大学(Waseda University)加藤一郎教授发明了全球首台带双脚的机器人 WABOT-1,配备了机械手脚、人工视觉以及听觉等传感器,外型与目前的人形机器人十分相似。WABOT-1 可以借助嘴巴交流常用日语,通过眼睛(视觉)感知距离和方向,依靠双脚双手进行移动以及搬运物体。之后,日本的本田公司研发出一台仿生机器人 ASIMO,它能够识别周围移动的物体,并根据内置传感器计算与目标之间的距离和方向,及时避让附近障碍物的同时保持自身平衡<sup>[17]</sup>。小朋友们可以根据乐高公司推出的机器人套装自行设计各种像人、动物甚至会飞的机器人,然后将每个小模块像积木一样拼出来,并且通过编程让机器人实现各种预设的动作。这一好玩有趣的产品使机器人开始走进儿童的世界。这几年,软银旗下的波士顿动力学公司一直不断研制出许多令人震撼的机器人。Spot 是该公司生产的一款敏捷的小型四足机器人,充满电后能够运行一个半小时,并且几乎不存在噪音。该机器人可以承受摔倒、被脚踢,在崎岖的地形上行走以及出现在轮式机器人无法到达的地方,因此可以帮助人们从事很多危险工作。

国内机器人相比西方国家起步较晚,1986 年中央计划将移动机器人作为国家自动化方向高新技术发展的重点之一,希望学习发达国家先进水平,主要以特种机器人作为研究的关键

<sup>[18]</sup>。1992 年国产自动导引车 (AVG) 首次应用于柔性生产线。我国对于机器人技术的研究大多集中在高校以及相关的实验团队, 北京理工大学在二十一世纪初开始研究仿人机器人, 不断推出了“汇童”1 到“汇童”6 仿人机器人, 在传统仿生结构的基础上能够实现机器人的稳定运动控制, 结合复杂的动作和有趣的表情。该仿人机器人能够表演中国的传统武术太极拳, 与真人进行乒乓球对打<sup>[19]</sup>; 2009 年大连理工大学副教授张永顺团队发明胶囊医疗微型机器人, 该机器人能够在肠道内自由活动、窥视、判断病因甚至喂药, 减轻患者病痛的同时加快了康复的时间; 2014 年南京理工大学研制出全球首台飞行吸附机器人, 该机器人能够长时间飞行和悬空, 能快速移动到目标位置, 实现在空间壁面不断吸附的任务, 可用在车站、机场以及大型广场等公共场所进行目标的监控和追踪; 2019 年天津大学王树新教授团队与中南大学共同合作研制出具有自主知识产权的“妙手 S”手术机器人, 使得医生在手术操作过程中更加方便和快捷, 手术质量也得到明显提升<sup>[20-21]</sup>。

随着国家对于智能机器人行业的大力支持以及人民对美好生活的追求, 越来越多的公司研发出各种各样的优秀机器人<sup>[22]</sup>。2018 年深圳优必选公司推出智能巡检机器人 ATRIS (安巡士), 以优必选的 U-SLAM 方案为基础, 结合 GPS、激光雷达以及深度摄像头等多个传感器进行融合, 用户可以自定义机器人的起点、运动路线以及行动时间, 实现更加准确的定位。ATRIS 具有强大的人脸识别系统, 能够对识别到的行人进行视觉追踪以及年龄性别的区分, 将识别的人脸与数据库进行对比从而找到可疑分子并产生告警<sup>[23]</sup>。

### 1.2.2 SLAM 研究现状

机器人想要通过自主移动来完成各种任务的前提就是在未知的场景下准确地构建可用于导航的地图, 所以机器人领域的专家学者和优秀企业一直都在致力于机器人 SLAM (同时定位与地图构建) 问题的研究。上个世纪八十年代末, Smith、Cheeseman 以及 Durrant-Whyte 团队对空间不确定性的表示和估计成为 SLAM 问题研究的开端。最开始发展的是基于激光雷达传感器的一系列概率滤波方法和后续出现的图优化方法, 主要包括 EKF (扩展卡尔曼滤波)、RBPF (粒子滤波)、FastSLAM、Gmapping、Karto-SLAM 以及 Cartographer 等。由于卡尔曼最初提出的滤波理论只适用于线性系统, 所以 Bucy, Sunahara 等人提出并研究了 EKF (扩展卡尔曼滤波)<sup>[24]</sup>。EKF 对非线性系统的函数进行泰勒展开, 忽略二次项及以上高阶项, 从而解决非线性问题并将 KF 理论引入非线性领域。然而 EKF 也带来了两个缺点, 第一是高度非线性系统不满足局部线性的条件, EKF 算法可能会使滤波不能收敛; 第二是在进行线性化过程要用 Jacobian 矩阵, 造成计算量大以及系统实时性差<sup>[25]</sup>。Hammersley 等人提出基于顺序重要

性采样 (SIS) 的粒子滤波 SLAM 方法, 通过从条件概率中抽取粒子来代表概率分布, 适用于所有状态下的空间模型。早期的 RBPf 算法是 Murphy、Doucet 等学者将 Rao-Blackwellization 算法与粒子滤波相结合, 通过降维来减少计算复杂度, 用于研究 SLAM 问题<sup>[26]</sup>。二十一世纪初, Montemerlo 等人首次将 Rao-Blackwellised PF (粒子滤波) 与 EKF 相结合进行机器人 SLAM 的研究, 也就是后来开源的 FastSLAM 算法<sup>[27]</sup>。这种方案是最开始就能实时输出栅格地图, 它将 EKF 和粒子滤波进行优势互补, 在保证较低的计算复杂度情况下具有广泛的适用性和稳定性, 极大地促进了 SLAM 技术的发展。2007 年, Giorgio Grisetti 等人提出一种实时 SLAM 的算法 Gmapping, 它的传感器采用激光雷达结合里程计数据, 加入自适应重采样技术解决粒子耗散的问题, 开辟了 2D 激光 SLAM 的里程碑<sup>[28]</sup>。随后 Blanco J L 等人在此基础上形成 Optimal RBPf-SLAM 方案, 一个粒子在传播结束后得到 N 个粒子, 然后取 N 个中的最优粒子传递下去, 有效避免 Gmapping 算法中最优粒子耗散的问题。

Konolige K 等人在 2010 年首次提出基于图优化框架的算法 Karto SLAM, 该方案通过非线性最小二乘法减少 SLAM 过程累计的误差, 能够基本达到滤波器 SLAM 方案的效果<sup>[29]</sup>。该方法如果直接进行全局匹配, 遇到大范围环境时系统处理效率将严重下降。2011 年, Kohlbrecher S 等人提出了 Hector-SLAM 方案, 将激光雷达传感器得到的每一帧数据利用高斯牛顿方法和地图相匹配, 里程计信息通过激光雷达数据进行估算, 所以需要激光雷达具有较高的扫描频率<sup>[30]</sup>。在建图过程中如果机器人运动速度过大或者大幅度旋转时, 机器人位置会发生偏移导致建图失败。Hector-SLAM 方案构建出来的地图效果相较 Gmapping 算法更好, 但是其对激光雷达的性能要求更严格, 适用于对地图要求较高的场合。Google 于 2016 年在 Karto-SLAM 的基础上进行优化提出 Cartographer 开源方案, 该方案的关键点是根据多源数据创建局部子图和闭环检测模块所涉及的配准算法。Cartographer 前半部分扫描匹配算法是通过相关扫描匹配与梯度优化共同来完成每个子地图完成后对应一次局部闭环检测; 当结束所有子地图构建, 利用分枝定界和占据栅格算法, 从而完成全局的闭环检测。

二十一世纪初, 随着计算机视觉的发展, 视觉传感器中相机具有技术成熟、生产成本低以及机器人装配便捷等优点, 越来越多的专家学者开始深入研究视觉 SLAM。相比激光雷达传感器, 视觉传感器在现实场景中因其复杂的结构和优秀的性能采集到更加丰富全面的环境数据, 同时能够更好地表现和还原周围的真实环境。在众多视觉 SLAM 方案中, 主要涉及到的视觉传感器是单目相机、双目相机以及 RGB-D 深度相机。2007 年, 视觉 SLAM 领域的开拓者 A.J. Davison 教授首次提出基于单目相机的实时视觉 SLAM 系统—MonoSLAM, 该算法主要结合 EKF (扩展卡尔曼滤波) 进行后端优化; 2007 年, Klein 和 Murray 等人提出 PTAM (Parallel

Tracking and Mapping) 视觉 SLAM 算法, 该算法引入关键帧机制, 通过几个关键图像进行串联, 对其轨迹和地图进行优化。PTAM 首次采用前端和后端分开的双线程模式, 为后续视觉 SLAM 的发展奠定基础; 2015 年, Raul Mur-Artal 等人在 PTAM 的基础上提出一个基于特征点的实时单目的 ORB-SLAM (Oriented FAST and Rotated BRIEF SLAM) 方案, 将实时追踪特征点的 Tracking、局部 Bundle Adjustment (光束法平差) 的优化以及全局 Pose Graph 的回环检测与优化三个线程并行实现完整的 SLAM 过程。后续的 ORB-SLAM2 在此前基础上, 又新增双目相机和 RGB-D 相机两种模式, 具有良好的鲁棒性。直接法视觉 SLAM 中最具代表性的是 J.Engle 提出的 LSD-SLAM (Large Scale Direct monocular SLAM) 方案, 该算法成功重建出现实环境的半稠密地图。

国内对基于激光雷达的 SLAM 研究也在不断发展, 浙江大学的何衍等人提出添加约束的 EKF-SLAM 算法, 在传统 EKF-SLAM 算法的短时间内获得低精度估计, 将全局先验方向对该估计结果的约束和人工测量信息结合, 在减少计算时间的同时极大提高了估计精度; 天津理工大学的梁雪慧等人提出一种改进的 Fast-SLAM 算法, 使用自适应粒子群优化算法替换原算法中的粒子滤波部分<sup>[31]</sup>。利用粒子的后验位姿提议分布改善算法开始粒子的分布, 更新粒子的全局和历史最优, 提升粒子的质量, 从而在 SLAM 估计精度和计算效率方面都得以改善; 天津大学的郑兵等人针对传统 Gmapping 算法出现因粒子耗散导致定位准确度下降的问题提出一种萤火虫算法优化 Gmapping 的方案, 粒子采样过程中加入萤火虫算法, 分别处理高似然和低似然区域的粒子, 使粒子逼近真实状态, 在解决粒子退化问题的同时提升定位的精准度; 哈尔滨工业大学的朴松昊等人提出在特征点比较的前一步引入模糊 K-均值聚类来改进传统的 ORB-SLAM 算法, 该方法把像素聚成 k 类具有相似像素点的区域, 有效提升了 ORB 特征点匹配精准度和匹配效率<sup>[32]</sup>; 西安工程大学雷皓玮等人提出一种混合 SLAM 算法, 通过分析采样传感器获取到的图像有针对性地采取不同的方法进行定位与建图, 从而提升 SLAM 系统的稳定性以及位姿精度<sup>[33]</sup>。同时加入目标检测功能, 使系统可以自主确定运动物体的坐标位置, 在估计机器人状态之前将去除对应的移动像素点, 从而减轻移动物体对整个 SLAM 系统的干扰。

### 1.2.3 多传感器融合研究现状

多传感器数据融合技术能够一定程度的降低系统的不确定性, 根据不同传感器的观测得到准确的结果和多角度信息, 已经在移动机器人领域获得普遍关注和广泛应用, 多传感器融合的 SLAM 将是一种必然的趋势。移动机器人要能够随时自主感知周围环境的变化, 利用各

个传感器之间冗余和互补的特性来获取更加丰富的环境信息<sup>[34]</sup>。随着多传感器融合 SLAM 技术的不断发展,越来越多的国内外专家学者在这个领域做出了研究和贡献。2011 年 O.Aycard、Q.Baig 等人提出一种基于双目相机和激光雷达融合的 SLAM 方案;2015 年 Ji Zhang、Sanjiv Singh 等人提出了视觉里程计和激光雷达数据相结合的系统,一方面改进激光雷达的扫描匹配算法,提升了点云匹配的效果,另一方面从视觉里程计的角度增加了机器人运动估计的准确度;2018 年 Y.Nava 等人利用激光雷达和深度相机传感器,通过模式选择和信息融合技术优化 SLAM 的跟踪模块,从而在室内场景得到精确的地图;2016 年, Xiao Liang、Haoyao Chen 等人提出室内场景下的激光雷达和视觉融合方案,在激光雷达传感器的基础上加入基于 ORB 特征的视觉回环检测模块,有效提升了系统的稳健性;2018 年 Silva、Roche 等人发表无人驾驶领域激光雷达与视觉传感器融合(激光雷达传感器辅助视觉 SLAM)进行环境感知的论文,得到两个传感器之间的转换关系,采用高斯过程回归的算法进行数据插值,解决了分辨率匹配的问题;2019 年 W.Shao、S.Vijayarangan 等人提出了双目视觉惯性激光雷达(VIL)SLAM,该算法通过将紧耦合的双目视觉惯性里程计(VIO)、激光雷达建图以及辅助的视觉闭环检测功算法融合,能够在退化的粒子中表现更好,能够实时生成回环修正的雷达姿态和稠密地图,展示出更高的准确性和鲁棒性<sup>[35]</sup>。

张恒等人提出一种激光雷达与 Kinect 深度相机融合的 SLAM 方法,先对 Kinect 深度数据进行降维处理,接着使用卡尔曼滤波对激光数据与降维后的二维数据进行融合,最后使用贝叶斯估计将激光雷达与 Kinect 各自生成的二维局部栅格地图进行融合<sup>[36]</sup>;西安交通大学的樊建彪等人提出了基于磁力计、惯性测量单元(IMU)以及单目视觉传感器融合的 SLAM 算法,用以估计视觉尺度因子、重力、速度、IMU 参数和载体姿态,解决单目视觉在解算位姿时因纯旋转而出现漂移的情况,同时具有较高的精准度和广泛的适应性<sup>[37-38]</sup>。当加入视觉传感器后,匹配深度相机采集的三维数据和环境图像,用激光雷达和 Kinect 相机融合后的信息作为贝叶斯滤波的观测值来更新地图。

### 1.3 本文研究内容及章节安排

SLAM(同时定位与地图构建)技术对于移动机器人、无人车以及其他智能移动体的自主性至关重要。目前应用最广泛的就是激光雷达 SLAM 和视觉 SLAM,但是只采用单一传感器进行同自主定位导航时定位与地图构建存在很大局限性。在实际复杂环境中因缺失部分环境信息导致建图不准确,激光 SLAM 中使用 2D 激光雷达只能扫描到一个平面内的环境信息,在又长又直、两侧是墙壁或带有玻璃墙面的环境时性能较差,而 3D 激光雷达因其价格昂贵导致

短时间内很难大规模应用；视觉 SLAM 中传感器受环境光的影响较大，在光线暗处或无纹理区域（比如洁白的墙面）很难工作。基于此，本课题在保证准确性和低成本的情况下融合 2D 激光雷达和视觉信息构建更加精准的栅格地图。本文具体的章节安排如下：

第一章：绪论。本章主要介绍了论文的研究背景及意义，然后调研了国内外对于机器人、同时定位与地图构建以及多传感器融合等领域的研究现状，最后提出了本文主要的研究内容和各章节安排。

第二章：系统建模与传感器数据处理。本章主要描述里程计的运动学模型以及使用直接线性法标定里程计，然后给出激光雷达的数学模型以及使用传感器辅助法去除运动畸变，最后介绍视觉相机的数学模型、内参标定以及激光雷达和相机的外参标定。

第三章：基于激光雷达的 SLAM 算法研究。本章主要介绍了 SLAM 的相关内容，然后研究了几种经典的基于滤波器的 SLAM 算法，在 Gmapping 算法的基础上提出粒子数的自适应方案，最后对上述的 SLAM 算法进行实验以及分析。

第四章：基于 Kinect 的视觉 SLAM 算法研究。本章主要介绍了视觉 SLAM 的框架，然后对 ORB-SLAM 算法进行仔细研究以及实验分析，最后在 ORB-SLAM2 算法的基础上构建可用于导航的二维栅格地图。

第五章：激光雷达与视觉融合 SLAM 及实验分析。本章主要介绍了多传感器信息融合的相关内容，然后基于贝叶斯估计的融合方法提出激光雷达和视觉 Kinect 的融合方案，最后介绍实验平台环境以及实验结果的分析。

第六章：总结与展望。本章主要是归纳总结前面五章的内容，然后分析本文对于移动机器人 SLAM 领域做出的工作以及还可以继续改进的地方，最后针对不足提出解决想法。

## 第二章 系统建模与传感器数据预处理

移动机器人在空间中运动需要正确的位姿估计，同时后续的地图构建以及自主导航也对系统中的传感器精度有着很高的要求。里程计的机械标称值和实际值之间存在差距，需要对其标定以后使用。激光雷达的数据获取需要一定时间，造成的运动畸变将导致数据与真值之间出现误差，影响后续扫描匹配的精度，因此去除激光雷达的运动畸变显得尤为重要。为了使深度相机采集的数据更为准确，需要根据相机数学模型分析产生畸变的原因，从而计算内部参数和畸变参数来减小出现的误差<sup>[39]</sup>。激光雷达与视觉融合的 SLAM 系统需要这两个传感器之间建立严格的数学关系，通过两者的联合标定得到激光雷达坐标系与相机坐标的转换关系，为最终的融合提供基础。

### 2.1 机器人运动模型及里程计标定

本文采用实验室现有的机器人移动底盘 Yunjin Kobuki 如图 2.1 所示，是韩国 Yunjin 公司开发的两轮差分底盘的移动机器人平台。Yunjin Kobuki 移动底盘配备两个驱动轮和两个车轱辘，单轴陀螺仪，碰撞、跌落以及悬崖传感器，高精度编码器等。该底盘支持 ROS（机器人操作系统）、支持 Linux 及 Windows 操作系统，因其结构简单、价格便宜、模型简单以及可扩展性受到广泛使用。



图 2.1 Kobuki 移动底盘

#### 2.1.1 机器人运动模型

Kobuki 两轮差速底盘如图 2.2 所示是一个欠驱动系统，有两个驱动轮位于底盘左右两侧分别独立控制速度，同时在底盘前后配有两个辅助支撑轮。移动底盘的运动是耦合的，两个驱动轮以  $r$  为半径  $o$  为圆心做圆弧运动，左右两轮距底盘中心的距离为  $d$ ，左右两轮的线速度

和角速度分别为 $v_L$ 、 $v_R$ 和 $\omega_L$ 、 $\omega_R$ ，下面详细推导底盘中心的线速度 $v$ 和角速度 $\omega$ ：

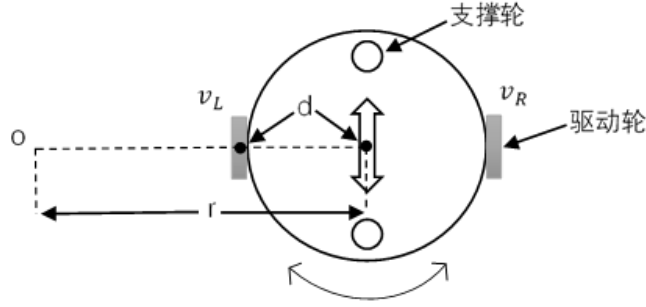


图 2.2 两轮差速移动底盘

根据左右两轮角速度相等可得：

$$\omega_L = \frac{v_L}{r-d} = \omega_R = \frac{v_R}{r+d} \quad (2.1)$$

将式 (2.1) 化简，得到用 $v_L$ 、 $v_R$ 以及 $d$ 表示的半径 $r$ ：

$$r = \frac{(v_R + v_L)d}{(v_R - v_L)} \quad (2.2)$$

以右轮为例表示出底盘中心的角速度：

$$\omega = \frac{v_R}{r+d} \quad (2.3)$$

$$r+d = \frac{(v_R + v_L)d}{(v_R - v_L)} + \frac{(v_R - v_L)d}{(v_R - v_L)} = 2 \frac{v_R d}{(v_R - v_L)} \quad (2.4)$$

将式 (2.4) 代入式 (2.3) 的分母中得到底盘中心角速度：

$$\omega = \frac{v_R - v_L}{2d} \quad (2.5)$$

根据线速度和角速度之间的关系得到底盘中心线速度：

$$v = \omega * r = \frac{(v_R - v_L)}{2d} \frac{(v_R + v_L)d}{(v_R - v_L)} = \frac{v_R + v_L}{2} \quad (2.6)$$

在移动机器人系统中主要涉及三个重要的坐标系：机器人坐标系、传感器坐标系以及世界坐标系。根据移动机器人里程计模型对其进行建模如图 2.3 所示。

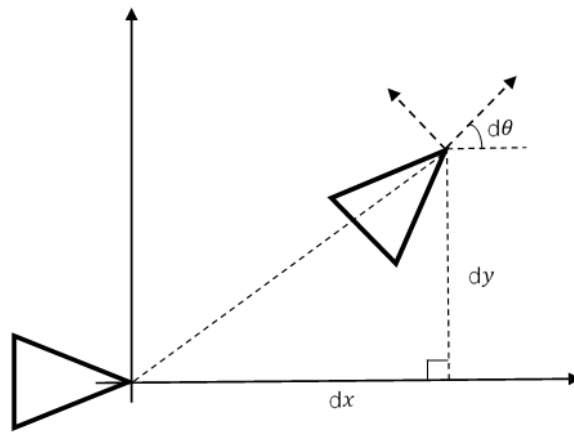


图 2.3 移动机器人航迹推算



假设当前时刻机器人在世界坐标系下的位姿为  $(x', y', \theta')$ ，上一时刻机器人在世界坐标系下的位姿为  $(x, y, \theta)$ ，其中  $(x, y)$  表示机器人在世界坐标系下的二维坐标， $\theta$  表示航向角。里程计测量到的机器人坐标系下的运动学解算增量为  $(dx, dy, d\theta)$ ，同时该增量需要从机器人坐标系转换到世界坐标系下。进一步推算出移动机器人的运动轨迹：

