分 类 号: P225.2　　密　　级:

学　　号: 201510532

西安科技大学
XI' AN UNIVERSITY OF SCIENCE AND TECHNOLOGY

硕士学位论文

Thesis for Master's Degree

室内测绘机器人SLAM技术的研究与实现

申请人姓名：李维鑫

指 导 教 师：陈晓宁（校内） 王小平（校外）

类　　别：全日制专业学位硕士

工 程 领 域：测绘工程

研 究 方 向：室内测绘机器人

2018 年 6 月

西安科技大学

学位论文独创性说明

本人郑重声明：所呈交的学位论文是我个人在导师指导下进行的研究工作及其取得研究成果。尽我所知，除了文中加以标注和致谢的地方外，论文中不包含其他人或集体已经公开发表或撰写过的研究成果，也不包含为获得西安科技大学或其他教育机构的学位或证书所使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中做了明确的说明并表示了谢意。

学位论文作者签名：李维鑫　　日期：2018.6.8

学位论文知识产权声明书

本人完全了解学校有关保护知识产权的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属于西安科技大学。学校有权保留并向国家有关部门或机构送交论文的复印件和电子版。本人允许论文被查阅和借阅。学校可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。同时本人保证，毕业后结合学位论文研究课题再撰写的文章一律注明作者单位为西安科技大学。

保密论文待解密后适用本声明。

学位论文作者签名：李维鑫　　指导教师签名：王…平 陈晓宁

2018年6月8日

论文题目：室内测绘机器人 SLAM 技术的研究与实现

工程领域：测绘工程

硕 士 生：李维鑫　　（签名）李维鑫

指导教师：王小平　　（签名）王小平

陈晓宁　　（签名）陈晓宁

摘 要

近年来，随着人工智能在各个学科领域的广泛应用，机器人学实现了从理论到实际应用的一个飞跃。同步定位与建图（SLAM）算法作为机器人领域的重要研究方向，是机器人在未知环境下自主导航和完成复杂智能任务的基础，也是机器人的感知能力和智能水平的重要依据。测绘机器人是测绘行业新的发展方向，对实现测绘自动化、信息化、智能化具有十分重大的意义。由于信号遮挡等因素的影响，传统的 GPS 技术并不能进行室内定位测量，针对这一问题，本文自主设计测绘机器人系统，应用 SLAM 技术实现测绘机器人自主绘制二维室内地图和定位导航功能。主要研究内容如下：

（1）通过在测绘机器人平台上搭载激光雷达、轮式里程计、IMU 等传感器获取室内环境信息，分别利用 Hector SLAM、Gmapping、Cartographer 算法进行测绘机器人实时绘制室内地图实验。结果表明：三种算法均可实时有效地得到室内二维地图。相比较传统测量时间大大节省人力资源和时间成本。

（2）通过对三种 SLAM 算法的对比分析，结果表明：Cartographer 算法的测图精度明显优于 Hector SLAM 和 Gmapping 算法，测图精度到达 5cm，能够满足基本的室内服务需求；同时因其具有闭环检测能力，使得 Cartographer 算法具有很好的鲁棒性。

（3）本文创新性地采用深度学习方法来解决机器人定位问题，通过对机器人定位过程进行训练学习，采用 FFNN 及 CNN 神经网络模型得到机器人位姿信息。结果表明：神经网络进行机器人定位有效可行，同时表明了深度学习在机器人定位领域存在巨大潜力。

关 键 词：测绘机器人；同步定位与建图；Hector SLAM；Gmapping；Cartographer；机器人定位；深度学习

研究类型：应用研究

Subject : Research and implementation of SLAM technology for Indoor surveying and mapping robot.

Specialty : Surveying and Mapping Engineering

Name : Weixin Li **(Signature)** Weixin Li

Instructor : Xiaoping Wang **(Signature)** Wang Xiaoping

Xiaoning Chen **(Signature)** Chen Xiaoning

ABSTRACT

In recent years, with the wide application of artificial intelligence in various disciplines, robotics has realized a leap from theory to practical application. The SLAM algorithm is an important research direction in the field of robotics. It is the basis for the robot to autonomously navigate and complete complex intelligence tasks in an unknown environment, and is also an important basis for the robot's perception ability and intelligence level. Surveying and mapping robots are the new development direction of the surveying and mapping industry and have great significance for the realization of surveying and mapping's automation, informationization and intelligence. Due to the influence of signal occlusion and other factors, traditional GPS technology can't perform indoor localization and measuring. In view of this problem, this paper independently designs the mapping robot system and applies SLAM technology to realizes automatic drawing of two-dimensional indoor maps and navigation and positioning functions. The main research content is as follows:

(1) Acquire indoor environmental information by using sensors such as laser lidar, wheel odometer, and IMU on the mapping robot platform, and use the Hector SLAM, Gmapping, and Cartographer algorithms to perform real-time indoor mapping experiments for the mapping robot. The results show that the three algorithms can obtain indoor two-dimensional maps in real time. Compared with the traditional measurement, it can save human resources and time costs.

(2) Through the comparison and analysis of the three SLAM algorithms, the results show that the Cartographer algorithm's mapping accuracy is obviously better than the Hector SLAM and Gmapping algorithm, and the mapping accuracy reaches 5cm, which can meet the basic indoor service requirements; meanwhile, because of its closed loop capability makes the

Cartographer algorithm very robust.

(3) This paper innovatively uses the deep learning method to solve the robot localization problem. Through the training and learning of robot localization process, and used FFNN and CNN neural network models to complete the position of robot. The results show that the neural network is effective and feasible for robot positioning. It also shows that deep learning has great potential in robot localization.

Key words: Surveying and mapping robot; Simultaneous localization and mapping (SLAM) ; Hector SLAM ; Gmapping; Cartographer; Robot localization; Deep learning

Thesis: Application Research

目录

1 绪论

1.1 研究背景与意义

1.1.1 室内测量机器人的研究背景与意义

测绘作为国民经济和社会发展的基础，在各个领域都发挥着重要的作用。测量技术的与时俱进是十分必要的[1]。近年来，人工智能技术迅速发展，已经渗透至各行各业当中。学者们越来越重视人工智能技术在测绘行业中的应用。在 2017 年南京举办的中国测绘地理信息学会上，李建成院士在《人工智能对于测绘学科发展带来的机遇与思考》报告中提出未来五到十年测绘的发展方向将会是测绘机器人、移动测量机器人、飞行测量机器人等等，实现人机一体化测绘体系。可见开展新一代信息技术下的测绘、遥感、导航技术体系研究是十分必要的。

随着科学技术及互联网的高速发展，人们的生活变得越来越便捷，足不出户就能解决很多需求。据统计，目前人类的活动 80%是在室内进行的，人们对于室内环境信息的需求越来越迫切，对于室内环境的认知依赖于室内位置信息，但由于信号衰弱等因素，传统的 GPS 定位技术仅能提供室外的高精度位置信息，因此如何进行室内定位与环境感知成为了越来越多学者的重点研究方向。本文设计了一款用于测量室内二维地图的测绘机器人，结合 SLAM 技术，以轮式机器人为平台，搭载激光雷达、轮式编码器、IMU 等传感器，通过 4G 通讯技术进行数据传输，实现测绘机器人实时室内地图的绘制与定位任务。相比传统测绘技术，测绘机器人更高效更便捷，且其精度满足基本室内服务。同步定位与制图（SLAM）技术，指的是机器人在一个未知的环境中，对于自身位置未知的情况下，通过机器人运动对环境进行观测，增量式的创建环境地图，并同时确定自身在地图中的位置的技术[2]。对于机器人的运动轨迹以及路标位置的估计都是在线实时进行的，不需要任何的位置的先验知识。我们借助 SLAM 技术，应用于室内测量领域，可以有效的解决室内测图与定位问题，弥补因为 GPS 信号衰弱无法进行室内定位问题；同时随着测绘机器人的研发与应用，测绘机器人可代替人们进行一些繁琐的重复性工作，一定程度上解放了劳动力，促进了测绘行业更加自动化智能化。

1.1.2 SLAM 技术的研究背景与意义

目前，移动机器人的智能化水平是衡量一个国家科技发展水平和综合国力的重要标志。对于移动机器人的研究涉及机械运动、人工智能、模式识别、传感器技术、计算机通讯等多个学科领域，其中机器人利用传感器进行自主定位与环境感知是机器人完成其

他任务的一个重要前提[3]。SLAM 技术正是解决这一问题的重要技术手段，同时也是实现机器人高智能化的核心技术。近年来，SLAM 算法已经被应用于机器人、无人机、无人驾驶、AR 等多个领域，学者针对这一问题展开大量研究与实验。就室内机器人这一行业的 SLAM 应用而言，最主流的 SLAM 算法主要包括基于视觉 SLAM 算法和基于激光 SLAM 算法两大类[4]：激光 SLAM 最早是由激光测距的定位方法发展而来，激光 SLAM 技术是通过对机器人在不同时刻采集的环境中一系列分散的、具有准确角度和距离信息的点云进行匹配，计算激光雷达相对运动的距离和姿态变化，从而实现机器人对于环境的感知与自身定位；早期的视觉 SLAM 是基于滤波理论，通过搭载在机器人上的视觉传感器，如单目摄像头、双目摄像头、深度相机等，从环境中获取大量的、丰富的视觉信息、纹理信息来实现同步定位与建图工作，其对于环境具有超强的辨识能力。由于激光雷达的测距准确、模型计算简单、对于不同光照强度运行稳定等优点，使得激光 SLAM 理论的研究较视觉 SLAM 成熟，这也是目前国内外一些团队的主要研究方向。

1.2 国内外研究现状

1.2.1 测绘机器人的国内外研究现状

机器人作为代替人类生产的一种工具，最早是于上世纪六十年代开始并以惊人的速度迅速发展。各国对于机器人行业十分重视，大量投入资金给予支持。最早的机器人较多被应用于工业领域，但近年来，机器人逐渐向着智能化方向发展。相比较传统的工业机器人，智能化移动机器人具有更强的自主性和适应能力，可以更好地完成任务。第一台自主移动机器人 Shakey 诞生于上世纪 70 年代的斯坦福大学，它可以完成自主定位、路径规划和导航等任务，这也是世界首个能够自主运动的机器人[5]。此后，机器人应用范围越来越广泛，比如英国的“龙行者”，它可以代替人们进行拆弹这样的危险工作；Willow Garage 公司的 PR2 机器人，国内的科沃斯“地保”吸尘器机器人都是在家政服务中表现出色的机器人。

针对测绘行业，用于测量领域的机器人也取得十分可观的成绩。过去几年 Google 一直致力于室内测绘技术，先后推出 Trekker 和 Cartographer 两款背包测量机器人。背包人员在建筑物中穿行时，搭载的传感器会实时自动的生成建筑平面图，同时使用者还可以在地图上标注兴趣点。2016 年，中海达将激光扫描仪、全景相机、里程编码器进行集成，突破基于激光的 SLAM 技术，推出了他们的 HiScan-SLAM 室内移动测量系统，在商场、地下室等卫星信号较弱的地方，也可以完成室内定位、室内空间获取与成图等测量任务[6]。武汉大学测绘遥感信息工程国家重点实验室罗斌教授研制的一款集制图与导航于一体的移动机器人，可轻松实现机器人对于室内环境地图的绘制，同时具有自主导航和避障等功能，目前已成功应用于电厂、图书馆等场景。

(a)HiScan-SLAM 测量机器人

(b) cartographer 背包机器人

(c) PR2 机器人

(d)可佳机器人

(e)防爆机器人

(f)地宝机器人

图 1.1 国内外发展成熟的机器人系统

1.2.2 SLAM 技术的国内外研究现状

SLAM 问题又被称为 CML（Concurrent Mapping and Localization）问题，被誉为自主移动机器人研究领域的“圣杯”[7]。 SLAM 算法发展至今，已形成多种 SLAM 体系，且均取得了相当丰富的成果。最早的 SLAM 概念是由 R.Smith 和 P.Cheeseman 提出的[8]，他们建立了用于描述特征与不确定性几何物体之间关系的统计学基础，认为环境中对于不同路标的位置估计是具有高度相关性的，并且这些相关性会随着连续观测而不断增加，运用卡尔曼滤波对机器人位姿和环境中特征进行估计；Dieter Fox[9]和 Sebastian Thrun[10]提出利用概率学解决机器人定位问题，成为基于概率 SLAM 算法的模型基础；卡内基梅隆大学的 Michael Montemerlo[11]改进了卡尔曼滤波的数据关联和计算复杂度问题，提出了 FastSLAM 算法，该算法基于粒子滤波不仅大大减少了计算量而且在大型的、地标模糊的环境中，依然可以获取精准的地图。杜克大学的 Austin Eliazar[12]提出一种新的地图创建方法，称为分布式粒子（DP）地图，即直接利用传感器信息创建占有地图，这样就避免了关于地标的定义和相应的数据关联问题，同时提出了向三维空间扩展的方法；Daniel T.Savaria[13]将立体摄影视觉系统、里程计和声纳等多个传感器进行融合进行 SLAM 实验，并采用 SURF 算法描述地图特征。Yan Ma, Hehua Ju[14]通过建立似然域 LF 获得似然函数对 RBPF-SLAM 算法进行改进，并利用激光、惯导结合里程计等传感器将

其用于月球探测车。

在国内，南京理工大学的郭建辉[15]博士分别用多种形式的 SLAM 算法对悉尼大学的“Victoria Park”数据集进行实验，研究其中的相关性，改进了数据关联等技术。此外，祝继华等[16]采用最邻近迭代法 ICP 计算机器人的运动方程，改进了 RBPF-SLAM 里程计的建议分布函数，同时改进粒子重抽样策略，大大减少了粒子个数。王彭林[17]等人利用 SIFT 特征匹配算法，将提取的摄像机的旋转角度与里程计计算的角度信息结合进行 SALM 实验，一定程度上提高了机器人的定位精度。袁夏[18]提出一种点-面匹配的三维 SLAM 算法，将 6 个自由度的三维问题简化到 5 个自由度，实验结果证明该方法可以有效解决三维 SLAM 问题。蔡则苏等[19]人利用激光数据和里程计数据，得到基于 EKF 算法的机器人导航定位方案。柯江胜[20]等人通过调整先验估计误差和噪声协方差得到一种鲁棒的 SLAM 算法。在激光 SLAM 和视觉 SLAM 结合方面，梁潇[21]采用单目辅助激光的形式进行 SLAM 实验，通过建立 ORB 特征词袋，用视觉信号激发激光 SLAM，进行闭环检测，实现机器人位姿图优化。 庄严[22]等人结合单目摄像头和激光测距仪提出针对结构化室内环境的 SLAM 方法。张毅[23]等人使用深度相机和激光传感器，基于贝叶斯方法提取地图中一致性特征，进行特征信息融合，有效提高算法精度。

1.3 本文的关键技术

本文自主设计的测绘机器人平台，可实现自主运动，快速通过激光等传感器数据完成室内二维地图构建与定位任务。这是一个十分具有挑战性的课题。为了搭建测绘机器人平台，本文对所需硬件进行长时间测试与筛选，最终实现测绘机器人能够稳定运行。同时为了使机器人能够自主绘制二维室内地图，本文深入研究了目前最主流的 SLAM 算法，并将其应用到测绘机器人平台，经过长时间的测试，最终实现测绘机器人快速自主的绘制室内二维地图和定位导航。

本文的技术难点在于同步定位与建图涉及多个学科领域，包括室内地图绘制、机器人在地图中的定位、传感器数据获取等多个方面，同时还涉及概率学、信号传输处理、传感器信息融合、多元数据的解析、点云扫描匹配以及多种滤波器等[3]。本次论文跨多个学科领域，因此需要多个专业的知识支持。

本课题的研究任务主要分三个阶段完成。第一阶段主要进行测绘机器人平台的设计与构建，包括机器人运动底盘的设计，采用双轮差速驱动；激光测距仪、编码器等传感器设备的安装和调试；传感器数据的采集；第二阶段主要进行 SLAM 在测绘机器人平台上的测试工作，其中包括对于激光扫描仪扫描角度、测距范围、点云数据处理、机器人运动速度、地图描述方法以及每个 SLAM 算法中具体的阈值参数的测试与调整；第三阶段结合传感器数据，利用不同的 SLAM 算法进行室内二维地图的绘制，对实验结果进行对比分析，从而为实际工作提供一定的指导意义。在成功绘制地图基础上，实现测绘机

器人导航定位功能。

1.4 本文的主要研究内容

室内测量作为测绘行业的新兴方向，对于人们的工作和生活具有重要意义[24]。本文通过自主设计搭建测绘机器人平台，介绍目前最主流三种 SLAM 算法以及定位算法，并在测绘机器人平台上进行相关实验。共分为七章，各章节的主要内容如下：

第一章绪论，主要介绍了本文的研究意义与背景，包括 SLAM 算法的研究意义与背景，同时介绍了 SLAM 算法在测绘行业中应用的背景与研究意义；然后介绍了测绘机器人和 SLAM 算法国内外的研究现状；最后对于本文的研究技术路线和主要研究内容进行阐述。

第二章主要针对 SLAM 算法的基本理论，介绍了包括 SLAM 地图的表示，SLAM 算法的主要理论框架；然后介绍了测绘机器人实验平台的设计与搭建过程，主要从硬件和软件两个方面给出了阐述。为后续进行测绘机器人 SLAM 实验奠定基础。

第三章对 Hector SLAM 算法的基础理论进行详细说明；构建测绘机器人仿真模型以及仿真环境；在仿真环境下对 Hector SLAM 算法进行实验，完成其绘制二维地图任务；接着，进行真实环境场景实验，实现 Hector SLAM 在测绘机器人平台上的测图任务。

第四章具体介绍了 Gmapping 算法的基础理论；在机器人模型上添加里程计模型，同样在仿真环境和真实环境下对 Gmapping 算法进行实验，目视对比其与 Hector SLAM 测图结果。

第五章在首先对 Cartographer 算法原理进行详细阐述；前两章的基础上，在机器人上加入 IMU 传感器，利用 Cartographer 算法再次进行相同环境下的仿真实验和真实物理环境实验，并结合上述两章实验结果进行具体的对比分析与评价。

第六章介绍了测绘机器人自主定位导航的方法，介绍了传统基于概率模型的自适应蒙特卡洛定位方法和基于深度学习神经网络的定位方法，并进行实验验证其定位精度。

第七章对本文进行总结，说明了本文对于实际工作的指导意义并提出不足之处；对接下来进一步的工作给出了展望。

2 SLAM 基本原理

2.1 地图的表达方式

地图是我们用来表示环境的主要方式。绘制地图是测绘行业的一项基础业务。目前针对机器人自主绘制地图主要包括栅格地图、几何特征地图和拓扑地图。我们通常是根据机器人所处环境的实际情况、同时定位与建图（SLAM）算法的适用条件以及机器人的需求来选择地图的表示方式。选择合适的地图表示方式不仅可以满足建图的需求，同时更加有利于我们在地图的基础上开展其他工作，例如导航避障等。

2.1.1 栅格地图

栅格地图是最基础的地图表示方式，其主要思想是：将环境分为若干个相同大小的格网，对于二维栅格地图，每个格网都有两种状态：被障碍物占用和未被障碍物占用。我们根据 SLAM 算法赋予每个格网（0，1）之间的值，该值表示这个栅格被占用的概率，对应实际环境中该格网处存在障碍物的概率。对于三维地图，则是在二维地图的基础上，给每个格网添加了高度来表示实际对应物体的高度。

栅格地图最大的优点在于易于创建和维护，我们只需要创建简单的二维或三维数组即可实现，对于环境中的动态信息，比如行人等信息，也可以很容易实现地图的更新工作。尤其适用于使用激光、超声波数据创建的地图。但是，地图的分辨率与栅格数量呈正比关系，提高地图分辨率必然会增加栅格的数量，即增大计算量[7]。而且大型环境中，随着环境的逐渐扩增，栅格信息的存储会占据很大的空间，因此栅格地图仅适用于较小的环境。

