

硕士学位论文

基于 SLAM 的室内移动机器人导航技术研究

**RESEARCH ON NAVIGATION TECHNOLOGY OF
INDOOR MOBILE ROBOT BASED ON SLAM**

李想

哈尔滨工业大学

2018 年 6 月

国内图书分类号：TP242.6

学校代码：10213

国际图书分类号：621

密级：公开

工程硕士学位论文

基于 SLAM 的室内移动机器人导航技术研究

硕 士 研 究 生：李想

导 师：白清顺 副教授

申 请 学 位：工程硕士

学 科：机械工程

所 在 单 位：机电工程学院

答 辩 日 期：2018 年 6 月

授 予 学 位 单 位：哈尔滨工业大学

Classified Index: TP242.6

U.D.C: 621

Dissertation for the Master Degree in Engineering

RESEARCH ON NAVIGATION TECHNOLOGY OF INDOOR MOBILE ROBOT BASED ON SLAM

Candidate:	Li Xiang
Supervisor:	Associate Prof. Bai Qingshun
Academic Degree Applied for:	Master of Engineering
Speciality:	Mechanical Engineering
Affiliation:	School of Mechatronics Engineering
Date of Defence:	June, 2018
Degree-Conferring-Institution:	Harbin Institute of Technology

摘 要

移动机器人导航技术处于当前科学研究领域的前沿，象征着高新技术的发展方向，基于同步定位与地图构建理论(Simultaneous Localization and Mapping, SLAM)的导航技术更是机器人领域的研究热点。为了让室内导航技术更加自主化与智能化，本文将对基于 SLAM 的室内导航技术进行研究，并对其中的 SLAM、定位以及路径规划三个问题进行深入研究。由于本文采用模块化设计方式，导航系统的可移植性强，可将其应用在服务机器人、工业运输机器人等移动机器人上。本文的完成有望对国内自主导航领域的发展起到促进作用。

首先，针对传统的基于滤波理论的 SLAM 算法所构建的地图质量不佳的问题，应用基于图优化理论的 SLAM 算法进行机器人的同步定位与地图构建；并对其中的数据关联、闭环检测、后端优化等关键步骤进行深入研究；设计了一种主动 SLAM 技术，应用基于边界的主动探索模型将 SLAM 与路径规划结合在一起，使得机器人能够自主地进行未知环境的地图构建。

其次，研究了基本的蒙特卡洛定位算法及自适应蒙特卡洛定位算法，并通过仿真实验对定位算法的收敛过程进行分析；对 A*全局路径规划算法进行研究，并通过仿真实验与其它算法进行对比分析，验证了其优越性；采用了一种新颖的局部路径规划算法，以实现实时地轨迹优化，并对其进行仿真实验以验证其有效性；为了实现室内复杂环境下对小尺寸物体的避障，采用基于三维点云的避障策略，以实现室内复杂环境下的三维避障。

最后，通过实验对各个理论进行验证与分析。主动 SLAM 实验证明了主动 SLAM 技术的可行性；通过两种 SLAM 算法所构建的地图间的对比分析，以证明本文所用 SLAM 算法的优越性；通过地图尺寸与实际环境尺寸之间的对比分析，来验证地图的精确程度。通过自主导航实验验证了定位算法与路径规划算法的可靠性，并进行了导航精度的分析。静态避障与动态避障实验验证了导航系统的三维避障能力及动态避障能力，证明路径规划算法具有良好的实时性与鲁棒性。

本文通过对 SLAM、定位以及路径规划三个导航关键问题的研究，最终得到一套稳定的、自主能力较高的、导航精度较高的导航系统。

关键词：移动机器人；室内自主导航；SLAM；定位；路径规划

Abstract

Navigating autonomously of mobile robots is a symbol of the development direction of high and new technology, which is in the forefront of the field of scientific research. And, navigating autonomously based on Simultaneous Localization and Mapping (SLAM) is a problem that has attracted a great deal of interest in mobile robotics. In order to make the indoor mobile robot more autonomous and intelligent, this topic will study the indoor navigating autonomously based on SLAM, and further study the three problems of SLAM, localization and path planning. Due to its modular design, it has many different applications such as service robots and transport robot. The completion of this thesis will promote the development of autonomous navigation in China.

Firstly, in view of the shortcomings of the traditional filter-based SLAM algorithm, a SLAM algorithm based on graph optimization is studied in this thesis, and utilizes it to estimate the pose of the robot and the location of the features. Key steps in SLAM such as data association, loopback detection, and backend optimization are analyzed. In this paper, an active SLAM technology is designed, and the active exploration method based on the boundary is used to combine the SLAM with the path planning, so that the robot can build the map of the unknown environment autonomously, so that the robot can be truly autonomous.

In addition, the basic Monte Carlo localization algorithm and adaptive Monte Carlo localization algorithm are studied. And the convergence process of localization algorithm is analyzed through simulation experiments. The global planner algorithm A* is studied and compared with other algorithms through simulation experiments to verify its superiority. A novel local path planner algorithm is applied to achieve real-time trajectory optimization, and simulation experiments are performed to verify its effectiveness. In order to navigate safely in indoor environment, the thesis presents a method for achieving 3D obstacle avoidance in indoor complex environments, which is based on the three dimensional point cloud.

Finally, all the algorithms presented in this thesis have been tested and verified in real world experiments. Active SLAM experiments prove the feasibility of active SLAM technology. By comparing and analyzing the maps obtained by the two SLAM methods, the superiority of the SLAM algorithm used in this paper is explained. A comparison of the key dimensions between the map and the actual environment verifies the accuracy of the map. Autonomous navigation experiments verify the performance of localization algorithm and path planning algorithm, and the navigation accuracy is analyzed. Through the obstacle avoidance experiment, the navigation system's 3D obstacle avoidance ability

and dynamic obstacle avoidance ability are tested. It is proved that the real-time and robust of the path planning algorithm is good.

Through the research on SLAM, localization, and path planning, this thesis has finally obtained a stable navigation system with high autonomous capabilities and high navigation accuracy.

Key words: mobile robot, indoor autonomous navigation, SLAM, localization, path planning

目 录

摘 要	I
Abstract	II
第 1 章 绪论	1
1.1 研究背景及意义	1
1.1.1 课题来源	1
1.1.2 研究背景及意义	1
1.2 自主导航技术综述	2
1.2.1 SLAM 理论	3
1.2.2 定位技术	4
1.2.3 路径规划技术	5
1.3 自主导航关键技术的研究现状	6
1.3.1 SLAM 算法的国内外研究现状	6
1.3.2 基于 SLAM 的导航技术国内外研究现状	7
1.4 主要研究内容	9
第 2 章 移动机器人的系统模型及架构	11
2.1 引言	11
2.2 传感器选型及实验平台搭建	11
2.2.1 传感器分析	11
2.2.2 实验平台搭建	13
2.3 机器人运动模型	14
2.4 传感器观测模型	15
2.5 环境地图模型	15
2.6 导航系统整体架构	16
2.6.1 机器人操作系统 ROS	16
2.6.2 Navigation 导航软件包	17
2.6.3 室内导航系统整体结构	18
2.7 本章小结	20
第 3 章 基于图优化理论的 SLAM 算法研究	21
3.1 引言	21
3.2 基于图优化理论的 SLAM 算法研究	21

3.2.1 激光扫描匹配	23
3.2.2 闭环检测	24
3.2.3 位姿图优化	26
3.3 主动 SLAM 技术	28
3.3.1 基于边界的主动探索	28
3.3.2 主动 SLAM 技术架构	30
3.4 本章小结	31
第 4 章 移动机器人的定位与路径规划算法研究	32
4.1 引言	32
4.2 移动机器人的定位算法研究	32
4.2.1 蒙特卡洛定位算法研究	32
4.2.2 自适应蒙特卡洛定位算法研究	34
4.2.3 定位算法仿真实验	35
4.3 全局路径规划算法研究	37
4.3.1 全局路径规划算法	37
4.3.2 全局路径规划算法仿真实验	38
4.4 局部路径规划算法研究	39
4.4.1 局部路径规划算法	39
4.4.2 基于点云的避障策略	41
4.4.3 路径规划模块仿真实验	42
4.5 本章小结	43
第 5 章 移动机器人导航系统的实验验证	44
5.1 引言	44
5.2 室内环境的地图构建实验	44
5.2.1 主动 SLAM 实验	44
5.2.2 SLAM 算法对比实验	45
5.2.3 地图精度分析	47
5.3 移动机器人的定位及自主导航实验	48
5.3.1 已知地图的定位及自主导航实验	48
5.3.2 移动机器人的导航精度分析	49
5.4 移动机器人的避障实验	51
5.4.1 移动机器人的静态避障实验	51
5.4.2 移动机器人的动态避障实验	53

5.5 本章小结	54
结 论	55
参考文献	56
哈尔滨工业大学学位论文原创性声明和使用权限	60
致 谢	61

第1章 绪论

1.1 研究背景及意义

1.1.1 课题来源

本文为芜湖哈特机器人产业技术研究院有限公司的研发项目“基于激光雷达的室内 AGV 自主导航系统研究”的子课题。

1.1.2 研究背景及意义

1956 年，乔治·德沃尔发明创造了世界上第一台电子可编程的工业机器人，并且于 1961 年发表了该项专利。之后随着世界上第一家机器人制造工厂 Unimation 公司的成立，机器人技术开始引起国内外研究学者的广泛关注，从而带动了机器人相关领域的发展，包括控制理论、材料学、仿生学、机电一体化、机器人学以及人工智能等领域^[1]。

移动机器人导航技术处于当前科学研究领域的前沿，象征着高新技术的发展方向。移动机器人导航根据决策方式可以分为遥控、半自主决策、自主决策三种类型。在各种复杂环境下依然能够进行自主定位、导航的移动机器人被称为自主机器人^[2]。机器人的自主导航即为移动机器人能够自主地到达人为指定的目标点。而机器人想要完成导航任务，首要条件就是能够感知周围环境，只有这样机器人才能知道“我在哪”，“我去哪里”^[3]。并且，机器人想要安全的抵达目标点还需要拥有感知障碍物与躲避障碍物的能力。因此，研究移动机器人的导航技术是一个非常具有研究价值及实际意义的研究方向。

本文基于这一课题背景进行室内移动机器人导航技术的研究，利用 2D 激光测距仪与 Kinect 相机进行环境的检测，实现机器人自主构建未知环境的地图，并在完成自主导航任务时能够自主躲避障碍物。机器人的自主导航技术具有可移植性强的特点，可将其应用于服务机器人点餐、送快递的运输机器人、工厂里的巡检机器人、在工厂搬运货物的移动机器人等。本文的完成将实现同步定位与地图构建理论在移动机器人导航领域的应用，有望对自主导航领域的发展起到促进作用。本文的研究领域与国家发布的于 2020 年实现有条件的高度自动驾驶的政策十分迎合，对自动驾驶的导航技术有望起到促进作用。

1.2 自主导航技术综述

移动机器人实现自主导航需要解决 Durrant-Whyte HF 与 Leonard JJ 提出的三个基本问题：机器人的定位、构建地图与路径规划^[4]。定位是以一种准确的度量描述机器人在当前环境中的位置和姿态，地图构建是将通过各个传感器感知到的局部环境组合成为一个单一的、一致的地图，路径规划则是在已知地图中为机器人计算出一条能够到达目标点的最优路径^[5]。

最初，机器人的地图构建与定位的研究是分开进行的。后来，研究人员发现这两个问题无法独立解决。在机器人构建地图之前，它需要知道自己是在哪个位置进行环境观察的。同时，在没有地图的情况下，机器人很难确定自己的当前位置。因此，为了解决这个问题国外学者提出了同步定位与地图构建理论（SLAM）。SLAM 一直被称为“鸡和蛋”问题：机器人自身位姿需要良好的地图，而地图构建的首要条件是机器人知道自身的位姿，才能计算出障碍物相对于自身的相对距离。图 1-1 展示了导航的三个关键步骤之间的关系，定位与地图构建融合形成 SLAM 理论，SLAM 理论与路径规划融合形成主动 SLAM 理论。

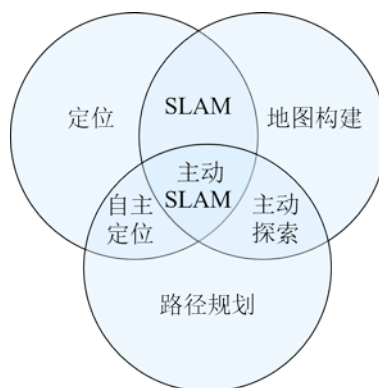


图1-1 导航的三个关键步骤

导航技术自从 20 世纪中叶被一些学者提出来，到目前为止已有 60 多年的历史。根据导航原理的不同，常见的导航方式主要有：惯性导航、磁导航、距离传感器导航、视觉导航及 GPS 导航等^[6]。

惯性导航是利用陀螺仪计算机器人转动角度，利用加速度计统计机器人的位置变化，从而实现机器人的自身定位与导航^[7]。这种导航方式很容易实现，但是由于加速度计等传感器只在短时间内精准，其误差会随着移动距离的增加显著增大，因此这种导航方式精度很低，难以用于工业生产。

磁导航是比较传统和实用的导航方式，目前国内很多自动导引运输车（Automated Guided Vehicle, AGV）都应用这种方式。它是通过在地上铺设磁带，

机器人始终跟随着磁带运动,并通过传感器检测偏离磁带的距离实时校正位姿,从而完成导航任务^[8]。这种导航方式需要改变周围环境,要求地面较平整,磁带一经铺下很难改变路径,因此应用范围有限。

视觉导航是利用安装在机器人上的摄像头感知周围环境,通过对周围环境进行特征提取、检测并识别从而实现避障与导航^[9]。这种导航方式可以获取到十分丰富的信息,但是也加大了计算复杂度,降低了实时性。而且摄像头对光线十分敏感,只能用在光线充足的环境中。目前这种技术一直处于研究阶段,真正应用的还比较少。

基于距离传感器的导航方式应用非常广泛,这种导航方式实现简单,实时性好。常用的测距传感器有声呐传感器、超声波传感器、红外传感器以及激光雷达。激光导航是采用测距定位技术来估计周围环境与机器人当前位姿从而实现导航过程^[10]。激光雷达精度非常高,能够以较高的频率提供大量的距离信息,而且不需要改变周围环境,路径灵活可变。在十多年前,国际上已经成功将基于激光雷达的导航技术应用于工业生产领域。然而,国内还没有获得成熟的应用,因此研究激光导航技术具有很大的研究价值。

1.2.1 SLAM 理论

同步定位与地图构建理论(SLAM)可以描述为:机器人能否在未知环境下在确定自身位姿的同时根据这个位姿逐步建立一个相符的环境地图;SLAM 问题的解决方案被视为移动机器人界的“圣杯”,因为它将提供使机器人真正自主的手段,已经成为了移动机器人研究领域的热点^[11]。