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} dx + \varepsilon x \\ dy + \varepsilon y \\ d\theta + \varepsilon \theta \end{bmatrix} \quad (2.7)$$

其中， $(\varepsilon x, \varepsilon y, \varepsilon \theta)$  表示里程计的系统误差。

### 2.1.2 里程计标定

最小二乘是一种数学优化方法，利用该方法能够快速解出待求变量，同时所得到的解与真实数据两者误差的平方和最小。几乎所有的工程问题都可以用  $Ax=b$  ( $A$  为  $m \times n$  的矩阵， $x$  为  $n \times 1$  的向量) 方程组来描述，而线性最小二乘法就是用来求解方程组的未知状态量  $x$ 。绝大多数情况为  $m > n$ ，此时称为超定方程组，且方程组只有最靠近真实解的最小二乘解：

$$x^* = (A^T A)^{-1} A^T b \quad (2.8)$$

常用的里程计标定方法有直接线性法和基于模型法，直接线性法因相较于模型法具有通用性强和实现简单等特点而受到广泛应用。直接线性法通过 PL-ICP 帧间匹配算法得到的激光雷达第  $i$  帧位姿  $\mu_i^*$  作为真值， $\mu_i^* = (\mu_{ix}^*, \mu_{iy}^*, \mu_{i\theta}^*)^T$ 。里程计数据为  $\mu_i$ ， $\mu_i = (\mu_{ix}, \mu_{iy}, \mu_{i\theta})^T$ ，假设它们之间成线性关系  $\mu_i^* = T * \mu_i$ ，即：

$$\begin{aligned} \mu_{ix}^* &= \mu_{ix} * t_{11} + \mu_{iy} * t_{12} + \mu_{i\theta} * t_{13} \\ \mu_{iy}^* &= \mu_{ix} * t_{21} + \mu_{iy} * t_{22} + \mu_{i\theta} * t_{23} \\ \mu_{i\theta}^* &= \mu_{ix} * t_{31} + \mu_{iy} * t_{32} + \mu_{i\theta} * t_{33} \end{aligned} \quad (2.9)$$

线性方程组中的  $T = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix}$ ，表示一个  $3 \times 3$  的矩阵。

为了使用式 (2.8) 的求解公式，需要对式 (2.9) 改写成：

$$\begin{bmatrix} \mu_{ix} & \mu_{iy} & \mu_{i\theta} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu_{ix} & \mu_{iy} & \mu_{i\theta} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mu_{ix} & \mu_{iy} & \mu_{i\theta} \end{bmatrix} \begin{bmatrix} t_{11} \\ \dots \\ t_{33} \end{bmatrix} = \begin{bmatrix} \mu_{ix}^* \\ \mu_{iy}^* \\ \mu_{i\theta}^* \end{bmatrix} \quad (2.10)$$

式 (2.10) 已经化成  $Ax=b$  的形式，根据式 (2.8) 求解出向量  $(t_{11}, \dots, t_{33})^T$  再转换为  $3 \times 3$  的矩阵  $T$ 。里程计观测值与激光雷达观测值形成对应关系，将每个里程计数据根据矩阵  $T$  进行矫正，即完成里程计的标定。在 ROS 系统中现有的数据包 `odom.bag` 进行实验，采用直接线性法对机器人的里程计进行矫正，最终在 RVIZ 中输出轨迹如图 2.4 所示，图中红色表示

激光雷达轨迹，绿色表示里程计轨迹，黄色表示矫正后的轨迹。

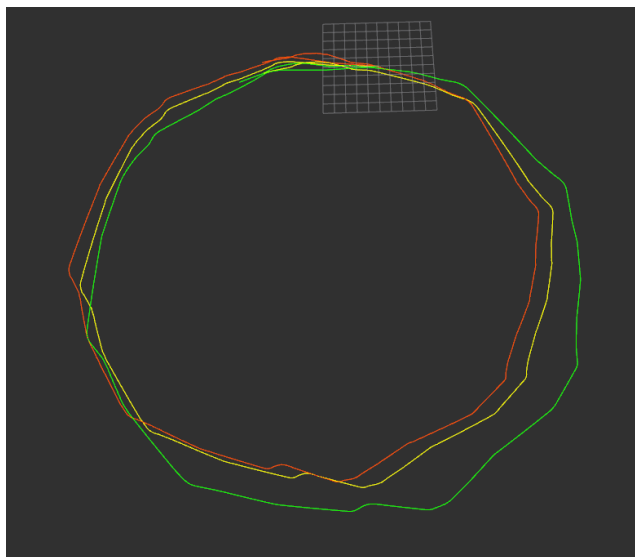


图 2.4 里程计矫正轨迹图

最终程序输出的矫正矩阵如下：

$$T = \begin{bmatrix} 0.943889 & 2.21271 & -0.0380446 \\ -0.0131005 & 6.68881 & 1.160112 \\ 0.0009834 & 23.0121 & 0.389921 \end{bmatrix} \quad (2.11)$$

## 2.2 激光雷达数学模型及运动畸变去除

思岚科技（SLAMTEC）生产的 RPLIDAR A1 激光雷达如图 2.5 所示，通过激光三角测距技术以及高速有效的视觉传感器模块，能够每秒 8000 多次地进行距离测量，进一步得到所处环境的轮廓图以及点云地图。RPLIDAR A1 激光雷达具体配置参数如表 2.1 所示，现已广泛应用于智能移动机器人、AGV 等场景下，尤其在扫地机器人领域更是能够快速、精确地构建地图。



图 2.5 RPLIDAR A1 激光雷达

表 2.1 RPLIDAR A1 激光雷达参数

项目	单位	最小值	典型值	最大值	备注
测距范围	米	/	0.15-12	/	基于白色高反光物体
扫描角度	度	/	0-360	/	/
测距分辨率	毫米	/	<0.5	/	测量物体在 1.5 米内
角度分辨率	度	/	≤1	/	5.5Hz 扫描时
单次测量时间	毫秒	/	0.5	/	/
测量频率	赫兹	2000	>4000	8000	/
扫描频率	赫兹	1	5.5	10	扫描一周 400 个采样点

激光雷达的测距方法主要有三角法和 TOF（飞行时间）法，三角法是激光器发射激光，照射到目标以后反射回来经 CCD 感光单元进行处理。激光器与探测器之间存在间隔，因此不同测量目标会成像在 CCD 的不同区域，最后通过公式得到被测物体的距离；TOF 法是激光器发射一个激光脉冲，同时需要记下发射的时间，接收器接收到其返回，并记下返回的时间。两个时刻的间隔就是光的“飞行时间”，在已知光速的情况下结合时间就可以求解出与目标之间的距离。

RPLIDAR A1 激光雷达所采用的三角测距原理如图 2.6 所示，A 和 B 两点是激光发射头， $\alpha$  和  $\beta$  是激光发射的两个角，L 表示两个激光发射点 AB 间的距离，根据角边角就可以确定一个三角形。其中  $\alpha$  和 L 已经测量出来的， $\beta$  是由 A 点所对应的  $\alpha$  求解得到，距离 d 的计算过程如下：

$$\begin{aligned} L_1 &= \frac{d}{\tan \alpha} \\ L_2 &= \frac{d}{\tan \beta} \end{aligned} \tag{2.12}$$

$$L = L_1 + L_2 = d \left( \frac{\cos \alpha}{\sin \alpha} + \frac{\cos \beta}{\sin \beta} \right) = d \frac{\sin(\alpha + \beta)}{\sin \alpha \sin \beta} \tag{2.13}$$

$$d = L \frac{\sin \alpha \sin \beta}{\sin(\alpha + \beta)} \tag{2.14}$$

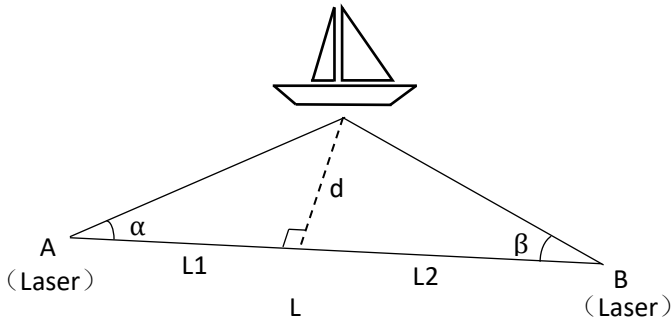


图 2.6 三角测距示意图

### 2.2.1 激光雷达数学模型

激光雷达的数学模型主要有光束模型 (beam model) 和似然场模型 (likelihood mode)。给定一个机器人的位置  $x_t$  和地图  $m$ , 得到这一帧观测值  $z_t$  中第  $k$  束激光的概率  $P(z_t^k | x_t, m)$ ,  $z_t^{k*}$  是传感器测量的均值, 表示测量的真实距离,  $z_{max}$  表示激光最大测量距离。光束模型分为四种测量模型, 如图 2.7 所示:

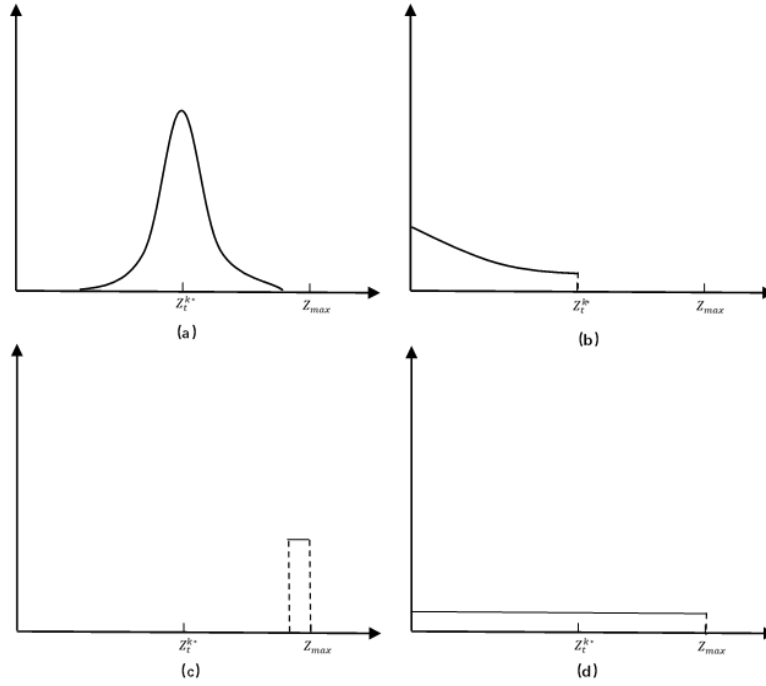


图 2.7 光束模型图

图 2.7 (a) 表示激光束击中障碍物时服从高斯分布, 此时  $P_{hit}(z_t^k | x_t, m)$  的权重为  $a_{hit}$ ; (b) 表示激光束被某些物体遮挡时服从指数分布, 此时  $P_{short}(z_t^k | x_t, m)$  的权重为  $a_{short}$ ; (c) 和 (d) 表示另外测量失败和随机测量两种都服从均匀分布, 权重分别为  $a_{max}$  和  $a_{rand}$ 。概率  $P(z_t^k | x_t, m)$  通过上述四种模型的概率值结合相应比重加权得到:

$$P(z_t^k | x_t, m) = \begin{pmatrix} a_{hit} \\ a_{short} \\ a_{max} \\ a_{rand} \end{pmatrix}^T \begin{pmatrix} P_{hit}(z_t^k | x_t, m) \\ P_{short}(z_t^k | x_t, m) \\ P_{max}(z_t^k | x_t, m) \\ P_{rand}(z_t^k | x_t, m) \end{pmatrix} \quad (2.15)$$

因为机器人大多处在动态变换的非结构化环境, 位姿出现细微的变动就会造成期望值出现剧烈变化, 现实情况下几乎不用此模型。

似然场模型如图 2.8 所示, 通过传感器噪声和机器人位姿噪声得到一个高斯分布, 用该

分布对整个图像进行模糊处理，对障碍物进行高斯平滑。该模型相较于光束模型直接省略了 raytracking（光线追踪）这一步的计算，通过计算量小的查表法即可得到结果。即使位姿发生一点改变，激光的观测概率值也不会突变，所以能够适应结构化环境和非结构化环境。

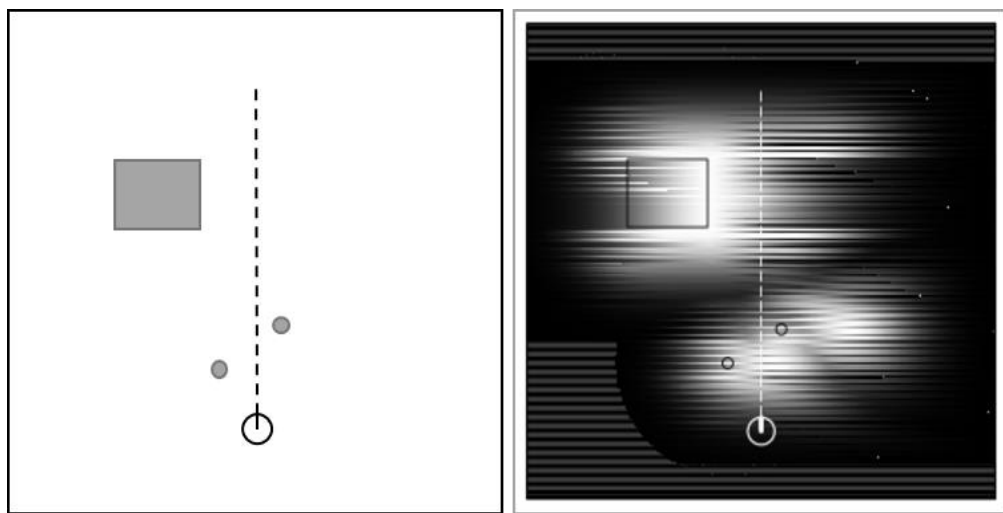


图 2.8 似然场计算模型图

### 2.2.2 运动畸变去除

激光雷达采集数据不会立刻完成，扫描过程中机器人会不断发生移动，激光雷达的帧率越低带来的运动畸变就越明显。运动畸变的主要解决方法有纯估计法、里程计辅助法以及两者的结合。纯估计法不借助任何外部设备，只依靠匹配算法去除运动畸变。纯估计法中使用较多的是 VICP 匹配算法，该算法是 ICP（迭代最近点）算法的变种，认为机器人做匀速运动且在进行匹配的同时计算机器人的速度。VICP 匹配的大致步骤是已知每一个时刻  $t_i$ ，然后反推出所有点的位姿，再结合每一个点的速度对采集到的数据进行修正。

里程计辅助法能够解决纯估计法中匀速运动这一假设不成立的情况，最大程度地描述机器人运动状态，实现预处理和状态估计的解耦。轮式里程计能够尽可能准确得到机器人的局部角度和位移，同时更新速度也较快。假设当前帧激光的开始时间为  $t_s$ ，结束时间为  $t_e$ ，时间间隔为  $\Delta t$ ，里程计数据保存在时间递增的队列中，同时满足最开始的里程计数据时间  $< t_s$  以及最后的里程计数据时间  $> t_e$ ，具体步骤如下：

(1) 计算当前帧激光数据中  $t_s$  和  $t_e$  时刻的位姿  $p_s$  和  $p_e$ 。ROS 中 TF 库定义 OdomList 为里程计数据队列，如果里程计队列正好与激光数据时间一致，即  $t_s$  对应第  $i$  帧激光数据， $t_e$  对应第  $j$  帧数据，则：

$$p_s = OdomList[i]$$

$$p_e = OdomList[j] \quad (2.16)$$

一般情况下里程计数据不会和激光数据正好同步，如果在 $t_s$ 时刻匹配不到里程计位姿，假设 $l$ 和 $k$ 时刻存在位姿，且 $l < s < k$ ，在 $l$ 到 $k$ 的时间内机器人做匀速运动，然后进行线性插值，其中 $LinearInterp()$ 为线性插值函数：

$$\begin{aligned} p_l &= OdomList[l] \\ p_k &= OdomList[k] \end{aligned} \quad (2.17)$$

$$p_s = LinearInterp\left(p_l, p_k, \frac{s-l}{k-l}\right) \quad (2.18)$$

同理当 $t_e$ 时刻匹配不到里程计位姿时，根据线性插值得到 $p_e$ 。

(2) 激光雷达扫描一圈的过程中假设机器人做匀加速运动，因此机器人的位姿是以时间 $t$ 为自变量的二次函数。用分段线性函数模拟逼近二次曲线，当分段数大于3时忽略近似带来的误差。可以通过线性插值法得到一帧激光数据中所有点对应的位姿 $\{p_s, p_{s+\Delta t}, \dots, p_e\}$ 。

(3) 结合得到的位姿重新计算激光数据的坐标，形成一帧新的激光数据发布出去。

本文采用轮式里程计辅助法在 ROS 系统中运行在两侧墙面平行的走廊环境下效果如图 2.9 所示，其中红色表示畸变矫正前绿色表示畸变矫正后。

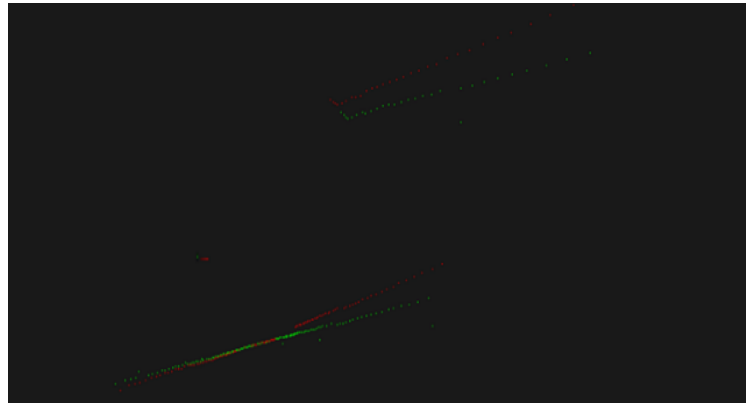


图 2.9 激光雷达运动畸变去除

## 2.3 相机数学模型及内参标定

Kinect 是美国微软公司于 2010 年 6 月发布的一款 XBOX-360 配套体感相机，该摄像机配有 RGB 彩色相机、红外发射器以及红外接收器三个摄像头，其中红外发射器和红外接收器组合在一起用于获取红外影像，该红外影像中存储距离远近的深度信息。Kinect 具有动作和图像识别、语音处理、社交互动等功能，同时开源了快速安装、传感器原始数据流、骨骼关节识别追踪、抑制噪音与回音消除等 SDK 供开发人员或玩家使用<sup>[40]</sup>。

2014 年 10 月又发布第二代产品 Kinect V2 如图 2.10 所示，它作为深度相机广泛应用于

移动机器人的导航和 RGB-D 视觉 SLAM 中，具体技术参数如表 2.2 所示：

表 2.2 Kinect V2 参数

配置	参数
彩色相机分辨率	1920 x 1080
深度相机分辨率	512 x 424
水平角度	70°
垂直角度	60°
骨骼关节数	25 个/人
检测范围	0.5-4.5m
音效	16bit, 16kHz



图 2.10 Kinect V2 摄像机

### 2.3.1 相机数学模型

本文采用 Kinect V2 深度相机进行视觉 SLAM 的研究，下面以最简单有效的针孔模型将三维世界中坐标点到二维像素平面坐标映射的过程进行建模，如图 2.11 所示。现实空间一点  $P$  通过小孔在成像平面为  $P'$ ， $O$  表示在相机坐标系  $O-x-y-z$  下相机的光心即小孔，成像平面坐标系  $O'-x'-y'$  中  $O'$  到光心  $O$  的距离表示焦距  $f$ 。假设空间中点  $P$  的坐标为  $[X, Y, Z]^T$ ，经过成像后点  $P'$  的坐标为  $[X', Y', Z']^T$ ，那么根据相似三角形定理可得：

$$\frac{Z}{f} = -\frac{X}{X'} = -\frac{Y}{Y'} \quad (2.19)$$

虽然从针孔模型中看小孔成的是倒像，但是现实中相机会对图片进行预处理成正像，所以将式 (2.19) 改写成：

$$\begin{aligned} X' &= f \frac{X}{Z} \\ Y' &= f \frac{Y}{Z} \end{aligned} \quad (2.20)$$