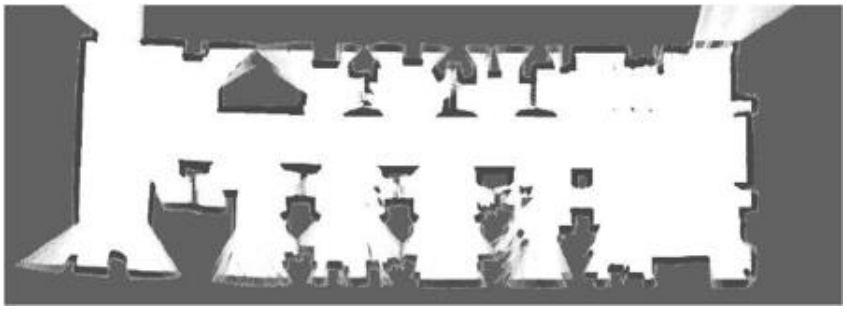


图 2.1 栅格地图

2.1.2 几何特征地图

几何特征地图是利用环境中物体的几何特征来表示环境地图，例如点、直线、曲线、曲面等特征。主要思想是：根据传感器的原始数据，从中提取环境物体的几何特征用来

近似表示该物体，从而用这些几何特征来构建环境地图[25]。

几何特征地图的优点在于易于识别，尤其是环境中物体的位置信息，一目了然。同时对于内存的占用也不是很大。但是这种表示方法存在一个很大的问题：数据关联问题，在进行地图创建时，需要将局部地图与全局地图相对应的环境特征进行关联匹配，它要求所提取的环境特征要具备一定的精准度，这样才可以保证地图的一致性。另外数据关联问题也会影响环境特征和全局地图的匹配，从而对机器人位姿估计造成影响。同时物体特征的提取是另一个难点，对于室内结构化的环境来说，物体特征较简单，大多是点、线等，较易于提取，但是对于室外非结构化的环境，就很难用简单特征去描述，因此几何特征地图更适合室内环境。

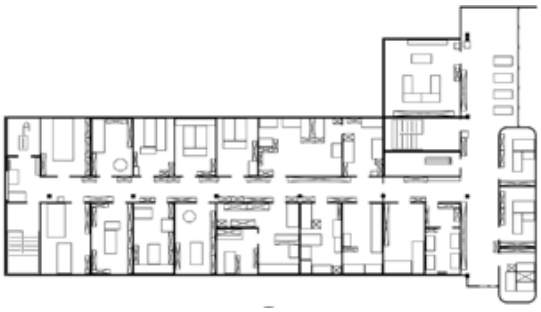


图 2.2 特征地图

2.1.3 拓扑地图

拓扑地图是基于数据结构中拓扑图的思想，使用节点与连接节点的边来表示地图，这与栅格地图和几何特征地图使用笛卡尔坐标表示位置信息是不同的。在拓扑地图中，每个节点对应环境中的一个物体，连接节点的边表示环境物体之间的邻近关系，可以通过边连接的两个节点说明这两个节点之间存在直接路径[26]。

拓扑地图的优点在于，我们可以在图上直接判断两节点是否连通，是否存在可以到达的路径，这对于在地图上进行导航和避障工作是十分有利的。同时拓扑地图不会因为地图规模增大而变得很复杂，在较大的环境下依然具有很高的计算效率。同样，这种地图表示方法也存在相应的问题，创建拓扑地图时需要利用环境特征寻找关键点与关键线，这需要每个特征都必须具有可以区别其他特征的唯一标识，但当复杂环境出现很相似的特征时，对于物体的识别就会出现偏差，机器人会不知道自己是否是处于同一地方，因此会对后续的工作带来影响。

当然，由于以上上三种地图表示方法均存在各自的优缺点，因此在进行 SLAM 工作时，也会结合实际环境条件，对上述三种地图表示方法取长补短，混合使用。例如一些学者提出将拓扑地图与几何特征地图相结合的混合地图表示方式。用几何特征地图表示局部地图中物体位置信息，然后建立这些局部地图的拓扑关系，形成全局拓扑地图，这样即可以清楚表示物体的位置信息，还可以很清楚的看出物体之间的相对关系。

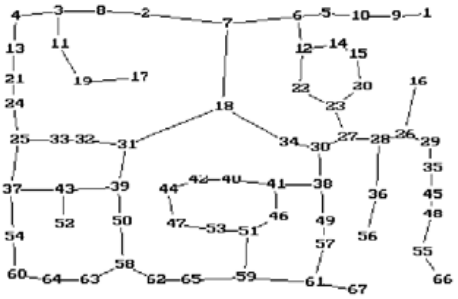


图 2.3 拓扑关系地图

2.2 SLAM 算法理论

同步定位与建图（SLAM）问题是机器人在建立环境地图的同时运用该地图进行定位的过程。关于机器人的跟踪和环境路标位置的估计都是在没有任何先验知识的情况下进行的。根据机器人的实际用途，SLAM 有不同的研究算法，目前 SLAM 方法大致分为两类：基于概率估计的 SLAM 方法和非概率估计的 SLAM 方法。其中，基于概率模型的方法应用较广泛，使用和研究最多的是基于贝叶斯模型的 SLAM 算法。常见的方法有：KF-SLAM、EKF-SLAM、PF-SLAM 算法等。下面给出了基于贝叶斯模型的 SLAM 算法的理论框架：

机器人在环境中自主运动是通过安装在机器人上面的传感器对于大量未知路标的关联观测实现的。如图 2.4 所示。

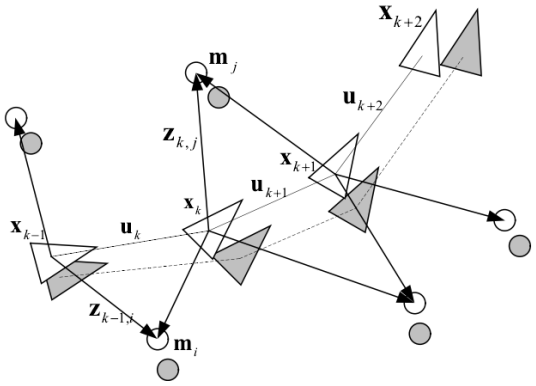


图 2.4 SLAM 问题示意图

在k时刻时，我们定义以下变量：

x_k：描述机器人位置和方向的状态向量。

u_k：控制向量，在 $k-1$时刻驱动机器人运动至 k 时刻的 x_k 状态。

m_i：描述第 i 个路标位置的向量，我们假设它的位置不随时间变化。

$z_{i,k}$：在 k 时刻机器人对第 i 个路标的观测，当同一时刻观测到多个路标时，或者出现与我们所讨论的不相关的特殊路标时，这些观测定义为 z_i。

另外，定义以下集合：

$X_{0:k}=\{x_0,x_1,...,x_k\}=\{X_{0:k-1},x_k\}$：机器人的历史姿态，即轨迹。

$U_{0:k}=\{u_0,u_1,...,u_k\}=\{U_{0:k-1},u_k\}$：机器人历史控制输入数据。

$m=\{m_1,m_2,...m_n\}$：所有的路标集合，即地图。

$Z_{0:k}=\{z_0,z_1,...,z_k\}=\{Z_{0:k-1},z_k\}$：所有路标观测值集合。

SLAM 问题可以看做是计算每一时刻的概率分布：$P(x_k,m|Z_{0:k-1},U_{0:k},x_0)$。即在给定 0 到$t$时刻的观测记录、控制输入以及机器人初始状态的情况下，计算环境路标位置和t时刻机器人状态的联合后验概率。实际上，采用递归的方法去解决 SLAM 问题是最好的。首先，由贝叶斯定理，根据k时刻的控制输入和观测值，去估计$t-1$时刻的后验概率分布$P(x_{k-1},m|Z_{0:k-1},U_{0:k-1})$，这个计算过程需要一个运动模型和一个观测模型，分别用来描述机器人的控制输入和观测[27]。观测模型描述的是当机器人位置和路标位置已知的情况下观测值为z_k的概率表示形式是：$P(z_k|x_k,m)$，假设在给定了地图和当前机器人状态条件下观测是独立的。运动模型可以描述为状态转换的概率分布，其形式是$P(x_k|x_{k-1},u_k)$。状态转换是一个蒙特卡罗过程，就是下一时刻的状态x_k只依赖于上一时刻状态$x_{k\text{-}1}$和运动控制u_k，是独立于观测值和地图的[28]。

SLAM 算法分为两个标准步骤来实现：递归的预测和更新矫正：

$$P(x_k,m|Z_{0:k-1},U_{0:k},x_0)=\int P(x_k|x_{k-1},u_k)P(x_{k-1},m|Z_{0:k-1},U_{0:k-1},x_0)dx_{k-1} \tag{2.1}$$

$$P(x_k,m|Z_{0:k},U_{0:k},x_0)=\frac{P(z_k|x_k,m)P(x_k,m|Z_{0:k-1},U_{0:k},x_0)}{P(z_k,Z_{0:k-1}U_{0:k})} \tag{2.2}$$

上式提供了关于递归地计算联合后验概率$\mathrm{P}(x_k,\mathrm{m}|Z_{0:k},U_{0:k},x_0)$，即在 0 到k时刻所有观测值和控制值已知的情况下计算k时刻的系统状态和地图。递归过程是机器人运动模型和观测模型的函数。

值得注意的是，地图构建问题可以被表示为计算条件密度分布$P(m|X_{0:k},Z_{0:k},U_{0:k})$的过程。假定机器人任意时刻的位置是已知的，地图是根据不同位置的观测值构建的。相反地，关于定位问题可以视为计算概率分布$P(x_k|Z_{0:k},U_{0:k},m)$，假设环境路标的位置都是确定的，目的是估计机器人相对于这些路标的位置信息。

为了高效地得到上述的先验概率和后验概率，便产生了多种 SLAM 解决方案。本文在接下来三个章节会依次介绍目前三种主流的 SLAM 算法，并给出相关的实验结果与对比分析。

2.3 测绘机器人系统设计与构建

本文自主设计了一款用于测量室内二维地图的测绘机器人平台。平台外观图如图 2.5 所示。该平台主要包括机器人运动底盘、传感器及软件平台三部分，下面对这三个部分分别给出具体的介绍。

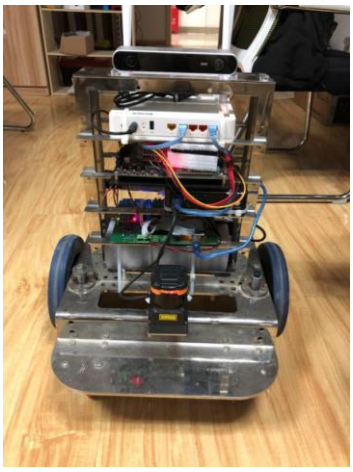


图 2.5 测绘机器人平台外观图

2.3.1 硬件元件

机器人主要由 Arduino Mega2560 单片机、工控主板、降压模块、电源电池、直流电机、电机驱动模块等硬件单元组成，同时安装二维激光扫描仪、姿态传感器、光电增量式编码器三个主要的传感器。表 2.1 主要硬件元件的详细阐述。

表 2.1 主要硬件元件

硬件元件		详细说明
UST-30LX 二维激光扫描仪		UST-30LX 2D 激光扫描测距产品拥有 30m，270°的测量范围。单周扫描时间为 25ms，测量方式为非接触式。精度可达±30mm、高分辨率、宽视场设计给机器人提供了良好的环境识别能力；紧凑型设计节约了安装空间，使其具备低重量、低功耗的特点。
IMU AHRS 姿态传感器		IMU AHRS 模块集成了 3 轴陀螺仪和 3 轴加速器、3 轴地磁传感器、气压高度计和 STM32F103T8 32-bit ARM CortexM3 处理器。采用 UART 异步串行接口，波特率可达 115200bps，支持 7 通道的 DMA 传送，集成了硬件循环冗余校验（CRC）以及单周乘法和硬件除法器，保证姿态结算的稳定性及通信效率。
光电增式编码器		490 线两相的增量式光电编码器作为里程计算的传感器，编码器通过 A、B 信号输出的脉冲个数和待测定的角度变化量成线性关系，通过 A、B 相信号的波形和时序关系，根据线性关系解算出被测机械的旋转方向和旋转角位移或者速率。

电机驱动模块		PWM 的取值范围在 0~255，用来控制电机的转速。ENA/ENB 接口用来控制电机的转动方向，ENA 置高电平，ENB 置低电平时，电机正传，机器人前进；反之机器人后退；均置低电平时，电机停止转动。
Arduino Mega2560 单片机		采用 USB 接口的核心电路板。处理器核心是 ATmega2560，同时具有 54 路数字输入/输出口，16 路模拟输入，4 路 UART 接口，一个 16MHz 晶体振荡器，一个 USB 口，一个电源插座，一个 ICSP header 和一个复位按钮，兼容 3 种供电方式。收集和打包传感器信息，接收计算机命令并控制电子元件。

2.3.2 测绘机器人底盘设计

采用低功耗双轮差分式驱动结构，为了增加平台的稳定性，在两个驱动轮的基础上安装四个万向轮。同时加入减震弹簧装置，增强平台在不平坦地区的抗冲击、抗震动性能。整个平台采用金属不锈钢材质，保证具备一定的负载能力。机器人的底盘尺度68 × 60cm，轮子直径 12cm，轮间距 54cm。图 2.6 给出机器人底盘的模型：

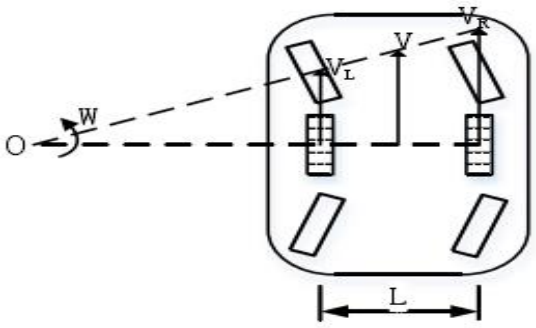


图 2.6 测绘机器人底盘模型

图中，L 表示机器人的轮间距，V_L, V_R 分别表示机器人左轮和右轮的线速度。V, W 为机器人的线速度和角速度，由此可以给出机器人的运动模型：

$$V = \frac{V_L + V_R}{2} \tag{2.3}$$

$$W = \frac{V_R - V_L}{L} \tag{2.4}$$

其中左右两轮的速度可以由增量式光电编码器来获得。在已知平台的初始位置和运动速度的条件下，可以利用微分思想对机器人运动进行解析，实时的解算机器人的里程计数据，即机器人的位姿信息 (x, y, θ) 和速度信息 (v, ω)。图 2.7 表示的是机器人的里程计模型：

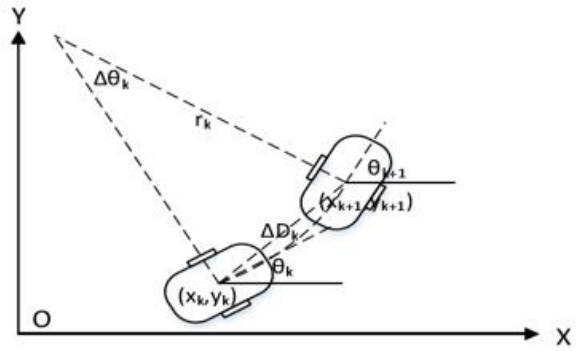


图 2.7 测绘机器人里程计模型

假设光电编码器的线数为 p，即轮子转动一圈编码器输出 p 个脉冲。在 Δt 时间内光电编码器输出 N 个脉冲，轮子半径为 r，那么机器人在 Δt 时间内的位移 Δd 可以表示为：

$$\Delta d = \frac{2\pi r \cdot N}{p} \tag{2.5}$$

假设从 k 时刻到 $k+1$ 时刻机器人左右轮的位移分别为 Δd_L 和 Δd_R，则机器人的在这段时间间隔内移动的距离 ΔD_k 和角度 $\Delta\theta_k$ 为：

$$\Delta D_k = \frac{\Delta d_R - \Delta d_L}{2} \tag{2.6}$$

$$\Delta\theta_k = \frac{\Delta d_R - \Delta d_L}{L} \tag{2.7}$$

机器人在 k 时刻的位姿 (x_k, y_k, θ_k)，k+1 时刻的位姿 $(x_{k+1}, y_{k+1}, \theta_{k+1})$，在很短的时间间隔内可以看成圆弧运动，该圆弧的半径可表示为 $r_k = \frac{\Delta D_k}{\Delta\theta_k}$，$\Delta\theta_k \neq 0$。据此我们得到机器人的位姿：

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k + r_k(\sin(\theta_k + \Delta\theta_k) - \sin\theta_k) \\ y_k + r_k(\cos(\theta_k + \Delta\theta_k) - \cos\theta_k \\ \theta_k + \Delta\theta_k \end{bmatrix}, \Delta\theta_k \neq 0 \tag{2.8}$$

特殊情况，当 $\Delta\theta_k$=0 时机器人做直线运动，这时的位姿可表示为：

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k + \Delta D_k \cos\theta_k \\ y_k + \Delta D_k \sin\theta_k \\ \theta_k \end{bmatrix}, \Delta\theta_k = 0 \tag{2.9}$$

假设机器人左右轮的速度分别为 V_L 和 V_R，根据上述的运动模型，可以得到机器人在 k 时刻的线速度和角速度分别为：

$$\omega = \frac{V_R - V_L}{L} \tag{2.10}$$

$$v_k = \begin{cases} \omega_k r_k & \Delta\theta \neq 0 \\ \dfrac{V_R + V_L}{2} & \Delta\theta = 0 \end{cases} \tag{2.11}$$

2.3.3 机器人操作系统 ROS

软件的部分采用基于 Linux 系统的机器人操作系统 ROS，实现上位机和下位机的信息通讯、数据处理、实验结果实时显示等工作。

ROS(robot operating system)是 2007 年由斯坦福大学人工智能实验室与 Willow Garage 机器人项目合作推出的一款开源机器人操作系统，是机器人研究领域应用十分广泛的机器人系统[29]。ROS 为广大机器人研究人员及爱好者提供一个开源的免费平台，作为一种分布式的操作系统，它将数据包封装到堆和栈中以便于数据的共享与分发；采用点对点的设计实现多线程和多主机同步运行；支持 Java/C++/Python/Lisp 等多种语言，可以实现各种语言自由的混合和配合使用；同时包括大量的小工具进行编译、运行、可视化等操作。

ROS 主要包括几个重要的概念：节点（node）、消息（massage）、主题（topic）、服务（service）[30]。

图 2.8　ROS 主要概念

a.节点：每个节点都是一个可执行文件，即计算机进行运算任务的进程。通常一个系统是由多个节点组成的，我们可以在 ROS 中单独运行一个节点，也可以同时运行多个节点。

b.消息：消息是负责节点之间通讯的，一个消息就是一个数据结构，这和 C 语言中的结构体 structs 类似。ROS 支持整数、浮点型、布尔型以及数组等数据类型。

c.主题：消息是通过订阅和发布主题的形式进行通讯的，节点可以在一个给定的主题上发布或订阅消息。ROS 也支持多个节点发布或订阅同一个主题上的消息。可以说发布者和订阅者之间是互相独立的，并不知道对方的存在。

d.服务：服务是 ROS 中除了基于主题的通讯模式外的另一种通讯模式。当不需要节点间的同步传输时，就可以采用服务这种通讯模式。用一个字符串和一对消息定义，一

个用于请求，一个用于回应。服务与话题的区别在于，只有一个节点可以用任意单独的名称广播一个服务。

ROS 的控制器称为 ROS Master，Master 通过对每个节点进行登记和注册，保证各个节点之间可以有条不紊的进行，不至于找不到其他节点，同时进行消息的交换和服务的调用。