导航过程中的很多场景是不可能提前拥有环境地图的,即便室内环境具有建筑的平面图纸,但由于各种摆设与家具的存在使得地图与建筑图纸相差很大。能够建立各个场景的地图使得机器人具有环境适应性,因此地图构建是自主机器人的核心之一。构建一个准确的地图受很多因素的制约,主要表现在以下几个方面:

(1) 定位与地图构建要同时进行:第一,定位问题。当机器人在环境中移动时,它的里程计会逐渐积累误差,使其不确定性逐渐增加,慢慢的会丢失掉它自身的位置。第二,建图问题。在知道机器人姿态的情况下构建地图十分容易。然而,在没有初始地图和精确姿态信息的情况下,机器人要同时做到这两点:构建地图及相对于该地图确定自身位姿,是十分困难的。

(2) 维度高:由于地图是在连续的空间上定义的,因此地图的空间有无限多个维度。即使在离散逼近的情况下,如网格逼近,地图也可以很容易的拥有 10^5 或更多的变量。这个超高维度的空间使得计算地图上的完整后验概率变得十分具有

挑战性。

(3) 大范围建图：相对于机器人的感知范围而言，环境范围越大，构建地图就越困难。

(4) 感知和驱动的噪声：噪声越大，误差越大，建图就越困难。如果机器人传感器和执行器没有噪声，那么建图将是一个十分简单的问题。

(5) 感知模糊：由于机器人是通过传感器的数据感知世界，在某些非常相似的地方传感器的数据可能非常接近，机器人就很难判断自己是否在运动。

(6) 闭环检测：绘制带有闭环的环境地图十分困难。当机器人走完一个闭环返回到起点时，由于里程计的误差，它构建的地图很可能不闭合，从而出现闭环误差导致地图不一致。

1.2.2 定位技术

定位是确定机器人相对于给定环境地图的姿态的问题，它通常被称为位置估计。定位技术是机器人感知中最基本的问题，几乎所有的机器人任务都需要知道自身的位姿^[12]。

移动机器人定位可以看作一个坐标变换的问题。地图是在全局坐标系中描述，而该坐标系与机器人自身坐标系不相关。因此，定位问题可以描述为建立全局坐标系与机器人自身坐标系之间对应关系的过程。利用这种坐标变换，机器人就可以在自身坐标系下描述周围环境的位置，这是完成导航任务的必要前提。

由于用于测量的传感器不可能没有噪声，机器人的位姿必须要从这些充斥着噪声的数据中推测出来。而且单一的传感器通常不足以确定其姿态（例如，激光雷达很难识别走廊环境）。

文献[12]根据初始位姿是否已知与周围环境性质将定位问题分为四个维度，分别如下：

(1) 局部定位与全局定位：初始位姿已知的定位问题被描述为“位姿跟踪”。位姿跟踪的不确定性只是在运动过程中产生的，局限于靠近机器人真实姿态的区域。初始位姿未知的问题被描述为“全局定位”。机器人不仅要进行位姿跟踪，还要确定相对于地图的初始位姿。机器人在运动过程中可能会遇到“绑架问题”，即突然被绑架并放到了其它位置，机器人很可能认为自己处在一个错误的位置上。绑架问题常常被用来测量定位算法在全局定位失败时的恢复能力^[12]。

(2) 静态环境与动态环境下的定位：静态环境，即周围环境中除了机器人外不存在其它的动态物体。动态环境下除了机器人自身在运动之外还存在许多其它的变化，如人，可移动家具或门等，动态环境下的定位更为复杂。

(3) 主动与被动定位：被动定位问题中，机器人的运动是被其它模块控制，定位模块仅仅是观察周围环境以确定自身位姿。主动定位中，定位模块将通过控制机器人运动的方式来主动地最大限度地减少定位误差与定位过程所需的成本。

(4) 单一或多个机器人定位：单个机器人的定位是最普遍的，通过机器人获得的传感器数据即可完成定位过程。多个机器人的定位不仅仅包括单个机器人的定位，还需要考虑多个机器人之间的交流以及相互之间的定位。

这四个维度描述了机器人定位问题的四个最重要的特征，当然还存在一些其它特性导致定位问题不是那么简单，如对称环境下的定位等。

定位的方式也有很多种类：GPS、航位推算法^[13]、基于测距传感器的定位（如超声波、声呐、激光雷达等）、基于视觉的定位^[14]、基于射频技术^[15]的定位（如RFID、蓝牙与WiFi等）。

1.2.3 路径规划技术

路径规划就是计算出一条符合机器人运动学约束的最佳无碰撞轨迹^[16]。然而，周围环境的地图很可能是不准确或者不完整的，如果机器人沿着这样的路径盲目移动，仍然会发生碰撞。因此，将路径规划分为基于已知地图的全局路径规划与基于传感器数据的实时局部路径规划，可以实现较为安全的导航^[17]。全局路径规划根据已知地图提供一个全局的可以到达目标点的路径。局部路径规划根据传感器信息进行实时的避障与路径实时规划，控制着机器人沿着全局路径规划移动^[18]。

全局路径规划的路径搜索方法主要有 Dijkstra 算法、可视图法、A*算法、D*算法^[19]等；局部路径规划主要有动态窗法（Dynamical Window Approach, DWA）、人工势场法、蚁群算法、粒子群算法^[20]等。

人工势场法是将机器人与障碍物上设置一个人造的引力场，通过各个引力见得吸引与排斥达到规划路径的效果。该算法具有数学描述简单、美观、生成的路径平滑且安全，但是存在靠近障碍物的目标点时震荡，在狭窄通道中摆动等缺点。

可视图法是将机器人与各个障碍物的周围顶点用直线连接起来，保证连线不穿过障碍物，通过找到这些直线的距离和的最小值来找到路径，但这种方法缺乏灵活性，实时性较差^[21]。

Dijkstra 算法通过以起点为中心逐层的向外扩展，直至找到目标点。这种算法能够找到最短路径，但是其搜索空间大，搜索时间较长。

A*算法采用启发式评价函数实现有导向性搜索，算法效率高，可以得到两点之间的最短路径。但当问题规模较大时，搜索的时间和空间复杂度较高。

DWA 算法将速度和加速度的限制设置到搜索窗口中，通过对各个速度与加速

度评分从而得到安全可靠的路径。但是这种算法存在局部最小问题。

除了这些传统算法外,很多新颖的智能算法也被引入到机器人领域中,如蚁群算法、神经网络等。

1.3 自主导航关键技术的研究现状

1.3.1 SLAM 算法的国内外研究现状

SLAM 问题起源于 1986 年在加利福尼亚州旧金山举行的 IEEE 机器人与自动化大会 (Robotics and Automation Conference) 上。

1987 年, Self 等学者最先发现机器人位姿估计与地标的位置估计是高度相关的,由此他们提出用状态向量同时表示路标的位置和机器人位姿,用该向量的协方差矩阵表示位姿与路标位置之间的关系,即为扩展卡尔曼滤波 (Extended Kalman Filter, EKF) [22]。

扩展卡尔曼滤波 (EKF) 是卡尔曼滤波器进行线性化的均值和协方差估计的非线性版本 [23]。由于 EKF 算法在减小误差上的优势,一经发表就成为当时最流行的 SLAM 方法 [24]。

针对 EKF 在非线性系统下的精度低的问题, Julier 等提出了一种改进的无迹卡尔曼滤波 (Unscented Kalman Filter, UKF), 该方法无需将方程线性化,而是使用无损变换将非线性系统做变换,使其适用于标准卡尔曼滤波体系 [25]。

相对于要求强线性,高斯噪声的卡尔曼滤波器及其改进算法,粒子滤波器 (Particle Filter, PF) 不受线性高斯噪声的限制,可以应用在非高斯非线性的系统下。由 Doucet 等人提出的 Rao-Blackwellized Particle Filter (RBPF) 算法是目前应用最广泛的粒子滤波算法 [26]。

基于 RBPF 算法, Michael Montemerlo 提出了 FastSLAM, 一种递归地估计机器人姿态和地标位置的完全后验分布的算法 [27], 并成功应用于机器人上。

目前, SLAM 的解决方法大致可以分为两类,一类是基于滤波理论的 SLAM 算法,另一类是基于图优化理论的 SLAM 算法。基于滤波理论 SLAM,如前所述的 EKF-SLAM 算法,系统的状态变量仅包含当前的机器人位姿和地图,并通过新的测量数据加入和增强状态估计。基于图优化理论的 SLAM 算法通过机器人从最开始所有的测量数据来对机器人所有的轨迹进行估计,应用后端优化使得误差最小化,被称为完全 SLAM。

基于图的 SLAM 算法由 Lu 与 Milios 在 1997 年提出。但是由于解决误差最小化问题的计算复杂度太高,使得这种方法在当时没有广泛应用。最近几年,随着多

内外学者对 SLAM 问题的结构的深入与稀疏线性代数领域的进展,产生了解决优化问题的有效方法,使得这种方法再次出现研究者的眼中。由于其在计算速度与准确性方面的巨大优势,基于图优化理论的 SLAM 已经成为最先进的理论框架。

国内在 SLAM 领域发展比较晚,但在这十几年内也取得了很多研究成果。

大连理工大学的庄严对多传感器数据融合的 SLAM 进行了研究,提出了在多传感器数据在特征层次的融合问题上新的解决方法并通过了实验验证^[28]。

国防科技大学的季秀才对 SLAM 过程中的数据关联问题进行了深入研究,提出了关于数据关联的新算法^[29]。

中南大学的陈白帆对动态环境下的移动机器人 SLAM 进行了研究,对 SLAM 解决方法、数据关联和动态目标的处理三个方面进行了改进,最后建立和实现了 SLAMiDE 系统并测试成功^[24]。

华南理工大学的李昀泽使用激光雷达和里程计融合,提出了改进的 RBPF-SLAM 算法,使得 SLAM 算法的性能更好,构建地图的质量更佳^[30]。

东南大学的韩明瑞提出了一种 Graph-based SLAM 框架下基于 3D 激光 LIDAR 扫描匹配的室外 SLAM 统一处理方法^[31]。

哈尔滨工业大学的邹谦通过将二维码位置信息加入到基于图优化的激光 SLAM 过程中,成功的解决了狭长走廊等单调场景中应用激光雷达难以检测闭环的问题^[32]。

多年来,国内外学者对 SLAM 问题做了大量的研究工作,国外的 SLAM 技术已经很成熟并应用到很多领域,并且在陆地、空中、水下、室内、室外等方向也有了各自的 SLAM 框架,例如 José-Fernán 等人就水下 SLAM 问题的准确性和鲁棒性提出了增广扩展卡尔曼滤波方法^[33]。

国内由于起步比较晚,在 SLAM 领域上一直处于研究阶段,还没有广泛的应用。目前,应用 SLAM 构建地图的精度达不到工业级别的要求。而且,建图的方式大多数为通过人为遥控的方式进行建图,这使得机器人不可能应用在环境未知的场景中,这使得拥有在未知环境中自主构建地图的能力十分重要。

1.3.2 基于 SLAM 的导航技术国内外研究现状

国外在移动机器人领域上的研究已经进行了很多年,他们的应用从最开始的航空航天,到现在几乎涉及到所有领域。Google 公司研发的全自动驾驶汽车可以在高速公路,城市和山区道路上行驶。它使用激光雷达、红外摄像头、惯性导航系统等传感器,融合 GPS 定位,应用 SLAM 技术为汽车提供地图更新,从而让它准确找到目的地,其原理图如图 1-2 所示。

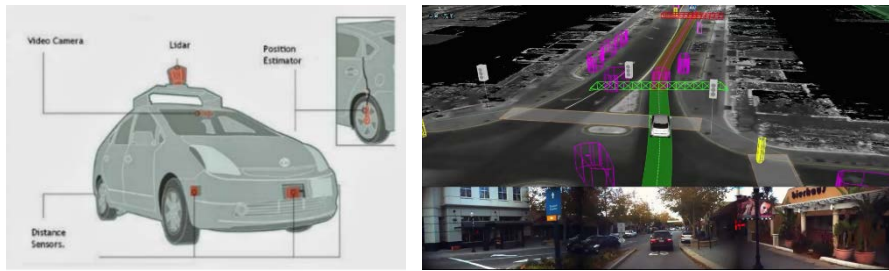


图1-2 Google 无人汽车工作原理图及其控制界面

美国 Omron Adept 技术公司推出的 Adept Lynx 是可以自主导航的自主室内小车 (Autonomous Intelligent Vehicle, AIV)。它的专有软件和控件系统允许其在拥有行动的人和不在地图中的障碍物的情况下依然能够进行导航, 而且不需要修改设备, 如地板磁贴或导航信标^[34], 如图 1-3 所示。它通过激光与声呐进行环境采集, 采集范围为 250°, 以“安全第一”为设计准则, 因此它可实现很好地人机合作。

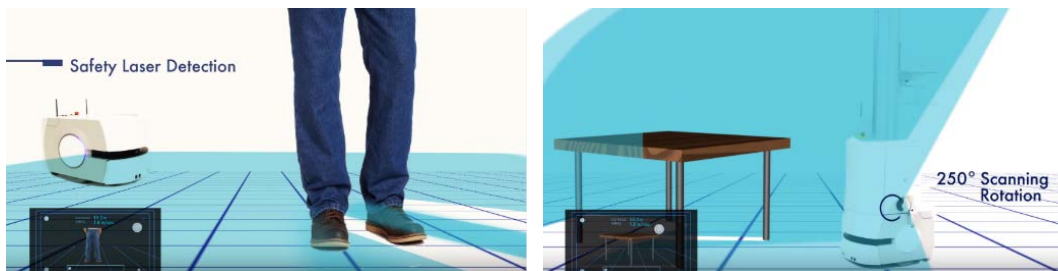


图1-3 Adept Lynx 移动平台

David V. Lu 等学者针对环境存在行人的导航问题进行研究, 分析了在代价地图中将行人对导航的影响设为高斯分布时, 高斯参数与路径规划算法元素之间复杂的相互作用, 图 1-4 展示了高斯参数不同时, 所计算路径的差别: 随着 A 值的增大, 计算的路径远离行人, 但当 A 值非常大时, 计算的路径将无限接近行人^[35]。



图1-4 高斯参数对导航的影响^[35]

国内对移动机器人基于 SLAM 理论的导航的研究时间较国际稍短一些, 但也取得了一定的成绩。

浙江大学的奚婉对电力巡检机器人的导航系统进行了研究, 成功的实现了地图构建以及遵循固定轨迹的导航, 实现了较高的导航精度^[37]。

哈尔滨工业大学的王超应用动态窗口法成功实现了动态避障, 如图 1-5 所示,

但是避障方式为二维避障，难以实现室内复杂环境下的安全避障。

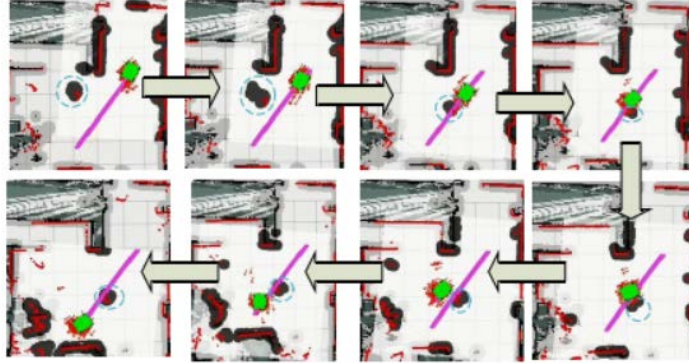


图1-5 DWA 动态避障过程图^[38]