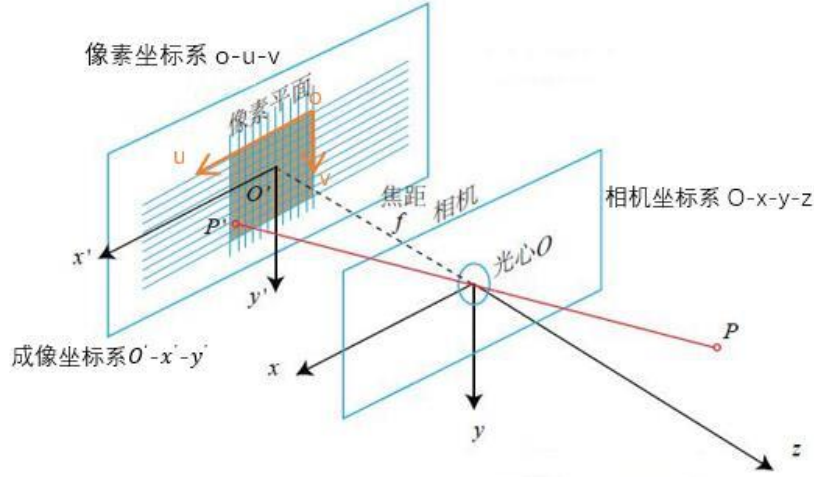


图 2.11 针孔相机模型

在实际的相机中，我们需要采样经过小孔模型后的成像，然后进一步量化成像素点，下面详细推导相机中光线到像素点转换的过程。像素坐标系中  $u$  轴和  $v$  轴分别是在成像坐标系  $O'-x'-y'$  的  $x'$  轴和  $y'$  轴上缩小了  $\alpha$  倍和  $\beta$  倍，同时原点  $o$  点是  $O'$  点平移  $[c_x, c_y]^T$  之后得到。假设像素平面坐标系  $o-u-v$  上  $P'$  点的坐标为  $[u, v]^T$ ， $P'$  点与像素坐标系之间的转换关系如下：

$$\begin{cases} u = \alpha X' + c_x \\ v = \beta Y' + c_y \end{cases} \quad (2.21)$$

将式 (2.20) 带入式 (2.21)，同时把  $\alpha f$  和  $\beta f$  合并成  $f_x$  和  $f_y$ ，得到：

$$\begin{cases} u = f_x \frac{X}{Z} + c_x \\ v = f_y \frac{Y}{Z} + c_y \end{cases} \quad (2.22)$$

其中  $f_x$  和  $f_y$  的单位为像素，进一步将式 (2.22) 改为矩阵形式：

$$Z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \triangleq K P \quad (2.23)$$

式 (2.23) 中  $K$  表示内参矩阵，当相机发生移动时，当前相机坐标系下位姿  $P_{uv}$  可以用世界坐标系下位姿  $P_w$  在相机的外参数作用下得到，其中  $R$  和  $t$  称为相机的外参数。

$$Z P_{uv} = Z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K(R P_w + t) \quad (2.24)$$

Kinect V2 相机通过 TOF（飞行时间）法来测量像素的距离，相机向物体发出脉冲光，然后计算从发射到接收之间所用的时间，从而确定目标到自身的距离。在测量完像素深度后，相机按照出厂时镜头之间的位置和距离完成深度信息和彩色图像之间的匹配，最后输出相对



应的深度图和彩色图。

### 2.3.2 相机内参标定

上述针孔模型的分析是在理想状态下，实际情况下通常在相机感光元件前面放透镜来提升成像的品质，但是会导致成像平面实际的成像与理论状态下成像之间出现一定误差。造成这种光学畸变的误差主要原因是：凸透镜的形状会改变光线的传播方向以及相机装配过程中像素点位置的偏移。

假设平面上一点  $P$  的坐标为  $[x, y]^T$ ，对应的极坐标系下坐标为  $[r, \theta]^T$ ，其中  $r$  表示原点到  $P$  点的距离， $\theta$  表示与水平轴之间的夹角。将透镜形状造成的畸变叫做径向畸变，它是坐标点朝着长度  $r$  方向偏移。径向畸变分为正向的枕型畸变和负向的桶型畸变两大类，主要的表现就是一条直线经过相机的透镜后变为曲线。

将相机透镜装配过程中存在的误差叫做切向畸变，它是  $\theta$  角度发生了偏移。径向畸变中像点的位置误差与它到光心的距离有关，可以用中心距离为自变量的多项式函数来表示。对于相机坐标系中的点  $P(X, Y, Z)$ ，我们通过不同的畸变参数对径向畸变和切向畸变进行矫正，具体步骤如下：

(1) 将空间中点  $P$  投影到归一化图像平面， $[x, y]^T$  表示  $P$  点的归一化坐标。

(2) 将平面上  $P$  点进行径向和切向畸变的矫正：

$$\begin{cases} x' = x(1 + k_1 r^2 + k_2 r^4) + 2p_1 xy + p_2(r^2 + 2x^2) \\ y' = y(1 + k_1 r^2 + k_2 r^4) + p_1(r^2 + 2y^2) + 2p_2 xy \end{cases} \quad (2.25)$$

其中， $[x, y]^T$  是畸变矫正前的坐标， $[x', y']^T$  是畸变矫正后的坐标。径向畸变涉及参数  $k_1$  和  $k_2$ ， $k_1$  用来矫正图像中心畸变程度较小的地方， $k_2$  则矫正图像边缘畸变程度大的地方。切向畸变用参数  $p_1$  和  $p_2$  来矫正。

(3) 将矫正后的点通过式 (2.22) 投影到像素平面，该点在图像中的正确位置：

$$\begin{cases} u = f_x x' + c_x \\ v = f_y y' + c_y \end{cases} \quad (2.26)$$

摄像机的内部构造比较精细复杂，很容易受到外部诸多环境因素的影响，生成的目标图像会出现或多或少的畸变。为了避免输入的数据源因畸变产生误差，需要对摄像机的内参和畸变参数进行标定。本文采用张正友标定法对深度相机进行彩色图像标定和红外图像标定，在 ROS 机器人操作系统中使用官方 `iai_kinect` 标定包来完成，选用内部角点  $8 \times 6$  以及格子边长为 24mm 的黑白棋盘格，下载打印黑白棋盘格并粘贴到硬纸板上便于标定。下面以彩色摄像头为例详细介绍标定步骤和标定结果：

- (1) 运行 ROS 系统，新建用于存储后续生成的文件和图片的文件夹 temp。
- (2) 启动 ROS 中深度相机的驱动，查看当前设备的串口号，在 Linux 系统中新建一个以该串口号命名的文件夹。
- (3) 开始标定彩色图像如图 2.12 所示，不断调整棋盘格的上下、左右、前后以及倾斜转动位置状态。按空格键保存图片，尽可能多地保存一些标定得到的数据。等数据采集完成后，打开新的命令终端，输入指令来计算相机参数，具体参数如表 2.3 和表 2.4 所示。
- (4) 打开新建的 temp 文件夹，可以看到标定过程中采集的图像和 calib\_color.yaml 文件中的一些相机参数。

表 2.3 相机内参数值

内部参数	$f_x$	$f_y$	$c_x$	$c_y$
数值	530.12178	529.36870	326.28367	250.13662

表 2.4 畸变矫正参数

畸变参数	$k_1$	$k_2$	$p_1$	$p_2$
数值	-0.023612	0.048551	0.001833	-0.002306

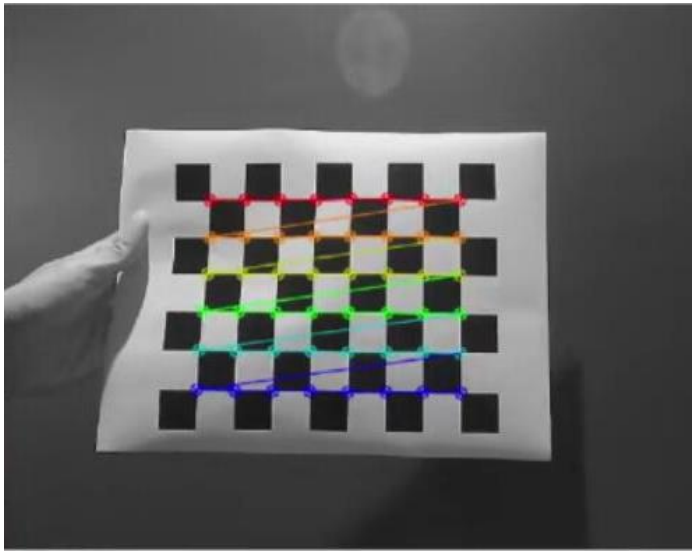


图 2.12 彩色摄像头标定

2.4 激光雷达与相机外参标定

本课题融合的定位与地图构建中涉及二维激光雷达 RPLIDAR A1 和深度相机 Kinect V2 两种不同的传感器，它们两个传感器对空间中的物体有不同的数据表现形式。在进行数据融合之前，需要将激光雷达获取的数据和深度相机获取的三维位姿数据进行转换，最终得到激

光雷达坐标系和深度相机坐标系之间的转换关系<sup>[41]</sup>。因此需要对激光雷达和深度相机进行联合的外参标定。如图 2.13 所示：

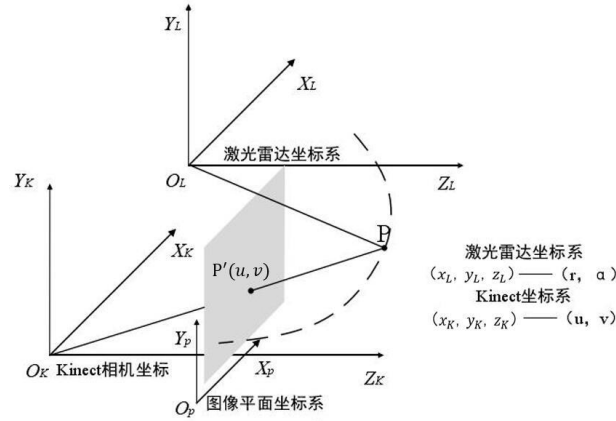


图 2.13 激光雷达和深度相机坐标

假设空间中一点  $P$  表示检测到的目标物体， $P$  点在激光雷达坐标系  $O_L-X_L-Y_L-Z_L$  下的坐标是  $(x_L, y_L, z_L)$ ，激光雷达传感器获取的数据信息是目标的距离和角度。在深度相机坐标系  $O_K-X_K-Y_K-Z_K$  下的坐标是  $(x_K, y_K, z_K)$ ， $P$  点在图像平面坐标系中  $O_P-X_P-Y_P$  下的投影点  $P'$  坐标是  $(u, v)$ ，深度相机获取到的数据是投影点的坐标位置及其深度信息。下面将详细介绍激光雷达和深度相机标定的具体内容：

目标点  $P$  点在激光雷达坐标系  $O_L-X_L-Y_L-Z_L$  下的坐标是  $(x_L, y_L, z_L)$ ，根据深度相机坐标系与激光雷达坐标系之间的几何位置关系得到  $P$  点在相机坐标系  $O_K-X_K-Y_K-Z_K$  下的坐标  $(x_K, y_K, z_K)$ ：

$$\begin{bmatrix} x_K \\ y_K \\ z_K \end{bmatrix} = R \begin{bmatrix} x_L \\ y_L \\ z_L \end{bmatrix} + T \quad (2.27)$$

其中  $R$  表示旋转矩阵， $T$  表示平移的向量。在 2.3.1 节的相机数学模型中，式 (2.21) 描述了点  $P(x_K, y_K, z_K)$  在成像平面坐标系与像素坐标系  $(u, v)$  中的转换关系：

$$Z_K \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} x_K \\ y_K \\ z_K \end{bmatrix} \quad (2.28)$$

内部参数  $K$  中的  $f_x$ 、 $f_y$ 、 $c_x$  以及  $c_y$  已经由 2.3.2 节相机的内参标定计算得到。将式 (2.27) 和式 (2.28) 合并可以得到点  $P$  在图像平面坐标系下的投影点  $P'$  坐标  $(u, v)$  和激光雷达坐标  $(x_L, y_L, z_L)$  的转换关系：

$$Z_K \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \left( R \begin{bmatrix} x_L \\ y_L \\ z_L \end{bmatrix} + T \right) \quad (2.29)$$

激光雷达的扫描平面为坐标系 $O_L-X_L-Z_L$ ，激光雷达的中心位于原点 $O_L$ ， $r$ 表示 $O_L$ 到目标的距离， $\alpha$ 表示与 $Z_L$ 轴的夹角，所以 P 点的坐标 $(x_L, y_L, z_L)$ 和极坐标下 $(r, \alpha)$ 的关系如下：

$$z_L = r \cos \alpha, x_L = r \sin \alpha \quad (2.30)$$

平面 $O_L-X_L-Z_L$ 和平面 $O_K-X_K-Z_K$ 之间的距离为 $h$ ，那么结合式(2.30)可以将式(2.29)改写成：

$$Z_K \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \left( R \begin{bmatrix} r \sin \alpha \\ h \\ r \cos \alpha \end{bmatrix} + T \right) \quad (2.31)$$

对于空间中目标点 P，深度相机能够得到 $Z_K$ 和 $(u, v)$ ，激光雷达可以获取 $(r, \alpha)$ 。使用 ROS 机器人操作系统中的开源标定功能包，尽可能多的采集需要的像素点，将相机的参数和激光雷达测量的数据式(2.31)中，通过最小二乘法得到矩阵 R 和向量 T，最终完成外参的标定。实验中得到的激光雷达坐标系与深度相机坐标系之间的关系为：

$$R = \begin{pmatrix} 0.4123 & 0.7321 & -0.5442 \\ 0.3972 & -0.6623 & 0.7324 \\ 0.5045 & 0.3908 & -0.3208 \end{pmatrix} \quad (2.32)$$

$$T = \begin{pmatrix} 0.8215 \\ -0.3446 \\ 0.0065 \end{pmatrix} \quad (2.33)$$

## 2.5 本章小结

本章详细推导了 Kobuki 两轮差速底盘的运动学模型以及使用直接线性法标定了里程计，提升了里程计在后续 SLAM 实验中的精度。然后介绍了激光雷达的测距原理以及光束模型和似然场模型，阐述了去除运动畸变的两种方法：VICP 匹配的纯估计法和里程计辅助法，采用里程计辅助法完成走廊环境下运动畸变的矫正。接着介绍了 Kinect 深度相机的数学模型，使用张正友法完成相机的内参标定。最后得到的一次激光雷达坐标系与深度相机坐标系之间转换关系，即进行外参的联合定标，为后续融合 SLAM 的研究奠定基础。

## 第三章 基于激光雷达的 SLAM 算法研究

在使用激光雷达传感器进行室内机器人的定位与地图构建研究中，基于滤波器的方案是解决 SLAM 问题最经典的方法<sup>[42]</sup>。滤波器的本质就是结合预测值与观测值得到尽可能精确的位姿后验值，其中贝叶斯滤波原理是概率估计方法的基础，在此基础上出现了许多诸如卡尔曼滤波、粒子滤波、RBPF-SLAM 以及 Gmapping 等算法。

### 3.1 同时定位与地图构建理论基础

定位描述的是在已知的地图环境中尽可能准确地估计机器人的位姿，地图构建表示在已知移动机器人的位姿情况下尽可能真实地构建环境地图。同时定位与地图构建（SLAM）表示移动机器人根据当前的位姿以及传感器得到的陌生环境信息数据，在进行自身定位的同时真实地构建导航需要的环境地图。

#### 3.1.1 SLAM 数学模型

定位问题和建图问题是 SLAM 系统最关键的两个部分。当移动机器人在陌生环境下运动时，自身携带的里程计或 IMU（惯性测量单元）等传感器将不断累计误差造成错误的位姿估计。如果此时能够在正确的局部地图上提供观测信息用于误差校正，那么机器人将得到更为准确的位姿，进而结合观测数据构建更为精确的地图。SLAM 系统需要同时的定位和建图，两者之间相互影响、密不可分。下面将从数学概率估计的角度出发，详细分析移动机器人 SLAM 系统的完整过程，如图 3.1 所示，图中三角形指代移动中的机器人，六角形指代环境中碰到的不同特征。

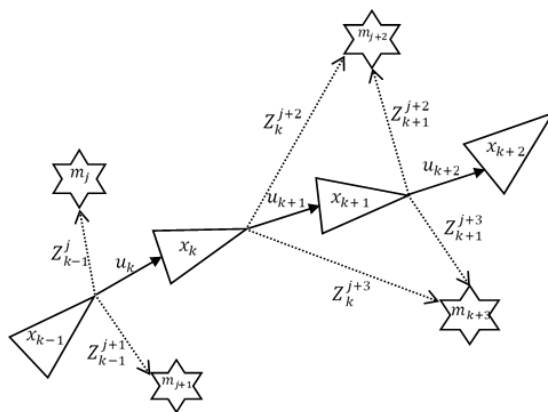


图 3.1 SLAM 系统状态

为了更好的用数学语言表达,将移动机器人运动的连续时间离散化为  $t=1, 2, \dots, k-1, k$ 。每一个离散时刻对应的移动机器人的位置用  $x_1, x_2, \dots, x_{k-1}, x_k$  表示,它们之间的连线代表移动机器人的运动轨迹。环境中的  $N$  个路标表示一幅完整的地图,每个路标对应的小地图分别为  $m_1, m_2, \dots, m_{N-1}, m_N$ 。在  $k$  时刻机器人处在  $x_k$  位置识别到第  $j$  个路标产生的观测信息用  $Z_k^j$  来表示,本课题中观测信息主要由激光雷达和 Kinect 深度相机提供。

移动机器人主要通过携带里程计和 IMU 等传感器来测量自身数据信息,包括位移、速度、加速度和角速度等。机器人在某个时刻的位置信息或者是状态信息可以用统一的运动方程来表示:

$$x_k = f(x_{k-1}, u_k, w_k) \quad (3.1)$$

其中  $u_k$  表示机器人从  $k-1$  时刻状态  $x_{k-1}$  转换到  $k$  时刻状态  $x_k$  所施加的控制量,也成为机器人的控制输入。 $w_k$  表示运动过程出现的噪声。

在  $k$  时刻机器人处在  $x_k$  位置识别到第  $j$  个路标产生的观测数据  $Z_k^j$  可以用抽象函数  $g$  表示:

$$Z_k^j = g(m_j, x_k, v_k) \quad (3.2)$$

其中  $v_k$  表示观测过程引入的噪声。

根据以上描述,运动过程由内部的里程计或 IMU 传感器提供,观测过程由外部传感器激光雷达或相机提供,则 SLAM 系统可以用运动方程与观测方程联合表示:

$$\begin{cases} x_k = f(x_{k-1}, u_k, w_k) \\ Z_k^j = g(m_j, x_k, v_k) \end{cases} \quad (3.3)$$

图 3.1 中 SLAM 系统的转移模型具体过程是移动机器人在  $k$  时刻通过内部传感器提供的控制输入量  $u_k$  和上一时刻的状态量  $x_{k-1}$  估算出机器人在  $k$  时刻的位姿,同时再结合  $k$  时刻外部传感器对路标  $m_j$  的观测数据  $Z_k^j$  来矫正估计的位姿  $x_k$ 。在系统的运动和观测两个过程中都将引入噪声,机器人对环境中路标的观测数据将绘制出完成的地图,移动机器人位姿状态随着上一时刻状态和控制输入的不断更新和观测数据的矫正过程称为其自定位。SLAM 问题可以总结为“估计”-“观测”-“矫正”-“再观测”的循环往复直至构建出最终地图的过程,SLAM 系统的核心转化为概率机器人的状态估计问题。

### 3.1.2 SLAM 分类

按照机器人携带的传感器不同可以将 SLAM 分为二维激光 SLAM、三维激光 SLAM、视

觉 SLAM、声纳 SLAM 以及融合 SLAM 等，如表 3.1 所示。二维激光雷达成本低、结构简单，能够满足绝大多数室内场景下的需要，三维激光雷达能够获取更丰富的信息，但是价格昂贵，多用于无人机或者自动驾驶等场景；视觉 SLAM 主要通过单目、双目或是 RGB-D 深度相机传感器实现<sup>[43]</sup>。基于 Kinect 深度摄像机的视觉 SLAM 采集三维点云数据得到与障碍物的距离；声纳 SLAM 需要搭载声纳传感器测距，主要用于水下场景绘制海图；融合 SLAM 主要有激光和视觉融合、IMU 和视觉融合等，通过多个传感器之间优势互补得到更加精准的地图。

表 3.1 不同传感器的 SLAM 分类

类型	适用场景	特点
二维激光 SLAM	室内（可解决 95%的问题）	成本低
三维激光 SLAM	室外、测绘	成本高 信息量较大
视觉 SLAM	室内室外通用	信息量丰富 易受光线影响
声纳 SLAM	水下，绘制海图	
融合 SLAM	室内室外通用	效果好 融合复杂有难度

根据 SLAM 后端优化方式的不同分为图优化方法、滑动窗口法以及滤波器法，如表 3.2 所示。图优化方法估计每个时刻机器人的位姿，某一时刻的位姿是一个节点或顶点，节点之间的不同关系构成边；滑动窗口法估计固定数量的位姿，随着 SLAM 系统不断遇到新的环境，系统状态量和残差项都会不断增多，计算量也会随着不断增加。该方法动态增加新的状态或利用边缘化消去老的状态变量，得到先验信息，保持状态量的个数在一定范围内；基于滤波的方法只估计当前位姿，滤波在 SLAM 系统中的作用是让移动机器人处于正确位置上所对应的概率提高，而把错误位置上的概率降低。滤波器的本质就是结合预测与观测，得到机器人最精确的位姿后验值。事实上预测与观测均从传感器而来，因此滤波器就是结合各传感器得到一个最好的融合结果。