2.3.4 数据获取与传输

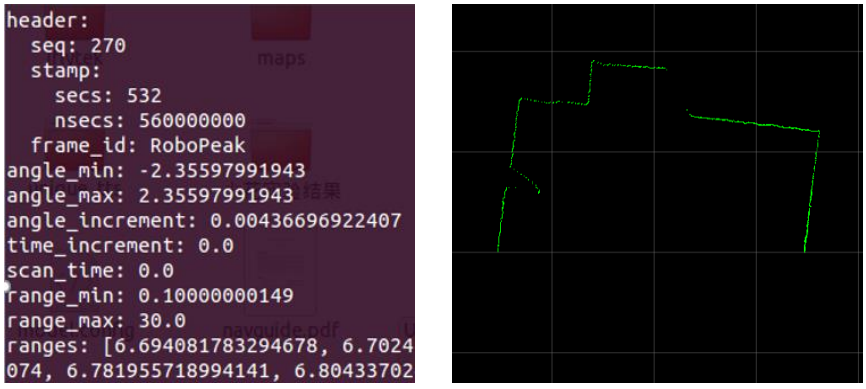
本文设计的测绘机器人主要通过搭载激光雷达、增量式光电编码器、IMU 姿态传感器来进行数据的获取。传感器通过 USB、串口等外接接口进行通讯，完成传感器数据的传输。

a.激光数据

在 2.3.1 小节我们对本文所采用的激光扫描仪进行了介绍，激光雷达的优点在于其不受光照的影响，可以近乎全方位高精度的采集环境数据。采用 USB 接口将 UST-30LX 2D 激光扫描仪与主板进行连接。在 Ubuntu 下执行如下命令，下载 hokuyo_node 包：

$ sudo apt-get install ros-indigo-hokuyo-node

ROS 通过订阅 scan 主题来进行激光数据的读取，scan 主题的消息数据类型为 sensor_msgs/LaserScan，图 2.9(a)是在 ROS 中查看 LaserScan 消息：



(a) LaserScan 消息　　(b)激光点云数据可视化

图 2.9 激光数据

其中 Header 是一个结构体，包含了 seq、stamp、frame_id，其中 seq 是扫描顺序增加的 id，stamp 包括了每次开始扫描的时间差，frame_id 是参考系名称；angle_min 和 angle_max 分别表示开始扫描和结束扫描的角度；angle_increment 是每次扫描增加的角度；time_increment 是测量的时间间隔；scan_time 是扫描的时间间隔；range_min 和 range_max 表示测量距离的最小值和最大值；ranges 是存储测量距离的数组；intensities 为强度数组，与设备类型有关。同时可以利用 ROS 中可视化工具 Rviz 查看激光数据，如图 2.9(b)。

b. 里程计数据

ROS 对移动机器人底层接口的运动控制主要通过两个消息机制实现：通过 cmd_vel 主题将数据传输给底层进行机器人的运动控制。cmd_vel 主题的消息类型是 geometry_msgs/Twist，在二维空间中只需要水平速度 linear.x、垂直速度 linear.y、角速度 angular.z；odom 主题表示的是里程计，实现数据的回传，关于机器人里程计模型，在 2.3.2 小节已经介绍过了，这里就不在赘述了。图 2.10 为 geometry_msgs/Twist 数据类型：

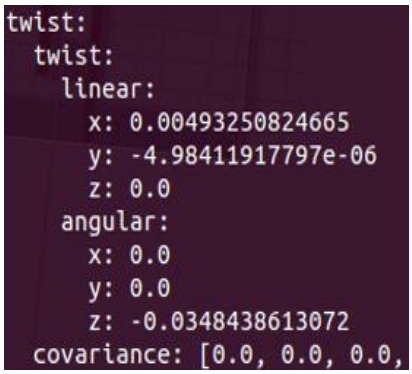


图 2.10 里程计数据

c. IMU 姿态数据

将 IMU AHRS 传感器通过串口连接到 Ubuntu 系统上，将读取的传感器返回的四元数信息，通过订阅 IMU_data 话题，就可以查看机器人的姿态信息，转动 IMU，在 rviz 中可以看到虚拟立方体和坐标轴也会随着转动。

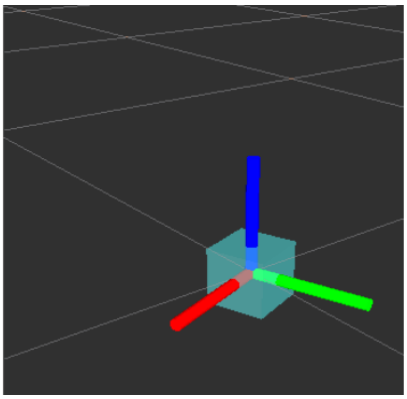


图 2.11 IMU 姿态可视化

2.4 本章小结

本章首先介绍了几种常用地图的表示方法，包括栅格地图、几何特征地图、拓扑关系地图；然后介绍了 SLAM 算法的主要框架；同时对于本次论文自主设计的测绘机器人平台的硬件部分，软件部分给出详细说明。为后续的 SLAM 实验搭建平台。

3 Hector SLAM 算法

Hector SLAM 是一种灵活的、可扩展的 SLAM 算法，已经被成功运用到无人地面车（unmanned ground vehicles ，UGV 和 unmanned surface vehicles USV) 和小型室内导航等系统。它具有低计算率、低功耗等优点，适用于小型机器人系统，最早便是应用到机器人足球世界杯援救竞赛中，在模拟的复杂崎岖的地震环境中搜救幸存者。

近十年来，关于 SLAM 算法取得了很有价值的研究，大多数 SLAM 算法在构建 2D 室内地图时取得不错的效果，但都依赖于精确和高效的里程计数据，对于激光雷达提供的高更新频率的激光数据没有很好的利用，对于一些非结构化环境的地图构建是十分困难的。Hector SLAM 算法不依赖里程计，仅利用高更新频率的激光数据便可完成 SLAM 任务。

3.1 Hector SLAM 系统概况

Hector SLAM 主要包括两个部分：2D SLAM 子系统和导航子系统。其中 2D SLAM 系统提供机器人的姿态信息，通过扫描匹配完成机器人绘制环境地图任务；导航系统采用导航滤波算法，融合 IMU 或其它传感器数据形成三维导航方案。这两个子系统是单独更新只进行松散耦合，这样保证这两部分是同步进行的。本文主要讨论 2D SLAM 子系统。

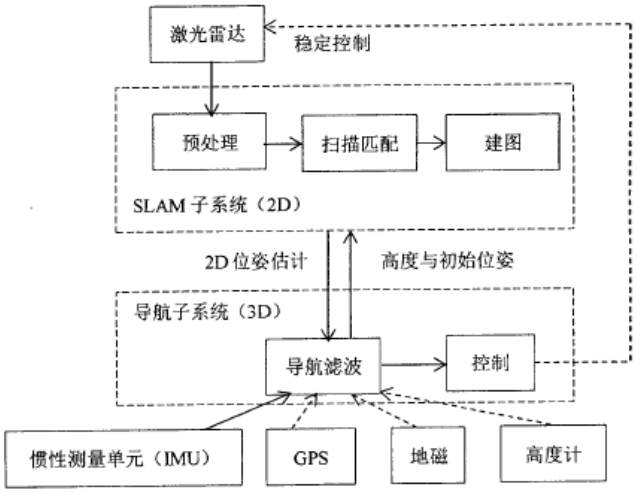


图 3.1 Hector SLAM 系统框架

依据右手坐标系原则定义一个导航坐标系统，以机器人的初始位置为坐标原点，Z 轴垂直向上，X 轴指向机器人初始的偏航角方向。机器人的 3 维坐标可以表示为：$x=(\Omega^T,P^T,V^T)$，其中，$\Omega=(\phi,\theta,\psi)^T$，表示的是机器人的翻转角，俯仰角和偏航角的欧

拉角表示；$P=(p_x,p_y,p_z)^T$，$V=(v_x,v_y,v_z)^T$ 表示机器人的位置信息和速度信息。惯导测量数据用 $u=(\omega^T,a^T)^T$ 表示,其中 $\omega=(\omega_x,\omega_y,\omega_z)^T$ 和 $a=(a_x,a_y,a_z)^T$ 分别表示机器人的角速度和加速度信息。于是，机器人的运动模型就可以写成如下的非线性系统：

$$\dot{\Omega}=E_{\Omega}\cdot\omega \tag{3.1}$$

$$\dot{P}=V \tag{3.2}$$

$$\dot{V}=R_{\Omega}\cdot a+\mathrm{g} \tag{3.3}$$

R_{Ω} 是从机器人坐标系向导航坐标系转换的方向余弦矩阵；E_{Ω} 是从机器人自身角速度到欧拉角的映射。g 是重力加速度。在这里，我们忽略了地球自转对于系统的影响。

由于传感器存在噪声，会导致速度和位置偏离真值，因此可能需要融合更多的传感器信息。在 Hector SLAM 中，这些信息是由 scan matcher 决定的，对于室内环境十分适用。

3.2 基于 Hector SLAM 的自主建图的算法

Hector SLAM 采用栅格地图的表示方式来绘制地图，建图过程主要分为两个阶段：地图获取和扫描匹配。即先用激光数据对环境进行描述，这个阶段是将连续的环境信息进行离散化的过程；然后再将这些离散的激光数据进行匹配，将局部地图融合形成全局环境地图[31]。根据激光的相对于机器人的安装位置，将激光数据转换到当地坐标系统下。我们将环境场景离散化转换为激光点云，根据具体的场景信息，对激光点云进行预处理，比如进行降采样或剔除异常值的点。Hector SLAM 算法是对 z 坐标进行滤波，这样只有激光点数据在扫描平面的某个阈值中，才会被用于扫描匹配过程。

3.2.1 地图获取

因为栅格地图的离散性会限制系统的精度，而且也不允许直接计算插值和导数，所以 Hector SLAM 采用双线性内插法进行插值，估计栅格的占用概率和计算导数。直观地说，就是栅格地图占据值可以看做是在连续概率分布函数中采样。

双线性插值，又称作双线性内插法。在数学上，双线性插值是有两个变量的插值函数的线性插值扩展，主要思想就是在两个方向上分别进行一次线性插值。在给定一个连续的地图坐标 P_m ，栅格的占用值为 $M(P_m)$，用相邻近的四个整数坐标 P_{11}，P_{10}，P_{01}，P_{00} 来估计梯度 $\nabla M(P_m)$，如图 3.2 所示。于是，x 方向和 y 方向的线性插值可以表示为：

$$\nabla M(P_m)\approx\frac{y-y_0}{y_1-y_0}\left(\frac{x-x_0}{x_1-x_0}M(P_{11})+\frac{x_1-x}{x_1-x_0}M(P_{01})\right)$$

$$+\frac{y_1-y}{y_1-y_0}\left(\frac{x-x_0}{x_1-x_0}M(P_{10})+\frac{x_1-x}{x_1-x_0}M(P_{00})\right) \tag{3.4}$$

梯度估计可以表示为：

$$\frac{\partial M}{\partial x}(P_m)\approx\frac{y-y_0}{y_1-y_0}(M(P_{11})-M(P_{01}))+\frac{y_1-y}{y_1-y_0}(M(P_{10})-M(P_{00})) \tag{3.5}$$

$$\frac{\partial M}{\partial y}(P_m)\approx\frac{x-x_0}{x_1-x_0}(M(P_{11})-M(P_{01}))+\frac{x_1-x}{x_1-x_0}(M(P_{10})-M(P_{00})) \tag{3.6}$$

这里所谓的采样点即地图的网格是位于1×1规则格网之中的。

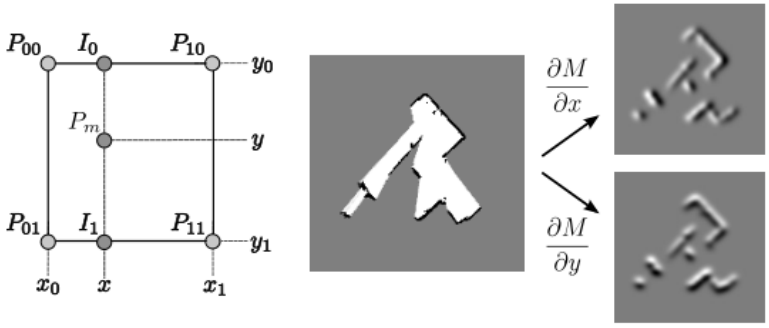


图 3.2 双线性内插和梯度下降法

3.2.2 扫描匹配

扫描匹配（scan matching）就是将激光传感器获得的数据与已有地图进行匹配的过程。Hector SLAM 要求激光扫描仪是低噪声且高扫描频率的。对于许多机器人系统，所用的激光扫描仪的精度和准确度要比里程计的精度高很多，因此 Hector SLAM 算法仅利用激光数据，根据牛顿-高斯迭代法进行扫描匹配[32]。

我们假设在静态的环境中且测量没有噪声，测量值应该是与地图高度一致的，那么：

$$\sum_{i=1}^{n}[1-M(S_i(\xi))]^2=0 \tag{3.7}$$

其中，$\xi=(p_x,p_y,\psi)^T$ 是世界坐标系下机器人的位姿；

$S_i(\xi)$ 是第 i 号激光束在机器人位姿为 ξ 时测量到的障碍物的世界坐标值；s_i 是激光束端点坐标；

$$S_i(\xi)=\begin{pmatrix}\cos(\psi) & -\sin(\psi)\\ \sin(\psi) & \cos(\psi)\end{pmatrix}\begin{pmatrix}s_{i,x}\\ s_{i,y}\end{pmatrix}+\begin{pmatrix}p_x\\ p_y\end{pmatrix} \tag{3.8}$$

$$s_i=(s_{i,x},s_{i,y})^T \tag{3.9}$$

$M(S_i(\xi))$ 表示的是某坐标是障碍物的概率，1 表示 100%为障碍物，0 表示 100%空闲区域。

很显然在没有测量误差的情况下，3.10 式是成立的。但是，实际的情况是，地图存在误差的，且测量也有噪声。为了使激光数据与地图之间达到最佳匹配，于是我们转求最小二乘：

$$\xi^* = \arg\min \sum_{i=1}^{n} [1 - M(S_i(\xi))]^2 \tag{3.10}$$

在给定 ξ 的初始估计的条件下，用下面的公式去估计 $\Delta\xi$，使测量误差最优化：

$$\sum_{i=1}^{n} [1 - M(S_i(\xi + \Delta\xi))]^2 \to 0 \tag{3.11}$$

对 $M(S_i(\xi+\Delta\xi))$ 进行一阶泰勒展开，得到：

$$\sum_{i=1}^{n} [1 - M(S_i(\xi)) - \nabla M(S_i(\xi)) \frac{\partial S_i(\xi)}{\partial \xi} \Delta\xi]^2 \to 0 \tag{3.12}$$

当 3.12 式最小时偏导数为 0，所以：

$$2\sum_{i=1}^{n} [\nabla M(S_i(\xi)) \frac{\partial S_i(\xi)}{\partial \xi}]^T [1 - M(S_i(\xi)) - \nabla M(S_i(\xi)) \frac{\partial S_i(\xi)}{\partial \xi} \Delta\xi)] = 0 \tag{3.13}$$

$$\Delta\xi = H^{-1} \sum_{i=1}^{n} [-\nabla M(S_i(\xi)) \frac{\partial S_i(\xi)}{\partial \xi}]^T [1 - M(S_i(\xi))] \tag{3.14}$$

其中：

$$H = [\nabla M(S_i(\xi)) \frac{\partial S_i(\xi)}{\partial \xi}]^T [\nabla M(S_i(\xi)) \frac{\partial S_i(\xi)}{\partial \xi}] \tag{3.15}$$

$$\frac{\partial S_i(\xi)}{\partial \xi} = \begin{pmatrix} 1 & 0 & -\sin(\psi) s_{i,x} - \cos(\psi) s_{i,y} \\ 0 & 1 & \cos(\psi) s_{i,x} - \sin(\psi) s_{i,y} \end{pmatrix} \tag{3.16}$$

由此，便可得到激光数据与地图之间达到最佳匹配效果，从而增量式的完成全局地图绘制。

3.2.3 多分辨率地图表示

几乎所有的梯度下降算法都会陷入局部最优解，因此，Hector SLAM 算法采用了一种类似于计算机视觉中图像金字塔的多分辨率地图的表示方法。使用多重分辨率去表示地图，下层地图分辨率是上层地图的一半，但和图像处理中利用高斯滤波进行降采样不同，多层次地图并不是只从单个高分辨率图像中产生，Hector SLAM 将低分辨率的地图位姿估计逐步带入高分辨地图，使不同层地图保持同步更新。本文采用双层分辨率来表示地图。

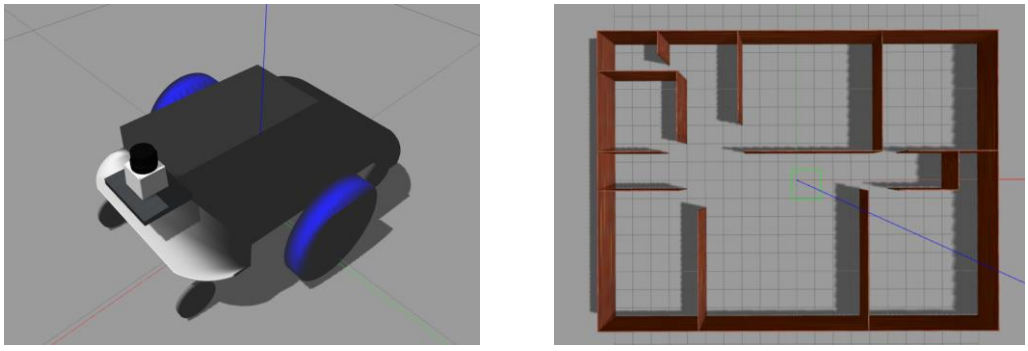
图 3.3 多分辨率地图

3.3 Hector SLAM 算法实验

3.3.1 仿真实验

a. 仿真实验平台构建

本文采用 Gazebo 仿真工具进行仿真实验，首先根据测绘机器人实验平台的尺寸形状，创建一个模拟机器人 3D 模型，在 ROS 中通过编写 URDF 文件来实现，URDF（Unified Robot Description Format），即标准化机器人模型，是一种用于描述机器人各个部件、关节、自由度等的 XML 文件。同时我们在机器人模型上添加一些激光雷达、里程计等传感器部件，图 3.4(a)是创建的机器人 3D 模型。利用 Gazebo 中另一个强大的 build model 功能来创建仿真环境地图。图 3.4(b)是本文在 Gazebo 中创建的环境模型。这个仿真环境将用于接下来的 SLAM 仿真实验。



(a)机器人 3D 模型　　(b)仿真实验环境 3D 模型

图 3.4 仿真实验平台构建

b. 实验过程

为了简化实验启动的步骤，本文将激光数据采集节点、地图加载节点、机器人运动控制节点、hector_mapping 建图节点、地图可视化节点 Rviz 等写入同一个 launch 文件，这样方便使用一个命令启动 SLAM 实验，避免出错且节省时间。

值得强调的是，由于激光扫描获取的特征信息是相对激光位置的，而我们创建地图

的信息是基于地图坐标系的，在 SLAM 过程中需要将特征信息转换到地图坐标系下；同时机器人还需要确定自身在地图坐标系下的位姿。在 ROS 中，特征点坐标转换是通过 tf 节点完成的。

通过键盘操控机器人在仿真环境中运动，随着机器人的运动，算法实时地计算机器人的位姿信息同时进行地图匹配，地图的自主绘制过程如下图，其中图 3.5 为 Hector SLAM 得到的仿真环境地图。ROS 中采用 map saver 节点保存地图，生成两个文件：.pgm 为环境地图；.yaml 文件是地图的配置文件。