哈尔滨工业大学的张明明应用 Kinect2.0 深度相机进行了三维地图的构建，并在三维地图下实现自主导航，实现了三维避障。但维护三维地图所需的成本非常大。

深圳乐行天下科技有限公司研发了一套完整的定位与导航的解决方案 NaviPack，可以实现室内机器人的定位、SLAM 以及智能避障，如图 1-6 所示。图 1-6 a) 所示为导航机器人移动底盘的结构图，装配了激光雷达、超声波传感器等传感器，图 1-6 b) 为通过人为遥控的方式构建室内环境的地图。

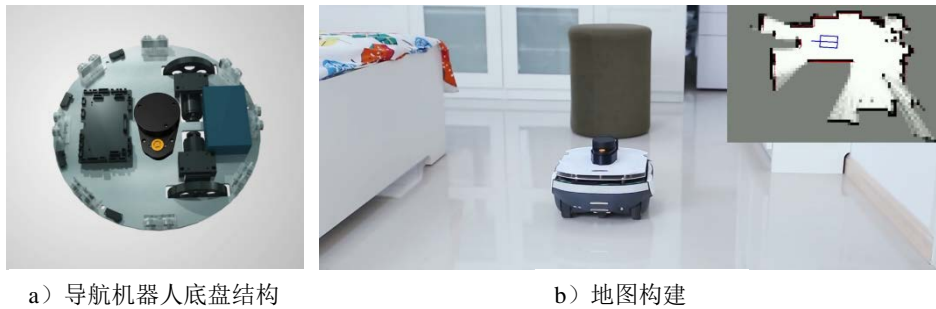


图1-6 搭载 NaviPack 的移动机器人

国内的导航技术相对于国际上还比较落后，移动机器人的导航方式主要以磁导航、基于反光板的导航为主，而基于 SLAM 的导航应用还比较少，因此，基于 SLAM 的导航非常具有研究价值。

1.4 主要研究内容

本文将对基于 SLAM 的室内移动机器人导航技术进行研究，在未知环境下，通过二维激光雷达扫描所处环境信息，应用主动探索算法完成主动 SLAM，得到当前环境地图，并在行进过程中通过基于 Kinect 相机的路径规划模块进行三维避障。最后定位算法根据激光雷达进行已知地图上的定位并通过路径规划完成人为指定的导航任务。

综上所述，本文的研究内容主要归纳为下面几个方面：

(1) 基于激光雷达的 SLAM 模块的研究

为了能够有效地在未知环境下进行复杂的任务，机器人必须拥有构建当前环境地图的能力。本文将选择合适的 SLAM 算法进行深入研究，并对 SLAM 中的数据关联、闭环检测、后端优化等关键步骤的理论过程进行分析，进行程序实现与调整参数，最终实现高效、高精度的室内环境地图构建。

(2) 基于 Kinect 相机的路径规划模块的研究

将分别就全局路径规划和局部路径规划算法进行分析，选择较合适的算法进行研究并进行程序实现，使机器人在执行导航任务时能够自主避障，从而安全地完成导航任务。通过将 Kinect 深度相机的深度数据转化成三维点云，设计了一种基于三维点云的避障策略，以实现在室内复杂环境下的三维避障。

(3) 机器人主动 SLAM 技术的研究

主动 SLAM 可以解释为：在未知的环境下，机器人能够在没有人为干预时自主地构建当前环境的栅格地图。主动 SLAM 技术将 SLAM 与路径规划结合在一起，使得机器人实现自主构建地图。本文将采用基于边界的主动探索模型，以实现主动 SLAM 技术。

(4) 定位模块的研究与导航技术实验

在已经拥有当前环境地图的情况下，机器人想要完成导航任务，首先要确定它在地图中的位置，即为定位。定位的准确与否将极大的影响导航精度，本文将根据全局定位与位姿跟踪能力选择定位算法并进行深入研究；最后通过实验对各个模块进行验证，分析其效果与精度。

第2章 移动机器人的系统模型及架构

2.1 引言

移动机器人的各个数学模型是定位、SLAM 算法实现的基础，因此，本章对文中将要使用的数学模型进行建模与定义。本章分为三个部分，首先对用于 SLAM 的各种传感器进行分析选择，搭建所用实验平台；其次，针对所用实验平台进行相应的数学建模；最后，对整个导航系统的架构进行阐述，分析了各个模块的功能以及导航系统的结构。

2.2 传感器选型及实验平台搭建

2.2.1 传感器分析

自主机器人的标准之一是能够感知其周围环境。可用于 SLAM 的典型传感器有测距型传感器，如声呐测距传感器、激光测距仪等，视觉型传感器如单目相机、双目相机、立体相机等，以及 RGB-D 传感器，也就是深度相机，如微软的 Kinect，华硕的 Xtion 深度相机等等^[39]。不同的 SLAM 方法需要不同类型的传感器，以适应机器人可能遇到的独特环境。SLAM 技术已经发展到了地面机器人，空中机器人，水下机器人，甚至陆空两用机器人。

机器人的传感器用于感知周围环境。传感器的选择不仅能影响机器人可用环境信息的质量和数量，而且还会影响到 SLAM 算法的选择。本文将对当前 SLAM 研究和应用中较为常用的传感器进行分析对比，以选择合适自己平台的传感器。

（1）声学传感器

声呐传感器主要应用在其它传感器难以工作的水下环境。声呐在海底区域可以提供更好的分辨率，探测范围广，并且以较低频率发出声呐还可以减小声呐的损耗。但是，海底区域的单一性意味着声呐信息存在较高的不确定性。

超声波传感器是移动机器人领域最便宜的空间传感器。它可以适应大多数物体的表面，无论是金属还是非金属，不论是否洁净或者透明，只要待测量表面具有足够的声学反射率它就可以测出距离。但是，它的空间分辨率很低，测量范围短，传播速度慢，以及对环境十分敏感等导致了它在 SLAM 领域无法普及。

（2）激光测距仪

激光测距仪是解决 SLAM 问题最常用的传感器之一。激光雷达能够以较高的

频率提供大量、准确的距离信息，并且测量范围远，不易受光线等环境因素影响，在室内和室外环境都可以获得可靠的结果，这使得激光雷达在解决 SLAM 问题上应用的越来越多。但是二维激光雷达只能扫描一个平面的信息，不能得到其它高度的信息。三维激光雷达可以扫描到不同高度平面的信息，然而，三维激光雷达的价格十分昂贵，是二维雷达的 10 倍左右，这也限制了三维激光雷达无法像二维激光雷达获得普遍的应用。

（3）立体相机

立体相机的结构与人眼的结构十分类似，通过图像匹配和计算获取环境中目标的精确坐标信息以及深度信息，得到的数据精度较高^[40]。但是在使用前要对其进行复杂的标定与校准，而且使用时需要对两侧的图像进行对比分析，运算量很大，难以保证数据实时性。

（4）RGB-D 传感器

RGB-D 相机价格低廉，性能稳定可靠，可同时获得深度图像信息与彩色图像信息，信息量大，可以建立 3D 模型。但是 RGB-D 信息数据量十分巨大，数据实时性较差。RGB-D 传感器不能在高强度的阳光下操作，测距表面为半透明或高反射表面时无法提供可靠的距离数据，并且有效范围有限，精度也有限。

激光雷达与 Kinect 的视角对比如图 2-1 所示。图 2-1 a) 为二维激光雷达的视角，仅检测平面上的物体，只能看到图中椅子的支持部位，而 Kinect 相机可以检测到前方不同高度下的障碍物，如图 2-1 b) 所示，Kinect 能够看到椅子的整体结构。因此，应用 Kinect 相机可以实现室内环境下的三维避障。

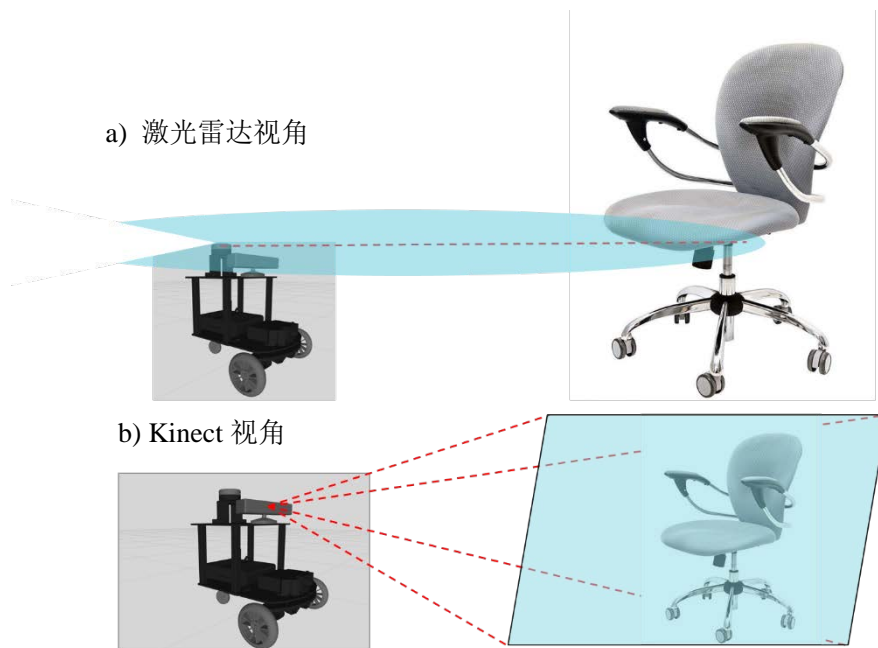


图2-1 激光雷达与 Kinect 视角对比

由于本文的研究环境为室内环境，室内环境复杂，充满着小箱子等不同尺寸的障碍物，应用二维雷达进行障碍物躲避是不安全的，因此，本文选用 Kinect 相机作为避障的传感器。

双目、RGB-D 等相机每刻采集的数据量太大，容易导致构建地图的实时性不好，而且用相机构建室内环境的二维地图将浪费很多数据，不能充分利用相机的三维数据。二维激光雷达的精度高于 RGB-D 相机等其它传感器，且价格适中，数据量显著少于 RGB-D 相机，因此本文选择二维激光雷达作为构建地图的传感器。

综上所述，本文最终选用二维激光雷达用于进行 SLAM 与定位，选用 Kinect 相机作为避障的传感器。

2.2.2 实验平台搭建

考虑到价格、精度、算法要求等因素，最终确定的传感器型号为：Hokuyo 公司的 URG-04LX 二维激光雷达，微软公司的 Kinect 1.0 相机。

激光雷达的配置参数如表 2-1 所示：

表2-1 激光雷达参数

产品名称	激光扫描仪
型号	URG-04LX
检测距离	60mm~4095mm（保证精度距离）
精度	距离：20~1000mm：±10mm 距离：1000~4000mm：测量值的±1%
分辨率	1mm
角度范围	240°
角度分辨率	0.36°
扫描速度	100msec/scan
噪声级别	≤25db（在 300mm 处）

Kinect 相机的视角范围为：水平视角 57.5°，垂直视角 43.5°，机身俯仰范围为 ±27°，色彩和深度的有效视距为 1.2~3.5m。

本文通过在小强 XQ4 通用 ROS 机器人开发平台上增加二维激光雷达与 Kinect 深度相机完成实验平台的搭建，如图 2-2 所示。该平台的驱动方式为差速驱动式轮式驱动，两个主动轮在前，一个万向轮在后，电机为 15W 直流电机，内置 i7-mini 工控机作为上位机向底层驱动发送运动指令，应用 STM32 作为电机驱动的控制板，将运动指令转换为电机的控制量。该平台拥有前置广角（178°）USB 摄像头、9 轴 IMU（Inertial measurement unit，惯性测量单元）、码盘等传感器，以及文本选用的二维激光雷达，Kinect 深度相机，可以满足地图构建与室内导航的要求。

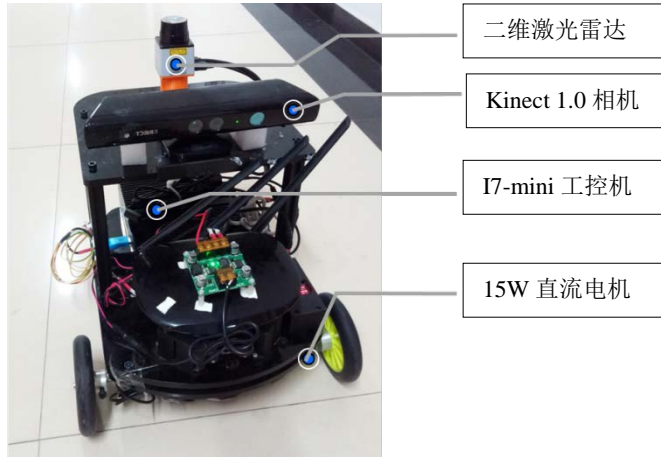


图2-2 移动机器人实验平台

2.3 机器人运动模型

由于控制噪声和许多未建模的外部效应的影响，机器人的运动很难以确定的形式精确描述。通过概率的形式来描述机器人运动学可以取得更好的效果。理论上，恰当的概率模型可以更加准确地模拟机器人运动和感知中存在的特定类型的不确定性。

机器人姿态由其相对于全局坐标系的位置和方向组成。刚性移动机器人拥有六个状态变量，三个表示它们在笛卡尔坐标系下的位置，三个表示它们的角度方向（pitch, roll, yaw）。对于本文研究的局限于平面运动的刚性移动机器人来说，姿态通常由三个变量描述，即它在平面中的两个位置坐标 (x, y) 和其航向 θ ，可以用向量 $x_t = (x, y, \theta)^T$ 描述， x_t 表示 t 时刻机器人的位姿。

控制命令表达机器人状态变化的信息，如机器人的速度。人们通常用里程计的数据代替控制命令。里程计是测量车轮转速的传感器，由于他测量了控制行为的影响，所以即使它是传感器，大家也用它作为控制数据，通常用 μ_t 表示。

通常，用概率论的微积分来表示机器人的不确定性。因此，移动机器人在控制 μ_t 的作用后的状态可由后验概率分布 $p(x_t | \mu_t, x_{t-1})$ 表示。