表 3.2 不同后端优化的 SLAM 分类

类型	特点
图优化方法(Graph-based)	估计全部位姿(最通用)
滑动窗口方法(sliding-window)	估计固定数量的位姿(VIO,VO)
滤波器方法(filter-based)	估计当前位姿(Pose-Fusion)

### 3.1.3 地图表示

根据机器人所处的不同环境以及任务场景的差异,采用不同的方式或通俗易懂的符号来构建符合要求的环境地图。在激光 SLAM 系统中构建出来的地图主要有几何特征地图、栅格地图以及拓扑地图等。结合机器人运动的起始点、结束点以及地图中障碍物位置,根据路径规划算法得到的路线完成相应的任务。在决定创建或使用具体哪种环境地图时要关注以下三点:一是机器人定位的精准度以及传感器的测量范围要和地图的特性吻合;第二点是机器人传感器返回的测量数据需要和地图所表示的特征类型相匹配;三是地图所表示的环境信息复杂度将影响机器人的定位建图、自主导航以及后续任务执行的效率。下面将对文中提到的三种不同地图表示方法一一介绍:

#### (1) 栅格地图

Elfes 和 Moravec 等人在 1985 年提出了占据栅格地图表示法,而后 Elfes 等人在此基础上继续研究形成完整的栅格地图理论。占据栅格法的原理是将机器人所处的未知环境分割成规格相同的栅格,栅格中概率值 1 表示被障碍物占据,0 表示没有被占据,0.5 表示概率未知。最开始构建栅格地图时,用概率值 0.5 初始化。

地图中每个栅格的信息都可以匹配实际环境空间中的局部区域,尽可能保留环境的重要信息。根据栅格地图的特性,可以正确地处理超声测量数据,完成移动机器人的自定位以及路径规划。然而,栅格地图的缺点是环境地图分辨率与栅格尺寸成反比,当对环境划分比较细致时需要减小栅格的尺寸,导致定位过程中处理的时间和复杂度有所提升,此时地图的维护和信息更新也将变得更加困难。同时该地图方法中每个栅格之间的概率值被认为是各不相关的,从而导致栅格地图和真实环境之间存在差距。本文所研究的机器人 SLAM 系统构建的就是二维的占据栅格地图。

#### (2) 拓扑地图

拓扑地图是由 Brooks 和 Mataric 等学者提出的一种较为抽象的地图表示方法。拓扑法将外部环境用一张带结点和不同边的拓扑结构图(graph)来描述。拓扑地图的优点是通用性高,适合障碍物不重复的大范围场景,便于后续移动机器人的路径规划;拓扑地图的创建和使用简单快捷,占用内存不多,计算速度快,可以利用很多现成有效的识别和搜索算法,节省开发成本。该方法的缺点表现在遇到两个相似度很高的场景时,很难判断两者之间的区别,所以在非结构化环境中很难作用。同时拓扑结构图只关注节点间是否连通,节点间的距离不一定最短,从而不能保证机器人的最优路径规划。



### (3) 几何特征地图

几何特征地图表示法是指移动机器人在所处环境中将外部传感器采集到的环境信息抽象出来,进一步转化点、直线以及曲线等几何特征。通过这些几何信息描述未知环境中的障碍物信息,使得地图中的信息更加简单直观,便于机器人的位姿估计以及特征提取。几何特征地图的缺点是要将环境特征数据关联起来,也就是全局地图和局部地图在更新后要完成对齐,在实现特征数据关联过程中很难做到一致性。在局部地图环境中,使用卡尔曼滤波法可以保证在一定计算量的基础上提升位置的精准度。但是在大范围全局环境中不能精确定位,而且几何数据在实际的环境中很难获取,所以几何特征地图的表示方法没有得到广泛的应用。

## 3.2 基于滤波器的激光 SLAM 算法研究

激光雷达 SLAM 方案主要有基于滤波器的 SLAM 方法和基于图优化的 SLAM 两种,早期滤波器方法是移动机器人 SLAM 的常用方案,但是随着回环检测算法的不断应用,基于图优化的 SLAM 具有更好的定位和建图效果。基于滤波器的 SLAM 需要根据外部传感器采集到的数据以及这个过程中引入的噪声来估计移动机器人的真实位姿,所以滤波器的本质就是结合预测值与观测值得到尽可能精确的位姿后验值。实际上预测值主要是根据 IMU 和编码器递推得到,观测值往往是从 GPS、激光雷达以及相机等传感器采集得到,因此滤波器的作用便是结合各传感器的数据得带一个最好的融合结果,即定位模块的输出。下面将主要介绍滤波器方法中几个经典的 SLAM 算法。

### 3.2.1 贝叶斯滤波算法研究

根据 3.1 节描述的 SLAM 系统状态模型,可以把它看作是移动机器人通过不同传感器的测量模型得到所处环境的数据,估计出机器人自身的真实状态。SLAM 系统状态指的就是移动机器人的位姿,可以用式(3.3)的运动方程与观测方程来联合表示。对于估算机器人的位姿,当只有运动方程而没有环境观测数据时,随着时间的增长机器人位置坐标的不确定性会逐渐累积,导致最终估计值与实际值相差较大。而当引入传感器对外部环境采集的观测方程之后,其位置的不确定性会被慢慢矫正,最终稳定在一个小范围内。

式(3.3)对于运动方程和观测方程的描述都是抽象不具体的。当移动机器人携带里程计以及激光雷达传感器时,假设空间中移动机器人的坐标 $x_k = (x, y, z)_k$ ,则可以得到一个时间间隔内的相对运动 $\Delta x_k = (\Delta x, \Delta y, \Delta z)_k$ ,激光雷达传感器观测到机器人距离目标点的距离和角度

信息  $Z_k = (r, \theta)_k, (L_x, L_y)$  表示路标点坐标，重新得到两个方程：

$$\begin{cases} x_k = x_{k-1} + \Delta x_{k-1} + w_k \\ Z_k = \begin{pmatrix} \sqrt{(L_x - x)^2 + (L_y - y)^2} \\ \tan^{-1} \left( \frac{L_y - y}{L_x - x} \right) \end{pmatrix} + v \end{cases}_k \quad (3.4)$$

根据运动方程和观测方程中  $f$  和  $g$  函数是否为线性系统，以及引入的噪声  $w_k$  和  $v_k$  是否服从高斯分布，将对应不同的滤波器函数模型，主要包括卡尔曼滤波和粒子滤波。其中贝叶斯概率模型在 SLAM 领域是其他模型的基础，它是一种已知先验概率然后求解后验概率的过程，如图 3.2 所示。

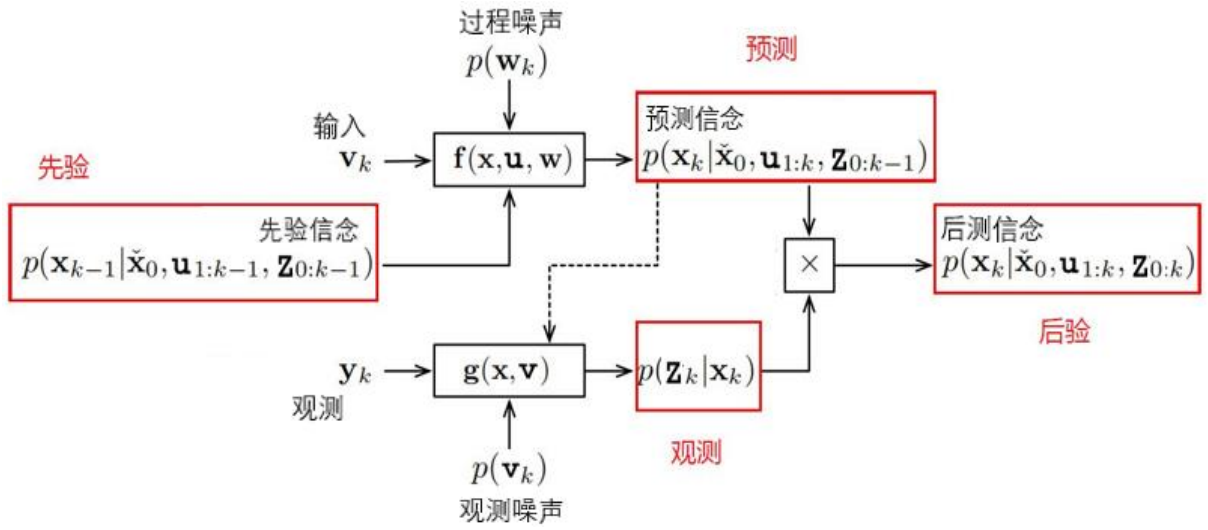


图 3.2 贝叶斯滤波示意图

在实际移动机器人的状态估计任务中，待估计的后验概率密度函数可以表示为  $p(x_k | \tilde{x}_0, u_{1:k}, z_{0:k})$ ，其中  $\tilde{x}_0$  表示机器人的初始状态， $u_{1:k}$  表示从时刻 1 到  $k$  的控制输入， $z_{0:k}$  表示从时刻 0 到  $k$  对环境信息的观测。因此滤波 SLAM 问题可以直观的理解为将初始状态值、控制输入以及观测值融合得到最终的估计位置。根据贝叶斯公式， $k$  时刻后验概率密度函数可以写成：

$$\begin{aligned} p(x_k | \tilde{x}_0, u_{1:k}, z_{0:k}) &= \frac{p(z_k | x_k, \tilde{x}_0, u_{1:k}, z_{0:k-1}) p(x_k | \tilde{x}_0, u_{1:k}, z_{0:k-1})}{p(z_k | \tilde{x}_0, u_{1:k}, z_{0:k-1})} \\ &= \eta p(z_k | x_k, \tilde{x}_0, u_{1:k}, z_{0:k-1}) p(x_k | \tilde{x}_0, u_{1:k}, z_{0:k-1}) \end{aligned} \quad (3.5)$$

在移动机器人的观测方程中，第  $k$  时刻的观测值  $z_k$  只与这个时刻机器人的位姿状态  $x_k$  有关，而与初始状态、控制输入以及之前的观测无关，所以上式可以化简为：

$$p(x_k | \tilde{x}_0, u_{1:k}, z_{0:k}) = \eta p(z_k | x_k) p(x_k | \tilde{x}_0, u_{1:k}, z_{0:k-1}) \quad (3.6)$$

概率机器人中用置信度表示机器人的位姿状态信息，置信度分布是以传感器观测到的数

据和控制输入信息为条件，关于机器人位姿状态的后验分布：

$$bel(x_k) = p(x_k | \tilde{x}_0, u_{1:k}, z_{0:k}) \quad (3.7)$$

机器人在上一状态  $bel(x_{k-1})$  经过控制量  $u_k$  到达  $x_k$  状态，此时可以预测这一状态即  $p(x_k | \tilde{x}_0, u_{1:k}, z_{0:k})$ ，令其等于  $\overline{bel}(x_k)$ ，表示  $x_k$  的预测概率分布。

$$\begin{aligned} \overline{bel}(x_k) &= p(x_k | \tilde{x}_0, u_{1:k}, z_{0:k-1}) \\ &= \int p(x_k, x_{k-1} | \tilde{x}_0, u_{1:k}, z_{0:k-1}) dx_{k-1} \\ &= \int p(x_k | x_{k-1}, \tilde{x}_0, u_{1:k}, z_{0:k-1}) p(x_{k-1} | \tilde{x}_0, u_{1:k}, z_{0:k-1}) dx_{k-1} \end{aligned} \quad (3.8)$$

通常认为 SLAM 系统服从马尔可夫模型，式 (3.8) 可以进一步化简为：

$$\overline{bel}(x_k) = \int p(x_k | x_{k-1}, u_k) p(x_{k-1} | \tilde{x}_0, u_{1:k-1}, z_{0:k-1}) dx_{k-1} \quad (3.9)$$

贝叶斯滤波算法按照上述过程可以分为两个步骤，第一步通过控制量做出预测  $\overline{bel}(x_k)$ ，第二步是结合观测量进行修正  $bel(x_k)$ ，最终化简之后的表达式：

$$\begin{aligned} bel(x_k) &= \eta p(z_k | x_k) \overline{bel}(x_k) \\ &= \eta p(z_k | x_k) \int p(x_k | x_{k-1}, u_k) p(x_{k-1} | \tilde{x}_0, u_{1:k-1}, z_{0:k-1}) dx_{k-1} \end{aligned} \quad (3.10)$$

其中  $p(z_k | x_k)$  表示观测对应式 (3.3) 中的函数  $g$ ， $p(x_k | x_{k-1}, u_k)$  表示预测对应式 (3.3) 中的函数  $f$ ， $p(x_{k-1} | \tilde{x}_0, u_{1:k-1}, z_{0:k-1})$  表示先验概率。

### 3.2.2 粒子滤波算法研究

粒子滤波是上一小节介绍的贝叶斯滤波的一种非参数化实现方式，能够处理非线性和非高斯的 SLAM 系统，同时能够解决多峰分布的情况。严格的粒子滤波是一个迭代估计的过程， $k$  时刻系统的更新需要上一时刻  $k-1$  的粒子集结合控制输入  $u_k$  以及观测值  $z_k$ ，是一个不断“采样”——“计算权重”——“重采样”的过程。粒子集用  $X = \{(x_k^i, w_k^i) | i = 1, 2, \dots, n\}$  表示， $n$  代表粒子的总个数， $x_k^i$  是机器人  $k$  时刻第  $i$  个粒子估算的位姿状态， $w_k^i$  是第  $i$  个粒子的权重，表示与地图的匹配度。粒子滤波的具体步骤如下：

(1) 用采样粒子进行下一时刻状态预测。根据式 (3.9) 可以知道状态传播的模型  $\overline{bel}(x_k)$ ，已知  $k-1$  时刻粒子的概率分布  $\{(x_{k-1}^i, w_{k-1}^i) | i = 1, 2, \dots, n\}$ ，根据  $k$  时刻的控制输入  $u_k$  得到此时粒子的概率分布  $p(x_k | u_k, x_{k-1}^i)$ ， $i = 1, 2, \dots, n$ 。

(2) 计算每一个粒子的权重。由于无法获取机器人真实的位姿分布，需要对提议分布

$p(x_k|u_k, x_{k-1}^i)$ 进行采样, 结合一定的权重来估计移动机器人的后验概率分布, 结合式 (3.10) 得到其中一个粒子的权重表达式定义如下:

$$W_k = \frac{bel(x_k)}{\overline{bel}(x_k)} = \eta p(z_k|x_k) \quad (3.11)$$

权重被用来衡量移动机器人预测的概率分布和真实概率分布之间的差距, 权重值越大说明两者之间差距越小。

(3) 根据所得权重进行重采样。权重大的粒子将被多次选中, 而权重小的粒子将被删除。重采样结束之后, 对每个粒子的权重做归一化处理, 至此粒子集已经可以尽可能接近目标分布。

在上述粒子滤波的算法中, 当提议分布与目标分布相差较大时, 必须通过大量的采样粒子来描述机器人的后验概率分布, 会出现占用过多系统资源以及粒子耗散等问题, 所以粒子滤波算法还可以进一步改进。

### 3.2.3 RBPf -SLAM 算法研究

RBPf-SLAM 算法就是基于 Rao-Blackwellized 粒子滤波的方法, 通过粒子滤波器来估计移动机器人的运动位姿, 环境地图的估计使用扩展卡尔曼滤波算法实现。SLAM 问题就是已知传感器采集到的数据信息, 在完成移动机器人自定位的同时构建相应的环境地图, 可以用式 (3.12) 表示:

$$p(x_{1:k}, m|u_{1:k}, z_{1:k}) \quad (3.12)$$

RBPf 算法将 SLAM 问题分解为两个相互独立的部分, 一个是移动机器人的自身定位, 另一个是基于已知机器人的位姿进行地图构建, 将式 (3.12) 进一步分解为:

$$\begin{aligned} p(x_{1:k}, m|u_{1:k}, z_{1:k}) &= p(x_{1:k}|u_{1:k}, z_{1:k})p(m|x_{1:k}, u_{1:k}, z_{1:k}) \\ &= p(x_{1:k}|u_{1:k}, z_{1:k})p(m|x_{1:k}, z_{1:k}) \end{aligned} \quad (3.13)$$

其中 $p(x_{1:k}|u_{1:k}, z_{1:k})$ 表示估计的移动机器人路径,  $p(m|x_{1:k}, z_{1:k})$ 表示结合给定机器人的位姿以及外部传感器观测到的数据信息进行地图的构建。

在初始状态时分配粒子权重, 将机器人运动模型的传递和环境地图的更新用概率表示, 计算出粒子权重来表示与当前地图环境的相似程度, 不断更新粒子集来完成机器人运动轨迹的估算以及环境地图的构建。RBPf-SLAM 算法的主要步骤如下:

(1) 粒子初始化。在最开始的陌生环境中无法获取机器人的位姿信息, 根据移动机器人

状态的先验概率  $p(x_0)$  选取  $N$  个粒子, 用  $x_0^i (i=1, 2, \dots, N)$  来表示, 每个粒子刚开始的权重设为  $1/N$ , 总和为 1, 从而得到  $(x_0^i, p_0^i)$ 。如果知道机器人初始位姿, 则将粒子分散在其周围。

(2) 对提议分布采样。由  $k-1$  时刻的粒子集  $\{x_{k-1}^i\} (i=1, 2, \dots, N)$  推出下一时刻  $K$  的粒子集  $\{x_k^i\} (i=1, 2, \dots, N)$ 。

(3) 计算权重并归一化。第  $i$  个粒子的权重:

$$W_k^i \propto W_{k-1}^i P(z_k | m_{k-1}^i, x_k^i) \quad (3.14)$$

$$\bar{W}_k^i = \frac{W_k^i}{\sum_{i=1}^N W_k^i} \quad (3.15)$$

(4) 重采样。权重大的粒子将被多次选中, 权重小的粒子将逐渐消失。

(5) 更新地图。结合粒子集表示的当前位姿以及外部传感器获取的观测数据更新地图  $p(m|x_{1:k}, z_{1:k})$ 。

### 3.2.4 Gmapping 算法研究

Gmapping 算法是 2007 年在 ROS 中开源的 SLAM 功能包, 它基于 3.2.2 小节中 RBPF 粒子滤波框架, 在目前基于滤波器 SLAM 算法中应用最为广泛。Gmapping 算法能够充分利用移动底盘中里程计数据提供机器人的先验位姿, 所以不需要很高的激光雷达频率。该方法可以实时构建室内地图, 在小范围环境中对地图的构建效果较好, 同时不涉及大量的计算。

在 RBPF 算法中每一个粒子都表示其对应的栅格地图, 随着环境范围的逐渐扩大, 有些粒子将会占用较大的内存, 从而增加了计算的复杂度。系统中的粒子数需要保持在一个较小范围内, Gmapping 算法改进了 RBPF 算法中的提议分布, 加入激光雷达最近一帧的观测数据, 因此可以用更少的粒子覆盖机器人的概率分布。

RBPF 算法存在粒子耗散的问题, 因为每一次进行重采样都存在一些随机性, 经过一次又一次的重采样过程, 粒子的多样性就会逐渐消耗, 即最终的所有粒子都从一个或几个粒子复制而来。Gmapping 算法在 RBPF 算法基础上加入自适应重采样, 定义一个变量  $N_{eff}$  见式 (3.16), 来表示当前估计和真实分布之间的差距, 当它低于设定的一个阈值时才进行重采样, 反之保持不变。通过自适应重采样的过程, 就可以极大减少重采样的次数, 有效地解决了粒子耗散问题。

$$N_{eff} = \frac{1}{\sum_{i=1}^N (w_k^i)^2} \quad (3.16)$$

结合上述两点的改进方法，Gmapping 算法的流程如图 3.3 所示，算法的主要步骤如下：

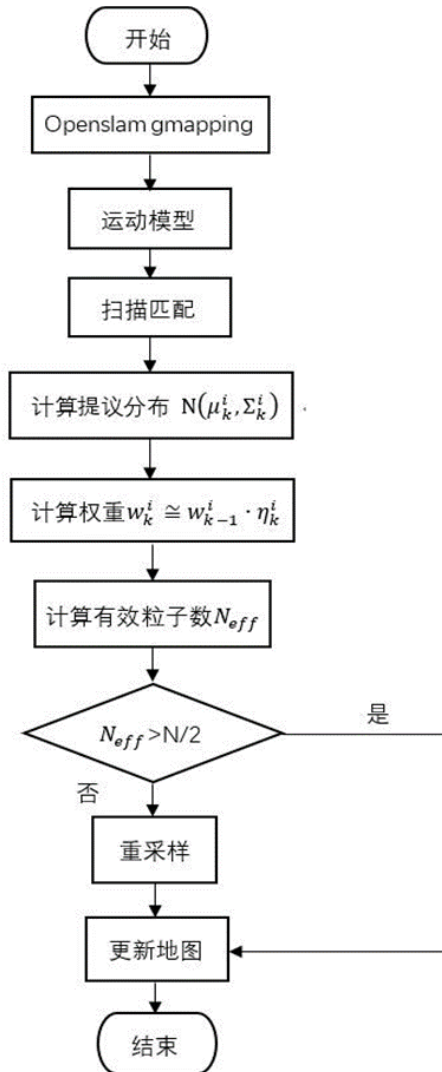


图 3.3 Gmapping 算法流程图

(1) 输入数据：将激光雷达和里程计采集到的数据输入到 ROS 机器人操作系统的 Gmapping 功能包中，为后续步骤提供基础。SLAM 系统初始化时粒子的权重全部设为  $1/N$ ，其中  $N$  表示粒子总数，Gmapping 算法中通常默认粒子总数为 30。