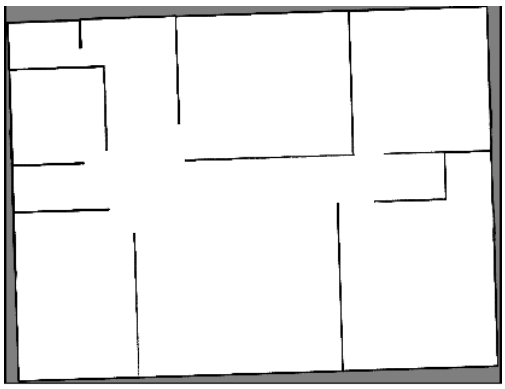
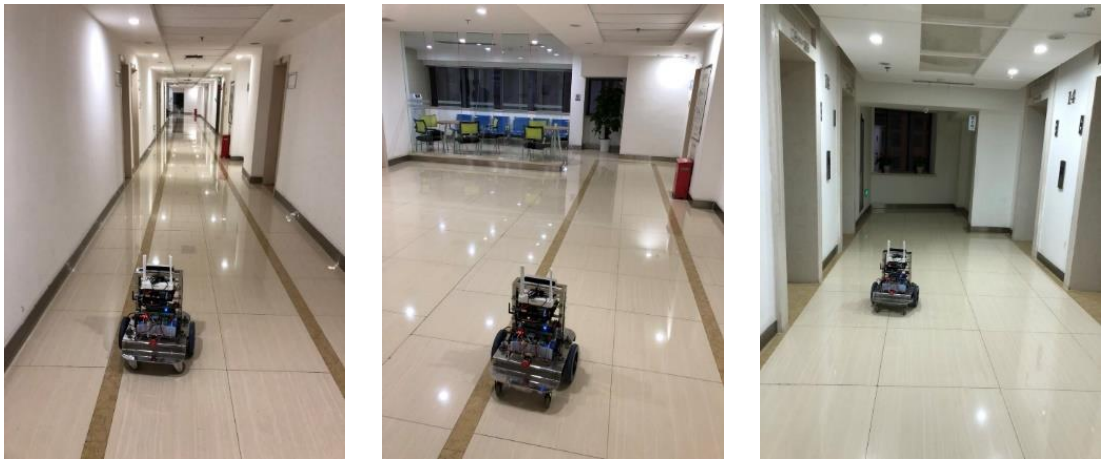


图 3.5 Hector SLAM 仿真实验结果

3.3.2 真实环境实验

实验在武汉中部创意大厦十四层楼室内进行，具体区域包括走廊、电梯间、茶水间，图 3.6 为机器人在现实场景中运动。启动 Hector SLAM 制图节点 hector_mapping，最终获得的真实环境地图如图 3.7 所示：其中白色区域为空白区域（free），灰色区域为未知区域（unknown），黑色就是障碍物所在的位置（occupied）。从目视效果上来看，建图效果还不错，不存在明显的结构误差，平整的墙壁和规则的门框可以得到很好地建图效果，但是在茶水间玻璃和反光金属材质的桌椅表面，激光会产生镜面反射，导致制图精度略差。同时在实验的过程中发现，Hector SLAM 由于仅依赖于激光匹配进行建图，因此制图过程中机器人的运动速度不宜太快，否则激光匹配会来不及更新，导致地图发生严重偏移的现象。



(a)走廊　　(b)茶水间　　(c)电梯间

图 3.6 机器人在实验环境中运行

图 3.7 Hector SLAM 真实环境建图结果

3.4 本章小结

本章主要介绍了 Hector SLAM 算法的主要理论基础，该算法不依赖除激光雷达之外的传感器就可实现 SLAM 制图工作，从仿真实验和真实环境实验分别验证了算法性能，通过实验结果可以看出，该算法可以达到较好的制图效果。

4 Gmapping SLAM 算法

Gmapping 算法是目前机器人邻域应用最广泛的 SLAM 算法，该算法是一种基于 Rao-Blackwellized 粒子滤波器（Rao-Blackwellized Particle Filter, RBPF）原理的 SLAM 算法[33]，绘制的地图是栅格地图。RBPF-SLAM 算法最早是于 1999 年由 Murph 提出来的，用来估计联合后验概率 $p(x_{1:k},m\mid z_{1:k},u_{1:k-1})$ 的，即计算在里程计测量值 (u) 和激光测量值 (z) 已知的情况下，计算机器人的实际位置 (x) 同时创建环境地图 (m)。这个后验概率可以进行因式分解为：

$$p(x_{1:k},m\mid z_{1:k},u_{1:k-1})=p(m\mid x_{1:k},z_{1:k})p(x_{1:k}\mid z_{1:k},u_{1:k-1}) \tag{4.1}$$

这样 SLAM 问题被分解为先根据里程计数据和激光数据来估计机器人的位置，再依据机器人的位置和激光数据来计算地图。因此建图过程非常依赖机器人位姿。

Gmapping 算法和 HectorSLAM 算法最大区别是增加了里程计数据，关于里程计推算机器人运动模型在 2.3.2 小节给出介绍，这里就不再赘述。

根据上述的因式分解，采用 2006 年由 Thrun, Burgard, and Fox 提出的 Occupancy grid mapping 算法和 Inverse sensor model 算法去计算 $p(m\mid x_{1:k},z_{1:k})$。因为栅格地图 m 是由有限个栅格单元组成，因此可以将整个地图的后验概率近似用边缘概率的乘积表示：$p(m\mid x_{1:k},z_{1:k})=\prod_i p(m_i\mid x_{1:k},z_{1:k})$。将二值贝叶斯滤波器应用于占用栅格地图构建，避免 0 和 1 附近的数值的不稳定性，

$$l_{k,i}=\log\frac{p(m_i\mid x_{1:k},z_{1:k})}{1-p(m_i\mid x_{1:k},z_{1:k})} \tag{4.2}$$

由此地图后验概率可以写成：

$$p(m_i\mid x_{1:k},z_{1:k})=1-\frac{1}{\exp(l_{k,i})} \tag{4.3}$$

而机器人运动轨迹的后验概率 $p(x_{1:k}\mid z_{1:k},u_{1:k-1})$ 是根据粒子滤波求解。下面对粒子滤波的基础理论做出具体介绍。

4.1 粒子滤波理论基础

粒子滤波是基于贝叶斯估计的滤波算法，关于贝叶斯滤波，假设系统的状态方程和测量方程：

$$x_k=f_k(x_{k-1},v_{k-1}) \tag{4.4}$$

$$z_k=h_k(x_k,n_k) \tag{4.5}$$

其中 x_k 是系统的状态方程，z_k 是系统的观测方程。f_k 是状态转移函数，h_k 是系统的

测量函数。v_k 是系统噪声，n_k 是测量噪声，噪声是独立同分布的。

根据贝叶斯理论，状态估计问题就是计算 x_k 在 k 时刻的置信度，在 k 时刻更新不同的观测值会得到不同的值，因此，这时就需要建立一个后验概率。假设系统初始状态向量的后验概率为 $p(x_0 \mid z_0) \equiv p(x_0)$，同时它也是先验概率（因为 0 时刻是没有测量值的），然后我们通过递归的方式获取后验概率，分为两步：预测和更新[34]。预测过程是根据系统的状态方程预测状态的先验概率密度，就是依据已有的先验知识对下一时刻的系统状态进行预测，即 $p(x_k \mid x_{k-1})$。更新的过程是根据最新的测量值对预测过程的先验概率密度进行修正，得到后验概率密度 $p(x_k \mid z_k)$。

假设，k-1 时刻的后验概率已知，预测过程就是根据系统状态模型由查普曼-科莫高洛夫方程 Chapman–Kolmogorov equation 获取 k 时刻系统的先验概率：

$$\begin{aligned} p(x_k \mid z_{1:k-1}) &= \int p(x_k, x_{k-1} \mid z_{1:k-1}) dx_{k-1} \\ &= \int p(x_k \mid x_{k-1}, z_{1:k-1})\, p(x_{k-1} \mid z_{1:k-1}) dx_{k-1} \\ &= \int p(x_k \mid x_{k-1})\, p(x_{k-1} \mid z_{1:k-1}) dx_{k-1} \end{aligned} \tag{4.6}$$

根据一阶马尔科夫假设，当前的状态只与前一时刻的状态有关，与历史观测值无关，即 $p(x_k \mid x_{k-1}, z_{1:k-1})$=p$(x_k \mid x_{k-1})$。$p(x_k \mid x_{k-1})$ 可由系统的状态方程得到。在 k 时刻，由贝叶斯定律得：

$$p(x_k \mid z_{1:k}) = \frac{p(z_k \mid x_k, z_{1:k-1})\, p(x_k \mid z_{1:k-1})}{p(z_k \mid z_{1:k-1})} = \frac{p(z_k \mid x_k) P(x_k \mid z_{1:k-1})}{p(z_k \mid z_{1:k-1})} \tag{4.7}$$

其中，归一化常数 $p(z_k \mid z_{1:k-1}) = \int p(z_k \mid x_k)\, p(x_k \mid z_{1:k-1}) dx_k$。$p(z_k \mid x_k)$ 可由系统的观测方程得到。对于后验概率的递归过程已经解决，但是上式中用到了积分，通常实际情况机器人的运动是非线性非高斯的，很难得到后验概率的解析解。为了解决这一问题，引入了蒙特卡洛采样。

蒙特卡洛采样的主要思想是用一系列带有权重的粒子估计后验概率分布，假设从后验概率分布中随机采样 n 个粒子。于是后验概率可以近似表示为：$p(x_{0:k} \mid z_{0:k}) \approx \sum_{i=1}^{N_s} \varpi_k^i \delta(x_{0:k} - x_{0:k}^i)$。其中，$\{x_{0:k}^i, \varpi_k^i\}_{i=1}^{N_s}$ 是从后验概率中随机产生的，$\{x_{0:k}^i, i = 0,1,...,N_s\}$ 是一系列的粒子，它们的权重为 $\{\varpi_k^i, i = 0,1,...,N_s\}$。$x_{0:k} = \{x_j, j = 0,1,...,k\}$ 是系统直到 k 时刻的所有状态。所有粒子的权重和为 1。

粒子的选择遵循重要性采样原则（importance sampling），因为后验概率并不知道，所以无法从后验概率分布中采样。那么，我们假设 $p(x) \propto \pi(x)$，$\pi(x)$ 是与 $p(x)$ 有相同分

布，它们的权重之间存在一定的比例。另外提出一个已知的分布函数 $q(x)$ 并从中采样，即 $x^i \sim q(x), i-1,...,N_s$。$q(x)$ 称为重要性概率密度函数。因此 $p(x)$ 可以近似表示为：$p(x) \approx \sum_{i=1}^{N_s} \varpi^i \delta(x-x^i)$。其中，$\varpi^i \propto \frac{\pi(x^i)}{q(x^i)}$ 表示第 i 个粒子权重的归一化。因此，如果从 $q(x)$ 中采样，那么粒子权重可以表示为：

$$\varpi^i \propto \frac{p(x_{0:k}^i \mid z_{1:k})}{q(x_{0:k}^i \mid z_{1:k})} \tag{4.8}$$

假设重要性概率密度函数可以分解为：

$$q(x_{0:k} \mid z_{1:k}) = q(x_k \mid x_{0:k-1}, z_{1:k}) q(x_{0:k-1} \mid z_{1:k-1}) \tag{4.9}$$

那么样本集 $x_{0:k}^i \sim q(x_{0:k} \mid z_{1:k})$，就是通过已有的粒子集 $x_{0:k\text{-}1}^i \sim q(x_{0:k\text{-}1} \mid z_{1:k\text{-}1})$ 和最近一时刻的粒子 $x_k^i \sim q(x_k \mid x_{0:k-1}, z_{1:k})$ 构成的。为了计算粒子权重的递归公式，

$$\begin{aligned} p(x_{0:k} \mid z_{1:k}) &= \frac{p(z_k \mid x_{0:k}, z_{1:k-1}) p(x_{0:k} \mid z_{1:k-1})}{p(z_k \mid z_{1:k-1})} \\ &= \frac{p(z_k \mid x_{0:k}, z_{1:k-1}) p(x_k \mid x_{0:k-1}, z_{1:k-1})}{p(z_k \mid z_{1:k-1})} \times p(x_{0:k-1} \mid z_{1:k-1}) \\ &= \frac{p(z_k \mid x_k) p(x_k \mid x_{k-1})}{p(z_k \mid z_{1:k-1})} p(x_{0:k-1} \mid z_{1:k-1}) \\ &\propto p(z_k \mid x_k) p(x_k \mid x_{k-1}) p(x_{0:k-1} \mid z_{1:k-1}) \end{aligned} \tag{4.10}$$

带入后得到：

$$\begin{aligned} \varpi_k^i &\propto \frac{p(z_k \mid x_k^i) p(x_k^i \mid x_{k-1}^i) p(x_{0:k-1}^i \mid z_{1:k-1})}{q(x_k^i \mid x_{0:k-1}^i, z_{1:k}) q(x_{0:k-1}^i \mid z_{1:k-1})} \\ &= \varpi_{k-1}^i \frac{p(z_k \mid x_k^i) p(x_k^i \mid x_{k-1}^i)}{q(x_k^i \mid x_{0:k-1}^i, z_{1:k})} \end{aligned} \tag{4.11}$$

如果 $q(x_k \mid x_{0:k-1}, z_{1:k}) = q(x_k \mid x_{k-1}, z_k)$，那么重要性概率密度函数就只与 x_{k-1} 和 z_k 有关。通常我们只需要对于每一时刻系统的状态进行更新，即只需计算 $p(x_k \mid z_{1:k})$，这时只需记录第 i 个粒子 k 时刻的状态，丢弃掉 0 到 k-1 时刻系统的状态和观测值。于是，修改权重公式：$\varpi_k^i \propto \varpi_{k-1}^i \frac{p(z_k \mid x_k^i) p(x_k^i \mid x_{k-1}^i)}{q(x_k^i \mid x_{k-1}^i, z_k)}$。因此后验概率可以写成：

$$p(x_k \mid z_{1:k}) \approx \sum_{i=1}^{N_s} \varpi_k^i \delta(x_k - x_k^i) \tag{4.12}$$

以上就是关于序贯重要性采样滤波的推导过程，包括对每个粒子权重的递归和每个采样点测量值的序贯接收。

序贯重要性采样算法就是粒子滤波的前身。可是在实际应用的过程中会遇见很多问题，最常见的问题就是粒子退化问题：即在进行几次迭代之后就会发现，一些粒子的权

重就会几乎变为 0，只有少数的粒子权重比较大。而且粒子权重的方差会随着时间不断增大，有效的粒子数目会越来越少，这样会导致大量的计算都浪费在对估计后验概率分布几乎没有作用的粒子上面，使得滤波性能下降。同时关于 $q(x)$ 重要性概率密度函数的选取问题也是一个难点。RBPF 算法在粒子滤波的基础上提出了一个精确的建议分布函数，不仅考虑到机器人的运动也顾及传感器的实时观测，有效的提高了 SLAM 问题在预测阶段机器人位姿的估计精度；更进一步提出一种自适应重采样方法减少了粒子衰减现象[35]。

4.2 RBPF 算法

4.2.1 RBPF 改进建议分布函数

上文提到的，我们需要在预测阶段从建议分布函数中采样以获得下一代粒子。很显然，建议分布函数越近似目标分布函数，滤波效果越好。比如说，如果可以直接从目标分布函数中采样，所有粒子的重要性权重都会相等，那么就不再需要进行重采样过程。但是事实上无法直接从目标分布中采样。一般的粒子滤波会将里程计运动模型作为建议分布函数从中采样，因为这种运动模型计算简单且适用于许多机器人。于是就可以根据观测模型 $p(z_t \mid m, x_t)$ 计算重要性权重，用运动模型替换 $\pi(x)$，得到：

$$\begin{aligned} \varpi_t^{(i)} &= \varpi_{t-1}^{(i)} \frac{\eta\, p(z_t \mid m_{t-1}^{(i)}, x_t^{(i)})\, p(x_t^{(i)} \mid x_{t-1}^{(i)}, u_{t-1})}{p(x_t^{(i)} \mid x_{t-1}^{(i)}, u_{t-1})} \\ &\propto \varpi_{t-1}^{(i)} \cdot p(z_t \mid m_{t-1}^{(i)}, x_t^{(i)}) \end{aligned} \tag{4.13}$$

但是，这个建议分布函数是一个次优化方案，尤其是当传感器信息明显比基于里程计进行机器人运动估计精确时。如图 4.1 所示，实线表示的是利用激光信息对于机器人位置的估计，可以看出其精度明显高于仅利用里程计运动模型作为建议分布函数的精度。

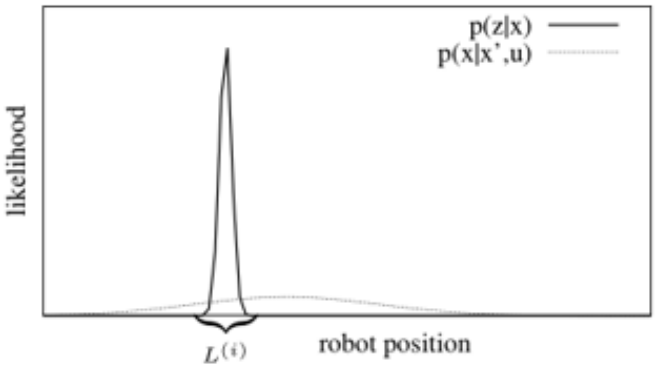


图 4.1 激光数据位置估计和里程计建议分布对比图

于是可以考虑利用最近一次的激光观测值来形成下一代粒子。通过将观测值整合到建议分布中可以集中在位置估计概率较大的区域进行采样，于是建议分布函数可以

写作：

$$p(x_t \mid x_{t-1}^{(i)}, m^{(i)}, z_t, u_{t-1}) = \frac{p(z_t \mid x_t, m^{(i)})\, p(x_t \mid x_{t-1}^{(i)}, u_{t-1})}{p(z_t \mid x_{t-1}^{(i)}, u_t)} \tag{4.14}$$

通常会对目标分布进行重定义：$\tau(x) = p(z_t \mid x_t, m^{(i)})\, p(x_t \mid x_{t-1}^{(i)}, u_t)$。但是仍存在一个问题：因为观测似然函数的形状是无法预测，因此无法得到建议分布函数的近似形状 。通常，观测似然函数通常只有一个峰值，我们可以在这个峰值附近采样，更进一步的说就是，通过扫描匹配算法在观测似然函数峰值附近来局部近似后验概率 $p(x_t \mid x_{t-1}^{(i)}, m^{(i)}, z_t, u_t)$。具体得做法是：首先，从里程计分布中进行下一代粒子采样，然后将当前的观测值z_t与已获取的局部地图进行匹配直到粒子的估计最接近机器人真实的位置。关于激光点云的匹配用到的是 ICP（Iterative Closest Point）算法，将两组点云数据通过最小二乘进行迭代匹配。为了更加有效的获取下一代粒子，我们计算基于获取数据的一个近似高斯分布并从中进行采样，然后根据目标分布评价这些采样点。在计算均值和方差时也会考虑里程计数据。

$$\mu^{(i)} = \frac{1}{\eta^{(i)}} \sum_{j=1}^{K} x_j \cdot p(z_t \mid m_{t-1}^{(i)}, x_j)\, p(x_j \mid x_{t-1}^{(i)}, u_{t-1}) \tag{4.15}$$

$$\sum\nolimits_t^{(i)} = \frac{1}{\eta^{(i)}} \sum_{j=1}^{K} p(z_t \mid m_{t-1}^{(i)}, x_j) \cdot p(x_j \mid x_{t-1}^{(i)}, u_{t-1}) \cdot (x_j - \mu_t^{(i)})(x_j - \mu_t^{(i)})^T \tag{4.16}$$