由于执行机构与里程计存在误差，以及其它未建模因素的影响，在给定机器人控制 μ_t 之后，机器人的位置很可能会和目标位置存在一定偏差。考虑到系统不确定性的机器人的状态关系可以由式（2-1）表示，该式表示在给定控制 μ_t 的作用下，机器人现在的状态估计 x_t 与初始状态 x_{t-1} 之间的关系。

$$x_t = f(x_{t-1}, \mu_t) + \omega_t \quad (2-1)$$

式中 $f(x_{t-1}, \mu_t)$ ——系统的状态转移函数；

ω_t ——表示系统动态噪声及系统建模本身的不确定性。

将控制量表示成 $\mu_t = (D_{t+1}, \gamma_{t+1})$ ， D_{t+1} 为平移量， γ_{t+1} 为旋转量。将控制量带入式（2-1）中可以得到基于控制量的移动机器人运动模型，如式（2-2）所示：

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{bmatrix} = \begin{bmatrix} x_t + \frac{D_{t+1}}{\gamma_{t+1}} \cos((\theta_t + \gamma_{t+1}) - \cos(\theta_t)) \\ y_t + \frac{D_{t+1}}{\gamma_{t+1}} \sin((\theta_t + \gamma_{t+1}) - \sin(\theta_t)) \\ \theta_t + \gamma_{t+1} \end{bmatrix} + \omega_{t+1} \quad (2-2)$$

2.4 传感器观测模型

通过对机器人的传感器进行建模，可以将传感器的数据转换为机器人所需的信息。然而在实践中，由于物理现象的复杂性，精确建模通常是不可能的。因此，依然通过概率的形式来模拟传感器测量过程中的噪声。形式上，测量模型被定义为条件概率分布 $p(z_t|x_t)$ ，其中 x_t 表示机器人姿态， z_t 为 t 时刻的测量值。

各种传感器观测环境得到的数据类型不同，本节将为二维激光雷达建立观测模型。传感器得到的数据通常都是相对于传感器自身而言的，因此，大多数测距传感器均采用以传感器为中心的极坐标系来描述数据。

观测模型表示了传感器观测到的地标在极坐标系下的距离、角度与该地标在全局坐标系下位置之间的对应关系，见式（2-3）。

$$\begin{pmatrix} r \\ \varphi \end{pmatrix} = \begin{bmatrix} \sqrt{(\lambda_x - x)^2 + (\lambda_y - y)^2} + v_r \\ \tan^{-1}\left(\frac{\lambda_y - y}{\lambda_x - x}\right) - \theta + v_\theta \end{bmatrix} \quad (2-3)$$

式中 r, φ ——传感器坐标系下的距离与角度；

λ_x, λ_y ——观测到的地标在全局坐标系下的位置；

x, y, θ ——机器人在全局坐标系下的位姿；

v_r, v_θ ——观测模型的距离角度误差。

通过观测模型，机器人可以将传感器坐标系下的数据转换到全局坐标系下，转换后的数据可以用来估计机器人自身的位姿与环境地图的位置。

2.5 环境地图模型

地图构建的过程中，需用采用特定的存储形式来保存地图以供机器人导航使用。不同应用场景及不同环境有不同的地图表征方式。常用的地图表达方式有三种：栅格地图、拓扑地图、几何特征地图，其基本特点如下：

（1）栅格地图是用大小相等的单元栅格表示环境，每个栅格被赋予一个 0 或

者 1 的值来表示此处的状态：1 表示存在障碍物，用黑色栅格表示；0 表示不存在障碍物，用白色栅格表示；0.5 代表未知状态，为栅格地图的初始值，用灰色栅格表示。每个栅格的边长被称为地图的分辨率。地图的分辨率越小，该地图越接近真实环境的地图，同时所需要的计算复杂度也越高。栅格地图如图 2-3 所示。

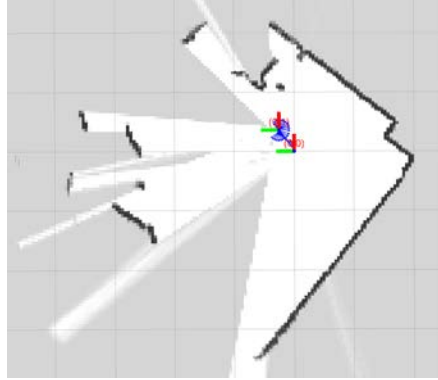


图2-3 栅格地图

(2) 拓扑地图是采用将重要部分抽象出来表示的方式，由代表环境汇总特定位置的节点与代表特定地点之间的路径信息的连线构成，对应的权值表示距离大小。拓扑地图紧凑、储存空间小，避免了特征提取、地图描述等难点，适合应用在一些复杂环境中。

(3) 几何特征地图是指通过提取传感器感知环境的原始测量的点、直线、拐角、多边形等几何特征来表示环境，与人对环境的认识最为相似。基于该地图的机器人定位简单，精度较高，但是该地图难以描述一些非结构化的环境。

本文研究的导航应用场景为室内环境，由于室内环境大多为结构化环境，而且大多数地面较平，描述环境的二维地图已经足够。由于栅格地图的精度高、表达方便、简洁易读等优点，在机器人领域被广泛应用。并且室内环境范围有限。因此本文选用二维栅格地图表示环境，可以充分发挥栅格地图直观易懂、细节描述清晰的优势。

2.6 导航系统整体架构

2.6.1 机器人操作系统 ROS

随着机器人技术的规模和范围不断扩大，为机器人编写控制软件也越来越复杂。不同种类的机器人的硬件型号很可能不同，不同的硬件需要的控制信号也可能不同，这使得代码重用变得非常困难。除此之外，机器人编写系统是非常庞大的，它包括了底层驱动、感知部分、抽象推理等等。由于机器人技术包含了很多领域的

知识, 研究人员将所有领域的知识都学会是非常困难的, 这就要求机器人的软件架构必须支持模块化的集成能力^[41]。

Willow Garage 公司发布了机器人操作系统 (Robot Operating System, ROS) 来支持机器人系统集成化、模块化。ROS 是面向机器人的开源的软件开发框架, 提供了很多的库和工具以帮助软件工程师开发属于自己的机器人, 能够提供如硬件抽象、装置驱动、库文件、可视化工具、消息传递管理、软件包管理等等功能。由于 ROS 的分布式系统、多语言跨平台、高度的集成特性、免费开源等优点, 现如今已经被广泛应用到机器人编程领域。而且 ROS 将一整套系统抽象成几个独立的、解耦的模块, 因此基于 ROS 设计的软件系统往往具有很好的可移植性, 通过一次编程即可适用很多个机器人, 极大的提高了开发效率。

因此, 本文所搭建的室内导航系统是基于 Ubuntu 环境下的机器人操作系统 ROS 开发的, 该导航系统具有很好地可移植性, 可以应用到不同的机器人中。

2.6.2 Navigation 导航软件包

本文的路径规划模块借助 ROS 提供的 Navigation 导航软件包集来完成路径规划算法与基于点云避障策略的实现。

图 2-4 为 Navigation 导航软件包集的框架, 可以看到该框架的输入为地图文件、里程计数据、传感器数据如激光雷达与 Kinect 等, 输出为控制机器人运动的线速度、角速度数据。该导航框架通过地图数据以及传感器数据生成供全局路径规划与局部路径规划的全局代价地图与局部代价地图。还提供了一个供导航失败的运动恢复软件包, 该软件包可以在导航失败时自动调整位姿以重新恢复导航。

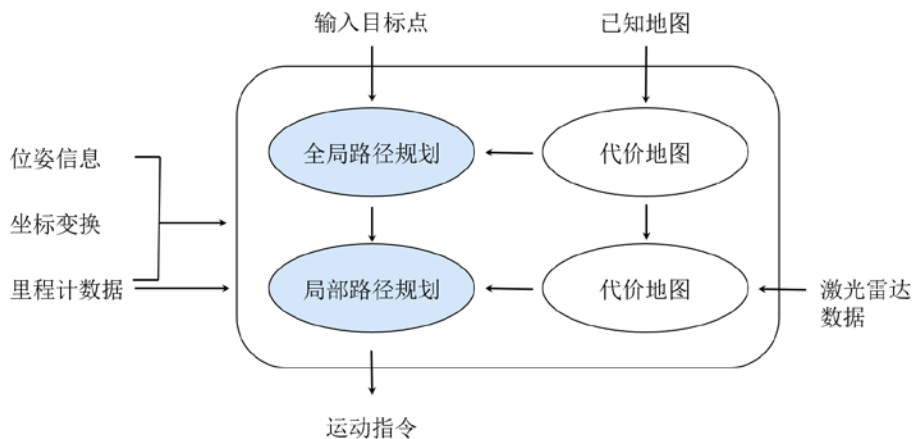


图2-4 Navigation 导航软件包集框架图

图 2-5 为 ROS 中的 3D 可视化仿真工具 Rviz 的软件界面。该软件通过订阅各个话题得到各种消息以及传感器数据, 然后将这些数据可视化到软件界面上以便

查看。该图展示了机器人模型、机器人轨迹、障碍物、代价地图、雷达数据点、目标点位姿等信息。通过该软件可以非常直观的观察机器人运行状态。本文将通过此软件进行机器人的控制，从而完成各个实验。

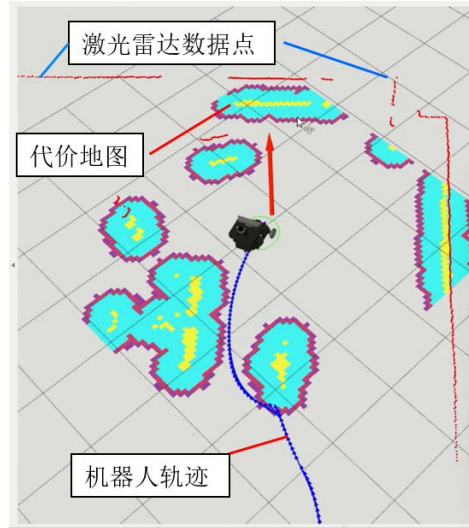


图2-5 Rviz 软件界面

代价地图储存了栅格地图中障碍物的信息以及机器人与障碍物碰撞的可能值。它通过使用传感器数据与静态地图的信息来储存和更新代价地图中障碍物的信息。代价地图中的每个栅格都存在三种状态，分别是占用、空闲及未知。并且每一个栅格都会分配到一个 0 到 254 间的数值，数值越大代表机器人越有可能和被占用的栅格碰撞，0 代表绝对不会发生碰撞。由于计算路径规划时不会考虑机器人的几何形状，而是将机器人当做质点进行处理。为了衡量计算出的路径与障碍物碰撞的可能性，代价地图将被占用的栅格进行膨胀，膨胀出来的栅格的数值随着距离的增加而减小。代价地图不会使用这些数值信息，而是将这些信息提供给路径规划算法，路径规划算法将考虑这些信息从而计算出最优路径。

本文的路径规划模块将应用 Navigation 导航软件包集框架进行程序实现。将 A*全局路径规划算法与时间弹性带局部路径规划算法融入该框架中，并应用独立设计编写的程序，将 Kinect 相机的深度数据转换成三维点云，根据三维点云进行避障，从而完成了对路径规划的实现。

2.6.3 室内导航系统整体结构

由于采用 ROS 作为系统框架，导航系统各项之间耦合度很低，本文采用模块化设计方式。根据导航系统的需求分析，将整个系统分为 SLAM 模块、主动探索模块、定位模块、路径规划模块。

图 2-6 为导航系统的整体架构图，从图 2-6 中可以看出本文的导航系统分为两个部分：未知环境地图时的主动 SLAM 部分与已知地图时的自主导航部分。主动 SLAM 实现未知环境下的自主地图构建，从而得到当前环境的二维栅格地图。在拥有了环境地图后，自主导航板块实现已知地图下的定位以及路径规划，从而完成人为指定的导航任务。

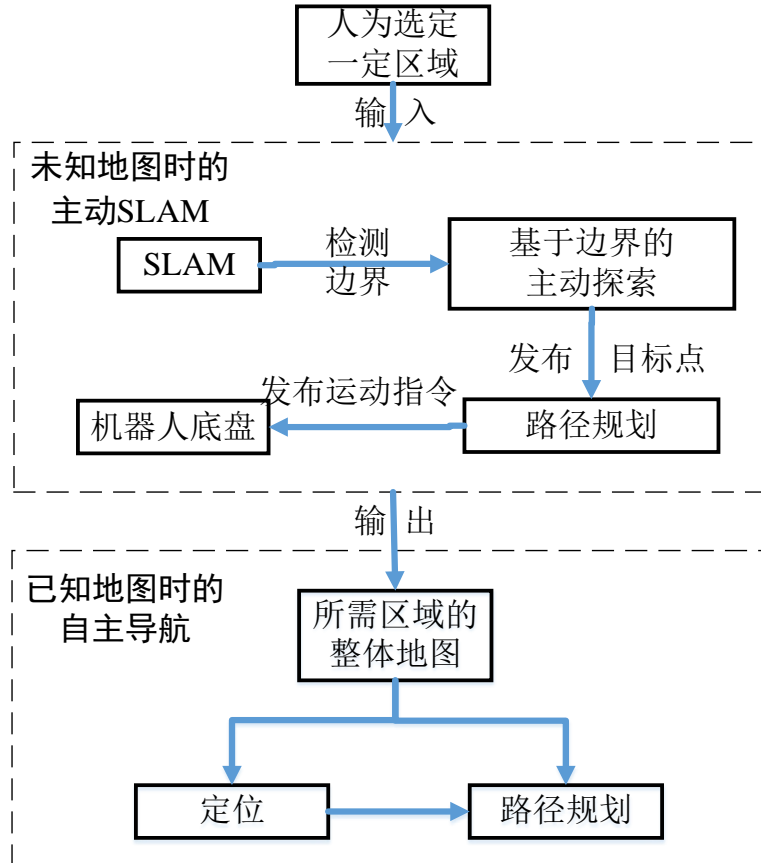


图2-6 导航系统整体架构

主动 SLAM 部分包含了 SLAM 模块、路径规划模块与主动探索模块三个部分。SLAM 模块应用二维激光雷达进行地图构建，路径规划模块实现路径的规划与机器人的运动控制，并应用 Kinect 相机进行室内复杂环境下的三维避障。基于边界的主动探索模块负责检测地图的边界并向路径规划模块发布目标点，使得机器人向着未知区域前进，直至构建完成规定区域的全部地图。

得到了环境地图后，SLAM 模块停止工作，并应用二维激光雷达检测环境以进行已知地图下的定位。在机器人明确其在地图中的位置后，基于 Kinect 相机的路径规划模块负责计算路径并进行三维避障，使得机器人安全地完成人为指定的导航任务。

2.7 本章小结

本章首先通过对用于 SLAM 的各个传感器进行对比分析，确定了本文所用传感器为：二维激光雷达作为构建地图的传感器，Kinect 相机作为避障的传感器。针对所用实验平台进行了相应的数学建模，分别建立了机器人运动模型，传感器观测模型以及环境地图模型。最后，对导航系统的整体结构进行分析，描述了各个模块之间的关系与功能。