(2) 运动模型。根据  $k-1$  时刻粒子的状态和里程计数据，估算  $k$  时刻粒子的位姿状态，同时考虑加入相应的高斯采样噪声。

(3) 扫描匹配。对系统中的全部粒子进行扫描匹配，这样做的目的是找到各个粒子在时刻  $k$  所处的最佳位置，为后续进行权重的更新提供基础；如果匹配出现错误，则根据运动模型采样粒子，根据传感器观测模型计算采样粒子权重。

(4) 提议分布。RBPF 算法中计算提议分布时只关注运动模型的数据，使得提议分布与

目标分布之间相差很多，只能使用更多的粒子来表示。改进后把最近一次的观测数据 $z_k$  结合到提议分布中，使之更接近机器人目标位姿的后验概率分布，从而使用更少的粒子就可以覆盖机器人的概率分布。提议分布如下：

$$p(x_k|m_{k-1}^i, x_{k-1}^i, z_k, u_{k-1}) = \frac{p(z_k|m_{k-1}^i, x_k) \cdot p(x_k|x_{k-1}^i, u_{k-1})}{p(z_k|m_{k-1}^i, x_{k-1}^i, u_{k-1})} \quad (3.17)$$

提议分布从里程计观测模型转为激光雷达观测模型，该模型的方差较小，假设其服从高斯分布：

$$p(x_k|m_{k-1}^i, x_{k-1}^i, z_k, u_{k-1}) \cong N(\mu_k^i, \Sigma_k^i) \quad (3.18)$$

下面对这个高斯分布进行求解，首先通过极大似然估计得到局部极值：

$$x_k^* = \underset{x_k}{\operatorname{argmax}} \{p(z_k|m_{k-1}^i, x_k) \cdot p(x_k|x_{k-1}^i, u_{k-1})\} \quad (3.19)$$

一般认为 $x_k^*$  在距高斯分布均值的小范围内，所以为了方便计算在 $x_k^*$  周围采样得到  $M$  个位姿：

$$\{x_j | |x_j - x_k^*| < \Delta\} \quad (3.20)$$

最后对采样得到的  $M$  个位姿计算 $p(z_k|m_{k-1}^i, x_j)$ ,  $j=1, 2, \dots, M$ 。同时假设这些采样点服从高斯分布，进而得到具体的高斯分布：

$$\eta_k^i = \sum_{j=1}^M p(z_k|m_{k-1}^i, x_j) \cdot p(x_j|x_{k-1}^i, u_{k-1}) \quad (3.21)$$

$$\mu_k^i = \frac{1}{\eta_k^i} \sum_{j=1}^M x_j \cdot p(z_k|m_{k-1}^i, x_j) \cdot p(x_j|x_{k-1}^i, u_{k-1}) \quad (3.22)$$

$$\Sigma_k^i = \frac{1}{\eta_k^i} \sum_{j=1}^M (x_j - \mu_k^i) \cdot (x_j - \mu_k^i)^T \cdot p(z_k|m_{k-1}^i, x_j) \cdot p(x_j|x_{k-1}^i, u_{k-1}) \quad (3.23)$$

其中， $\mu_k^i$  和  $\Sigma_k^i$  表示  $k$  时刻近似分布中的均值和方差， $\eta_k^i$  表示  $k$  时刻粒子归一化权重。

(5) 权重计算。更新系统中采样粒子的权重，系统在重采样前后都会重新计算权重。权重计算公式如下：

$$w_k^i = w_{k-1}^i \cdot p(z_k|m_{k-1}^i, x_{k-1}^i, u_{k-1}) \cong w_{k-1}^i \cdot \eta_k^i \quad (3.24)$$

(6) 自适应重采样。在重要性采样之前对系统当前的有效粒子数 $N_{eff}$ ，定义见式 (3.16) 进行判断，结合设定的阈值来决定重采样的执行。当有效粒子数小于预设的阈值 $N_{th} = N/2$  ( $N$  表示粒子数) 时，说明当前的估计和实际分布相差较大，此时需要执行重采样；反之当 $N_{eff}$  很

大时说明估计和实际分布基本相似, 就不再需要重采样, 因此极大地减少了粒子的重采样次数。

(7) 地图维护和更新。每个粒子都携带其对应的地图信息, 根据粒子的运动轨迹更新粒子的地图信息。判断最优粒子, 获取其运动轨迹从而更新地图信息。

### 3.3 改进的 Gmapping 算法与实验分析

#### 3.3.1 基于可变粒子数的 Gmapping 算法

通过 3.2 节的具体介绍可以发现, Gmapping 算法是在 RBPF-SLAM 方案的基础上对提议分布和重采样的方式进行了优化。一方面最大程度上减少了重采样的次数, 使得粒子保持一定的多样性, 从而解决了粒子耗散问题; 另一方面在提议分布中加入激光雷达的观测数据, 使得可以用更少的粒子覆盖机器人的概率分布。但是一般 Gmapping 算法中粒子数默认选取 30 个, 不能根据移动机器人所处环境的复杂程度动态调整粒子数。当机器人对复杂的外部环境构建地图时, 少量固定的粒子数不能完全覆盖所处区域的概率分布, 导致机器人定位有偏差, 建图的精度也会有所下降。而当面对结构简单的环境时, 只需要少量的粒子就可能满足地图的精度要求, 但是如果系统中固定的粒子数偏多将导致计算量偏大以及资源占用过多, 从而降低系统运行的效率。

针对上述算法中固定粒子数可能造成地图精度下降以及系统资源过度使用等问题, 本节在 Gmapping 算法的基础上提出可变粒子数的解决方案。该方案将粒子的数目与所处环境的复杂程度进行对应关联, 根据环境的差异使用不同的粒子数来完成位姿的估计以及后续在建图。处于复杂的环境中增加系统中的采样粒子数, 使得机器人的定位和地图构建满足后续导航的精度需要。处于简单的环境中能够适当减少粒子数目, 在保证建图精度的情况下降低系统的计算量, 使得系统运行速度以及资源利用率显著提升。

以 Gmapping 算法框架为主, 启动 SLAM 功能包以后, 在粒子释放之前运行激光雷达程序并接收其六米的扫描半径内点云数据, 系统进一步分析计算数据中点云的波动量。在障碍物多的复杂环境下, 激光的点云波动量大, 而在简单空旷的环境下, 点云波动的数值较小。通过不同场景下的反复实验, 找到激光雷达波动量与粒子数之间的对应关系, 粒子的数目随着点云波动量线性增长。实验表明, 在简单空旷的场景下, 激光点云的波动量一般小于六, 此时粒子数选取五倍的波动量就可以满足建图精度的需要。而随着环境的复杂度增加, 波动量几乎全部大于等于六, 此时粒子数需要增加到六倍的波动量才能达到误差需求。



综上所述,当激光雷达波动量  $n$  大于等于 6 时,粒子数  $N=6n$ ,当  $n$  小于 6 时,粒子数  $N=5n$ 。激光雷达运行以后会实时根据环境复杂程度计算出点云的波动量,从而不断改变所需的粒子数。通过可变粒子数的方案能够在保证建图精度的前提下提升系统的运行时间。该方案的具体流程如图 3.4 所示:

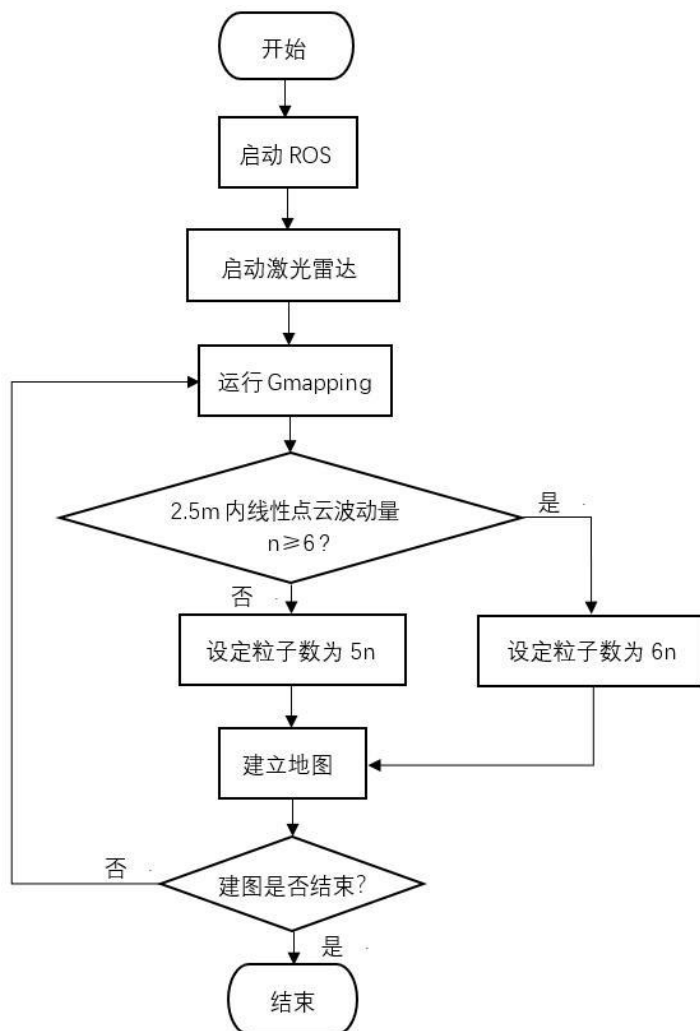


图 3.4 基于可变粒子数的算法流程图

### 3.3.2 算法仿真实验

上述提出的可变粒子数算法与传统 Gmapping 算法主要的不同就是在于粒子数的选取,这一点决定了该方案能否超过改进前的算法。在简单环境下系统中粒子数太多将不会再提升地图的精度,反而会加大计算量,严重影响系统资源的利用率。而当复杂环境下过少的粒子不能完全覆盖机器人的概率分布,影响地图构建的精度。改进之后的算法能够将系统中固定不变的粒子数根据不同环境动态调整,在保证建图效果的情况下提升系统运行时间。为了验证改进后的算法与传统 Gmapping 算法在实际环境中的有效性,使用 MATLAB 软件进行仿真

实验来对比改进前后对相同环境的定位精度。具体对比结果如图 3.5 和图 3.6 所示：

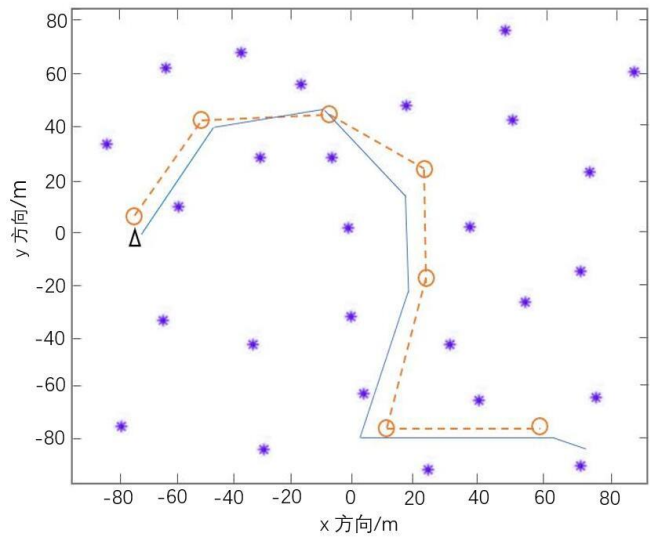


图 3.5 改进前算法仿真

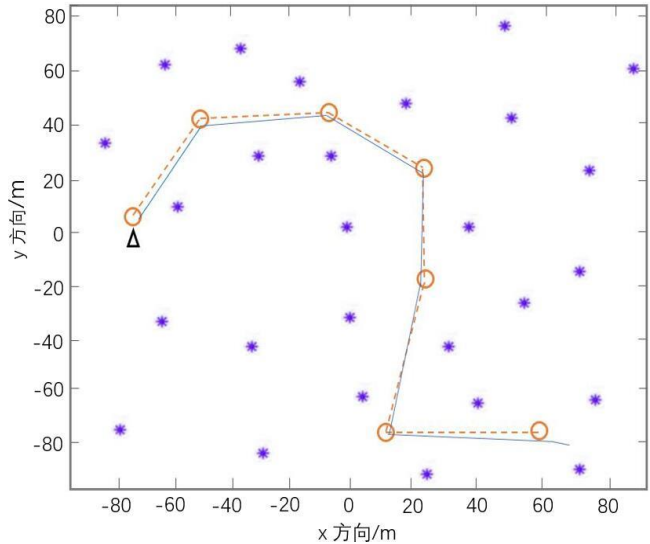


图 3.6 改进后算法仿真

图中紫色的星号表示真实环境下的特征点，橘黄色线条表示机器人理想的运动轨迹，蓝色线条表示经过改进的 **Gmapping** 算法之后输出的运动轨迹。从图中可以明显看出改进后的算法对地图中目标点的估计几乎接近真实，比改进前的精度有了明显的提高。

通过 **MATLAB** 计时器对两种算法的执行效率进行统计，统计结果如表 3.3 所示：

表 3.3 程序运行时间对比

方案	程序运行总时间（s）
传统 <b>Gmapping</b> 算法	2.034
改进 <b>Gmapping</b> 算法	1.879

从表 3.3 可以看出，改进后的滤波算法较原始 **Gmapping** 算法在执行效率上明显提升，

改进后的滤波算法优化了粒子迭代的运算复杂程度，从而使得系统运行效率得到提高。

### 3.4 本章小结

本章详细推导了移动机器人 SLAM 问题的数学模型，根据不同传感器以及不同后端优化算法对常用的 SLAM 方案进行分类，介绍了几种不同的地图表示方法。然后以贝叶斯滤波为基础，详细介绍了粒子滤波、RBPF-SLAM 以及 Gmapping 等基于滤波器的激光 SLAM 算法。最后针对传统算法存在的问题，在 Gmapping 的基础上提出可变粒子数的滤波算法，通过 MATLAB 仿真验证了改进后的算法在机器人定位精准度上的提升。

## 第四章 基于 Kinect 的视觉 SLAM 算法研究

目前通过 Kinect 深度相机采集外部环境，以此来完成机器人的定位与建图成为视觉 SLAM 领域的主流方法，此类方法因具有良好的环境适应性以及稳定性得到广泛应用。以视觉里程计作为整个 SLAM 框架的前端，根据特征点法或直接法粗略估计相邻图像间相机的运动。然后将获取到的相机位姿在后端通过 BA 或位姿或因子等图优化方法进行优化，减少视觉里程计造成的累计误差。最后通过回环检测，构建出全局一致的轨迹和地图。

### 4.1 视觉 SLAM 框架

首先进端通过视觉里程计采集所处环境的图像数据并进行预处理，在特征的提取与匹配过程中会因为环境中存在大量的重复纹理，使得特征描述非常相似从而造成误匹配的情况。如果只根据视觉里程计前后时刻估计运动轨迹，得到的相机位姿会存在一定误差。由于系统中含有噪声的原因，视觉里程计会将上一时刻图像之间存在的运动估计误差累计到下一时刻，造成求解出来的运动轨迹发生累计漂移现象。然后后端优化的首要任务就是解决整个系统中的噪声问题，从而消除累计误差。最后对比图像间的相似度进行回环检测，判断机器人是否到达已经过的位置，解决机器人估算位姿发生漂移的现象，得到全局一致的轨迹和地图<sup>[44]</sup>。整个经典视觉 SLAM 框架的具体流程如图 4.1 所示：

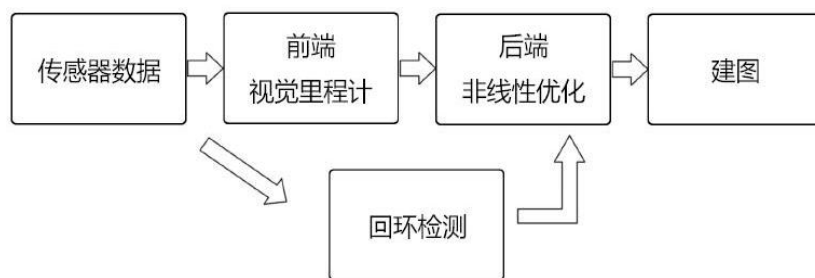


图 4.1 经典视觉 SLAM 流程图

#### 4.1.1 视觉里程计

视觉里程计（Visual Odometry）主要是估计相邻图像之间的相机运动，在整个系统中称为前端。根据 2.3 节对于相机模型与空间点之间几何坐标的转换，里程计只关注相邻时刻的运动，而与之之前的已发生数据不产生关系。根据各时刻估算出来的相机位姿坐标，求解出像素点在空间中的坐标，从而得到环境的局部地图。

然而仅凭视觉里程计估算出来的机器人运动轨迹，会发生不同程度上的累计漂移现象。因为视觉里程计会将之前时刻的误差继续累计到下一时刻，导致一段时间之后估算的运动轨迹误差增大。为了克服漂移现象，通过回环检测的方法识别机器人已到过的地方来消除累计误差，接着后端优化算法接收上述数据信息，矫正机器人的运动轨迹<sup>[45]</sup>。

### 4.1.2 后端优化

后端优化（Optimization）主要是解决视觉里程计估计过程中引入的噪声的问题，从而得到全局一致的运动轨迹和环境地图。实际情况中即使再精确的传感器在工作过程中，也会由于外部环境温度等因素或多或少地引入一些噪声。后端优化所解决的问题反映出 SLAM 系统的本质是对移动机器人位姿和所处环境空间不确定性的估计。后端优化算法主要有基于扩展卡尔曼滤波、以 BA 为主的图优化、位姿图优化以及因子图优化等。

### 4.1.3 回环检测

回环检测（Loop Closing）也叫闭环检测，主要作用是去除视觉里程计引起的位置估计随时间漂移的问题。后端根据接收到的新信息将运动轨迹和地图变换到与回环检测结果一致的状态。通过大量准确的回环检测，能够消除累积误差，从而输出全局一致的轨迹和地图。回环检测在地图构建过程中十分重要，能够保证估计的运动轨迹和地图处于较小的误差范围。当跟踪算法无法工作时，回环检测能够帮助机器人完成重定位，提升了机器人定位精度、可靠性以及稳定性。

### 4.1.4 地图构建

地图构建不像前后端或回环检测一样有固定的表达形式，一组空间中坐标点的集合可以叫做地图，一个完整的 3D 模型也可以是地图，又或是一个带有城镇、沙漠、道路、水流等标记的图片也是地图。地图的表现形式应该与应用场景相匹配，主要分为度量地图与拓扑地图两种。

## 4.2 ORB-SLAM2 算法研究

2017 年 Raul 等人在单目 ORB-SLAM 系统的基础上进一步支持双目相机和 RGB 相机，

形成更具鲁棒性和广泛适用性的 ORB-SLAM2 系统<sup>[46]</sup>。ORB-SLAM 主要分为特征追踪（Tracking）、局部建图（LocalMapping）以及回环检测（LoopClosing）三个线程完成最终的任务。追踪部分表示从采集到的图像中提取 ORB 特征，结合上一帧估算此时姿态，或是通过全局重定位进行位姿初始化。跟踪重建好的局部地图优化位姿，按照规则加入新的关键帧；局部建图主要是插入关键帧，检测最近生成的地图点并进行筛选从而生成新的地图点，经过局部 BA 算法调整后进行筛选去掉冗余的关键帧；回环检测部分是指检测出一批候选关键帧，接着使用 Sim3 算法得到变换关系，回环校正部分包括回环融合以及 Essential Graph 图优化。具体流程如图 4.2 所示：

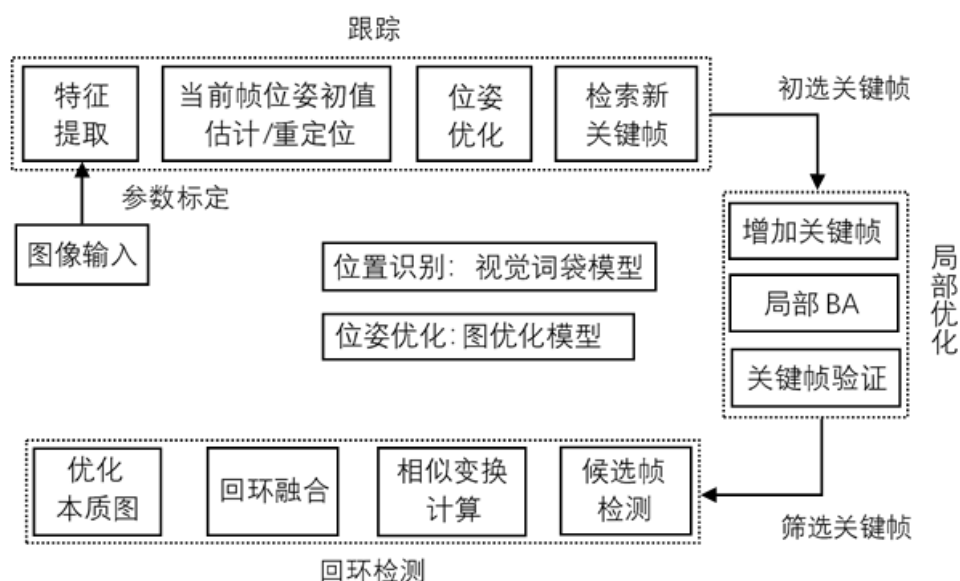


图 4.2 ORB-SLAM 算法流程图

### 4.2.1 特征追踪

ORB-SLAM2 系统采用 ORB 特征作为系统的特征提取和描述方式，ORB 特征只有旋转不变性而没有尺度不变性。ORB-SLAM 采取尺度金字塔的方式，首先在图像的不同尺度上提取 ORB 特征，将每个尺度提取出的特征组合到一起成为该图像提取的 ORB 特征。然后根据上一帧和运动模型进行计算或通过全局重定位估计相机位姿初值，将当前帧的特征点和地图点进行初步的匹配。最后跟踪局部地图并对该局部地图进行 BA 优化，如果当前帧含有重要信息，则将其作为关键帧插入地图并进行后续의局部处理。