接着进行归一化处理：

$$\eta^{(i)} = \sum_{j=1}^{K} p(z_t \mid m_{t-1}^{(i)}, x_j) \cdot p(x_j \mid x_{t-1}^{(i)}, u_{t-1}) \tag{4.17}$$

至此，便得到了最优建议分布函数，于是新的权重计算可以写作：

$$\begin{aligned} \varpi_t^{(i)} &= \varpi_{t-1}^{(i)} \cdot p(z_t \mid m_{t-1}^{(i)}, x_{t-1}^{(i)}, u_{t-1}) \\ &= \varpi_{t-1}^{(i)} \cdot \int p(z_t \mid m_{t-1}^{(i)}, x') \cdot p(x' \mid x_{t-1}^{(i)}, u_{t-1}) dx \\ &\cong \varpi_{t-1}^{(i)} \cdot \sum_{j=1}^{K} p(z_t \mid m_{t-1}^{(i)}, x_j) \cdot p(x_j \mid x_{t-1}^{(i)}, u_{t-1}) \end{aligned} \tag{4.18}$$

4.2.2 RBPF 自适应重采样

重采样过程是给权重很低的粒子重新分配较大的权重。重采样过程的实质是减少无效粒子、增加有效粒子。对各个粒子的权重进行比较，将重要性权值小的粒子舍弃，保留重要性权值较大的粒子，这样来适应动态系统的变化。一般比较常见的重采样方法有分层重采样、参差重采样、最小方差和系统重采样方法，他们的主要思想是根据粒子以及粒子对应的权重的后验概率分布函数，对他们的粒子权重进行重新分配以产生权值一样的新粒子集。因为这些方法原理较简单，易于实现且计算量低，因此被广泛运用[1]。图

4.2 表示的就是重采样的过程：在重采样前上面大小不一的实心圆表示不同粒子的权值大小，与直径成正比；经过重采样后，权重小的粒子会被舍弃，权重大的粒子保留被复制，最终所有粒子的权值都是 1/ M 。所有粒子权重一样，计算时间及复杂度就降低了。

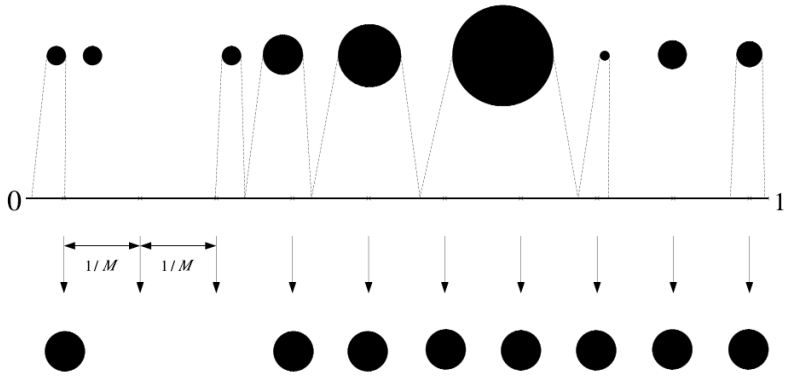


图 4.2 重采样过程

这一步是十分必要的，如果不进行重采样，那么就会出现仅有有限个粒子逼近目标分布，这样很容易出现粒子退化现象。但是，传统的粒子滤波在重采样过程中，是对所有粒子分配相同的权重，这样可能会移除一些好的粒子使滤波器贫化，同时，进行重采样的时机对于粒子滤波器的性能也是至关重要的。

RBPF 采用了一种自适应重采样过程，即根据目标函数去调整采样大小。根据公式 4.19 计算采样数量：

$$Neff = \frac{1}{\sum_{i=1}^{N}(\tilde{\varpi}^{(i)})^2} \tag{4.19}$$

其中 $\tilde{\varpi}^{(i)}$ 表示的是粒子 i 的归一化权重。

$Neff$ 的关键思想是：粒子近似越差，那么粒子之间的权重差异就越大。如果估计结果很好，那么粒子将具有相同的权重大小。具体的做法就是，当 $Neff$ 的值低于粒子总数的一半时，我们进行重采样步骤。

4.3 Gmapping 算法实验

4.3.1 仿真实验

仿真实验的环境采用 3.3.1 节中构建的仿真环境地图。编写 launch 文件去配置 slam_gmapping 功能包的坐标框架、激光数据阈值、采样粒子数目、地图更新频率等相关参数，使其更好地适应当前仿真环境。本文实验中使用的参数具体如下：

odom_frame: odom	linearUpdate: 0.5	astep:0.05
base_frame: base_footprint	angularUpdate: 0.3	iterations: 5

map_frame: map
map_update_interval: 1.0
throttle_scans: 1
maxUrange: 7.9
maxRange: 8.0
minmumScore: 3.0
linearUpdate: 0.5
angularUpdate: 0.3
temporalUpdate: -1.0
resampleThreshold: 0.5
temporalUpdate: -1.0
resampleThreshold: 0.5
xmin: -16.0
ymin: -16.0
xmax: 16.0
ymax: 16.0
delta: 0.05
sigma: 0.05
kernelSize: 1
lstep: 0.05
lsigma: 0.075
ogain: 3.0
lskip: 5
llsamplerange: 0.01
llsamplestep: 0.01
lasamplerange: 0.05
lasamplestep: 0.05
srr: 0.1
srt: 0.2

需要强调的是，我们将制图分辨率设置为 0.05(即 5cm)，地图大小为 640*640。同 Hector SLAM 一致，这样做的目的是方便后面对不同 SLAM 算法结果进行评价。

由于 Gmapping 算法需要提供里程计的信息，因此需要 map→odom→base_link→laser 的 tf 变换，而其中 map→odom 由 gmapping 提供。base_link→laser 的变换是固定变换，可由 robot_state_publisher 或者 static_transform_publisher 进行发布。而 odom→base_link 的变换可由 laser_scan_matcher 提供，配置好之后生成的 tf 变换树如图 4.3 所示。

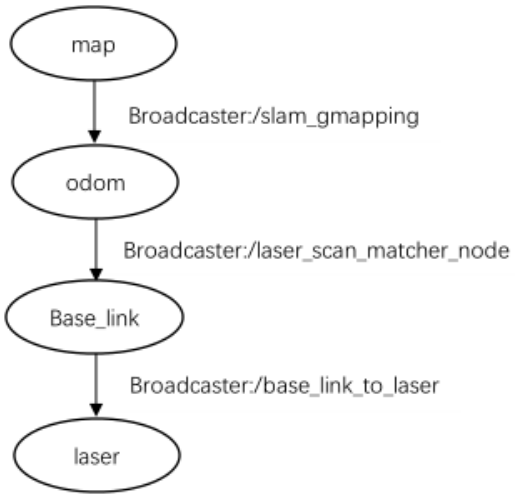


图 4.3 tf 坐标转换关系图

执行命令：roslaunch smart_robot slam_gmapping.launch，通过键盘控制机器人移动，最终绘制的仿真环境地图为：

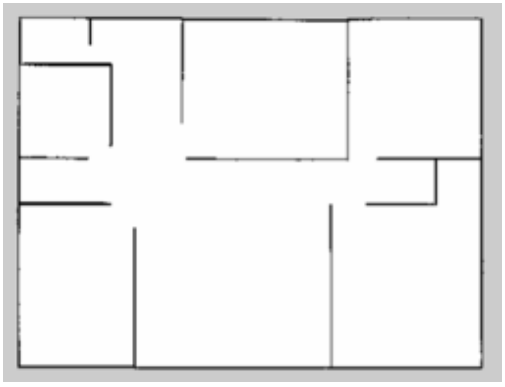


图 4.4 Gmapping 仿真实验结果

4.3.2 真实环境实验

同样，实验场地和上一章的相同，均在武汉中部创意大厦 14 楼室内进行，参数配置完成后，启动 slam_gmapping 节点进行制图实验，图 4.5 为 Gmapping 算法绘制地图。

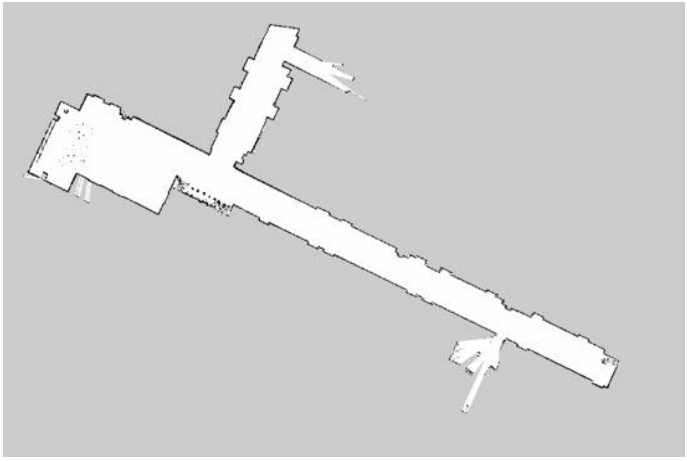


图 4.5 Gmapping 真实环境建图结果

因实验场地相同，从目视效果上来对比上一章节中 Hector SLAM 建图效果，可以看出对于像长廊这样的环境，用 Hector SLAM 获取的地图不是很精确，主要的原因是因为 Hector SLAM 只有激光雷达数据作为输入，而 Gmapping 算法不仅仅有激光雷达数据的输入，还有里程计的信息，所以其建出的地图相比 Hector SLAM 在细节部分的处理要精确。同时由于 Hector SLAM 依赖于高更新频率的激光数据，且不具有闭环的能力，因此其地图的鲁棒性没有 Gmapping 好。但 Hector SLAM 相比于 Gmapping 最大的优势就是其不需要里程计，这也降低了机器人的成本与重量。

4.4 本章小结

本章主要介绍了 Gmapping 算法的主要理论基础，从粒子滤波基础理论开始，进一步介绍 RBPF 关于粒子滤波建议分布函数的改进和自适应重采样方法；该算法除激光雷

达数据之外还添加了里程计数据去近似表示建议分布函数。最终从仿真实验和真实环境实验分别验证了算法性能，从目视效果对比上一章 Hector SLAM 算法，发现 Gmapping 算法的制图较 Hector SLAM 更精准。

5 Cartographer 算法

Cartographer 是 2006 年 10 月谷歌推出的一个实时同步定位与制图（SLAM）库，Google 设计一款背包，将传感器安装在背包上，通过操作人员在室内行走实时的绘制室内的二维格网地图。由激光雷达获得的每一帧 laser scan 数据，利用扫描匹配(scan matching)在最佳估计位置处插入到子图(submap)且 scan matching 只跟当前 submap 有关。Cartographer 没有采用粒子滤波的方法，通过定期优化来解决误差累计问题。当生成一个 submap 且没有新的 scan 插入时，会进行一次局部的回环(loop close)；submap 完成后，会自动封闭回环。如果激光扫描与当前估计的位姿很接近时，会在 submap 中寻找一致的 scan，如果找到了与当前位姿最佳的匹配结果，则将其作为闭环约束添加到优化问题中。采用分支定界和预先计算的网格来解决全局回环问题：每隔几秒进行一次优化，当机器人运动至重复地点时，扫描匹配会先于新扫描的生成，以此实现闭环检测。本文将该算法应用于测绘机器人平台上。绘制二维室内地图，并与 Hector SLAM 和 Gmapping 进行对比分析。

5.1 局部 2D SLAM 问题

Cartographer 分为局部和全局两部分来绘制二维地图。机器人的姿态包括平移 (x, y) 和旋转 ξ_θ 三个参数，表示为 $\xi=(\xi_x,\xi y,\xi_\theta)$，将当前姿态下的激光观测值叫做 scan。同时该算法会加入 IMU 数据，用于估计重力方向，保证激光位置与地面平行。利用非线性优化方案将一系列连续的激光观测值(scans)与子图(submap)进行匹配，这一过程称作扫描匹配（scan matching）。扫描匹配过程中的误差处理将在 5.2 节全局闭环检测方法中具体描述[36]。

a.激光数据

子图结构是激光数据重复在子图坐标系中匹配的迭代过程，我们将激光点信息写作 $H=\{h_k\}_{k=1,\ldots K}$，$h_k\in R^2$，初始激光点为 $0\in R^2$。测得的机器人位姿是基于激光位置的，因此由公式 5.1 将其转换到子图坐标系下：

$$T_{\xi_p}=\begin{pmatrix}\cos\xi_\theta & -\sin\xi_\theta\\ \sin\xi_\theta & \cos\xi_\theta\end{pmatrix}p+\begin{pmatrix}\xi_x\\ \xi_y\end{pmatrix} \tag{5.1}$$

b.子图

一系列的连续激光点被用来构建子图。子图采用概率格网的形式表示。构建的环境地图可以看做是一系列离散的具有一定数值的格网点组成。格网的值就是这个格网被占用的概率值。我们用相关的像素去近似表示每一个格网点，将激光点插入格网时，每一

个格网都有 hits 和 miss 两种情况。离激光点终点最近的格网为 hits，在激光点原点和终点之间相交的格网为 miss。如果是没有观测到的格网会分配一个概率值p_{hit}或者p_{miss}，如果是已经观测到的格网会对其进行概率更新。图 5.1 中，叉号所在的格网为 hits，与激光束相交的格网为 misses。

$$odds(p)=\frac{p}{1-p} \tag{5.2}$$

$$M_{new}(x)=clamp(odds^{-1}(M_{old}(x)\cdot odds(p_{hit}))) \tag{5.3}$$

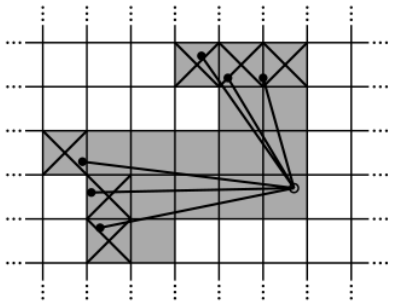


图 5.1 hits 和 misses 示意图

c. Ceres 扫描匹配

在将激光点插入子图之前，要对扫描姿态相对子图进行优化，这里采用基于 Ceres 扫描匹配方法，将求解扫描位姿问题转换为一个求解非线性最小二乘问题。

$$\arg\min_{\xi}\sum_{k=1}^{K}(1-M_{smooth}(T_{\xi}h_{\xi}))^{2} \tag{5.4}$$

上式中，T_{ξ}是根据扫描位姿将激光信息h_k从激光坐标系转换到子图坐标系的转换系数。 $M_{smooth}:R^2\rightarrow R$是对局部子图中概率值使用双三次插值法进行平滑处理，以此消除区间[0,1]之外数值的影响。

这个平滑函数的数学优化通常会比网格的分辨率具有更好的精度。由于最小二乘问题是局部优化，所以需要良好的初始估计（pose 初值），因此 IMU 被用来提供位姿初值的旋转变量。在缺乏 IMU 时，可以提高扫描匹配的频率或匹配精度。

5.2 闭环检测

由于扫描匹配仅与当前子图进行匹配，而地图又是由多个子图构成的，因此会不断地累积误差。Cartographer 遵循稀疏位姿调整方法（Sparse Pose Adjustment SPA）来优化所有的扫描和子图的姿态[37]：插入扫描处相对应的位姿会被存储在内存中用于闭环优化；除此之外，一旦子图不再改变，所有其他扫描和子图对应的位姿也会被用来作闭环；扫描匹配器是在后台运行的，如果找到合适的匹配项，会将其相应的位姿也添加到优化问题中。

5.2.1 SPA 优化问题

闭环优化问题和扫描匹配一样，都被认为是一个非线性最小二乘问题。采用 SPA 优化方法进行优化：

$$\arg\min_{\Xi^m,\Xi^S}\frac{1}{2}\sum_{i,j}\rho(E^2(\xi_i^m,\xi_j^s;\sum_{i,j},\xi_{i,j})) \tag{5.5}$$

其中 $\Xi^m=\{\xi_i^m\}_{i=1,\ldots,m}$，$\Xi^s=\{\xi_j^s\}_{j=1,\ldots,n}$ 表示的是子图姿态和扫描姿态根据一定的约束条件在世界坐标系中的优化，这些约束条件采用相对姿态 $\xi_{i,j}$ 和相对协方差矩阵Σ_{ij}的形式。对于子图i和激光扫描j，姿势 $\xi_{i,j}$ 描述了激光扫描在子图坐标系中的匹配时的位置。协方差矩阵Σ_{ij}可以采用例如 Ceres 中协方差估计特征这样的方法来评估。这个约束的残差 E 由下式计算：

$$E^2(\xi_i^m,\xi_j^s;\sum_{i,j},\xi_{i,j})=e(\xi_i^m,\xi_j^s;\xi_{i,j})^T\sum_{ij}^{-1}e(\xi_i^m,\xi_j^s;\xi_{i,j}) \tag{5.6}$$

$$e(\xi_i^m,\xi_j^s;\xi_{i,j})=\xi_{i,j}-\begin{pmatrix}R_{\xi_i^m}^{-1}(t_{\xi_i^m}-t_{\xi_j^s})\\ \xi_{i:\theta}^m-\xi_{j:\theta}^m\end{pmatrix} \tag{5.7}$$

使用 loss 函数ρ来减少扫描匹配在优化问题中增加不正确约束时可能出现的异常值的影响。这种异常值一般可能发生在类似办公室隔间这样对称环境中。

5.2.2 分支定界扫描匹配（BBS）

Cartographer 算法采用另一种称作分支定界扫描匹配（Branch-and-bound scan matching）的方法来进行闭环优化[38]。

$$\xi^*=\arg\min_{\xi\in W}\sum_{k=1}^{K}M_{nearest}(T_\xi h_k) \tag{5.8}$$

其中，W 是搜索窗口，$M_{nearest}$ 是对 M 通过四舍五入扩展到所有的激光数据集 R^2 上，首先从最近的网格点开始，将其值扩展到相应的像素上。通过选择合适的步进长度来提高效率。选择角度的增长值为 δ_θ，使得激光的最大测距范围 $d_{\max}$ 不会超过一个像素的宽度。由勾股定理可以得到：

$$d_{\max}=\max_{k=1,\ldots,K}\|h_k\| \tag{5.9}$$

$$\delta_\theta=\arccos(1-\frac{r^2}{2d_{\max}^2}) \tag{5.10}$$

根据搜索窗口的大小计算整数倍步进长度，使其可以覆盖整个搜索窗口。

$$w_x = [\frac{w_x}{r}] \quad w_y = [\frac{w_y}{r}] \quad w_\theta = [\frac{w_\theta}{\delta_\theta}] \tag{5.11}$$

公式 5.11 是以估计位姿ξ_0为中心形成搜索窗口：

$$\bar{W} = \{-w_x, ..., w_x\} \times \{-w_y, ..., w_y\} \times \{-w_\theta, ..., w_\theta\} \tag{5.12}$$

$$W = \{\xi_0 + (rj_x, rj_y, \delta_\theta j_\theta) : (j_x, j_y, j_\theta) \in \bar{W}\} \tag{5.13}$$

采用分支定界方法有效的计算ξ^*。主要思想是：将可能的子集的可能性表示为树的节点，树的根节点表示所有的可能的情况，也就是我们的搜索窗口W。每一个节点的子节点都是他们父节点的一个分区，因此他们具有相同的可能性且每个节点都是独立的，各自代表一个可行方案。该方法主要包括三个部分内容：节点的选择，分支规则以及上限的计算。

a.节点选择

使用深度首次搜索（depth-first-search DFS）方法进行节点的选择。DFS 方法对节点进行快速评估，为了避免将一些较差的匹配结果加入闭环检测，引入一个评分阈值，将小于该阈值的节点舍弃当。计算每一个节点的分数上限进行排序，选择分数最高的节点。图 5.2 是它的伪代码：