第3章 基于图优化理论的 SLAM 算法研究

3.1 引言

移动机器人的许多任务，如运输，搜索和救援等，都需要拥有周围环境的地图。机器人的定位与地图构建是相互依赖不可分割的，因此，SLAM 一直是机器人导航领域研究的热点。

目前流行的 SLAM 算法大致可以分为两类，一类为基于滤波理论的 SLAM 算法，一类为基于图优化理论的 SLAM 算法。基于滤波理论的 SLAM 只考虑当前的传感器数据，丢弃了之前时刻的数据，这使得在地图建立时不可能重新访问所有数据。基于图优化理论的 SLAM 保留所有观测数据直到结束，并且根据从最开始到现在的所有数据来矫正位姿，因而被称为完全 SLAM。本章将对基于图优化理论的 SLAM 算法进行研究。

3.2 基于图优化理论的 SLAM 算法研究

基于图优化理论的 SLAM 算法最早由 Lu 和 Milios 在 1997 年提出，由于使用标准技术解决误差最小化问题具有相当高的复杂性，因此在当时并没有受到重视。直到 2010 年左右，随着对 SLAM 问题结构的深入研究以及稀疏线性代数领域的发展，使得解决优化问题成为了可能。基于图优化理论的 SLAM 算法目前已经在速度和准确性方面属于最先进的技术之一^[42]。

该方法应用所有保存的观测数据估计机器人所有的轨迹，这解决了完全 SLAM 问题。该方法从观测数据提取一组软约束，用一个稀疏图表示，如图 3-1 所示，图的节点代表机器人在该时刻的位姿与环境地图，连接两个节点的边表示两个节点之间的非线性约束关系，实线为任意两个连续的机器人位姿间约束，虚线为位姿与机器人假定处于该位姿时观测到的特征的约束关系。

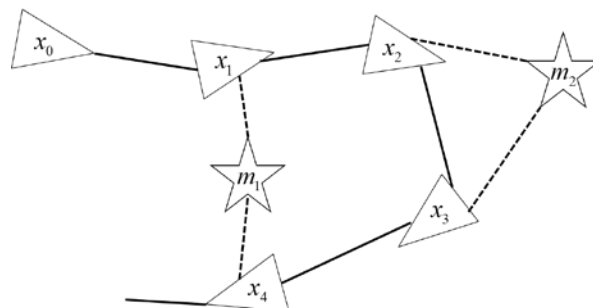


图3-1 位姿图表示

基于图优化理论的 SLAM 算法将 SLAM 问题划分为前端与后端两个部分。

前端根据传感器数据来构建图并计算约束，约束由两种形式构成，一种是通过相邻的几个连续观测的对应关系得到，另一种是通过闭环检测，检测机器人是否回到了曾经访问过的位置从而添加闭环约束。

后端根据前端提供的约束对图进行优化，它通过将这些约束条件解析为全局一致的估计值来获得地图和机器人路径。由于新的观测加入到已有的位姿图时中只是考虑最近的几次观测，因此位姿图会逐渐累积误差，通过全局优化问题可以将累积误差除去，从而得到在给定约束条件下的机器人运动轨迹的最优估计。

基于图的 SLAM 算法交错执行前端和后端，如图 3-2 所示。前端需要在部分优化的地图上进行操作，以限制误差的累积，当前位姿估计越精确，前端生成的约束条件的鲁棒性就越强，得到的地图就越精准。

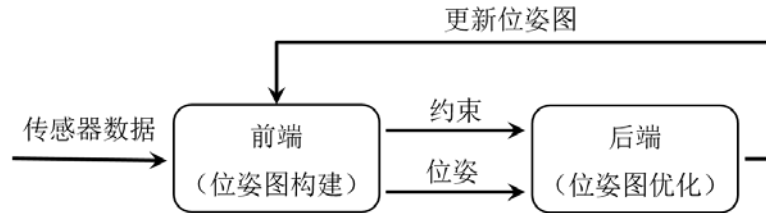


图3-2 基于图优化理论的 SLAM 框架

通过查阅各种算法的资料以及实验效果，本文最终采用的是基于图优化理论的 SLAM 算法：Cartographer。该算法是 Google 公司于 2016 年 10 月开源的基于激光雷达的 SLAM 算法。通过与应用非常广泛的 GMapping 算法进行实验对比，不论是地图准确性、鲁棒性还是地图精度，Cartographer 都优于 GMapping 算法，证明了该算法的优越性与实用性。

Google 公司的 Cartographer-SLAM 为配备激光雷达的背包构建室内地图提供了实时的解决方案，可生成分辨率为 5cm 的 2D 栅格地图。由于配备了惯性测量单元（Inertial measurement unit, IMU），该算法会将激光扫描到的数据投影到与重力垂直的方向。因此，系统的操作员在建筑物中行走即可构建地图。该算法的流程图由图 3-3 所示，该图将整个 SLAM 过程大致分为三个部分：激光扫描匹配、闭环检测以及位姿图优化。

在进行大范围地图构建时，维护整个地图的高阶协方差矩阵具有非常庞大的计算复杂度，而且距离较远的特征间的约束也不大。因此，将整个环境地图划分为一个个的子地图（Submap）可以减小计算复杂度，通过构建子地图以及子地图之间的位姿变换来构成整个环境地图。当子图构建完成后，不会再插入新的扫描，它将参与闭环检测的扫描匹配。

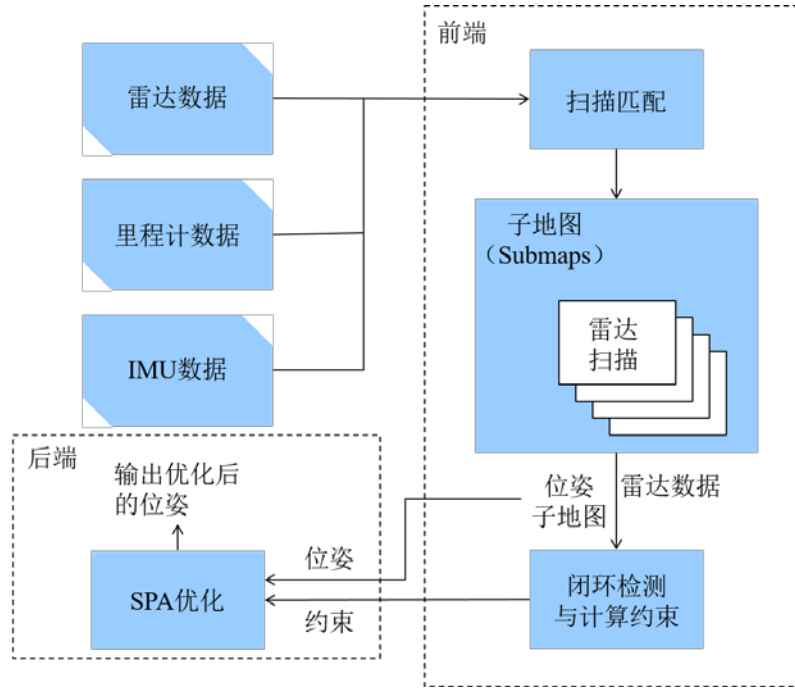


图3-3 Cartographer SLAM 理论过程图

3.2.1 激光扫描匹配

位姿图的构建主要是通过激光扫描匹配建立位姿节点之间的约束关系。扫描与扫描间的匹配（scan to scan）经常用于计算激光 SLAM 方法中的相对姿态变化。但是，这种方法累积误差很快。扫描与地图间的匹配（scan to map）有助于限制这种误差的积累，然而随着建图范围的增加，这种方法的计算复杂度将显著增加。

扫描与子地图间的匹配（scan to submaps）可以得到更好的效果，将每一帧的扫描数据通过扫描匹配插入子地图中的最佳估计位置处。由于机器人短时间内不会剧烈移动，因此扫描匹配是针对最近的子图进行的。扫描到子地图的匹配方式是使用非线性优化的方法，将每个连续扫描与子地图进行匹配，从而得到该扫描在子图中的最佳位置估计。

子地图的构建是反复匹配扫描坐标系与子图坐标系的迭代过程。在将扫描插入子图之前，扫描匹配器负责找到该扫描在子地图中概率最大的位姿，通过非线性最小二乘优化进行求解得到最优匹配，见式（3-1）^[43]。

$$\arg \min_{\xi} \sum_{k=1}^K (1 - M_{SMOOTH}(T_{\xi} h_k))^2 \quad (3-1)$$

式中 h_k ——代表该时刻第 k 个激光扫描数据；

T_{ξ} ——扫描数据 h_k 从激光雷达坐标系到子地图坐标系的坐标变换；

M_{SMOOTH} ——三次插值平滑函数，得到该扫描在子地图中位置的概率。

通过求解式 (3-1)，可以计算得到残差最小的位姿，再将当前扫描插入到该位姿处，从而实现最优的扫描匹配。但是这个过程始终存在误差，并且会不断累积，闭环优化可以消除该累积误差。

3.2.2 闭环检测

由于扫描匹配不可能完全准确，会随着时间的推移逐渐积累误差，需要通过闭环约束将其消除。闭环检测指的是通过传感器的数据判断是否来到了曾经探索过的地方，用以检测当前观测与已经构建好的地图之间的关系^[44]。如果匹配到相同的地方说明环境中存在闭环的结构，可以在位姿图中添加新的闭环约束。正确的闭环约束可以消除系统中的累积误差，从而得到一致性更好的位姿图。

子地图构建完成后，不再有新的扫描被插入，它将参与闭环检测的扫描匹配。所有完成的子地图和扫描都会被用来进行闭环检测。如果检测到当前扫描与子地图中某处足够相似，则将其作为闭环约束添加到优化问题中。每隔几秒进行一次匹配搜索，以使得在访问到曾经访问过的位置时，环路会立即闭合。因此，闭环扫描匹配的频率必须比扫描更新更快，否则会出现明显的滞后，可能会导致不一致的地图。通过使用分支定界方法（**Branch-and-bound Methods**）来加速这种搜索匹配可以实现较大的搜索空间上高效地计算匹配位姿。

分支定界法在混合整数线性规划背景下提出，其主要思想是将可行解的子集表示为树中的节点，其中根节点表示所有可行解空间，每个节点的子节点形成他们父节点的一个分支，称为分支节点，对应着一个可行解的部分集合，是该分支能否成为拓展分支的关键。整个分支的最底端被称为叶节点，它是全部可行解的集合。该算法结构如图 3-4 所示。

通常，该算法首先将全部的可行解集合不断的分成越来越小的集合，即为分枝；之后，为每个小集合计算出一个目标上界或者下界，即为定界；在进行分枝时，那些集合的界限超过了已知可行解界限的将不再进行分枝，即为剪枝，因为这个集合中不存在比目前最优的可行解更好的解。

对于 SLAM 过程，就是要找到最佳的匹配从而确定闭环，即确定当前扫描与以前构建的子地图为同一地点的最大概率，因而需要求得上界。解决 SLAM 闭环问题的分支定界算法过程如下：

第一步：根据计算复杂度以及上界质量设定边界。根据最初节点拓展其所有子节点，在当前可行解集合中选择边界，作为初始边界。接下来每步的上界由当前所有可行解集合的最大值决定（定界）。

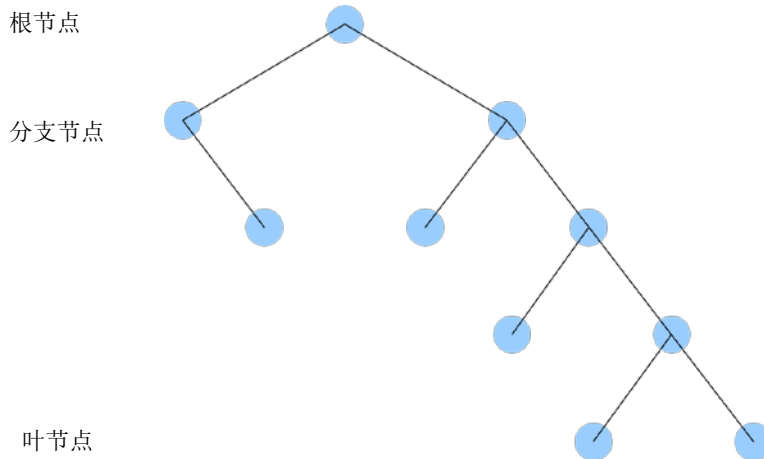


图3-4 分支界定算法的搜索树

第二步：每个未被拓展的节点有且仅有一次机会成为拓展节点，当其目标函数的值大于当前上界时即可成为拓展节点。对拓展节点进行分枝，生成当前拓展节点的所有子节点（分枝）。在所有子节点根据边界舍弃那些不可能产生最优解的节点（剪枝）。

重复第一步与第二步，在目前所有符合条件的节点中选取下一次的拓展节点。如此循环，直到找到问题的最优解或者待拓展节点列表为空。

匹配当前扫描与以往子地图的问题与上一节的扫描匹配十分类似，同样可以应用非线性最小二乘的方式解决。为了寻找到正确的闭环匹配，设定一个阈值函数。只有匹配得分大于该阈值才会被认为存在一个闭环。由于闭环在实际操作过程中不经常出现，这使得节点的选择策略与初始节点的选择并不是很重要。

闭环检测每隔几秒进行一次，不断在已经构建好的子地图中搜索与匹配，判断是否存在与当前扫描相似的子地图，这使得在重新访问已经来到过的地方时，环路会立即闭合。

图 3-5 展示了闭环优化的整体过程。机器人从起点开始绕着某房间运动了一圈。当机器人再次回到起点时，其路径中出现了图 3-5 a) 所示的闭环误差，即在同一位置处，新的激光扫描与最初构建的地图匹配不上。应用了分支定界扫描匹配的方式加快了激光数据的匹配过程，闭环检测能够每隔几秒进行一次，使得闭环误差一经出现就可以被检测到。图 3-5 b) 展示了下一时刻，进行了闭环优化之后的地图。可以看到整体地图在经过闭环优化之后，成功的消除了闭环误差，使得新得到的激光扫描数据与地图能够匹配上。