### 4.2.2 局部优化

接收从特征追踪线程插入的关键帧并进行预处理, 存储在地图中的地图点需要有较高的质量。如果在地图点后的三个关键帧时间内, 实际看到该地图点的帧数与应该看到的帧数之比小于四分之一就剔除该地图点, 通过三角化生成新的地图点。对上述观察到的所有地图点进行 BA 优化, 剔除地图中冗余的关键帧。

### 4.2.3 回环检测

接收局部优化线程送来的筛选处理后的关键帧, 先检测出一批候补关键帧。然后计算当前关键帧与候补关键帧之间的相似变换, 并据此确定最终的回环关键帧。接着矫正当前帧及与之相连的关键帧位姿。最后对本质图进行位姿图优化, 以便校正尺度漂移。

在 Ubuntu16.04 结合 ROS 系统的环境下使用 Kinect v2 深度相机运行 ORB-SLAM2 功能包的结果如图 4.3 所示:

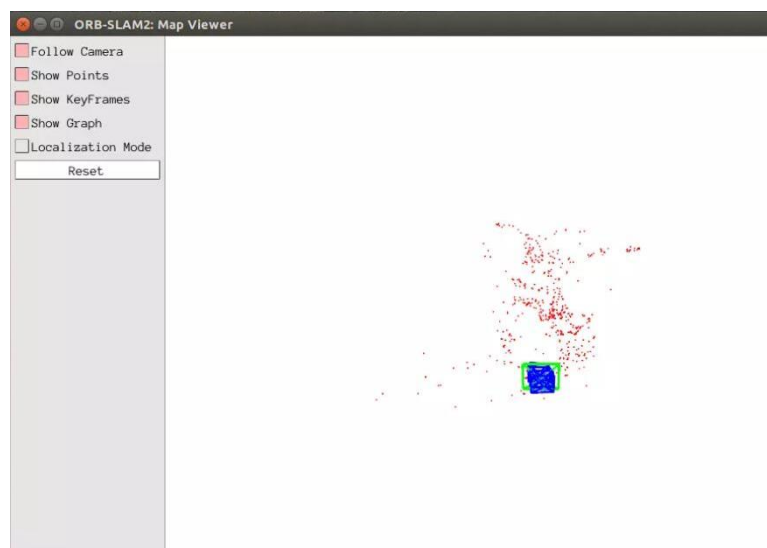


图 4.3 ORB-SLAM2 算法运行图

## 4.3 基于 ORB-SLAM2 的栅格地图构建

ORB-SLAM2 作为视觉 SLAM 中主流的开源算法, 具有很强的精度和鲁棒性。但该算法最终输出相机的运动轨迹以及稀疏点云地图, 所以无法构建用于机器人路径规划和自主导航的地图<sup>[47]</sup>。本节根据采集图像的关键帧、地图点以及机器人位姿得到能用于自主导航的二维栅格地图。首先需要建立一个根据传感器观测数据判断栅格是否被占用的逆传感器模型, 然后结合该模型重新推导地图构建的数学模型, 最后在 ORB-SLAM2 算法的基础上结合上述的逆传感器模型和栅格地图模型得到最终用于导航的二维栅格地图。

### 4.3.1 ORB-SLAM2 建图总体方案

基于 ORB-SLAM2 算法构建栅格地图的主要步骤是：首先在 ROS 中运行 ORB-SLAM2 功能包，然后发布系统生成的所有关键帧、地图点以及机器人位姿  $x_k$ ，接着将订阅到的数据当成逆传感器模型中所需要的信息，最后通过栅格地图模型构建所处环境的地图。具体算法过程如图 4.4 所示：

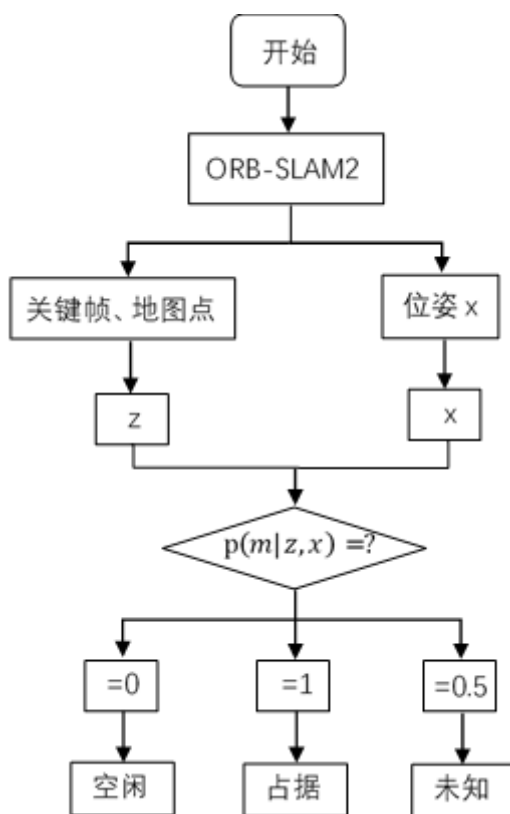


图 4.4 ORB-SLAM2 构建栅格地图流程图

### 4.3.2 逆传感器建模

逆传感器模型表示根据传感器所处环境观测到的数据信息建立对应的环境地图，同时认为观测过程中加入的噪声符合高斯分布。我们可以用一个概率值来表示地图中的小栅格是否存在障碍物，用  $m_i$  变量表示栅格单元是否被占据，其中 *occ* 代表占据，*emp* 代表空闲。根据栅格的两种状态得到：

$$p(m_i = occ) + p(m_i = emp) = 1 \quad (4.1)$$

构建机器人栅格地图时第一步需要根据传感器模型对其观测范围  $r$  进行解释，该模型用  $p(r|z)$  表示，其中  $z$  是传感器检测到的真实距离。第二步利用观测数据来更新栅格单元状态。



根据贝叶斯格式将第二步用式 (4.2) 表示:

$$p(m_i = occ|r) = \frac{p(r|m_i = occ)p(m_i = occ)}{\sum_{m_i} p(r|m_i) p(m_i)} \quad (4.2)$$

式 (4.2) 中的  $p(r|m_i)$  并不等于传感器模型  $p(r|z)$ , 它们之间的关系为:

$$p(r|z) = p(r|m_i = occ \wedge m_i = emp) \quad (4.3)$$

根据  $m_i$  可能出现的所有状态得到:

$$p(r|m_i = occ) = \sum_{\{M_{m_i}\}} p(r|m_i = occ, M_{m_i}) p(M_{m_i}|m_i = occ) \quad (4.4)$$

其中  $\{M_{m_i}\}$  表示所有栅格地图的集合, 同理可得:

$$p(r|m_i = emp) = \sum_{\{M_{m_i}\}} p(r|m_i = emp, M_{m_i}) p(M_{m_i}|m_i = emp) \quad (4.5)$$

式 (4.4) 和 (4.5) 中的  $p(M_{m_i}|m_i)$  的值由对应栅格的先验概率决定, 当取到最大熵先验时,  $p(m_i = occ) = p(m_i = emp) = 0.5$ 。

为了具体表示符合实际的逆传感器模型, 本文用克罗内克函数来表示理想传感器的概率分布  $p(r|z) = \delta(r - z)$ , 求解式 (4.2) 得到封闭解为理想状态下的逆传感器模型  $h(r)$ :

$$h(r) = \begin{cases} 0 & r < z - \frac{L}{2} \\ 1 & z - \frac{L}{2} \leq r < z + \frac{L}{2} \\ 0.5 & \text{else } r > 0 \end{cases} \quad (4.6)$$

其中  $L$  表示传感器观测范围  $r$  附近的一段区域宽度,  $h(r) = 0$  表示栅格空闲即无障碍物,  $h(r) = 1$  表示栅格被占据即有障碍物,  $h(r) = 0.5$  表示传感器观测不到的未知区域。式 (4.6) 中没有把概率值 1 对应到  $r$  而是在其附近的宽度  $L$  上, 这样能够保证在一定误差范围表示栅格被占据。真实环境中的逆传感器模型  $H(r)$  可以通过  $h(r)$  和高斯噪声模型  $f(r)$  通过卷积运算:

$$H(r) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(r-0)^2}{2\sigma^2}} (h * f)(r) \int_{-\infty}^{\infty} h(\tau) f(r - \tau) d\tau \quad (4.7)$$

结合式 (4.6) 中的分段函数对式 (4.7) 中  $(h * f)(r)$  进行分段卷积:

$$(h * f)(r) = -\frac{1}{2} k \left( \operatorname{erf}\left(\frac{r-b}{\sqrt{2}\sigma}\right) - \operatorname{erf}\left(\frac{r-a}{\sqrt{2}\sigma}\right) \right) \quad (4.8)$$

其中  $\operatorname{erf}(x) = 2/\sqrt{\pi} \int_0^x e^{-t^2} dt$  表示误差函数,  $(a, b)$  表示观测值  $z$  和范围  $r$  上的间隔,

$k \in \{0, 0.5, 1\}$  表示  $p(m_i)$  能取到的值。最终得到分段卷积的结果:

$$(h * f)(r) = -\frac{1}{4} \operatorname{erf}\left(\frac{r - z - \frac{L}{2}}{\sigma\sqrt{2}}\right) + \frac{1}{2} \operatorname{erf}\left(\frac{r - z + \frac{L}{2}}{\sigma\sqrt{2}}\right) - \frac{1}{4} \operatorname{erf}\left(\frac{-z}{\sigma\sqrt{2}}\right) \quad (4.9)$$

### 4.3.3 栅格地图模型

占据栅格地图将所处环境分成若干尺度相等的栅格单元，借助这些栅格单元来描述与之对应的外部环境，同时记录该栅格是否存在障碍物，即栅格单元状态是被占用、空闲或是无法获取。在占用栅格地图中，用概率值 0 来表示栅格空闲状态，用 1 表示栅格此时被占用，用 0.5 表示无法获取该状态。同时传感器在观测过程中不可避免会引入一些噪声，因此栅格地图模型中还需要考虑加入噪声等因素<sup>[48]</sup>。

在理想状态下，已知机器人 1 到 k 时刻传感器的观测数据  $z$  和机器人位姿  $x$ ，关于栅格地图  $m$  的后验概率写成  $p(m|z_{1:k}, x_{1:k})$ 。将地图  $m$  分成  $i$  个尺度相同的栅格单元  $m = \{m_i\}$ ，用  $p(m_i = 1)$  表示栅格处于占据状态，用  $p(m_i = 0)$  表示栅格处于空闲状态，用  $p(m_i = 0.5)$  表示栅格处于未知状态，可得：

$$p(m_i = 0) = p(\overline{m}_i) = 1 - p(m_i = 1) \quad (4.10)$$

根据马尔可夫过程的假设，每个栅格单元  $m_i$  之间的后验概率相互独立，所以  $N$  个栅格可是表示成：

$$p(m_1, m_2, \dots, m_N | z_{1:k}, x_{1:k}) = \prod_i^N p(m_i | z_{1:k}, x_{1:k}) \quad (4.11)$$

地图的更新需要不断加入相应栅格的状态信息，根据贝叶斯公式可以得到：

$$\begin{aligned} p(m_i | z_{1:k}, x_{1:k}) &= \frac{p(z_k | m_i, z_{1:k-1}, x_{1:k}) p(m_i, z_{1:k-1}, x_{1:k})}{p(z_k | z_{1:k-1}, x_{1:k})} \\ &= \frac{p(z_k | m_i, x_k) p(m_i, z_{1:k-1}, x_{1:k-1})}{p(z_k | z_{1:k-1}, x_{1:k})} \end{aligned} \quad (4.12)$$

将式 (4.12) 中的  $p(z_k | m_i, x_k)$  进一步使用贝叶斯公式，可得：

$$p(m_i | z_{1:k}, x_{1:k}) = \frac{p(m_i | z_k, x_k) p(z_k | x_k) p(m_i | z_{1:k-1}, x_{1:k-1})}{p(m_i) p(z_k | z_{1:k-1}, x_{1:k})} \quad (4.13)$$

同理可得：

$$p(\overline{m}_i | z_{1:k}, x_{1:k}) = \frac{p(\overline{m}_i | z_k, x_k) p(z_k | x_k) p(\overline{m}_i | z_{1:k-1}, x_{1:k-1})}{p(\overline{m}_i) p(z_k | z_{1:k-1}, x_{1:k})} \quad (4.14)$$

下面计算式 (4.13) 与 (4.14) 之比：

$$\frac{p(m_i | z_{1:k}, x_{1:k})}{p(\overline{m}_i | z_{1:k}, x_{1:k})} = \frac{p(m_i | z_k, x_k)}{1 - p(m_i | z_k, x_k)} \frac{p(m_i | z_{1:k-1}, x_{1:k-1})}{1 - p(m_i | z_{1:k-1}, x_{1:k-1})} \frac{1 - p(m_i)}{p(m_i)} \quad (4.15)$$

其中 $p(m_i|z_k, x_k)$ 表示栅格单元 $m_i$ 在 $k$ 时刻的后验概率分布,  $p(m_i)$ 表示地图的先验分布,  $p(m_i|z_{1:k-1}, x_{1:k-1})$ 表示 $k-1$ 时刻的递归项。

为了表达方便对式(4.15)取对数, 同时令 $L(m_i|z_{1:k}, x_{1:k}) = \log(p(m_i|z_{1:k}, x_{1:k})/(1 - p(m_i|z_{1:k}, x_{1:k})))$ , 可以将(4.15)改写成:

$$L(m_i|z_{1:k}, x_{1:k}) = L(m_i|z_k, x_k) + L(m_i|z_{1:k-1}, x_{1:k-1}) - L(m_i) \quad (4.16)$$

其中 $L(m_i|z_k, x_k)$ 就是 4.3.1 小节所说的逆传感器模型。按照最大后验概率进行地图构建:

$$\begin{aligned} m^* &= \operatorname{argmax}_m p(z_1, z_2, \dots, z_k | m, x_1, x_2, \dots, x_k) \\ &= \operatorname{argmax}_m \prod_{k=1}^K p(z_k | m, x_k) \end{aligned} \quad (4.17)$$

#### 4.3.4 实验与结果分析

本文需要根据实验分析 4.3.1 小节定义变量  $L$  的取值,  $p(m_i = 1)$  的值随着  $L$  的增大也逐渐增加, 由此可以看出随着  $L$  的增大, 越来越多的栅格单元被认为处于被占据状态。当  $L$  的值增大到一个阈值时, 所处环境中实际没有障碍物的地方也会被判定为有障碍物。通过大量实验得出当  $L$  的取值为栅格单元对角线的长度时, 出现错误的概率最小。

在 Kobuki 移动底盘上搭载 Kinect V2 深度相机进行栅格地图构建, 地点选在学校工程中心的会议室, 如图 4.5 所示, 最终构建出的栅格地图如图 4.6 所示:



图 4.5 会议室真实环境

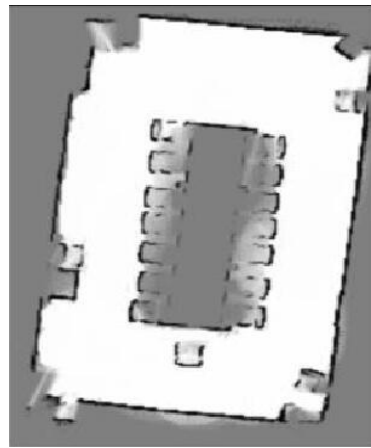


图 4.6 会议室二维地图

### 4.4 本章小结

本章详细介绍了经典视觉 SLAM 框架中的视觉里程计、后端优化、回环检测以及地图构

建等部分。然后对开源的 ORB-SLAM2 算法中特征追踪、局部优化以及回环检测三个线程进行分析，在 ROS 系统中运行该算法得到所处环境的点云以及相机运动轨迹图。最终在 ORB-SLAM2 算法的基础上对逆传感器进行建模，结合栅格地图模型进行建图，在学校工程中心会议室场景下得到可用于导航的二维栅格地图。

## 第五章 基于激光雷达和视觉融合的 SLAM 算法研究

移动机器人越来越广泛的应用在工业领域以及我们的日常生活中，其应用场景也逐渐变的更加复杂和多样，这也导致移动机器人需要更加准确的自身定位以及精度更高的外部环境地图。二维激光雷达只能扫描到一个平面内的环境信息，而视觉 SLAM 中传感器受环境光的影响较大，利用两者的冗余性和互补性进行数据融合，能够进一步提升建图的准确性和系统可靠性。数据融合在移动机器人上的应用已经取得了很大的进展，相较于单一传感器表现出无法比拟的优越性。

### 5.1 多传感器信息融合

随着科学技术的不断发展，20 世纪 70 年代首先在军事领域产生了多传感器信息融合这一新兴学科，后来迅速得到关注和应用。从 20 世纪 80 年代开始，许多科研专家开始在移动机器人领域应用融合技术。1997 年到达火星的旅居者号（Sojourner）移动机器人搭载英特尔处理器、太阳能板、无线电装置、相机以及光谱仪等设备，结合信息融合技术顺利完成探测任务。信息融合也称为数据融合，是对所有信息进行的一种特殊的处理方式，从而获取更加全面丰富的环境信息<sup>[49]</sup>。经过近 40 年的不断发展，信息融合技术已经在军事、移动机器人、智能交通以及远程医疗等领域得到了广泛的应用。将来多传感器融合技术的突破主要表现在原有算法的改进和新算法的产生、传感器性能和精准度的提升以及多层次的数据信息融合。

#### 5.1.1 信息融合层次

多传感器数据融合的层次结构是指在一个系统对获得的数据使用不同的方式进行多个层次的数据处理。根据数据抽象的三个层次，将多传感器数据融合分为数据层（像素级）融合、特征层融合以及决策层融合。

数据层融合是处于最底层的融合，如图 5.1 所示，指获取传感器观测到的原始数据没有经过太多处理就进行分析和融合。该层融合的好处是尽可能地保留环境中的原始信息，能够提高该层仅有的细节信息数据。但是系统需要处理的信息量大，导致处理过程耗时久和实时性差。数据层融合要求数据信息只能来自同一类传感器，同时系统的抗干扰能力也较差。

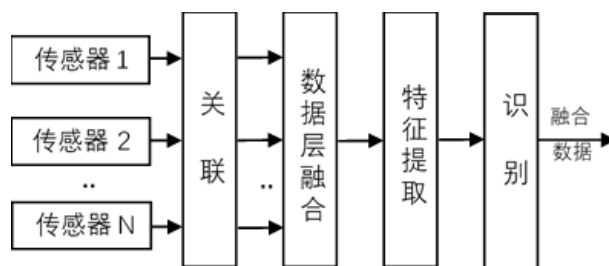


图 5.1 数据层融合

特征层融合是处于中间层的融合，如图 5.2 所示，指从传感器采集到的原始数据中进行特征信息提取，然后对这些特征进行分类和综合处理，提取到的特征包括环境的方向、拐角以及楼梯。该层融合的好处是在保存一定原始数据的情况下提高了数据处理过程的实时性。但是存在一些数据信息的丢失，需要进行有效的数据预处理。

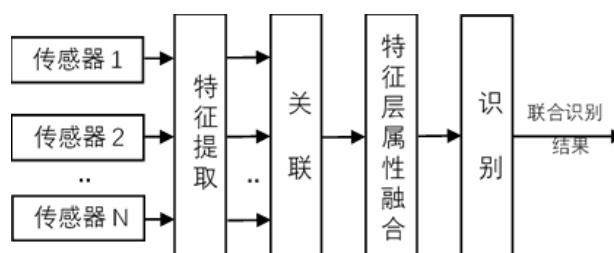


图 5.2 特征层融合

决策层融合处于上层融合，如图 5.3 所示，指根据传感器的信息对目标属性做出决策，然后融合得到的决策结果，最终得出整体一致的结果。该层能够支持不同种类的传感器融合，具有很强的稳定性，但是在获取各自传感器的决策结果时将增大数据的转换量。