Algorithm 3 DFS branch and bound scan matcher for (BBS)

```
best_score ← score_threshold
Compute and memorize a score for each element in C_0.
Initialize a stack C with C_0 sorted by score, the maximum
score at the top.
while C is not empty do
  Pop c from the stack C.
  if score(c) > best_score then
    if c is a leaf node then
      match ← ξ_c
      best_score ← score(c)
    else
      Branch: Split c into nodes C_c.
      Compute and memorize a score for each element
      in C_c.
      Push C_c onto the stack C, sorted by score, the
      maximum score last.
    end if
  end if
end while
return best_score and match when set.
```

图 5.2 BBS 伪代码

b.分支规则

树中的每个节点都由一个整数元组来描述：$c=(c_x,c_y,c_\theta,c_h)\in Z^4$。节点高度$c_h$结合了$2^{c_h}\times 2^{c_h}$种可能的转换，具体特定的转换是：

$$\bar{\bar{W}}_c=(\{(j_x,j_y)\in Z^2:\begin{matrix}c_x\le j_x<c_x+2^{c_h}\\ c_y\le j_y<c_y+2^{c_h}\end{matrix}\}\times\{c\theta\}) \tag{5.14}$$

$$\bar{W}=\bar{\bar{W}}\bigcap\bar{W} \tag{5.15}$$

节点的高度$c_h=0$时的解决方案：$W\ni\xi_c=\xi_0+(rc_x,rc_y,\delta_\theta c_\theta)$。对父节点进行分支，分为一系列的初始化节点$c_0$，这些初始化节点都具有固定高度且覆盖整个搜索窗口：

$$\bar{W}_{0,x}=\{-w_x,2^{h_0}j_x:j_x\in Z,0\le 2^{h_0}j_x\le 2w_x\} \tag{5.16}$$

$$\bar{W}_{0,y}=\{-w_y,2^{h_0}j_y:j_y\in Z,0\le 2^{h_0}j_y\le 2w_y\} \tag{5.17}$$

$$\bar{W}_{0,\theta}=\{j_\theta\in Z,-w_\theta\le j_\theta\le w_\theta\} \tag{5.18}$$

$$c_0=\bar{W}_{0,x}\times\bar{W}_{0,y}\times\bar{W}_{0,\theta}\times\{h_0\} \tag{5.19}$$

给定一个节点 c 的节点高度$c_h>1$，我们将其分支为四个高度为c_{h-1}的子节点：

$$c_c=((\{c_x,c_x+2^{c_h-1}\}\times\{c_y,c_y+2^{c_h-1}\}\times c_\theta)\bigcap\bar{W})\times\{c_h-1\} \tag{5.20}$$

c.上限计算

BBS 方法最后的一步就是计算每个内部节点的上限。上限由下面的公式可得：

$$\begin{aligned}score(c)&=\sum_{k=1}^{K}\max_{j\in\bar{\bar{W}}}M_{nearest}(T_{\xi j}h_k)\\&\ge\sum_{k=1}^{K}\max_{j\in\bar{W}}M_{nearest}(T_{\xi j}h_k)\\&\ge\max_{j\in\bar{\bar{W}}}\sum_{k=1}^{K}M_{nearest}(T_{\xi j}h_k)\end{aligned} \tag{5.21}$$

为了得到最大值，可以预先计算格网$M_{precomp}^{c_h}$。每个预先计算格网的可能高度为c_h，可以从一定数量的激光点中计算 score。同时还需要计算最大的$\bar{\bar{W}}$，它可能会比搜索窗口中相邻分支中$\bar{W}$都大。

$$score=\sum_{k=1}^{K}M_{precomp}(T_{\xi c}h_k) \tag{5.22}$$

$$M_{precomp}^{h}(x,y)=\max_{\substack{x'\in[x,x+r(2^{h}-1)\\ y'\in[y,y+r(2^{h}-1)}} M_{nearest}(x',y') \tag{5.23}$$

其中 ξ_c 来自之前的节点。$M_{precomp}$ 和 $M_{nearest}$ 具有相同的像素结构，每个像素中存储当前的像素框中 $2^h \times 2^h$ 的最大值。

为了使预先计算网格的计算量最小，需要等到概率网格不会再进一步更新时，再进行下一组格网的计算，并进行扫描匹配。对于每一个预先计算的格网，我们会从每一个像素的宽开始计算 2^h 的最大值。使用中间结果，然后构建下一个预先计算网格。连续的最大值被存储在双端队列中，它是一个包括所有当前值的最大值的集合，以及所有值的连续最大值的列表。对于一个空集，它的列表也是一个空列表。使用这种方法可以在$\mathcal{O}(\mathrm{n})$中计算预先计算网格，其中 n 是每个预先计算格网中的像素个数。

5.3 Cartographer 算法实验

5.3.1 仿真实验

Cartographer 仿真实验的环境同样采用 3.3.1 小结中构建的仿真环境。地图分辨率同样为 5cm。与前两章实验不同的地方是，我们加入 IMU 数据，使得更加精确地估计测绘机器人的位姿。图 5.3 是 Cartographer 的仿真实验结果：

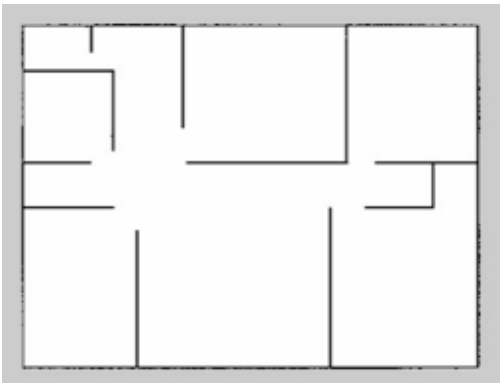


图 5.3 Cartographer 仿真实验结果

5.3.2 真实环境实验

选择和前两种 SLAM 算法相同的真实实验环境，调整配置合适的参数后，我们启动 Cartographer 节点进行制图实验，图 5.4 为 Cartographer 算法绘制室内二维地图：

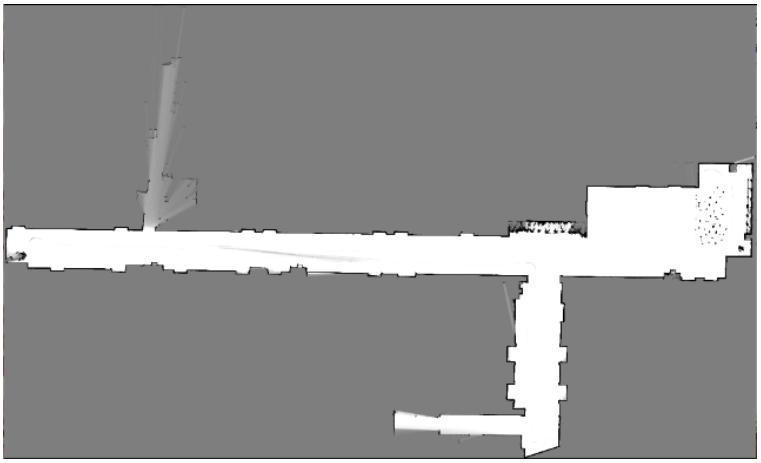


图 5.4 Cartographer 真实环境建图结果

5.4 Cartographer 和 Hector SLAM、Gmapping 算法的对比分析

5.4.1 仿真实验对比分析

对以上三种 SLAM 算法的建图精度进行对比分析：对比实验是基于 MATLAB 平台的，为方便对不同算法的实验结果进行对比，本文在进行 SLAM 算法建图实验时，将制图大小设置与仿真环境地图一致，均是 640*640，三种 SLAM 算法的地图分辨率均为 0.05(图像上一个像素代表实际 5cm)。首先根据仿真实验环境的尺寸在 MATLAB 中绘制图 5.5 的 ground truth 用来在评价基准。然后对绘制的地图进行二值化，图像像素值为 0 表示障碍物，像素值为 1 表示非障碍物；

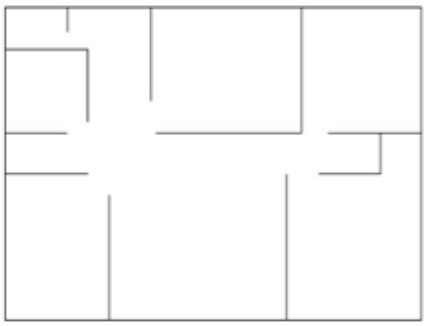


图 5.5 ground truth

采用 KNN（K-nearest-neighbor）算法[39]对 SLAM 算法绘制地图和 ground truth 进行匹配，KNN 算法的思想很简单：如果一个样本的特征空间中最邻近（欧式距离进行判断）的 k 个点大部分都属于某一类，那么这个样本就属于这一类。实际中 k 的取值会对分类结果产生很重要的影响。在这里因为只存在障碍物这一类型，因此取 k=1，即计算绘制地图中距离 ground truth 中每个障碍物最近的距离，同时进行归一化处理，将这些距离求和后，除以仿真环境地图中障碍物的总个数。本文对三种 SLAM 算法分别进行三次仿真

实验，消除因为机器人移动过程中操作等外在因素的影响。计算地图匹配结果，并求其均值及方差，最终的实验结果如表 5.1 所示：

表 5.1　KNN 算法评价仿真实验结果

序号	Hector SLAM	Gmapping	Cartographer
1	0.8897	0.4909	0.3165
2	0.6807	0.5804	0.2660
3	1.0180	0.6509	0.2855
均值	0.8628	0.5741	0.2893
方差	0.0193	0.0043	0.00042

从上述仿真实验对比结果可以看出：Cartographer 的制图精度明显优于 Hector SLAM 和 Gmapping 算法，且它的方差最小，说明 Cartographer 算法不仅精度比前两种高，且具有很好地稳定性；实验结果与本文对前两章的实验结果目视对比结果一致。主要原因在于，Cartographer 在进行制图时，在传感器数量上具有明显优势，其利用激光和里程计数据的同时结合 IMU 数据对机器人在制图过程中的位姿进行优化，使其得到比 Hector SLAM 和 Gmapping 更好的制图效果。这也验证了多传感器的结合使用对于测绘机器人制图效果有明显的优势。

5.4.2 真实环境实验对比分析

在仿真实验的基础上，本文对 Hector SLAM、Gmapping、Cartographer 在武汉中部创意大厦的真实场景下的实验进行评价。我们采用传统测量方法测量包括门框、电梯口、走廊宽度、花坛、柱子等物体的实际距离。图 5.6 是根据传统测绘手段在 CAD 中绘制的实验场地示意图。同样基于 MATLAB 平台，分别获取三组 SLAM 算法所绘制地图上的地物像素距离，比较其与传统测量方法的误差[40]。

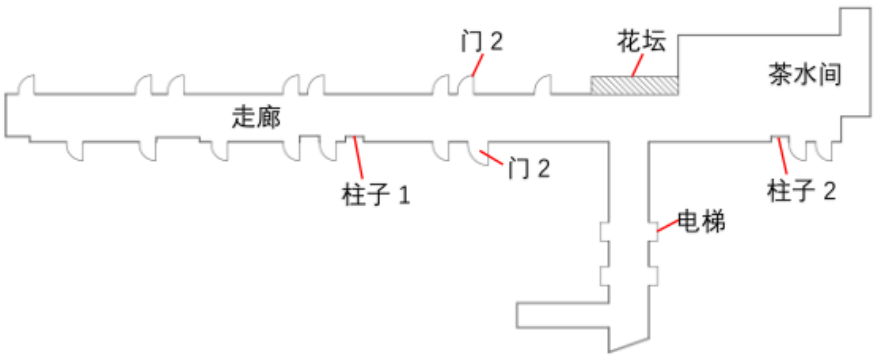


图 5.6　实验场景示意图

表 5.2 表示 Hector SLAM 制图得到的室内特征地物对比结果。表 5.3 表示的是 Gmapping 算法制图得到的室内特征地物对比结果。表 5.4 表示的是 Cartogarpher 算法制图得到的室内特征地物对比结果。在表 5.5 中对 HectorSLAM、Gmappig、Cartographer 三种算法制图结果最大差值、最小差值以及中误差的对比。

表 5.2 Hector SLAM 室内特征地物对比 单位：cm

序号	特征地物	传统方法测得长度	Hector SLAM 制图长度	差值
1	电梯门	101	100	1
2	门 1	106	115	9
3	门 2	90	90	0
4	走廊	260	225	35
5	花坛	485	505	20
6	柱子 1	113	110	3
7	柱子 2	105	110	5
8	消防器材	33	40	7

表 5.3 Gmapping 室内特征地物对比 单位：cm

序号	特征地物	传统方法测得长度	Gmapping 制图长度	差值
1	电梯门	101	98	3
2	门 1	106	110	4
3	门 2	90	89	1
4	走廊	260	234	26
5	花坛	485	501	16
6	柱子 1	113	112	1
7	柱子 2	105	99	6
8	消防器材	33	49	16

表 5.4 Cartographer 室内特征地物对比 单位：cm

序号	特征地物	传统方法测得长度	Cartographer 制图长度	差值
1	电梯门	101	100	1
2	门 1	106	115	9
3	门 2	90	85	5
4	走廊	260	265	5
5	花坛	485	500	15

6	柱子 1	113	110	3
7	柱子 2	105	110	5
8	消防器材	33	35	2

表 5.5 三种 SLAM 算法制图误差对比 单位：cm

	Hector SLAM	Gmapping	Cartographer
最大差值	35	26	15
最小差值	0	1	1
均值	10	9.125	6.25
中误差	14.9583	12.5050	4.1218

从真实环境实验对比结果可以看出：Hector SLAM 算法的最大误差达到 35cm，这和扫描匹配过程中的误差累计有很大关系。因为 Hector SLAM 其极度依赖于激光扫描匹配结果且没有使用里程计数据，所以当环境中特征的数量较少时会影响 Hector SLAM 的精度。在像本文实验场地这样的长廊环境 Gmapping 制图效果要比 Hector SLAM 好，精度可以控制到 12cm 范围内。同时，因为 Hector SLAM 对于激光的性能要求很高，要求激光具有高精度且高更新频率，本文实验所用的 UST-30LX 2D 激光扫描仪满足其要求，但如果使用的激光传感器性能略差会对制图效果产生影响。通过实验对比数据可以看出，Cartographer 测图精度是最好的，可以达到 5cm 以内。这是因为 Cartographer 在算法上没有采取粒子滤波算法，关于激光数据扫描匹配采用 SPA 和 BBS 算法进行优化，虽然从原理上来看运算量上要大于 Hector SLAM 和 Gmapping，但速度比想象中的要快。同时，Cartographer 比较 Hector SLAM、Gmapping 算法另一大的优势在于它具有回环检测，在实验的过程中发现，Cartographer 会依据全局地图去纠正局部地图，所以当局部地图出现偏差时，其闭环检测会将局部地图重新拉回全局地图进行匹配，这也使得 Cartographer 具有很好鲁棒性。

5.5 测绘机器人在项目中的应用

项目任务要求为武汉德圣堂变电站研发一款巡检机器人，代替工作人员对变电站进行仪器表检测，变电站环境异常检测等任务。要完成该任务首先需要为机器人提供导航的二维室内地图，本文将自主设计的测量机器人应用到该项目中，完成对变电站室内环境地图的绘制。变电站实地环境是一个中间集中为电表等仪器设备，四周是人员通道的矩形场地，图 5.7 是变电站环境实景图。

图 5.7 变电站实际环境

为了得到更好的制图效果，采取的具体操作步骤为：让测绘机器人在该环境下从一点出发，运动一个环形后回到起点处，完成一次制图后，再将机器人倒着沿之前的路径回到起始点，保证机器人可以对电厂环境每个区域都能观测到。

图 5.8 是 Cartographer 对该环境的制图效果，Cartographer 先由一定数量的 laser scan 构建 submap，再由 submap 拼接成地图，间隔一定数量的扫描后对所有 submap 进行一次图优化，从而完成闭环检测过程。对于旋转速度高的突然转向，也没有出现制图错误，利用回环检测消除累计误差，从实验结果来看，Cartographer 鲁棒性好，制图稳定且效果很好。为实现机器人巡检提供了很好的地图基础。整个制图过程仅需十分钟，相比较传统测量大大节省了时间。

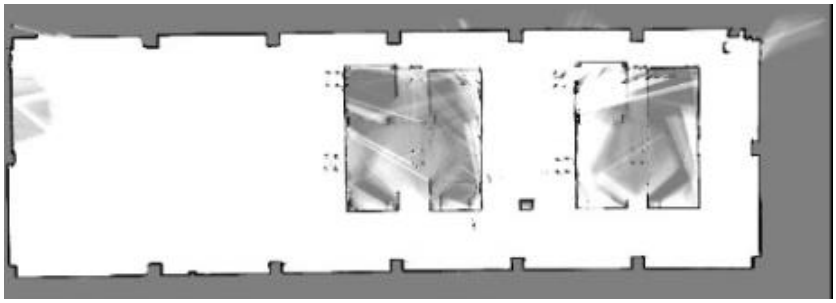


图 5.8 Cartographer 绘制变电站室内地图

5.6 本章小节

本章首先从 2D 局部建图和闭环优化两个方面对 Cartographer 算法的理论框架进行介绍，Cartographer 是一种在前端采集传感器数据后端进行非线性优化的 SLAM 算法；重点介绍了 Cartographer 中稀疏位姿调整和分支定界扫描匹配算法，最后分别进行仿真环境实验和真实环境实验，给出制图结果；结合上两章节，与 Hector SLAM 算法和 Gmapping 算法进行对比，结果表明：Cartographer 较前两种算法累计误差明显减少，且其回环检测功能，使得该算法具有很好的鲁棒性；最后将测绘机器人基于 Cartographer 制图应用于变电站巡检项目中，取得很好地效果。

6 测绘机器人定位与导航

测绘机器人的另一个重要功能是在成功绘制环境地图上进行测绘机器人自主定位。自主定位是衡量室内测绘机器人智能化程度的重要指标，同时也是机器人完成其他任务的前提条件。测绘机器人自主定位的实质就是利用传感器从环境中获取环境特征信息，根据环境特征同环境的对应关系进行数学运算，从而确定机器人在环境中的位置。一般解决机器人定位问题都是采用基于贝叶斯滤波器，比如卡尔曼滤波和粒子滤波。但这些定位方法都存在各自的问题，本章节首先介绍了应用较广泛的基于概率模型的自适应蒙特卡罗定位方法，该算法是在粒子滤波的基础上，改进其建议分布函数和重新采样过程来实现机器人定位的；接着提出一种新的定位方法，将深度学习理论运用到机器人定位问题上，输入激光、里程计等传感器数据，对神经网络进行训练，最终网络输出机器人的位置信息，完成机器人的定位工作。

6.1 自适应蒙特卡洛定位

6.1.1 自适应蒙特卡洛定位原理

自 1993 年粒子滤波被首次提出之后，得到各个领域的学者广泛应用，在机器人定位问题上同样取得不错的效果。自适应蒙特卡洛定位（Adaptive Monte Carlo Localization AMCL）方法同样是通过粒子滤波来进行机器人位姿估计的方法，在本文第四章对于粒子滤波计算后验概率的问题已经给出详细说明，这里就不再赘述。