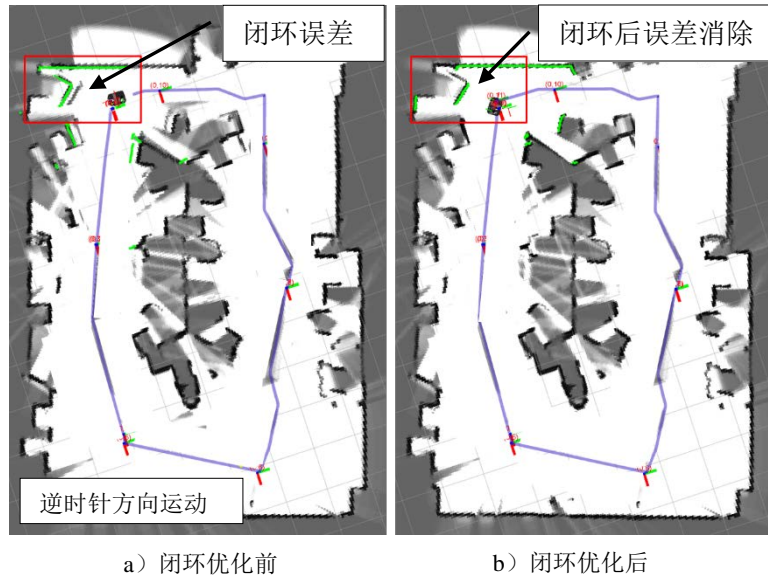


图3-5 闭环优化过程

3.2.3 位姿图优化

应用上述算法虽然可以完成位姿图的构建，但是为了应对由约束引入的累积误差，需要定期对位姿图进行优化，以便将误差最小化从而获得一致的地图。可以应用稀疏姿态调整（Sparse Pose Adjustment, SPA）来进行后端优化。该方法使用众所周知的 Levenberg-Marquardt（LM）方法作为框架，利用位姿图的稀疏性将地图特征与机器人间的约束转换为机器人姿态中的直接约束来创建更紧凑的位姿图，从而实现姿态和约束条件的优化。SPA 优化器具有以下优点^[45]：

- （1）它使用了约束中的协方差信息，从而得到更加准确的解；
- （2）SPA 方法对初始化具有很强的鲁棒性，对于增量和批量处理都具有非常低的失败率；
- （3）因此它只用 LM 方法进行了较少次数的迭代，使它的收敛速度非常快；
- （4）SPA 是完全非线性的。在每一次迭代中，它将当前姿态周围的所有约束线性化。

优化的目的就是在充满误差的位姿集合中，求得误差最小的位姿集合。位姿图中的节点表示见式（3-2）：

$$c_i = [t_i, \theta_i] = [x_i, y_i, \theta_i]^T \quad (3-2)$$

式中 c_i ——表示位姿图中的一个节点，代表机器人的位姿；

t_i, θ_i ——表示该节点在全局坐标系下的坐标与角度。

位姿图中的约束即为两个节点之间的变换关系，用式（3-3）表示：

$$h(c_i, c_j) \equiv \begin{Bmatrix} R_j^T(t_j - t_i) \\ \theta_j - \theta_i \end{Bmatrix} \quad (3-3)$$

式中 $h(c_i, c_j)$ ——观测方程，表示任意两个节点位姿之间变换的估计值；

R_i^T ——机器人角度 θ_i 的 2×2 的旋转矩阵。

因此，约束的误差公式可用式（3-4）表示：

$$e_{ij} \equiv \bar{Z}_{ij} - h(c_i, c_j) \quad (3-4)$$

式中 \bar{Z}_{ij} ——节点 c_i 到节点 c_j 间变换的测量值（真实值），带有精确度矩阵 Λ_{ij} （协方差矩阵的逆）。

整体位姿图的误差可用式（3-5）表示：

$$x^2(c, p) \equiv \sum_{ij} e_{ij}^T \Lambda_{ij} e_{ij} \quad (3-5)$$

通过最小化总体误差式（3-5）即可求得位姿集合 c 的最优配置。SPA 方法通过 Levenberg-Marquardt（LM）算法解决该问题，LM 系统用式（3-6）表示：

$$\begin{aligned} (H + \lambda H_{diag}) \Delta x &= J^T \Delta e \\ J &\equiv \frac{\Delta e}{\Delta x} \\ H &\equiv J^T \Lambda J \end{aligned} \quad (3-6)$$

式中 λ ——为一个较小正系数，该系数控制了 LM 算法的步长；

H ——由位姿图稀疏性构造出来的具有压缩存储数据结构的上三角矩阵；

H_{diag} ——取整体观测矩阵 H 对角线上的值构成的一个对角矩阵。

求解上述线性方程将产生一个增量 Δx ，通过将其重新加回到位姿 x 从而得到优化后的位姿，如式（3-7）所示：

$$\begin{aligned} t_i &= t_i + \Delta x \\ \theta_i &= \theta_i + \Delta \theta_i \end{aligned} \quad (3-7)$$

SPA 算法过程如下^[46]：

（1）首先初始化 λ 的值，若为0则应用上一次迭代储存的值；

由于传感器的距离限制，机器人只能观测到附近一定距离的特征，因此，位姿图呈现出非常稀疏的连接结构。该算法针对位姿图的稀疏性，通过只储存和处理非零元素可以大幅度降低储存空间以及计算复杂度。非零元素要通过压缩存储数据结构来储存，因此需要将由位姿图节点 c 和约束 e 创建出的稀疏上三角矩阵 H 转换成这种格式。

（2）建立线性系统式（3-6），通过求解该式得到增量 Δx 与 $\Delta \theta$ ；

（3）根据式（3-7）更新位姿，得到更新后的位姿；

（4）如果误差变小了，返回更新后的位姿，并把 λ 的值缩小一半，并保存；

(5) 如果误差变大了, 返回原始位姿, 并把 λ 扩大一倍, 并保存。

(6) SPA 算法输入为位姿 c 、约束 e 以及对角线增量 λ , 输出为更新后的位姿 c 。该算法的每次迭代是寻找一个合适对角线增量 λ , 当 λ 很小时, 算法就变成了 Newton-Euler 法的最优步长计算式, λ 很大时, 算法就蜕化为梯度下降法的最优步长计算式。

SPA 算法依赖于高效的线性矩阵构造和稀疏非迭代 Cholesky 分解, 以有效地表示和解决大型稀疏姿态图, 从而大幅度降低存储空间需求以及计算复杂度, 显著提高计算性能。

3.3 主动 SLAM 技术

机器人的导航需要使用地图, 但很少有机器人可以自主构建导航区域的地图。通常, 人们必须提前构建好地图, 提供障碍物的确切位置 (度量地图) 或者表示两个区域的连通性的图形 (拓扑地图)。因此, 大多数机器人在未知环境中无法有效导航。

自主探索可以让机器人突破这种限制。自主探索可以定义为机器人在未知环境中移动并构建出可用于后续导航的地图的一种行为。好的探索策略可以在合理的时间内生成完整或近乎完整的地图。虽然室内环境大多呈现出一种看起来简单直接的结构, 如矩形办公室, 直走廊, 以及无处不在的直角。但实际情况往往截然不同, 在真实场景下, 机器人需要在摆满家具的房间内进行导航, 墙壁也可能隐藏在书桌和书架的后面。因此室内的探索将要面对非常复杂的环境。

3.3.1 基于边界的主动探索

自主探索需要解决的问题可以总结为: 根据最初对周围环境的了解, 机器人可以在哪里获得尽可能多的新信息; 起初, 机器人仅仅知道在它初始位置可以感知的东西, 那么它如何才能尽可能快的构建一个尽可能完整的地图。基于边界的探索策略^[47]可以有效解决这个问题。边界这个概念是在机器人探索领域中提出的, 是指已知区域与未知区域间的交界。

该算法理论过程如下所述:

该算法首先利用类似于计算机视觉中的边缘检测和区域提取来查找边界, 通过检测栅格地图中不同的概率值来检测出边界。栅格地图将每个小栅格赋予一个 0、1 或者 0.5 的值来表示此处的状态: 1 表示被占用即存在障碍物, 0 表示没有被占用即不存在障碍物, 0.5 为栅格的初始状态表示该栅格未被探索。已探索区域与未

探索区域之间的栅格被称为边界栅格，连续的边界栅格共同组成边界区域，只有超过某个最小尺寸（机器人尺寸）的边界区域才被认定是一个有效的边界。

图 3-6 展示了该算法的边界提取过程，图 3-6 a) 显示了一个真实机器人在走廊建立的栅格地图，该位置周围存在的两个打开的门，图 3-6 b) 显示了已知区域与未知区域间的边界，图 3-6 c) 显示了大于最小边界尺寸的有效边界，十字线标记了该区域的中心，该位置同样是导航任务的目标点。边界 0 和边界 1 对应着两个打开的门，边界 2 对应着未探索的走廊^[47]。

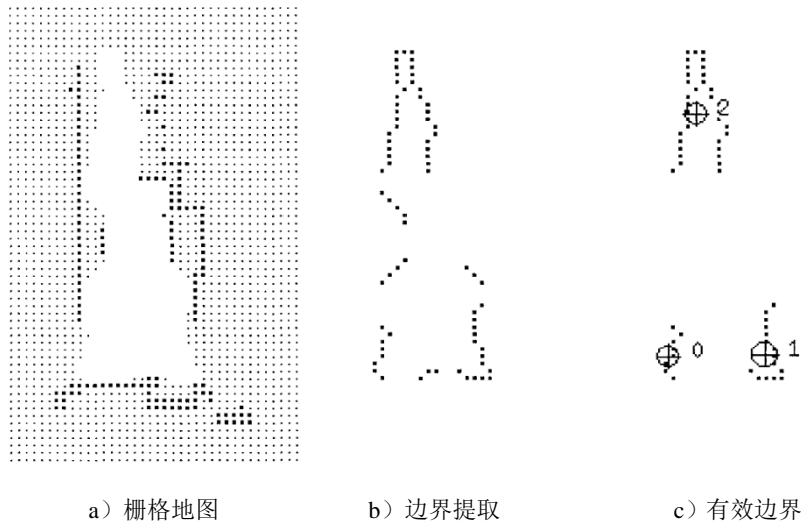


图3-6 边界提取^[47]

一旦在栅格地图中检测到边界，机器人就会尝试导航到最近的、未访问的边界。该算法会作为路径规划器的输入，为其提供导航目标点，之后由路径规划模块计算出一条无碰撞的路径并控制机器人沿着该路径到达目标点。由于室内环境是一种动态环境，因此路径规划模块还需要控制机器人躲避那些原本不在地图中的障碍物，这是避免与四处走动的人发生碰撞的必要条件。

当机器人到达目的地时，该位置将被添加到已访问的边界列表中。如果机器人无法向其目的地前进，经过一段时间后，机器人将确定目的地不可访问，并且其位置将被添加到无法访问的边界列表中。

机器人实时的进行传感器扫描，不断更新栅格地图，重新检测边界以及发布目标点，之后再尝试导航到最近的可访问的，未访问的边界。通过不断循环以上过程，直至访问完当前环境所有边界，完成当前环境的地图构建。

基于边界的探索策略的核心思想是让机器人移动到边界去获得更多的新信息。当机器人朝向边界移动时，它可以探测到未开发的区域并将其信息加入到地图中。随着机器人朝着边界前进，地图将不断扩大，边界也将继续向外扩张，新的边界将始终在远处，并提供一个新的探索目的地。通过这种方式，使用基于边界探索的机

机器人最终将探索完所有区域。

3.3.2 主动 SLAM 技术架构

目前大多数学者的地图构建方式都是采用人为遥控、或者采用机器人跟随人等被动的建图方式。本文将基于边界的探索设计一种自主建图方式，将路径规划、地图构建结合在一起，通过设置目标区域，让机器人在该区域内自主探索，完成该区域的地图构建。图 3-7 为整个主动 SLAM 的架构图，从中可以看出主动 SLAM 包含了 SLAM 模块，路径规划模块，主动探索模块三个部分。

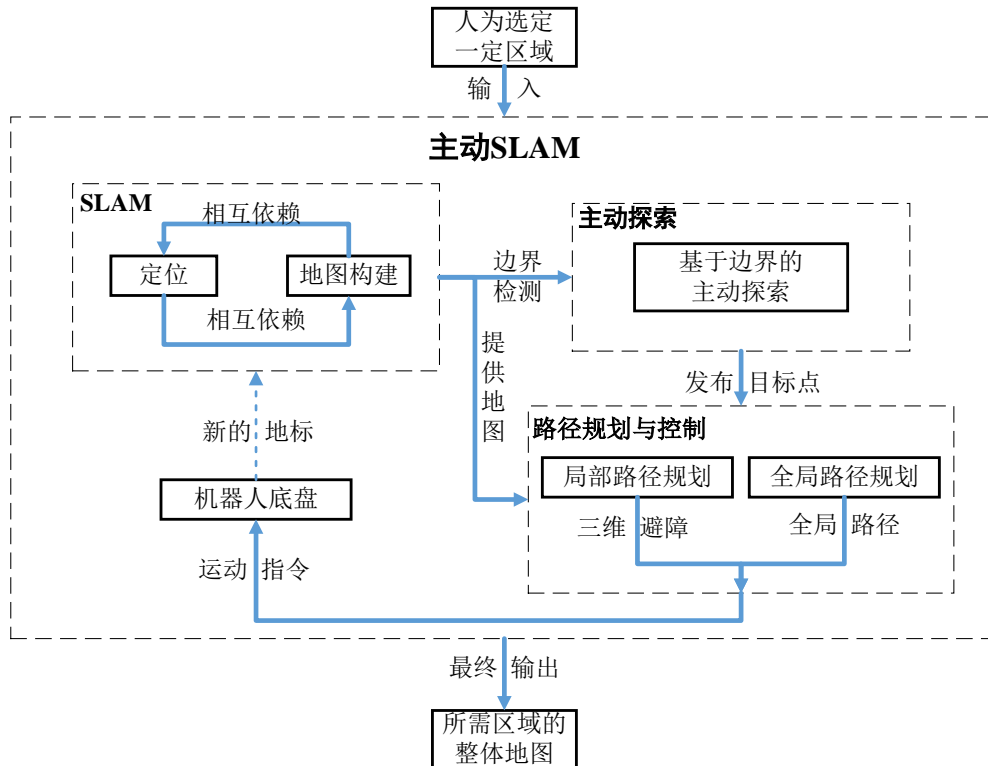


图3-7 主动 SLAM 技术整体架构

SLAM 模块完成实时的地图构建任务，为主动探索模块提供地图输入。主动探索模块检测该地图边界，并且根据边界选取策略选取下一个要探索的边界，同时为路径规划模块发布导航目标点。路径规划模块根据已经构建好的部分环境地图以及主动探索模块发布的目标点计算路径，并在运动过程中根据 Kinect 相机进行实时的避障。主动 SLAM 部分最终输出的是指定区域的环境地图。

整个主动 SLAM 系统的输入为一个区域的边框，人为的通过 Rviz 可视化仿真软件输入进去，将想要获得地图的区域框选出来，该区域由几条封闭的直线段组成，可以为矩形、五边形等形状。搭配了主动 SLAM 系统的机器人将自主把目标区域

的地图构建完整,最终得到一个可以供后续定位、导航任务使用的二维栅格地图。

3.4 本章小结

本章首先对基于图优化理论的 SLAM 算法进行研究,包括扫描匹配、闭环检测以及位姿图优化三个部分;之后对基于边界的主动探索模型进行研究,根据该模型设计了主动 SLAM 技术,使得机器人能够自主地完成地图构建,最后对主动 SLAM 技术的整体结构进行分析,阐述了主动 SLAM 技术各个组成部分的联系与功能。