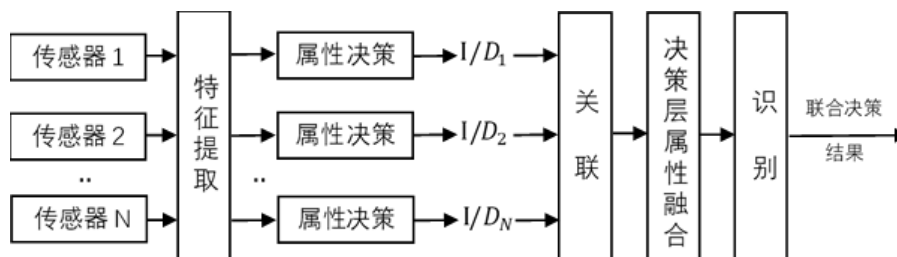


图 5.3 决策层融合

根据上述对融合过程三个层次的了解，融合的信息越接近信息源，融合后的结果精准度就越高。随着融合层次的提高，系统的容错能力以及抗干扰能力将提高，但是传感器信息中的细节将会丢失，导致数据的精度下降<sup>[50-51]</sup>。因此需要根据系统实现的可能性、处理能力以及计算能力等情况，综合权衡后再选择合适的融合层次。

### 5.1.2 融合方法分析

融合算法能够将多个传感器采集到的数据信息进行处理的具体方式体现出来,所以选择适合的融合算法至关重要。我们需要将获取到的传感器数据根据实际应用场景、结合恰当的融合层次以及合适的数学方法进行处理。在选择具体算法的过程中需要考虑方法的稳定性和可靠性、运行的效率和精度、与系统预处理和后续识别模块的衔接以及对传感器原始数据的要求等多方面因素。一般情况下多传感器数据融合方法包括随机法和人工智能法,随机法中主要包括加权平均、卡尔曼滤波、贝叶斯估计以及 D-S 证据推理等,人工智能法主要有模糊逻辑理论、人工神经网络等。下面对上述几种常用的融合算法进行介绍分析:

#### (1) 加权平均法

加权平均法将传感器处理后的信息根据相应比重进行加权平均,所得结果作为融合值,需要在融合前确定每个传感器所占比例。该方法常用于动态环境中直接对原始数据进行处理,十分简单方便。

#### (2) 卡尔曼滤波

卡尔曼滤波常用来融合处理较低层次的动态数据,该方法能够估计出当前系统的状态量并预测下一时刻的状态。卡尔曼滤波中状态的递推决定了其对数据的处理将占用较小内存,使得系统运算效率高、实时性好。

#### (3) 贝叶斯估计

贝叶斯估计常用于静态环境中融合较高层次的数据信息,在使用该方法之前需要尽量准确地给出系统的先验概率分布,否则得出的融合结果与实际相差较大。

#### (4) D-S 证据推理

D-S 证据理论可理解成贝叶斯估计的升级,主要对传感器数据进行定性融合。该方法的主要步骤是首先将多个传感器采集到的信息汇总,然后把获得观测信息推断成对应的目标报告,最后一步需要通过尽可能来自相同传感器的连续报告结果替换原有数据。在 D-S 证据理论实际使用中有时很难满足证据之间相互独立这一要求。

#### (5) 模糊逻辑理论

模糊逻辑理论常用于高层次的数据融合,逻辑模糊表示多值逻辑,使用 0 到 1 之间的数字描述目标的真实度,推理的过程也就表示传感器数据在进行融合。该方法中涉及的隶属函数主要根据先前的经验、认知以及推理等人为因素决定,所以对于数据信息的表示和处理缺乏客观性。它能够克服一些概率论方法所存在的缺点,对信息的描述以及处理更符合人类的思维方式。

#### (6) 人工神经网络

神经网络通过算法对样本数据不断训练得到相应的逻辑推理，然后结合得到的逻辑进行不断地学习与融合。首先根据样本数据的相似性得到分类标准，然后通过算法的不断训练得到不确定性推理机制，最后使用神经网络的处理和推断能力得到数据融合的结果。

## 5.2 激光雷达和视觉融合方案

### 5.2.1 贝叶斯估计融合法

贝叶斯估计是一种根据后验概率或条件概率来估计未知状态的数据融合方法。该方法通过观测得到已知的状态向量  $Z$  对一个  $n$  维状态向量  $X$  进行估计，并且已知向量  $Z$  中含有未知向量  $X$  的信息。贝叶斯估计给出了计算后验概率的方法，假设  $K$  时刻的概率是  $x_k$ ，传感器观测到的状态向量  $Z^k = \{z_1, z_2, \dots, z_k\}$ ，可得：

$$p(x_k|Z^k) = \frac{p(z_k|x_k)p(x_k|Z^{k-1})}{p(z_k|Z^{k-1})} \quad (5.1)$$

其中  $p(z_k|x_k)$  表示传感器的观测模型的似然函数， $p(x_k|Z^{k-1})$  表示先验概率。

在更新栅格地图的过程中， $O$  代表被传感器观测到障碍物事件，即栅格被占据。 $\bar{O}$  表示栅格空闲。 $E$  表示存在障碍物事件，根据贝叶斯估计可以得到：

$$p(E|O) = \frac{p(O|E)p(E)}{p(O|E)p(E) + p(O|\bar{E})p(\bar{E})} \quad (5.2)$$

$$p(E|\bar{O}) = \frac{p(\bar{O}|E)p(E)}{p(\bar{O}|E)p(E) + p(\bar{O}|\bar{E})p(\bar{E})} \quad (5.3)$$

其中  $p(E)$  表示先验概率， $p(E|O)$  和  $p(E|\bar{O})$  表示观测模型。

根据 4.3.3 节对栅格地图模型的推导，结合贝叶斯估计方法可以得到栅格地图融合的公式：

$$P^o = \frac{P_s^o P_m^o}{P_s^o P_m^o + (1 - P_s^o) + (1 - P_m^o)} \quad (5.4)$$

其中  $P_m^o$  表示栅格地图处于占据状态的先验概率， $P_s^o$  表示通过传感器返回的栅格处于占据的概率， $P^o$  表示经过贝叶斯估计融合后的结果。

### 5.2.2 激光雷达与深度相机融合方案

本文的融合方案采用贝叶斯方法，以第三章提出的改进 Gmapping 算法构建的局部栅格地图为主，将第四章提出的基于 ORB-SLAM2 生成的栅格地图按顺序加入来更新已知地图，



最终融合得到可用于导航的环境地图。融合过程如图 5.4 所示：

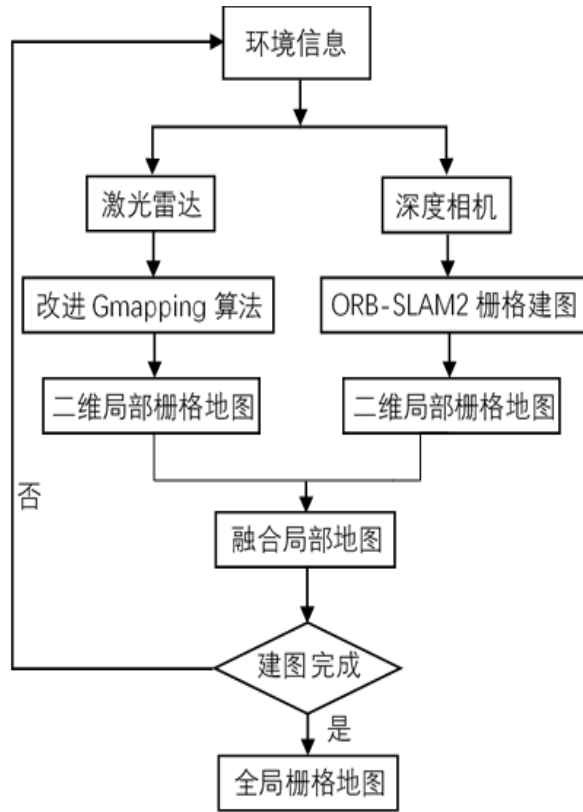


图 5.4 激光与视觉 SLAM 融合流程图

每个构建出来的栅格地图单元包含传感器原始数据地图以及局部更新图，其中按照贝叶斯估计更新出来局部地图是全局地图更新的基础，新栅格地图按顺序更新相应的旧地图。

可以根据式（5.4）进行激光雷达和视觉传感器构建栅格地图，对于同一坐标下的局部栅格地图进行融合，融合规则如表 5.1 所示：

表 5.1 激光雷达与深度相机融合规则

激光雷达	Kinect 深度相机		
	占据	空闲	未知
占据	占据	占据	占据
空闲	占据	空闲	空闲
未知	占据	空闲	未知

当栅格单元被占据状态的值 $P^o$ 大于我们设定的阈值  $T$ ，则将 $P^o$ 改为  $1$ ，否则该值不变。局部栅格地图的融合过程如下：

第一步是得到激光雷达和深度相机两个传感器所对应各个栅格单元概率值：

$$P_{n=1,2}^o = \begin{cases} 1 & P^o > T \\ P^o & else \end{cases}$$

(5.5)

其中 $P_{n=1}^o = P_1^o$ ， $P_{n=2}^o = P_2^o$ 分别代表激光或视觉传感器对应的栅格单元概率值，阈值  $T$  设为 0.5。

第二步是将已知的先验概率结合贝叶斯估计得到融合后栅格单元概率：

$$P^o = \frac{P_1^o P_2^o}{P_1^o P_2^o + (1 - P_1^o) + (1 - P_2^o)} \quad (5.6)$$

### 5.2.3 融合算法仿真实验

以 ubuntu16.04 下的 ROS-kinetic 为调试环境，因此需要把 KITTI 数据集中的数据制作成 ROS 中的 bag 文件，通过 pip 工具下载安装开源功能包 kitti2bag 进行数据转换。借助已有开源精度评价算法 EVO 分别对单一激光雷达 SLAM 和融合的 SLAM 进行仿真验证，最终得到运动轨迹如图 5.5 (a) 和 (b) 所示，其中灰色虚线表示真值。

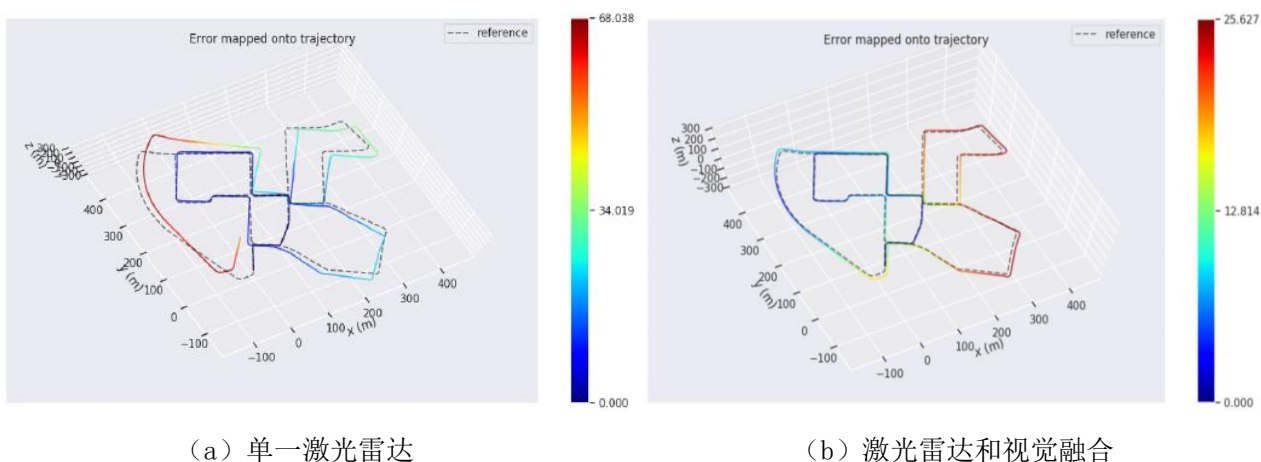


图 5.5 运动轨迹图

EVO 算法评价数据有 evo\_rpe 和 evo\_ape 两种模式，前一种代表每段距离内的误差，后一种代表绝对误差随路程的累计。根据上面的运动轨迹使用 evo\_ape 指令得到总的累计误差，如图 5.6 和 5.7 所示：



图 5.6 单一激光雷达的总累计误差图

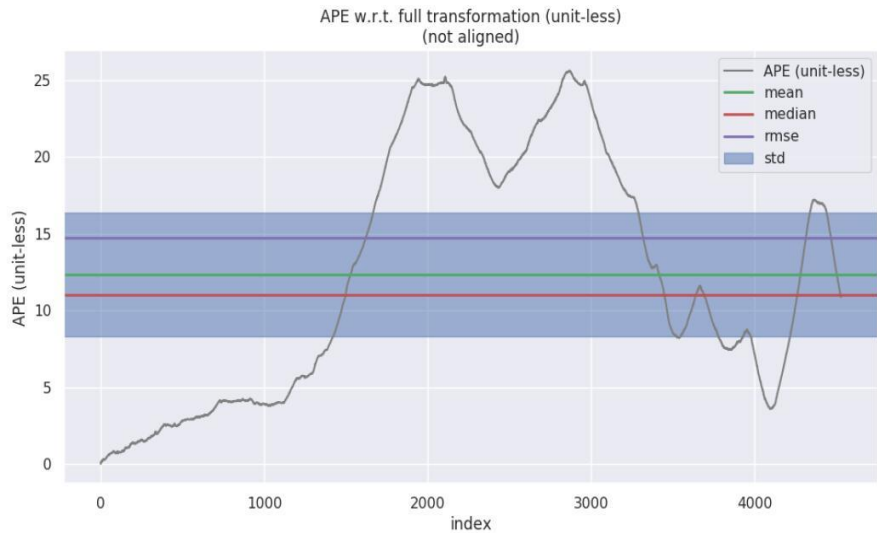


图 5.7 融合后的总累计误差图

从上图 5.6 和 5.7 中得到单一传感器总累计误差的最大值为 68.038200、均值为 22.565926，融合后的总累计误差的最大值为 25.518332、均值为 12.312251。融合后的累计误差远远小于单一激光雷达的误差，融合后的运动轨迹与真实轨迹更加接近。

5.3 实验平台搭建

本文提出了基于 ROS 操作系统的融合激光雷达与视觉联合 SLAM 的系统架构，如图 5.8 所示。该系统主要从上到下主要分为：人机交互层、SLAM 算法实现层以及物理层。

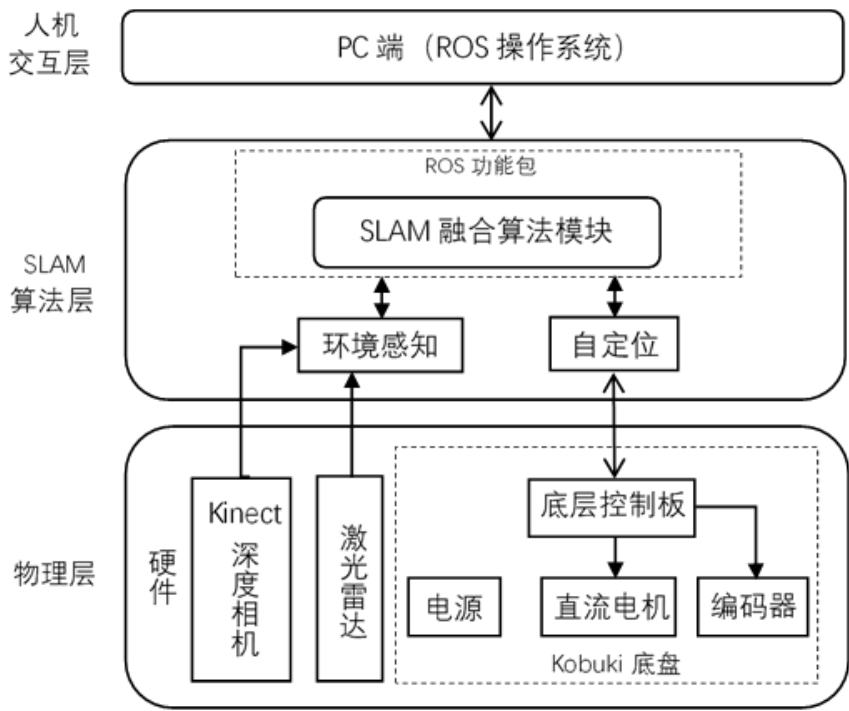


图 5.8 系统设计结构图

其中最下面的物理层主要由 Kinect 深度相机、二维激光雷达以及 Kobuki 移动底盘组成，为上一层提供机器人的运动控制信息以及采集到的环境信息；SLAM 算法层只要将第三章和第四章提出的激光雷达和视觉传感器 SLAM 进行融合，构建出可用于导航的更加准确的环境地图；人机交互层主要是装有 ROS 系统的 PC 端可以与工控机进行数据传输，可向机器人发送指令也可以接收机器人的状态信息。

### 5.3.1 硬件环境

本文实验所采用的机器人硬件平台如图 5.9 所示，主体采用 Kobuki 两轮差速移动底盘，系统控制器采用现有的 XCY 工控机，传感器使用 RPLIDAR A1 激光雷达和 Kinect V2 深度相机。移动底盘使用串口连接到工控机来控制机器人的运动，同时为传感器提供工作电压。工控机与激光雷达和深度相机分别连接来接收传感器采集到的环境信息。工控机中的网口与路由器相连，PC 上位机联网后能够在 ROS 下进行一系列可视化操作。激光雷达相较于底盘的安装高度为 40cm，Kinect 深度相机相较于底盘的安装高度为 22cm。



图 5.9 移动机器人平台

### 5.3.2 软件环境

本文实验采用的软件平台是 Ubuntu16.04 下安装的 ROS-Kinetic 机器人操作系统。ROS 系统是用以编写机器人相关代码程序的软件架构，能够支持多种编程语言便于代码的移植，同时提供了很多免费开源的功能包和软件包。ROS 中采取点对点设计，将每个节点作为一个

小模块，节点间可以单独编译处理，有效的提高了系统的集成度以及代码的复用性。

本文融合的 SLAM 系统需要处理激光雷达、深度相机以及机器人运动控制的数据信息，因此将软件系统框架分为机器人运动模块、激光 SLAM 模块以及深度相机 SLAM 模块，如图 5.10 所示。

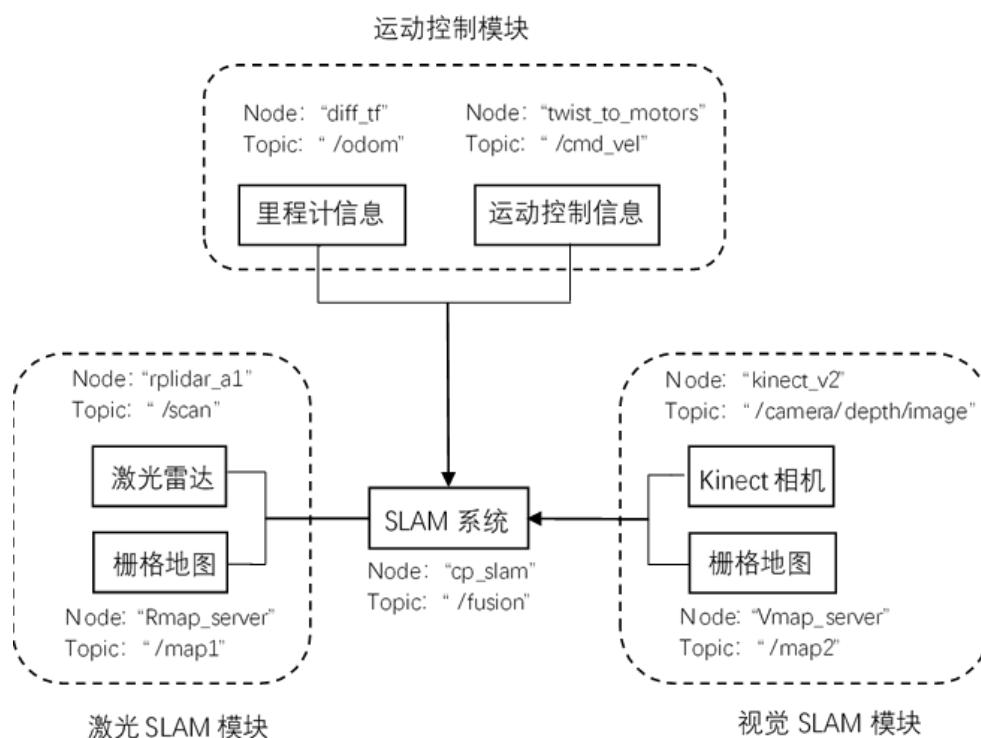


图 5.10 系统软件结构图

在上述的软件结构图中 SLAM 系统模块将订阅激光雷达发布的“/scan”话题、Kinect 深度相机发布的“/camera/depth/image”话题以及运动控制模块的“/odom”话题，将两个传感器构建的局部栅格地图进行融合，最终发布融合地图话题“/fusion”供可视化软件查看。

## 5.4 实验和结果分析

大部分的移动机器人室内地图构建只通过单一的传感器来获取外部环境信息，然后通过现有的激光或视觉 SLAM 算法完成最终的建图。本文的建图思路是使用激光雷达与深度相机分别构建局部栅格地图，采用贝叶斯估计的数据处理方法将两者进行融合。激光 SLAM 方面改进传统 Gmapping 算法，视觉 SLAM 方面在 ORB-SLAM2 的基础上提出实时栅格地图构建方法，最终得到精度更高，环境信息更加丰富的可用于导航的地图。

本文的 5.2 节已经通过 KITTI 数据集验证了所提融合算法的有效性。为了进一步验证在实际环境中的建图效果，在一个 4m\*5.5m 的室内空间中放置各种常见的障碍物进行实际环境