自适应蒙特卡洛是由蒙特卡洛定位（Monte Carlo Localization MCL）改进得到的。它在已知地图的基础上，结合激光数据和里程计数据，使用有限的粒子数去表示机器人在环境中的可能位姿的分布状态，每一个粒子都表示一个可能的机器人状态。当接收到新的传感器数据，每个粒子将通过检查其在当前姿态下接收这种传感器读数的可能性来评估其精度。接下来，该算法会重新采样粒子以得到更精确的粒子。继续重复迭代机器人移动、获取传感器数据和重采样步骤，当定位成功时所有粒子群会聚到机器人真实姿态附近。相比较蒙特卡洛定位，自适应蒙特卡洛解决了当机器人在全局未知环境中全局定位失败问题，以及受到外界干扰而导致机器人绑架问题，一定程度上提高了定位的鲁棒性。自适应蒙特卡洛采用 KLD 采样方法动态调整粒子数目[41]，在一定程度解决了因为增加粒子数目而导致的计算量严重增加的问题，KLD 采样实质就是通过观测两个概率分布不同的 KL 距离。KL 距离概率分布 p 和概率分布 q 之间的逼近误差：

$$K(p,q)=\sum_x p(x)\log\frac{p(x)}{q(x)} \tag{6.1}$$

6.1.2 实验与分析

本文首先基于 MATLAB 平台进行 AMCL 仿真定位实验。让机器人按照设定好的路径进行运动。如图 6.1 所示，实线表示的是为机器人设定的路径，蓝色星号线表示的是根据 AMCL 算法估计得到的机器人行走路线，我们可以看出，两条轨迹基本吻合没有出现明显的偏差，AMCL 可以取得较为精确的机器人定位结果。图 6.2 是对机器人在不同时刻的定位信息分别从 x 方向、y 方向以及偏航角三个分量进行分析对比，其中蓝色实线是机器人的真实位置分量，红色虚线表示的估计出来的机器人位姿分量。求取不同分量上的误差值得到如图 6.3 所示的结果：估计的位置与真实位置之间偏差为 2~5dm。角度误差不大于 6 度。

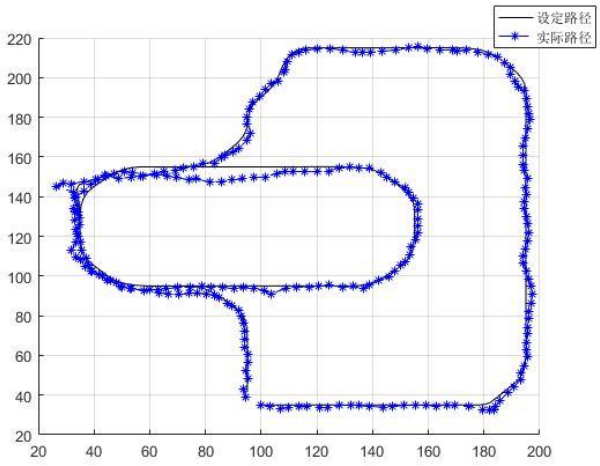


图 6.1 机器人运动路径对比

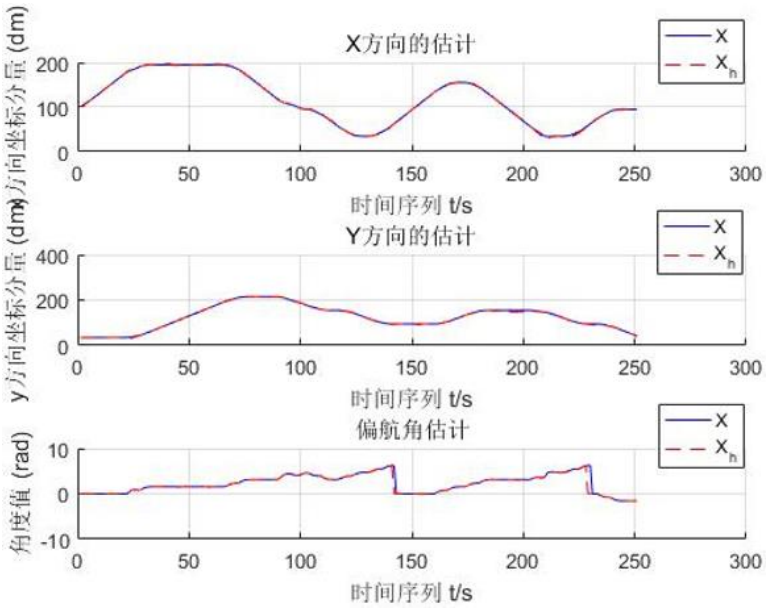


图 6.2 机器人定位结果对比

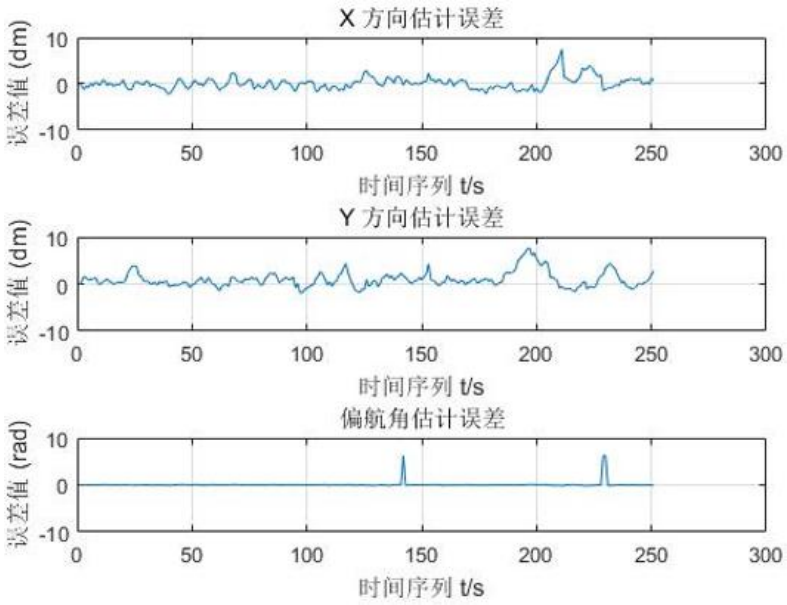


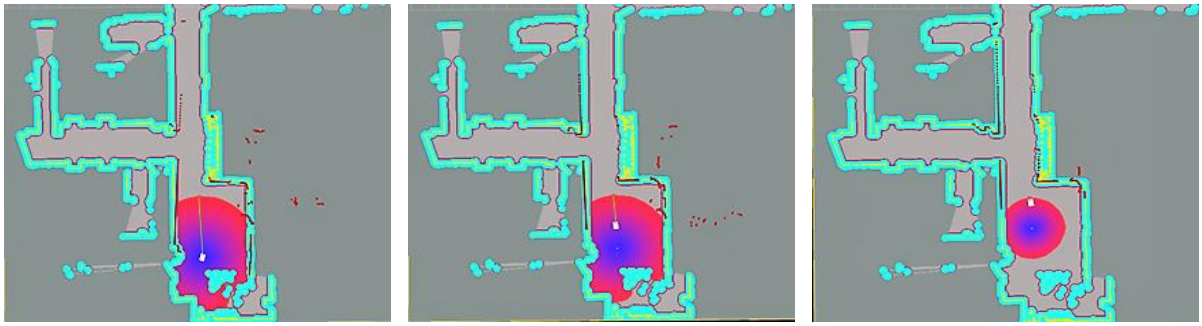
图 6.3 机器人定位误差分析

在仿真实验的基础上，基于 ROS 平台在第五章绘制的创意大厦地图上进行机器人真实场景定位导航实验。我们可以利用 ROS 中可视化工具 Rviz 观察机器人定位的过程。启动 AMCL 功能模块，接收地图、激光扫描数据和 tf 变换消息后，输出位姿估计。AMCL 根据提供的参数初始化其粒子滤波器。在 Rviz 的图形界面点击”2D Pose Estimate”，在地图中指定机器人的大致位置，我们会看到机器人会移动到了点击的图上位置，查看图上的激光点位与实际障碍物的轮廓是否吻合，如果方向相差太多，可以旋转机器人本体进行方向矫正，直至机器人位姿基本正确。图 6.4 为机器人初始化定位过程的示意图，其中红色的箭头表示的是粒子，起初粒子随机分布在机器人附近，当我们小范围前后左右移动机器人时，红色的粒子逐渐缩到一起，最终粒子全部聚集在位置就是机器人的当前位置，即表示机器人初始化定位成功了。

图 6.4 粒子初始化定位过程

把初始位置选好之后，就可以通过图形界面给机器人指定一个目标位置。在这里，会出现一个绿色的箭头，按照你的需要，在地图里面选择一个合适的位置，值得注意的是这个位置不能是机器人不能到达的位置，比如地图外部或者障碍物中。在放置点位的

时候可以通过拖动图标来设置最终导航目标的到达时机器人的朝向。完成目标定位点的设置后，将通过 global_planner 功能包找出一条由起点到目标点的合法路径，之后机器人会顺着这条路开始运动，如果在这期间环境没有什么变动的话，机器人就可以一直保持较为直接的状态到达指定的位置。如果在这期间环境发生了变化，例如中途出现动态的物体挡到了机器人原先规划好的路径。那进行局部规划的 local_planner 就会根据这中间出现的物体重新确定路线，避开这些原先并不存在的运动物体，具体的导航定位过程如图 6.5 和图 6.6 所示：



(a)起始点　　(b)向目标点运动　　(c)到达目标点

图 6.5 Rviz 可视化机器人定位导航过程

(a)起始点　　(b)向目标点运动　　(c)到达目标点

图 6.6 现实场景机器人定位导航过程

6.2 基于神经网络的定位方法

深度学习作为机器学习的分支，是一种试图使用包含复杂结构或者由多重非线性变换构成的多个处理层对数据进行高层抽象的算法。至今已有数种深度学习模型，例如深度神经网络、卷积神经网络、递归神经网络和对抗神经网络等。深度学习在计算机视觉领域应用已经十分成功，但关于机器人定位的应用却寥寥无几。本文应用一种创新性的

思想，利用深度学习中的神经网络来解决机器人定位实验。根据两种不同的定位方式结合两种不同的神经网络模型进行机器人定位实验。下面给出具体的阐述。

6.2.1 基于前向反馈网络（FFNN）的定位方法

机器人定位问题就是找到机器人每一时刻在地图上的位置。如果我们假设里程计没有噪声，那么便可以直接根据机器人运动模型计算机器人的位姿，但现实是不存在没有噪声的传感器，那么这时，就需要加入额外的传感器信息来获取环境信息完成定位。这里介绍一种定位方法称为：Naïve。这种定位方法是将所有有用的信息作为输入值输入至神经网络中，将机器人的位姿信息作为标签值，进行学习，最终网络会输出机器人的位置信息。这种方法比较像一个模拟的贝叶斯滤波器。下面网络输出输入值的描述：

输入：

x_{t-1}：上一时刻机器人的位姿

u_{t-1}：含有噪声的里程计控制信息

z_t：含噪声的激光观测值

输出：

x_t：机器人当前时刻的位姿。

本文首先采用一种比较简单的神经网络模型来进行机器人定位：前向反馈网络（FFNN）。这是一种信息从前向后传播训练（分别是输入和输出）的网络模型。图 6.7 是关于前向反馈神经网络的模型。其中，这个网络包括一个输入层，一个输出层和三个隐藏层。每个单层内神经元之间是相互独立不连接的，每个相邻层之间的神经元进行全连接。本文将尺寸为 362*1 的含噪声激光数据作为网络的输入值，将机器人的位姿信息，即一个 3*1 的向量作为标签，也就是预期的输出。从前向后进行传播训练，这个过程类似图像处理中的监督分类问题，最终输出机器人当前的位姿信息。其中三个隐含层是用来建立输入和输出之间的关系。采用这种模型的原因是它模型简单易于学习。

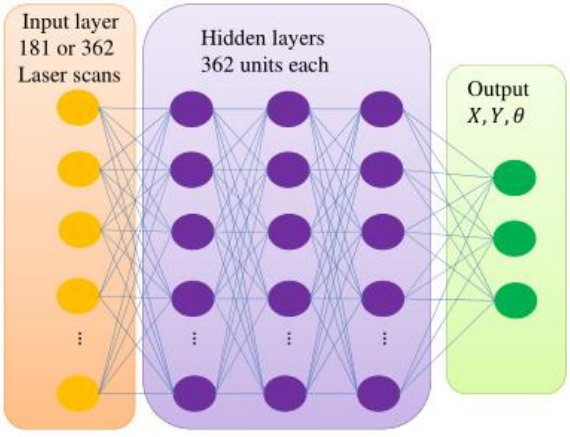


图 6.7 FFNN 模型

6.2.2 基于卷积神经网络（CNN）的定位方法

上文也提到，当里程计不存在噪声时根据航迹推算理论上是可以得到完美的机器人位置，基于这一理论，本文采用第二种 Correction 方法来解决定位问题，即将激光观测到的原始噪声观测量和由在里程计模型估计位置上模拟得到激光数据一同作为网络的输入值，这里采用一种叫 Fast Ray Cast 方法来模拟激光数据。总体思路是将空间分割成区域，在机器人所在位置发射射线并检查射线和障碍物之间的交点，从最接近射线的原点区域开始，当找到最近的交点时该点成为射线的终点。

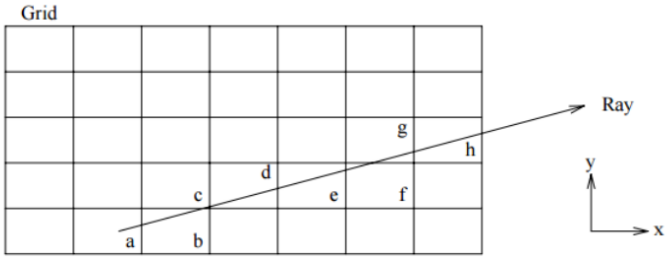


图 6.8 Fast Ray Cast 方法模拟激光

将里程计的改正数Δu作为网络的输出值，最后将改正数加入里程计模型由运动模型得到更精确的机器人的位姿。图 6.9 用一个一维模型去近似解释 Correction 方法。其中，深绿色表示机器人真实位置，浅绿色表示的是由运动模型估计出来的位置。绿色箭头表示的是机器人实际运动，红色箭头是里程计给出的运动。机器人的实际的视场范围为 FOV，在里程计估计的位姿上的模拟测距信息为 Map patch。Correction 方法就是将实际观测的激光数据和模拟的测距信息作为网络输入值，比较他们的差值，将里程计的改正数作为输出量。图 6.9 这个例子中里程计的改正数为-1。最终将改正之后的里程计数据加入运动模型完成机器人的定位。

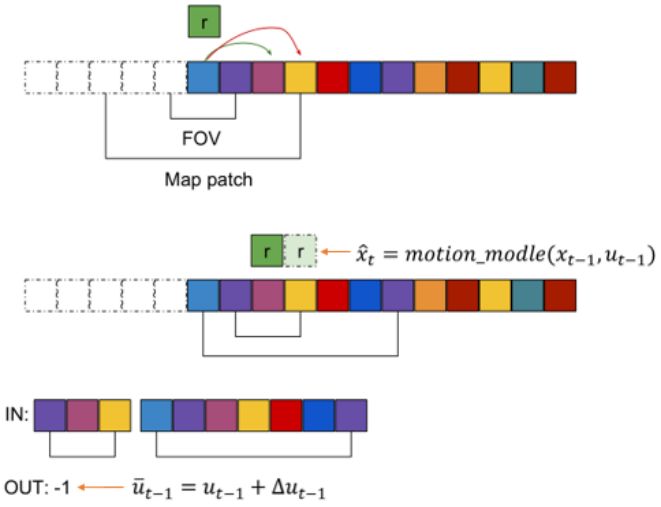


图 6.9 Correction 方法

对于 Correction 方法，本文采用卷积神经网络进行机器人定位实验。卷积神经网络因其易于提取局部变形、缩放等特征信息，因此被广泛应用于图像分类问题，后来也被部分学者用来处理音频数据等。卷积神经网络包括一个有卷积层和子采样层构成的特征抽取器。在卷积神经网络的卷积层中，一个神经元只与部分邻近神经元连接。在卷积神经网络的一个卷积层中，通常包含若干个特征平面，每个特征平面由一些矩形排列的神经元组成，同一特征平面的神经元共享权值，这里共享的权值就是卷积核。卷积核一般以随机小数矩阵的形式初始化，在网络的训练过程中卷积核将学习得到合理的权值。共享权值（卷积核）带来的直接好处是减少网络各层之间的连接，同时又降低了过拟合的风险。子采样也叫做池化（pooling），通常有均值子采样（mean pooling）和最大值子采样（max pooling）两种形式。子采样可以看作一种特殊的卷积过程。卷积和子采样大大简化了模型复杂度，减少了模型的参数。卷积神经网络由三部分构成：第一部分是输入层。第二部分由 n 个卷积层和池化层的组合组成。第三部分由一个全连结的多层感知机分类器构成。

图 6.10 是本文采用的卷积神经网络的模型结构。其中包括 5 个卷积层、3 个池化层以及 3 个全连接层。这个网络的目的是改正里程计数据。网络结构是将大小为 362*1 的来自真实位置的激光数据和来自里程计估计位姿处的模拟激光数据作为网络的输入值；用 8 个 3*1 的模板对输入值进行卷积，再进行 2*1 的最大池化处理，再继续进行两次 16 个 3*1 模板的卷积处理，然后进行一次最大池化，接着在重复两次卷积一次池化处理，之后进行三次全连接；最终得到一个 3*1 的向量，即里程计的改正值。

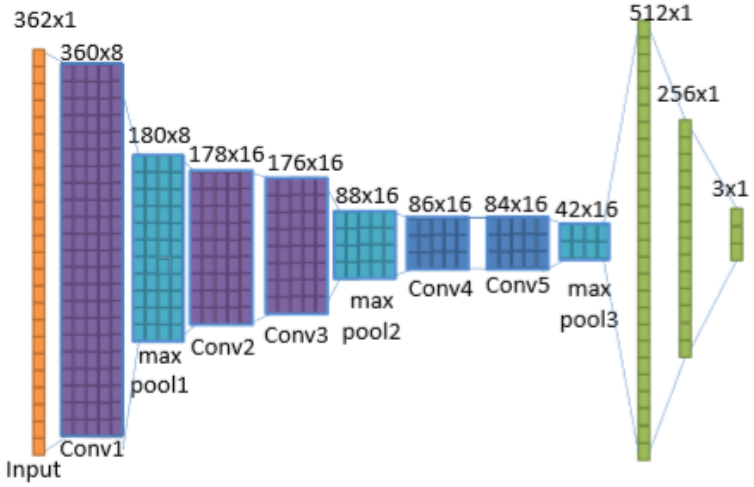


图 6.10 CNN 模型

6.3 神经网络定位实验

6.3.1 训练数据集获取

本文采用的仿真实验地图来自弗赖堡大学 Denis Tananaev 在 Gazebo 中绘制的模拟地图，地图大小为 41*52m。我们在 Rviz 中添加绘制路径的插件，在地图中绘制一条路径，为确保机器人可以安全运行，路径选择尽量位于地图中部，并且保证机器人可以到达地图的各个位置。然后编写脚本文件，使机器人在绘制的路径上产生随机点作为机器人的导航目标点。采用 ROS 中 DWA 导航包让机器人自主导航至这些目标点，其中 global cost maps 和 local cost maps 用于机器人避障。在机器人运动过程中，获取激光及里程计数据，订阅激光数据/scan、里程计数据/odom 的 topic，经过坐标转换，由里程计数据计模拟激光后，将这些数据分别输入两个网络模型进行训练，形成训练模型用于后续的定位测试。整个数据采集过程历时 6 个多小时。