第4章 移动机器人的定位与路径规划算法研究

4.1 引言

前述章节解决了移动机器人的导航问题中的地图构建问题。机器人拥有了环境地图就可以根据该地图进行定位与路径规划，从而完成各种导航任务。定位即在已知环境地图中如何确定机器人位姿，通常被称为位姿估计；有了良好的定位之后，机器人可以根据计算出的路径精准的完成导航任务。路径规划分为基于已知地图的全局路径规划与基于传感器数据的实时局部路径规划两个部分。全局路径规划根据已知地图计算出一条可以安全到达目标点的最短路径。局部路径规划控制机器人沿着全局路径规划移动，同时根据传感器信息进行实时的避障。

本章将分别对定位算法、全局路径规划以及局部路径规划算法理论过程进行研究，并通过仿真实验进行验证；同时为解决二维激光雷达由于视角问题导致室内环境下导航不安全的问题，设计了三维避障策略，实现室内复杂环境下的三维避障。

4.2 移动机器人的定位算法研究

4.2.1 蒙特卡洛定位算法研究

定位问题根据初始位姿是否已知分为位姿跟踪全局定位。初始位姿已知的定位问题被描述为位姿跟踪，也称为局部定位，即能够在已知初始位姿的条件下一直准确跟踪自身位姿的问题。全局定位问题的初始位姿未知，机器人首先要在整个环境地图下确定自身初始位姿，之后再进行位姿跟踪。机器人在运动过程中可能会遇到“绑架问题”，即突然被移动并放到了其它位置，机器人很可能认为自己处在一个错误的位置上。

图 4-1 描述了移动机器人定位问题的图形模型，图中阴影节点代表已知量：地图 m ，测量值 z 和控制量 u 。定位即为在已知地图 m 中根据传感器数据 z 与控制量 u 推断机器人在已知地图中的位姿 x 。

蒙特卡洛定位算法（Monte Carlo Localization, MCL）可以解决上述问题，它应用粒子滤波器（Particle Filter）来估计机器人位姿的后验概率密度。尽管这个算法推出时间还比较短，但是由于它非常容易实现，已经成为机器人领域最流行的定位算法之一。

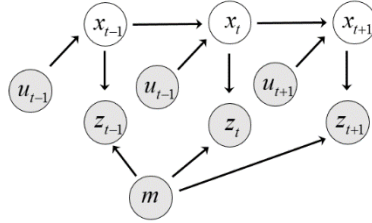


图4-1 移动机器人定位的图形化模型

基础蒙特卡洛定位算法的核心思想为利用一组带权值的粒子集来表示该时刻机器人在真实世界中可能所处的位姿，每一个粒子都代表着机器人处于该位姿的概率 $bel(x_t)$ 。该算法包括粒子的初始化、运动预测、观测更新以及重采样等步骤。具体过程如下：

(1) 粒子初始化：在此阶段，移动机器人仅仅拥有当前环境地图，却不能确定自身处在该地图的哪个位置，即初始位姿未知。该算法通过在整个地图空间中随机散布一组粒子 $X_t = \{x_t^1, x_t^2, \dots, x_t^{1m}\}$ 来进行全局定位从而估计初始位姿，并将每个粒子的权值均分为 $w_t^m = M^{-1}$ 。

(2) 运动预测：根据上一时刻粒子集合 X_{t-1} 与机器人运动控制输入 u_t ，通过运动模型 $p(x_t|u_t, u_{t-1})$ 来预测下一个时刻的假设的粒子集合 X_t ，得到的新粒子的概率为 $\overline{bel}(x_t)$ 。

(3) 感知更新：通过 t 时刻的观测 Z_t 为预测得到的粒子重新分配权值，权值即为粒子 x_t^m 在观测 Z_t 下的概率，由 $w_t^m = p(z_t|x_t^m)$ 得到。这组粒子的权值类似于贝叶斯滤波中的后验概率密度。

(4) 重要性采样：它是达尔文适者生存理念的的概率实现。该步骤以每个粒子的权值作为概率重新对 X_t 中的 M 个粒子进行散布，权值高的再次出现的几率大，权值低的出现的几率小。通过将重要性权值合并到重采样过程中，粒子的分布发生了变化：重采样之前，粒子根据 $\overline{bel}(x_t)$ 分布；重采样之后，粒子根据 $bel(x_t) = \eta p(z_t|x_t^m)\overline{bel}(x_t)$ 进行分布。通过丢弃了那些可能性低的粒子，并复制可能性高的粒子，通过这种方式使得最终留下来的粒子都是最接近真实位姿的粒子。

重复第 2-4 步，随着不断接收到新的传感器数据，粒子将聚集在一个较小的范围内，该算法将越来越有信心确定机器人在已知地图中的真实位姿。

MCL 算法通过将粒子分布在整个地图上来进行全局定位，随着机器人的不断运动，不正确的粒子不断被丢弃，最终将只剩下代表真实位姿的粒子，从而完成全局定位。在完成全局定位后，继续根据传感器数据筛选粒子，使得粒子始终聚集在一起，从而完成位姿跟踪。

4.2.2 自适应蒙特卡洛定位算法研究

基础形式的 MCL 算法解决了全局定位问题，但无法从机器人“绑架”问题或者全局定位错误中恢复过来。由上一节可知，随着机器人移动，粒子逐渐聚集在最有可能的位置上，其它地方的粒子逐渐消失。如果这个最有可能的位姿不正确，那么该算法将无法恢复与修正。而且，实际上这种随机算法也可能在重采样过程中将正确位姿附近的粒子丢弃，当粒子的数量很少时或者粒子分布在大面积时（如全局定位），这个问题尤其重要。随着重采样的进行，粒子将逐渐耗尽，导致粒子耗尽问题。

通过在每次迭代中添加随机粒子可以解决上述问题，并且这种随机粒子还可以解决“绑架”问题。它将在运动模型中产生一个小部分的随机状态，增强了整体算法的鲁棒性。添加随机粒子需要解决两个问题：首先，在算法的每次迭代中应该添加多少个粒子，其次，应该按照哪种分布生成这些粒子。

第一个问题可以通过监控传感器测量的概率并将其与平均测量概率解决。由于粒子的权值即为观测概率的随机估计，因此，随机粒子的数量的近似值可以通过将粒子的权值与总体的观测概率进行匹配来获得，如式（4-1）所示^[12]：

$$\frac{1}{M} \sum_{m=1}^M w_t^m \approx p(z_t | z_{1:t-1}, u_{1:t}, m) \quad (4-1)$$

第二个问题可以根据姿态空间上的均匀分布绘制粒子，并用当前观测值对粒子进行加权。

上述描述了基本的蒙特卡洛定位算法的改进方法。通过添加随机粒子的 MCL 算法改进了基本 MCL 算法，被称为自适应的蒙特卡洛定位算法（Adaptive Monte Carlo localization algorithm, AMCL），图 4-2 显示了其伪代码过程。该算法是自适应添加粒子的，因为它追踪似然函数 $p(z_t | z_{1:t-1}, u_{1:t}, m)$ 的短期和长期平均值。其第一部分与 MCL 算法相同：使用运动模型（第 5 行）从旧粒子中采样新粒子，并根据测量模型（第 6 行）设置其重要性权重。AMCL 算法第 8 行计算了历史观测的平均权值，并在第 10 行和第 11 行保持这种平均权值的短期和长期平均值。参数 a_{slow} 和 a_{fast} 是分别用于估计长期和短期平均值的指数滤波器的衰减因子，且要求 $0 \leq a_{slow} \ll a_{fast}$ 。

该算法的自适应性由在第 13 行中提供：在重采样过程中，随机粒子的概率由式（4-2）提供：

$$\max\{0.0, 1.0 - w_{fast}/w_{slow}\} \quad (4-2)$$

式中 w_{fast} ——观测信息的短期平均权值；

w_{slow} ——观测信息的长期平均权值。

```

1:  Algorithm Augmented_MCL( $\mathcal{X}_{t-1}, u_t, z_t, m$ ):
2:      static  $w_{slow}, w_{fast}$ 
3:       $\bar{\mathcal{X}}_t = \mathcal{X}_t = \emptyset$ 
4:      for  $m = 1$  to  $M$  do
5:           $x_t^{[m]} = \text{sample\_motion\_model}(u_t, x_{t-1}^{[m]})$ 
6:           $w_t^{[m]} = \text{measurement\_model}(z_t, x_t^{[m]}, m)$ 
7:           $\bar{\mathcal{X}}_t = \bar{\mathcal{X}}_t + \langle x_t^{[m]}, w_t^{[m]} \rangle$ 
8:           $w_{avg} = w_{avg} + \frac{1}{M} w_t^{[m]}$ 
9:      endfor
10:      $w_{slow} = w_{slow} + \alpha_{slow}(w_{avg} - w_{slow})$ 
11:      $w_{fast} = w_{fast} + \alpha_{fast}(w_{avg} - w_{fast})$ 
12:     for  $m = 1$  to  $M$  do
13:         with probability  $\max\{0.0, 1.0 - w_{fast}/w_{slow}\}$  do
14:             add random pose to  $\mathcal{X}_t$ 
15:         else
16:             draw  $i \in \{1, \dots, N\}$  with probability  $\propto w_t^{[i]}$ 
17:             add  $x_t^{[i]}$  to  $\mathcal{X}_t$ 
18:         endwith
19:     endfor
20:     return  $\mathcal{X}_t$ 
    
```

图4-2 AMCL 算法伪代码过程图^[12]

通过测量权值的短期和长期平均值来决定随机粒子的数量。如果观测信息的短期平均权值大于或等于长期平均权值($w_{fast}/w_{slow} \geq 1$)，则不会添加随机样本；如果短期平均权值比长期平均权值小($w_{fast}/w_{slow} < 1$)，则需要增加随机粒子，增加的随机粒子数量与他们之间的商成正比。当机器人被绑架时，观测信息的短期平均权值 w_{fast} 会突然减少，使得增加粒子的比例变大，迭代过程中增加较多的随机粒子，从而根据环境信息解决了机器人的“绑架问题”^[48]。

AMCL 算法是对 MCL 算法的改进，同样需要进行采样、运动预测、权值更新、重采样等步骤。该算法与基本蒙特卡洛定位算法相比解决了全局定位失败的恢复问题以及机器人被绑架的问题，由于增加的随机粒子，使得该算法不存在粒子耗尽问题，增加了算法的鲁棒性与稳定性。

因此，本文采用自适应的蒙特卡洛定位算法作为机器人的定位算法。

4.2.3 定位算法仿真实验

完成了室内环境下的地图构建任务之后，要在所构建的地图上进行导航，首要

任务就是要确定机器人在已知地图上的位置。因为，路径规划模块是通过已知位姿计算路径的，如果自身处在地图中位置与实际不符，继续沿着路径前进可能会导致导航不正确。因此，只有确定了自身在地图中的位置，路径规划模块计算的路径才是有意义的。

本节将进行定位算法的仿真实验。应用 ROS 下的仿真工具 Stage 进行仿真环境的搭建，仿真所用传感器为二维激光雷达，来验证基于粒子滤波理论的自适应蒙特卡洛算法的定位能力。

AMCL 算法通过将粒子分布在整个地图上来进行全局定位，随着机器人的不断运动，不正确的粒子不断被丢弃，最终将只剩下代表真实位姿的粒子，从而完成全局定位。只是这个过程需要较长的时间，因此该算法在程序实现的过程中做了改进，通过人为指定一个不是特别精确的初始位姿来对粒子的散布进行影响，将粒子主要分布在给定位姿附近，从而大幅度缩短全局定位的时间，使之较快的完成定位过程。

实验过程图如图 4-3 所示，图中的小箭头即为粒子，每个粒子都代表着一个可能的机器人位姿，箭头的起始点为机器人的位置，方向为机器人的朝向。图 4-3 a) 展示了给定初始位姿之后的粒子状态，可以看到，此时代表机器人可能位姿的粒子分布的非常松散，说明在此时刻定位算法还不能得到一个较准确的位置信息。

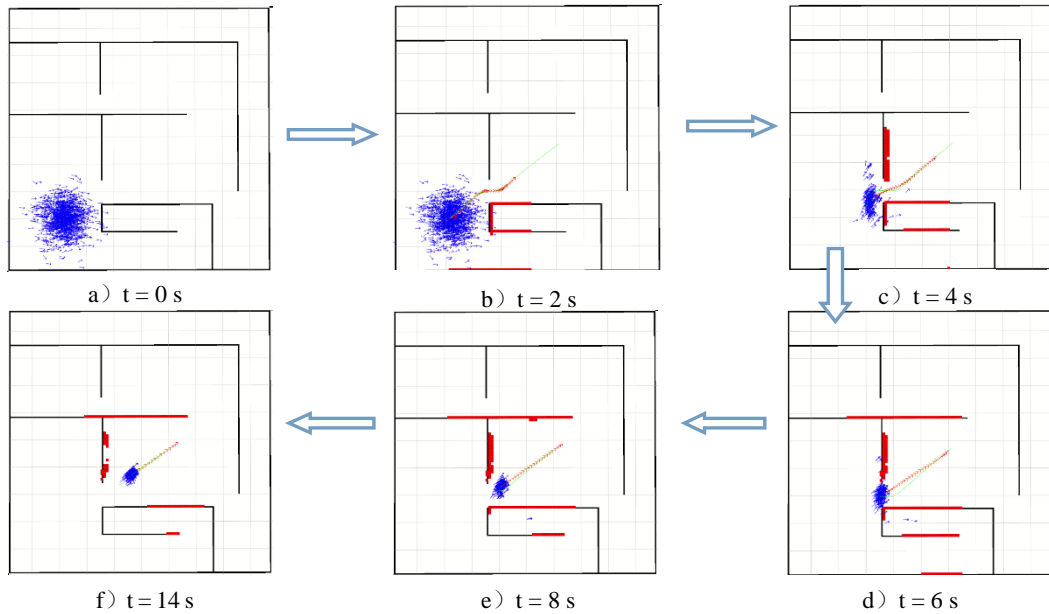


图4-3 定位算法的仿真实验

图 4-3 b) 到图 4-3 e) 为粒子收敛的过程，从中可以看到，随着机器人不断感知环境，粒子的分布范围越来越小，粒子逐渐收敛于较小的范围内。在每次迭代中，代表错误位姿的粒子不断被淘汰，而代表真实位姿的粒子被留下，直到最后只留下

了正确的粒子使得粒子收敛在一个较小的范围内并保持稳定，如图 4-3 f) 所示，说明此时定位算法有非常大的把握可以确定机器人在地图中的真实位姿。

定位算法的仿真实验表明：在给定已知地图后，AMCL 定位算法在感知环境一段时间后，粒子集能够收敛在一个较小的范围内，证明了 AMCL 定位算法具有在地图中找到机器人真实位姿的能力，

4.3 全局路径规划算法研究

4.3.1 全局路径规划算法

上述节解决了机器人的定位问题，机器人有了准确的位姿估计，就可以根据该位姿进行路径规划，从而精确的到达目标点。路径规划分为全局与局部两个部分，全局路径规划根据已知地图提供一个全局的可以安全到达目标点的最优路径。本节将对路径规划算法 A* 进行研究。