的搭建，在相同环境下分别完成单一传感器 SLAM 以及融合 SLAM 算法建图。

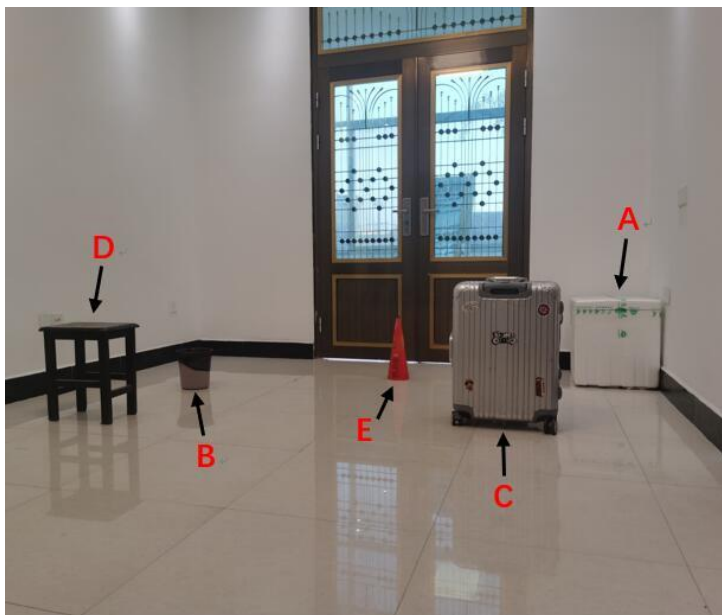


图 5.11 实际测试环境

实际环境如图 5.11 所示,右上角放置高为 50cm 的障碍物泡沫箱 A,障碍物 B 为高度 20cm 的圆形垃圾桶,障碍物 C 为高度 62cm 的行李箱,障碍物 D 为高度 43cm 的木板凳,障碍物 E 为高度 24cm 的烟花。

控制机器人进行匀速运动,第一次实验以改进 Gmapping 算法进行地图构建,最终结果如图 5.12 所示;第二次实验基于 ORB-SLAM2 算法进行地图构建,最终结果如图 5.13 所示;第三次实验在激光雷达和 Kinect 深度相机联合情况下进行地图构建,最终结果如图 5.14 所示。

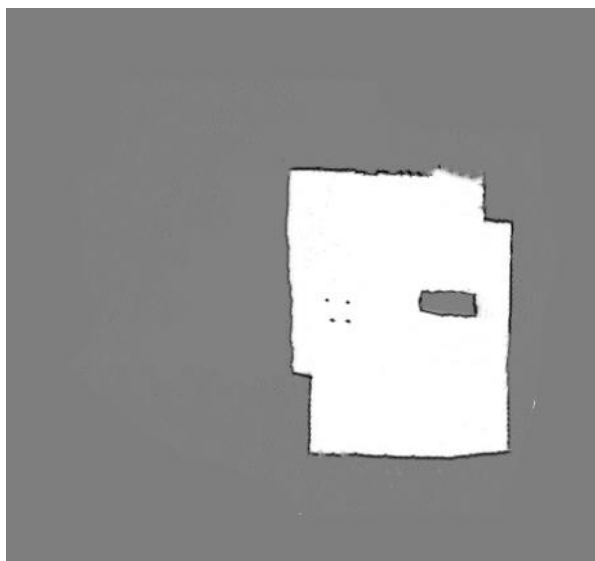


图 5.12 基于改进 Gmapping 栅格地图

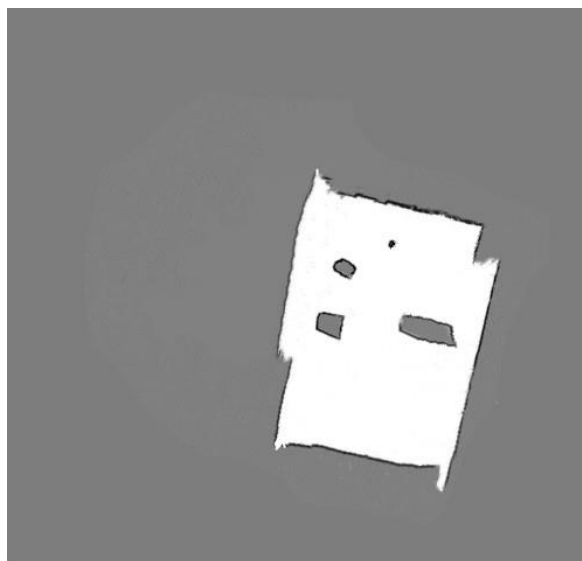


图 5.13 基于 ORB-SLAM2 栅格地图

上述栅格地图中黑色表示存在障碍物即机器人不能通过的区域,白色表示空闲区域即不

存在障碍物，灰色表示未知区域即无法判断是否存在障碍物。图 5.12 在激光雷达作为传感器的方案下进行建图，因为激光雷达只能检测所处平面的障碍物，所以低于激光雷达高度的障碍物 B 和 E 没有被发现，障碍物 D 也只能检测到四条板凳腿。图 5.13 在 Kinect 深度相机作为传感器，能够很好地检测到所有的障碍物，但是在拐角和边缘处参差不齐，不能满足机器人的导航需求。

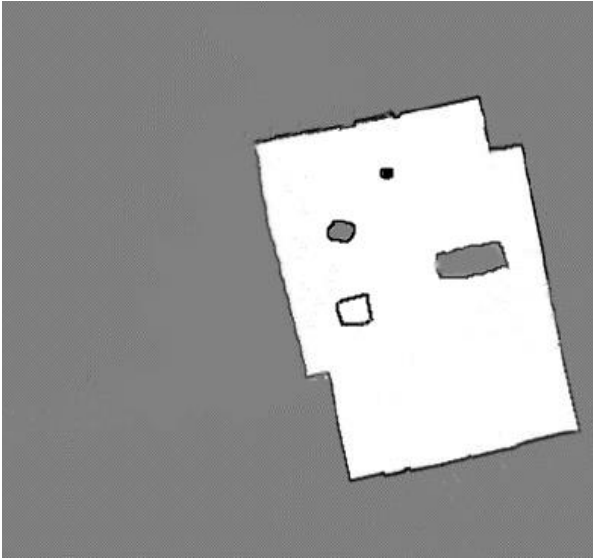


图 5.14 激光雷达与视觉融合栅格地图

图 5.14 在融合二维激光雷达和视觉 SLAM 的方案下构建出来的地图将五个障碍物全部检出，正是因为加入了 Kinect 深度相机提供的环境信息，解决了激光雷达只能扫描平面的问题，同时构建出来的地图边缘更加准确。综上所述，本文提出的融合激光雷达与深度相机的 SLAM 方案构建出来的地图明显优于单一传感器，栅格地图中障碍物检测全面，地图边缘清晰精准，证明该方案具体有较高的一致性和实用性。

本文实验的实际环境面积约为  $22m^2$ ，移动机器人在运行过程中的轨迹约为 30m，两次地图构建实验所运行的时间如表 5.2 所示：

表 5.2 地图构建所用时间

方案	时间(s)	速度
激光雷达	291	0.103m/s
视觉相机	300	0.100m/s
融合	312	0.096m/s

通过上述表格计算的时间，机器人在通过激光雷达与视觉进行融合时所处理的数据信息更多，比单一传感器中速度最快的激光雷达方案慢了 0.007m/s，但是考虑到融合后栅格地图的效果更好且仍能满足移动底盘的运动要求。

## 5.5 本章小结

本章详细介绍了多传感器信息融合的基本内容和融合层次，分析了多种不同融合方法的原理以及性能。然后仔细推导了贝叶斯估计融合算法在栅格地图构建中的应用步骤，提出了激光雷达与深度相机融合 SLAM 的具体方案，在 KITTI 数据集中进行了算法仿真实验，验证了融合算法的有效性。接着搭建了实际环境下的实验平台，分别从软硬件环境的角度给出总系统架构以及系统软件结构图。最后在实际室内环境下进行单一传感器建图和融合 SLAM 建图之间的对比实验，明显发现融合后的栅格地图更加准确清晰，验证了本方案的有效性。



## 第六章 总结与展望

### 6.1 全文总结

随着移动机器人应用场景的不断扩大以及环境更加复杂多样，只采用单一传感器进行同时定位与地图构建存在很大局限性，因缺失部分环境信息导致建图不准确。本课题采用贝叶斯估计融合激光雷达和 Kinect 视觉信息来构建更加完整精准的地图。本文完成的研究内容具体如下：

(1) 介绍传感器融合 SLAM 领域的研究背景和意义以及国内外的研究现状。推导 Kobuki 底盘的运动学模型，完成里程计标定。介绍激光雷达的光束模型和似然场模型，完成走廊环境下运动畸变矫正。推导 Kinect 深度相机的数学模型，完成相机的内参标定以及激光雷达和相机的联合外参标定。

(2) 推导 SLAM 问题的数学模型，对常用的 SLAM 方案进行分类，介绍了几种不同的地图表示方法。以贝叶斯滤波为基础，详细介绍了粒子滤波、RBPF-SLAM 以及 Gmapping 等基于滤波器的激光 SLAM 算法。针对传统算法存在的问题，提出可变粒子数的滤波算法，通过 MATLAB 仿真验证了改进后的算法提升了机器人定位的精准度和效率。

(3) 介绍了经典视觉 SLAM 框架中的视觉里程计、后端优化、回环检测以及地图构建等部分。对开源的 ORB-SLAM2 算法中特征追踪、局部优化以及回环检测三个线程进行分析，在 ROS 系统中运行该算法得到所处环境的点云以及相机运动轨迹图。以 ORB-SLAM2 算法为基础对逆传感器进行建模，结合栅格地图模型在学校工程中心会议室场景下进行建图。

(4) 提出了激光雷达与深度相机融合 SLAM 的具体方案，在 KITTI 数据集中进行了算法仿真实验，验证了融合算法的有效性。接着搭建了实际环境下的实验平台，在实际室内环境下进行单一传感器建图和融合 SLAM 建图之间的对比实验，明显发现融合后的栅格地图更加准确清晰，验证了本方案的有效性。

### 6.2 研究展望

本文提出的基于激光雷达与视觉融合的 SLAM 系统相较于单一传感器的 SLAM 方案，能够构建出更加清晰准确的栅格地图。但在研究过程中发现仍然有些地方需要进一步思考：

(1) 本文是在干净平整的室内地面进行 SLAM 实验，当移动机器人遇到复杂不平的路

面时，很有可能出现机器人的轮子打滑或剧烈颠簸现象，此时仅依靠移动底盘的编码器得到的里程信息将会出现严重偏差。所以我们进一步考虑需要融合 IMU 惯性测量单元的信息来改善出现的误差，从而构建出更加精准全面的环境地图。

(2) 本文系统中仅通过一个移动机器人进行定位与地图构建，传感器融合的算法比单一传感器建图速度稍慢，且搭建的室内实验环境的面积和复杂度都不是很大。随着实验场景的加大以及复杂度的提升，我们可以考虑借助多机器人之间协同来完成 SLAM，将会进一步提升地图的准确度和构建地图的时间。

## 参考文献

- [1] 王昊宇. 现代机器人的应用分析[J]. 集成电路应用, 2020, v.37;No.317(02):68-69.
- [2] 谷明信, 赵华君, 董天平. 服务机器人技术及应用[M]. 西南交通大学出版社, 2019:176.
- [3] 雷旭昌, 王定勇, 王旭. 工业机器人编程与操作[M]. 重庆大学出版社, 2018:160.
- [4] 托斯. 人工智能时代[M]. 人民邮电出版社, 2017:307.
- [5] Li H J. Intelligent Robots[M]. Nova Science Publishers, Inc.:2019-03-26.
- [6] Spyros G, Tzafestas. Roboethics[M]. Springer, Cham:2016-01-01.
- [7] 李姿杞, 姚天冲. 基于人工智能的机器人伦理危机[J]. 集成电路应用, 2021, 38(01):52-53.
- [8] 贾佳. 基于多传感器的智能自主灭火机器人的设计[J]. 自动化与仪表, 2020, 35(12):35-39.
- [9] Zhao L, Liu Z, Chen J, et al. A Compatible Framework for RGB-D SLAM in Dynamic Scenes[J]. IEEE Access, 2019, PP(99):1-1.
- [10] Zhang J, Singh S. Laser-visual-inertial Odometry and Mapping with High Robustness and Low Drift[J]. Journal of Field Robotics, 2018, 35(8):1242-1264.
- [11] 蒋江红, 张锬石. 基于 SLAM 技术的双控制器智能机器人[J]. 计算机技术与发展, 2020, 30(06):40-43.
- [12] Delaune J, Bayard D, Brockers R. Range-Visual-Inertial Odometry: Scale Observability Without Excitation[J]. IEEE Robotics and Automation Letters, 2021, PP(99):1-1.
- [13] 欧阳仕晗, 刘振宇, 赵怡巍, 秦圣然, 刘潇. 移动机器人三维激光 SLAM 算法研究[J]. 微处理机, 2020, 41(05):58-64.
- [14] Zhu Z, Yang S, Dai H. Enhanced Visual Loop Closing for Laser-Based SLAM[J]. IEEE International Conference on ASAP, 2018:1-4.
- [15] 刘力玮, 马振兴, 张志辉, 谢淋东. 一种基于 SLAM 算法的移动机器人[J]. 科学技术创新, 2020(27):89-90.
- [16] Zhang B, Zhang X, Chen X. Grid Map Guided Indoor 3D Reconstruction for Mobile Robots with RGB-D Sensors[C]. IEEE/ASME International Conference on AIM, 2018:498-503.
- [17] Hsiao M, Westman E, Zhang G. Keyframe-based dense planar SLAM[C]. IEEE International Conference on Robotics and Automation (ICRA), 2017:5110-5117.
- [18] Fei W, Xiao G R, Peng D, et al. A Micro SLAM System Based on ORB for RGB-D Cameras[J]. MATEC Web of Conferences, 2018, 160:07001.
- [19] Pumarola A, Vakhitov A, Agudo A, et al. PL-SLAM: Real-time monocular visual SLAM with points and lines[C]. IEEE International Conference on Robotics & Automation. IEEE, 2017.
- [20] Jaimes M, Kerl C, Gonzalez-Jimenez J, et al. Fast Odometry and Scene Flow from RGB-D Cameras based on Geometric Clustering[C]. IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017.
- [21] Huu M L, Lin C, Thanh N D, Soo J P. Design and modelling of a variable stiffness manipulator for surgical robots[J]. Mechatronics, 2018, 53.
- [22] 周亚丽, 张奇志, 徐鑫鑫, 张万杰, 律晔. 家庭服务机器人研究与人才培养[J]. 机器人技术与应用, 2019(01):43-48.
- [23] 张颖川. 我国物流机器人发展概况及未来机遇[J]. 物流技术与应用, 2019, 24(09):88-91.
- [24] José A. Castellanos, Ruben Martinez-Cantin, Juan D. Robocentric map joining: Improving the consistency of EKF-SLAM[J]. Robotics and Autonomous Systems, 2007, 55(1).
- [25] Mur-Artal R, Tardos J D. ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras[J]. IEEE Transactions on Robotics, 2017, 33(5):1255-1262.
- [26] 郭剑辉, 赵春霞, 陆建峰. 基于 Rao-Blackwellised 粒子滤波 SLAM 的一致性研究[J]. 系统仿真学

- 报,2008,20(23):6401-6405.
- [27] Aditya B, Koolwal, Federico B, et al. A Fast Slam Approach to Freehand 3-D Ultrasound Reconstruction for Catheter Ablation Guidance in the Left Atrium[J]. *Ultrasound in Medicine & Biology*,2011,37(12).
- [28] Zhao C, Pan W, Hu H. Interactive indoor environment mapping through visual tracking of human skeleton[J]. *International Journal of Modelling Identification & Control*, 2013, 20(4):319-328.
- [29] Han D, Li Y, Song T, et al. Multi-Objective Optimization of Loop Closure Detection Parameters for Indoor 2D Simultaneous Localization and Mapping[J]. *Sensors (Basel, Switzerland)*, 2020, 20(7).
- [30] 孙琪, 王世峰, 王开鑫, 陈森, 刘传义. 基于单线激光雷达的室内环境建图方法研究[J]. *长春理工大学学报*,2020,43(04):37-42.
- [31] Alsayed Z, Bresson G, Verroust-Blondet A, et al. 2D SLAM Correction Prediction in Large Scale Urban Environments[C]. *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018.
- [32] 梁雪慧, 张瑞杰, 赵菲, 程云泽. 基于自适应粒子群优化 FastSLAM 算法的改进[J]. *天津理工大学学报*,2020,36(05):11-15+19.
- [33] 钟秋波, 项彦澍, 朴松昊. 基于改进的 ORB-SLAM 智能移动机器人设计与实现[J]. *黑龙江大学自然科学学报*,2020,37(02):230-234.
- [34] Shin Y S, Park Y S, Kim A. Direct Visual SLAM Using Sparse Depth for Camera-LiDAR System[C]. *IEEE International Conference on Robotics and Automation (ICRA)*, 2018:1-8.
- [35] 徐海兵. 基于 SLAM 的移动机器人建图与导航研究[D]. 哈尔滨理工大学,2020.
- [36] 张恒, 徐万红, 张禹. 激光雷达与深度相机融合的 SLAM 技术研究[J]. *机械工程师*,2020(07):22-24+29.
- [37] 李杨. 基于 ROS 的室内全向自主导航机器人研究[D]. 中国矿业大学, 2020.
- [38] 罗荣华, 洪炳镭. 移动机器人同时定位与地图创建研究进展[J]. *机器人*,2004(02):182-186.
- [39] Diego R L, Fernando M. Local map fusion for real-time indoor simultaneous localization and mapping[J]. *Journal of Field Robotics*,2006,23(5).
- [40] Toshiki S, Isao K, Yosuke. Multi-robot SLAM via Information Fusion Extended Kalman Filters[J]. *IFAC PapersOnLine*,2016,49(22).
- [41] Wu M, Li L L, Wei Z H, et al. Moving Target Tracking with Robot Based on Laser-Camera Data Fusion[J]. *Advanced Materials Research*, 2014, 1023:141-144.
- [42] 刘雅彬, 陈万米. 融合激光与视觉信息的自主移动机器人的 SLAM 研究[J]. *工业控制计算机*,2015,28(09):70-72+74.
- [43] Wang X, Marcotte R, Ferrer G. ApriISAM: Real-Time Smoothing and Mapping[C]. *IEEE International Conference on ICRA*, 2018:2486-2493.
- [44] 高翔, 张涛. 视觉 SLAM 十四讲[D]. 电子工业出版社, 2017.
- [45] 王海. 基于视觉 SLAM 建图的无人机路径规划[D]. 北京化工大学, 2020.
- [46] Vallvé J, Solà J, Andrade-Cetto J. Graph SLAM Sparsification With Populated Topologies Using Factor Descent Optimization[J].*IEEE Robotics and Automation Letters*, 2018:1322-1329.
- [47] Azartash H, Lee K R, Nguyen T Q. Visual odometry for RGB-D cameras for dynamic scenes[M]. 2014.
- [48] Song Y, He B. Feature-Based Real-Time Visual SLAM Using Kinect[J]. *Advanced Materials Research*, 2014, 989-994:2651-2654.
- [49] Mitchell H B. Multi-Sensor Data Fusion[M]. Springer, Berlin, Heidelberg:2007-01-01.
- [50] Hartmann J, Forouher D, Litza M, et al. Real-Time Visual SLAM Using FastSLAM and the Microsoft Kinect Camera[C]// *Robotics Robotik, German Conference on. VDE*, 2012.
- [51] Essmaeel K, Gallo L, Damiani E, et al. Comparative evaluation of methods for filtering Kinect depth data[J]. *Multimedia Tools and Applications*, 2015, 74(17):7331-7354.

## 附录 攻读硕士学位期间申请的专利

[1] 李晓飞, 陈鹏. 一种基于改进的 Gmapping-SLAM 地图更新方法及系统, 202110086610.8, 2021.1.22, 待授权;

## 致谢

伴随着牛年刚刚开始，我的两年半研究生旅程就要接近尾声。回想在三牌楼的点点滴滴，遇到过课程的疑惑、生活的不顺以及实验的不理想等问题。但是最终以较好的成绩完成所有课程，在实验和工程实践方面也取得一些成果，所有经历的困难都成为宝贵的财富。想必日后回忆起这段求学时光，一定会充满怀念不舍和开心感动。

刚进入课题组就开始接触移动机器人方向，非常感谢我的导师李晓飞教授在这个方向对我的帮助。李老师从刚入学对于课题方向的选择、研究的角度、学习的方法以及实验的具体细节都给予我很多的指导，让我可以更快适应研究生的节奏。在论文的开题、中期检查以及撰写初稿期间，没有李老师提供的有针对性和宝贵的意见，论文也不会完成的如此顺利。在此真心的感谢李老师在学习和生活上对我的关心和帮助。

感谢工程中心课题组的师兄师姐、师弟师妹以及同学们，在学校的大部分时间都是我们在一起讨论学习上的问题，研究实验系统的搭建，大家有问题都是互相帮助。感谢同寝室的舍友们，能够包容我的一些缺点，在生活上答疑解惑。认识这些同学是我研究生期间最大的收获。

感谢一直支持我学习的父母，能够尽最大可能满足我的物质需求，还可以在精神方面给我鼓励，让我没有任何担忧的努力学习。在今后的生活中唯有奋斗才能报答父母的恩情。

再次感谢所有关心帮助过我的老师、同学以及朋友，未来希望我们大家一切都好。