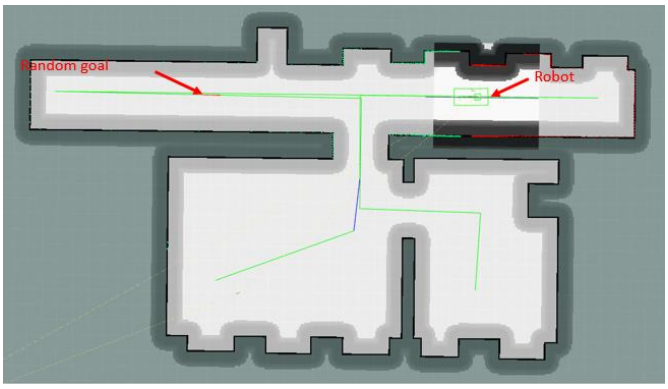


图 6.11 训练数据集获取

值得注意的是，我们采用ReLU(x)作为激活函数和 MSE 作为损失函数。因为 ReLU(Rectified Linear Unit) 相比较 sigmod 函数和 tanh 函数计算速度要快很多。

$$\mathrm{Re}\,LU(x)=\max(x,0) \tag{6.2}$$

$$MES=\frac{1}{N}\sum_{x\in X}^{N}(x_p(x)-x_g(x))^2 \tag{6.3}$$

6.3.2 基于神经网络的定位实验

通过 Rviz 图形界面给机器人指定一个目标位置，让机器人自主运动至目标点进行测试。图 6.12 分别是 FFNN 和 CNN 得到的定位实验结果。其中图 6.12(a)中，绿色方框表示的是机器人的真实位置，红色方框表示的是由 FFNN 估计的机器人位置，其中 MSE=0.0176；图 6.12(b)中，绿色方框表示的是机器人的真实位置，蓝色方框表示的是未改进里程计模型得到的机器人的位置，红色方框表示的是由 CNN 改正后的机器人位置，

MSE=0.0003。看出这两种方法均可得到精确的定位结果，且 CNN 的定位精度略高于 FFNN。

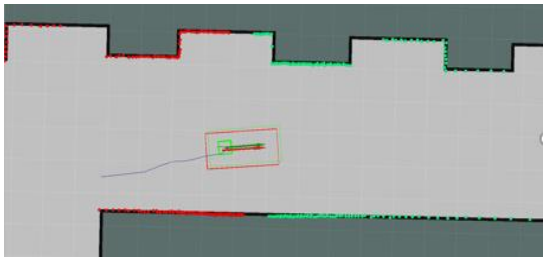


图 6.12(a) FFNN 定位结果

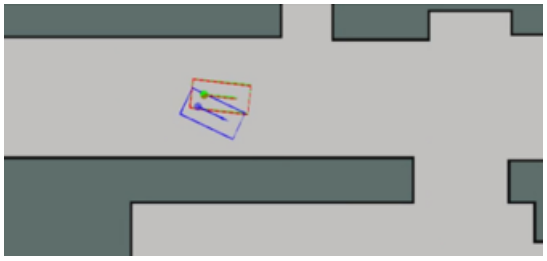
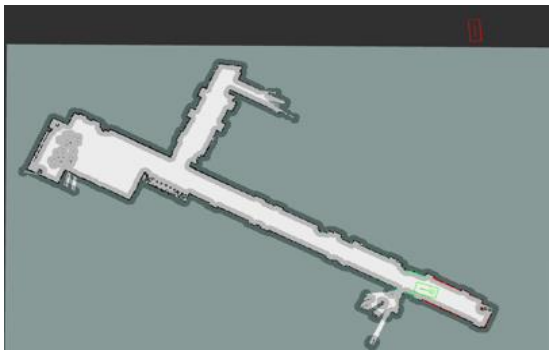


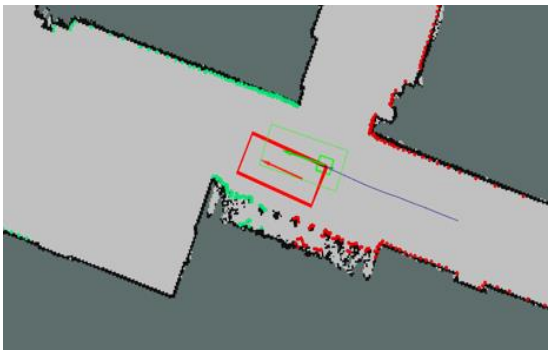
图 6.12(b) CNN 定位结果

图 6.12 基于神经网络的定位实验结果

之后将这两种神经网络模型应用于在 Gmapping 得到中部创意大厦的地图进行机器人的定位实验，图 6.13(a)为 FFNN 的定位实验结果，可以看出定位失败，这说明 FFNN 这种网络模型并不是适用于所有的地图，如果要在新的地图上定位需要重新对新地图进行训练。图 6.13(b)为 CNN 的定位结果，其中 MSE=0.395。可以看出基本可以确定机器人的位置，这说明 CNN 优于 FFNN，该方法可以应用于不同的环境地图完成机器人定位。但同样存在误差，这是因为我们的训练样本数目较少导致的。



(a) FFNN 定位结果



(b) CNN 定位结果

图 6.12 中部创意大厦地图定位实验结果

6.4 本章小结

本章主要介绍了测绘机器人自主定位导航的方法，首先介绍了传统基于概率模型的自适应蒙特卡洛定位方法，通过仿真实验和真实环境实验，分析验证机器人的定位精度。接着创新性地将深度学习神经网络应用于机器人定位问题上，通过实验分析，表明基于神经网络的定位方法有效可行。

7 总结与展望

7.1 总结

本文通过分析 SLAM 算法在测绘领域的应用前景，结合国内外关于测绘机器人和SLAM 的相关研究，给出了本文研究的现实意义。研究的主要内容涵盖了测绘机器人系统平台的搭建、测绘机器人模型的分析、主流 SLAM 算法的原理及实验测试、测绘机器人定位导航等各个方面。对于人工智能在测绘行业的应用具有一定的指导意义。

本文首先对于常见的 SLAM 地图表示方法进行详细的介绍，结合实际算法需求，采用栅格地图来创建地图；随后本文从宏观角度对 SLAM 算法的主要流程给出介绍；然后对本文设计的测绘机器人平台做出详细介绍，包括测绘机器人所使用的硬件元件、测绘机器人底盘的设计与搭建以及测绘机器人的软件平台 ROS，为测绘机器人进行室内测绘建立了良好的实验平台。

接着本文主要采用三种不同的 SLAM 算法进行测绘机器人室内测量。不同的算法在绘制 2D 室内地图表现出不一样的效果：

（1）进行 Hector SLAM 算法介绍，利用高斯牛顿方法解决激光扫描匹配问题，仅利用激光数据就可绘制出 2D 室内地图，但其对于激光传感器的性能要求较高，需要测量噪声小、更新频率高的激光雷达。实验表明：机器人在比较低的速度模式下运行，才能取得较好的制图效果，稳定性不佳。

（2）加入里程计数据，采用 Gmapping 算法进行室内 2D 地图绘制。Gmapping 算法利用粒子滤波进行制图，将里程计数据和激光数据同时用来建立建议分布函数，同时引入自适应重采样方法，减少粒子数量，降低了计算量。从实验结果表明，Gmapping 算法的制图精度高于 Hector SLAM 算法，更适合用于长廊及低特征的场景。

（3）将 Cartographer 算法应用到本文的测绘机器人绘制室内地图中。Cartographer 同时利用激光数据、里程计数据和 IMU 数据来对机器人在建图过程中的位姿进行估计和优化。实验结果表明：Cartographer 算法因其采用 SPA 和 BBS 算法，在制图精度上明显优于前两种算法，且其具有回环检测功能，使其算法鲁棒性较 Hector SLAM 算法和 Gmapping 算法都明显增强。

（4）对三种 SLAM 算法分别从仿真实验和真实环境实验进行验证和分析。得出结论：Cartographer 算法无论在制图精度、算法鲁棒性、计算速度上都由于 Hector SLAM 算法和 Gmapping 算法。这与其所利用的传感器的数量也有直接关系，这也说明多传感器的结合对于测绘机器人制图精度的影响。但现实情况往往需要控制传感器的成本，这时，可以采用 Hector SLAM 算法或 Gmapping 算法，同样可以达到较好的建图效果。从本文

的真实环境实验结果可以看出：本文所设计的测绘机器人的制图精度基本可以媲美传统测量结果，同时大大减少了人力成本。最后选择 Cartographer 应用于变电站巡检项目中。

最后，在已知地图的基础上进行机器人的定位实验，首先根据比较成熟的自适应蒙特卡洛算法完成机器人的定位实验，接着本文创新性的将深度学习神经网络应用到机器人的定位问题上，分别采用前向反馈神经网络和卷积神经网络进行实验，实验结果表明深度学习可以有效解决机器人定位问题，这也表明了深度学习在机器人定位领域存在巨大潜力。

7.2 创新点

本文对测绘机器人 SLAM 技术进行研究，主要创新点如下：

（1）本文自主设计搭建了一款用于测量室内二维地图的测绘机器人，结合 SLAM 技术，采用轮式结构，搭载激光扫描仪、轮式编码器、IMU 等传感器，实现测绘机器人实时室内地图的绘制与定位功能。相比传统测绘技术，测绘机器人更高效更便捷，且其精度满足室内服务需求。

（2）针对机器人定位问题，提出基于深度学习的定位方法。分别构建前向反馈神经网络（FFNN）模型和卷积神经网络（CNN）模型来进行机器人定位实验。实验结果表明该方法可以准确的估计机器人的位置，完成测绘机器人定位任务。

7.3 展望

测绘机器人作为一个较新的课题，涉及的学科领域很多，因此本课题是一个很有挑战性的课题。本文自主设计的测绘机器人虽然可以完成基本的测图工作，但还是仅针对室内的二维环境地图，同时其测图精度还待进一步完善。并且本文所涉及解决相关问题的方式偏向于工程应用，还未能很好地上升至理论改进的层面。在学术研究层面可能略显单薄，有待深化。

对于未来工作的展望：

（1）SLAM 算法作为一个庞大的理论体系，算法种类很多，本文仅针对最主流的算法进行介绍，在接下了的工作将继续深入研究 SLAM 算法，希望可以在其他学者的算法基础上提出自己的改进之处，提高测绘机器人的测图精度。

（2）尝试在测绘机器人上搭载摄像头，结合激光与视觉 SLAM 算法进行实验，建立三维环境模型，使得测绘机器人的可适用于室内外等多种环境。

（3）继续深入研究深度学习解决测绘机器人定位问题，尝试不同神经网络模型、不同训练方法、增加训练样本数量进行实验，提高测绘机器人定位精度。

致谢

时光匆匆，转眼间三年就过去了。回想起研究生刚入学的情景还历历在目，紧张而充实的研究生生活，让我从很多方面都成长了不少。它也必将成为我人生中最为难忘的回忆。在论文完成之际，谨向这三年来为我提供指导帮助，陪伴支持我的老师、同学、朋友和家人致以最诚挚的谢意。

重点感谢我的指导老师王小平老师和陈晓宁老师。两位老师严谨的治学态度，精湛的学术水平，和蔼的为人，做事认真负责的精神，在学习研究和生活等各个方面都值得我学习。感谢王小平老师的谆谆教诲，从我入学第一次见到王老师时，便教育我们业精于勤，同时用自己的创业治学经历告诉我们，做人要志存高远，严于律己。不能空有大抱负却不付诸行动，教育我们思考问题的思维模式是十分重要的。要将想法和实践结合，不断的磨练自己。告诉我们勤能补拙的同时要劳逸结合。在我碰到困难想放弃的时候给了我很多的鼓励，告诉我做研究便是吃得苦中苦方为人上人。同时还要感谢陈晓宁老师在我研究生期间对我的指导，他专业的学科素养在我们论文完成上提出很多建设性意见，同时陈老师无私奉献的精神总让我有高山仰止的体会，他的榜样力量使我明白学无止境，只有秉承学习进取的精神，才能更好地发现、解决问题。同样感谢教研室里其他各位老师以及教导过我的老师们。

特别感谢罗斌教授，在我为自己的研究课题上遇到难题时，给我去武汉大学学习的机会，为我提供良好的学习研究平台。在武大学习的这段时光里，我深刻认识到成长就是一条不断磨炼的道路，明白人之所以优秀，这与他背后付出的努力是密不可分的。感谢罗老师在我的毕业设计上所花费的心血。不仅为我指明了课题的研究重点，在论文结构、写作技巧等细节方面给予了很大的指导，让我的毕业设计能够由粗到精，最后装订成册。

感谢师们的师兄师弟师姐师妹们。感谢毛智辉师兄、刘洋师兄，在我实习期间对于我的帮助，从很小的软件使用，代码编程等方面均给予我很大的帮助。感谢荆超凡师弟，江丹师妹，教研室的程梦鸽、黎娟、葛艳、胡航、陶浩然同学，在我在校外学习阶段帮我处理很多学校的事务，同时能和大家同在一个教研室学习进步。很感到很幸福，祝他们学业进步。

感谢武汉大学的兄弟姐们。感谢张云师兄、刘军师兄、尹露师兄、赵青师兄、王伟师兄、姜尚杰、陈俊伟、陈警师弟、常云鹏、陈勇、邹建成、王晨捷、胡楚笛师弟、赵茜师姐、周媛媛、张婧、彭忻怡师妹。我们同在的小组就像一个大家庭一般，在武汉大学这段时光里大家互相帮助，共同学习进步，对我未来的人生道路具有很大的启示，我很珍惜与大家一起的时光。

感谢我的室友刘畅、王慧珍、薛玫娇、许雪敏同学，她们待人平和，为人乐观直率，让我十分享受在宿舍的休息时间。祝他们工作顺利，事业有成。

最后感谢我的家人，赋予我生命，抚育我成长，培养我，为了让我接受更好的教育，默默付出的一切。这么多年来对我无微不至地照顾，他们是我最坚强的后盾。

我的毕业设计能够顺利完成，离不开各位的支持。衷心地感谢你们。

参考文献

[1] 彭晟远. 基于激光测距仪的室内机器人 SLAM 研究:[毕业论文]. 武汉: 武汉科技大学, 2012.

[2] 周武等. 基于全局观测地图模型的 SLAM 研究[J]. 机器人,2010(05):647-654.

[3] 赵季. 基于激光测距仪的同步定位与建图研究[毕业论文]. 武汉: 武汉科技大学,2008.

[4] 应鹏. 室内机器人平台设计及 SLAM 研究[毕业论文]. 浙江: 浙江大学,2010.

[5] 廖方波. 基于传感器融合的移动机器人定位及地图构建技术的研究[毕业论文]. 北京: 北京交通大学,2014.

[6] 余建伟等. 基于 SLAM 的室内移动测量系统及其应用[J]. 测绘通报,2016(6):146-147.

[7] 李昀泽. 基于激光雷达的室内机器人 SLAM 研究[毕业论文]. 广州: 华南理工大学,2016.

[8] Garulli A, Giannitrapani A, Rossi A, et al. Mobile Robot SLAM for Line-Based Environment Representation[C]. Proceedings of the 44th IEEE Conference on Decision and Control, 2005.

[9] Dieter Fox, Wolfram Burgard, Sebastian Thrun. Markov Localization for Mobile Robots in Dynamic Environments [J]. Journal of Artificial Intelligence Research, 1999,11: 391-427.

[10] Sebastian Thrun. Probabilistic Algorithms in Robotics [J]. AI Magazine, 2000, 21(4):3-109.

[11] Michael Montemerlo, Sebastian Thrun, Daphne Koller, Ben Wegbreit. FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges.2003.

[12] Austin Eliazar. DP-SLAM [D]. Durham: Duke University, 2005.

[13]Savaria D.T, Balasubramanian R. V-SLAM: Vision-Based Simultaneous Localization and Map Building for an Autonomous Mobile Robot[A]. 2010 IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)[C]. Salt Lake City, UT: IEEE,2010.

[14] Ma Yan, Ju Hehua, Cui Pingyuan. Research on Localization and Mapping for Lunar Rover Based on RBPF-SLAM[A]. International Conference on Intelligent Human-Machine Systems and Cybernetics,2009. IHMSC'09[C]. Hangzhou,Zhejiang: IEEE, 2009:306-311

[15] 郭剑辉. 移动机器人同时定位与地图创建方法研究[毕业论文]. 南京: 南京理工大学,2008.

[16] 祝继华等. 基于 ICP 算法和粒子滤波的未知环境地图创建[J]. 自动化学报,2009(08):1107-1113.

[17] 王彭林等. 基于 SIFT 算法的移动机器人同时定位与地图创建[J]. 宁波大学学报(理工版),2008(01):68-71.

[18] 袁夏等. 一种基于点—面匹配的 3D-SLAM 方法[J]. 机器人,2011(02):215-221.

[19] 蔡则苏等. 基于激光测距传感器的家庭机器人导航仿真[J]. 哈尔滨工业大学学报,2004(07):902-904.

[20] 柯江胜等. 一种改进的鲁棒 SLAM 算法[J]. 计算机与现代化,2016(02):104-106.

[21] 梁潇. 基于激光与单目视觉融合的机器人室内定位与制图研究[毕业论文]. 哈尔滨: 哈尔滨工业大学,2016.

[22] 庄严等. 移动机器人基于激光测距和单目视觉的室内同时定位和地图构建[J]. 自动化学报,2005(06):113-121.

[23] 张毅等. 一种融合激光和深度视觉传感器的 SLAM 地图创建方法[J]. 计算机应用研究,2016(10):2970-2972.

[24] 曲丽萍. 移动机器人同步定位与地图构建关键技术的研究[毕业论文]. 哈尔滨: 哈尔滨工程大学,2013.

[25] 高丽华. 移动机器人在线地图构建与定位技术研究[毕业论文]. 南京: 东南大学,2007.

[26] 刘芳. 动态未知环境下移动机器人同时定位与地图创建[毕业论文]. 哈尔滨: 哈尔滨工业大学,2015.

[27] Hugh Durrant-whyte, Tim Bailey. Simultaneous Localization and Mapping: Part I [J] IEEE Robotics & Automation Magazine,2006.

[28] 赵越. 实时增强现实中运动目标及场景的跟踪注册关键问题研究[毕业论文]. 沈阳: 东北大学,2015.

[29] 夏汉均. 基于 ROS 的机器人即时定位与地图构建技术的研究[毕业论文]. 沈阳: 东北大学,2013.

[30] 张建伟等. 开源机器人操作系统 ROS[M].科学出版社,2012.

[31] 王硕. 基于无线传感器网络的移动机器人导航研究[毕业论文]. 沈阳: 东北大学,2014.

[32] Stefan Kohlbrecher ,Oskar von Stryk, Johannes Meyer, Uwe Klingauf. A Flexible and Scalable SLAM System with Full 3D Motion Estimation [J]. IEEE,2011.

[33] 沈俊. 基于 ROS 的自主移动机器人系统设计与实现[D]. 2015.

[34] Arnaud Doucet. A Tutorial on Particle Filtering and Smoothing: Fifteen years later [J].2012

[35] Giorgio Grisetti, Cyrill Stachniss, Wolfram Burgard. Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling .

[36] Wolfgang Hess, Damon Kohler, Holger Rapp, Daniel Andor. Real-Time Loop Closure in

2D LIDAR SLAM [J]. google,2016.

[37] Giorgio Grisetti, Rainer Kummerle, Wolfram Burgard, Regis Vincent. Efficient Sparse Pose Adjustment for 2D mapping [J]. IEEE RSJ International Conference on Intelligent Robots and Systems,2010.

[38] J. Clausen, Branch and bound algorithms-principles and examples, Department of Computer Science, University of Copenhagen, pp. 1–30, 1999.

[39] Joao Machado Santos, David Portugal, Rui P. Rocha. An Evaluation of 2D SLAM Techniques Available in Robot Operating System [J]. IEEE,2013.

[40] 黄鹤等. 3D SLAM 激光影像背包测绘机器人精度验证[J]. 测绘通报,2016(12):68-73.

[41] S. Thrun, W. Burgard and D. Fox, Probabilistic Robotics. Cambridge, MA: MIT Press, 2005.

攻读硕士学位期间论文发表与科研情况

1. 发表论文

[1] 基于 Visual Studio 的图像特征点提取研究与实现[A]. 测绘与空间地理信息, 2017, 40(12):193-199.（第一作者）。