Dijkstra 算法是从起始点逐渐向外扩展搜索，它将以起始点为中心逐层地向外扩展，直至找到目标点，并对从起始点到目标点的实际距离进行排序，从而找到最短路径。

Greedy Best-First 算法只考虑当前的最优解，优先搜索最接近目标的位置，所以它能较快的找到路径，但是由于它是对距离目标点的估计距离进行排序，因此它找到的将不是最短路径。

A* 算法是由 Hart 等学者在 1968 年在 Dijkstras 算法的基础上提出的，结合了 Dijkstras 算法与局部最优算法 (Best-First Search, BFS) 的优点，是目前最为流行的一种图搜索算法之一。

A* 算法是一种典型的启发式搜索算法，它将启发信息融入到估价函数中，根据代价 (距离最短、时间最短等) 最小或者接近最小生成到目标点的路径。该算法的估价函数 (见式 4-3) 使 A* 算法既可以寻得最优路径，也可以利用启发式搜索来减少搜索时间。

$$f(n) = g(n) + h(n) \quad (4-3)$$

式中 $g(n)$ ——为从起始节点到当前节点 n 的代价值；

$h(n)$ ——为从当前节点 n 到目标点的代价值，又称为启发函数；

$f(n)$ ——为从起始节点到当前节点 n 的估价函数值。

A* 算法在每次迭代中根据式 4-3 找到 $f(n)$ 值最小的点，并将该节点作为下一次搜索的起始点。该算法具体过程如下：

首先，该算法建立两个集合，分别称为 OPEN 集合与 CLOSE 集合，并将起

始点加入 OPEN 集合中。CLOSE 集合为不需要再次考虑的节点的集合，OPEN 集合是一个待检查的节点集合。将从其中寻找每次循环的起始点。

重复循环如下步骤^[49]：

(1) 寻找 OPEN 集合中 $f(n)$ 值最小的节点，将它作为当前节点。

(2) 将当前节点放入 CLOSE 集合中。

(3) 检查当前节点相邻的空闲节点（即不是障碍物也不在 CLOSE 集合中），如果相邻节点不在 OPEN 集合中，则将它添加进去，并将当前节点作为它的父节点，计算由当前节点到该节点的 $g(n)$ 值与 $f(n)$ 值（由于 $h(n)$ 值始终不变，只需在最初计算一次即可）；如果相邻节点已经在 OPEN 集合中，则进行 $g(n)$ 值比较。计算由当前节点到该节点的 $g(n)$ 值，比较此时 $g(n)$ 值与该节点原先 $g(n)$ 值的大小，若 $g(n)$ 值变小，则将该节点的父节点设置为当前节点，并重新计算 $g(n)$ 与 $f(n)$ 值；若 $g(n)$ 值不变或变大，则保留该节点的父子关系以及 $g(n)$ 与 $f(n)$ 值。

(4) 当终点加入到 OPEN 集合中时，则路径已经找到，算法停止；若 OPEN 集合为空时依然没有找到终点，则路径规划失败，算法停止。找到路径时，从终点开始，每个节点沿着其父节点连线至起点，即为所需路径。

$h(n)$ 即为启发函数，它是使 A* 算法区别于 Dijkstras 算法的关键因素。根据 $h(n)$ 与从起始点到目标点间实际距离间的关系，可以得到算法的不同效果。

$h(n)$ 若为 0，则 A* 算法将退化为 Dijkstras 算法，丧失其优化的能力，可以找到最短路径，搜索空间很大，计算时间长； $h(n)$ 值若小于实际距离，A* 算法能够找到最短路径， $h(n)$ 值越小，A* 算法越接近 Dijkstras 算法，搜索空间越大； $h(n)$ 若大于实际距离时，A* 算法的搜索空间变小，可以很快找到路径但该路径不是最短路径； $h(n)$ 值若等于实际距离，则此时 A* 算法将直接沿着最短路径进行搜索，搜索空间最小，因为此时最优路径上的 $f(n)$ 值均小于其它路径上的 $f(n)$ 值，但是在各种情况下保持 $h(n)$ 等于实际距离是不现实的^[50]。

$h(n)$ 可由多种方法确定，常用的有欧式距离、曼哈顿距离以及切比雪夫距离，通过计算上述距离作为启发函数的值，从而对 A* 算法进行搜索方向的引导。

4.3.2 全局路径规划算法仿真实验

A* 算法结合了 Dijkstras 算法与局部最优算法的优点，现将通过仿真实验对三个算法进行对比分析，以验证 A* 算法的效果。

仿真实验如图 4-4 所示，图中五角星表示起始点，X 形状表示目标点，二者中间设置了一个 U 型的墙壁以验证算法在处理局部最小值时的能力，带有数字的方块表示算法的搜索空间，浅色方块表示算法最终找到的路径。方块中的数字为各自

算法的依据, Dijkstras 算法的数字代表距离起始点的距离值, 局部最优算法的数字代表距离终点的距离值, A*算法的数字代表由起点到搜索点的距离与启发函数值之和。

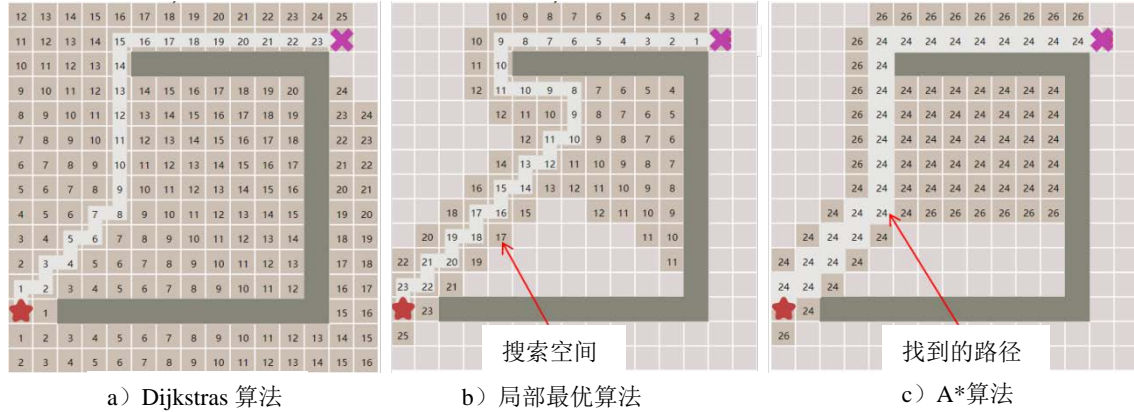


图4-4 路径规划算法仿真实验

从图 4-4 中可以看出, Dijkstras 算法搜索空间非常大, 可以找到最短路径。Greedy Best-First 算法的搜索空间较小, 但并没有找到最短路径, 只是向着最接近目标的方向搜索。A*算法的搜索空间较小, 同时也找到了最短了解, 证明 A*算法的优越性。

4.4 局部路径规划算法研究

在室内导航过程中很可能突然出现些原本地图中不存在的静态障碍物或者动态障碍物, 如桌子或椅子的位置变动、行人的突然出现, 使得机器人再沿着全局路径规划前进变得不安全。局部路径规划就是为了解决这个问题而产生的。

局部路径根据传感器的信息实时地生成接下来几秒前进的路径, 且更新频率很高。局部路径让机器人沿着全局路径前进, 在遇到突然出现的障碍物时局部路径重新规划, 生成绕过障碍物的局部路径使机器人进行避障, 从而安全地到达目标点。

4.4.1 局部路径规划算法

目前主流的局部路径规划算法为动态窗法 (Dynamical Window Approach, DWA), DWA 算法将速度和加速度的限制设置到搜索窗口中, 通过对各个速度与加速度评分从而得到安全可靠的路径。但是这种算法的计算复杂度较高, 实时不好, 难以实现动态环境下的避障。因此, 本文应用一种新颖的局部路径规划算法时间弹性带算法 (Timed Elastic Band, TEB) 以实现稳定的动态环境下的避障。

弹性带算法 (Elastic Band) 最初由 Richard Durbin 与 David Willshaw 为解决旅

行商问题 (Traveling Saleman Problem) 于 1987 年提出。该算法将规划处的路径假设为一条具有弹性的弹性带，带子是由很多点依次相连组成。带子上的每个点都受到三个力的作用，一个是城市点对该点的拉力，使弹性带上的点向城市移动，以及由于弹性带的弹力产生的该点两侧的点对该点的拉力，这两个力使弹性带的长度尽量缩短^[51]。弹性带上的每个点都受这两种力的作用，为了描述这两种力的作用与关系，该算法定义了一个能量函数，通过最小化能量函数来求得最优解。

之后弹性带算法被成功应用在机器人的路径规划问题上，众多学者对该算法进行了优化，其中 Christoph Rösmann 等学者基于弹性带算法提出的时间弹性带算法由于其优越的性能受到了广泛的关注。

经典的弹性带路径规划算法使全局路径规划产生的最短路径变形，同时避免与障碍物接触，它不直接考虑底层机器人的任何动态约束。而时间弹性带算法 TEB 为弹性带算法附加了时间信息，根据动态约束（如有限的机器人速度和加速度）通过将运动时间最小化来得到最快路径。

TEB 算法将 n 个机器人姿态 $x = (x_i, y_i, \beta_i)^T$ 表示成式 (4-4) 所示：

$$Q = \{X_i\}_{i=0 \dots n} \quad (4-4)$$

并且，它将每两个姿态之间用时间差 ΔT_i 来连接，见式 (4-5)，每个时间差表示从一个姿态转换到下一个姿态所用的时间，TEB 结构如图 4-5 所示。

$$\tau = \{\Delta T_i\}_{i=0 \dots n-1} \quad (4-5)$$

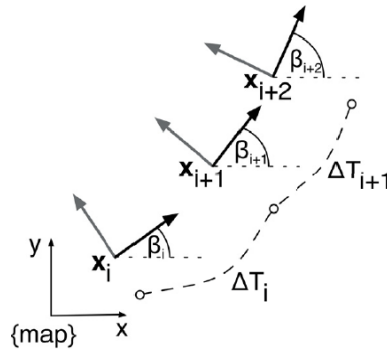


图4-5 TEB 结构图^[52]

TEB 算法的核心思想是根据姿态和时间间隔通过实时加权多目标优化来优化 TEB，从而得到最优轨迹。该算法首先将各种约束，如速度与加速度、全局路径与障碍物、非完整约束的运动学以及运动时间，通过式 (4-6) 进行加权求和得到总体约束；再将整体目标函数转化为超图，通过图优化的统一框架 g2o 进行总体优化，求得最优解 B^* ；经过优化的 TEB 将被验证是否违反了严格的约束条件，如果违反了约束条件，机器人就要停止运动或者重新进行路径的计算，若不违反约束条件则发布运动指令。

$$\begin{aligned}
 B &:= (Q, \tau) \\
 f(B) &= \sum_k \gamma_k f_k(B) \\
 B^* &= \arg \min_B f(B)
 \end{aligned} \tag{4-6}$$

式中 B ——代表“时间弹性带”（TEB）；

$f(B)$ ——加权多目标总体的目标函数；

B^* ——优化后的 TEB。

优化的 TEB 经过验证后，TEB 算法通过图 4-5 所示的位姿与时间关系，根据式（4-7）计算出机器人的平均线速度与平均角速度，并作为运动命令发送给机器人。在每次新的迭代之前，初始化阶段检查全局路径以及新的障碍物，从而保证算法的实时性。

$$\begin{aligned}
 v_i &\simeq \frac{1}{\Delta T_i} \left\| \begin{pmatrix} x_{i+1} - x_i \\ y_{i+1} - y_i \end{pmatrix} \right\| \\
 \omega_i &\simeq \frac{\beta_{i+1} - \beta}{\Delta T_i}
 \end{aligned} \tag{4-7}$$

仿真和实验的结果表明，该方法在实时生成最佳机器人轨迹方面具有鲁棒性和较高的计算效率。因此，本文采用 TEB 算法作为局部路径规划以及运动控制的算法。

4.4.2 基于点云的避障策略

目前国内大多数学者采用的避障方式为基于激光雷达的避障方式，二维雷达只能扫描到某个平面上的障碍物，然而由于室内环境复杂，充满着各种变截面物体。基于激光雷达的二维避障方式仅能针对物体的一个截面进行避障，而检测不到该物体其它截面的形状，使得机器人的导航变得不安全。基于激光雷达的二维避障的缺点如图 2-1 所示。而 RGB-D 相机 Kinect 可以观测到前方 1-4 米，垂直方向 43.5° 内，水平视角 57.5° 内的所有障碍物，能检测到变截面物体的各个截面形状，因而将其应用于障碍物检测可以实现室内复杂环境下的三维避障。

因此，为解决二维激光雷达在室内环境下由于视角原因导致导航失败的问题，本文采用了一种三维避障策略，使用 Kinect 相机作为避障传感器来进行复杂环境下的三维避障。通过独立设计编写的将深度数据转化为三维点云的程序，将三维点云输入到 ROS 中的 Navigation 导航堆栈框架中，该软件包将得到的三维点云转化供机器人路径规划可用的代价地图，之后路径规划模块根据代价地图计算出全局路径规划与局部路径规划，最终成功实现复杂环境下的三维避障。

由 Kinect 相机的深度数据转换得到的三维点云如图 4-6 所示，该图左下角为

机器人上的 RGB 相机照的实景图像，右侧为三维点云，图中颜色的变化代表着不同的深度值。从图 4-6 中可以看到，Kinect 相机可以检测到前方视角范围内的所有障碍物，包括了地面与墙面。

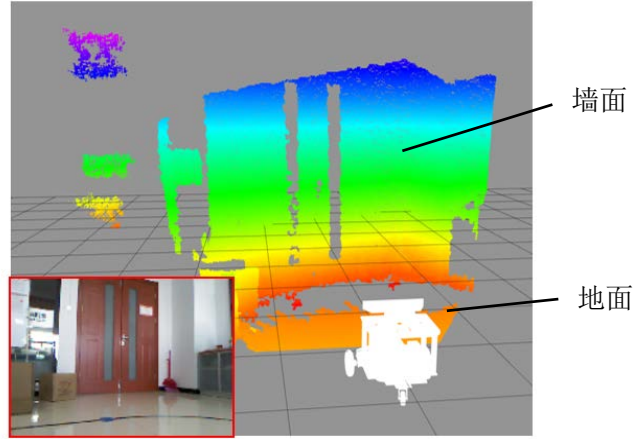


图4-6 由深度数据转换得到的三维点云

4.4.3 路径规划模块仿真实验

为了验证路径模块的效果，应用 ROS 下的仿真工具 Stage 进行仿真环境的搭建，对路径规划模块进行仿真分析。仿真地图为结构化的室内地图，仿真所用机器人为差速机器人。实验过程如图 4-7 所示，通过给定机器人目标点位置与姿态后，路径规划模块成功的计算出了全局路径以及局部路径。

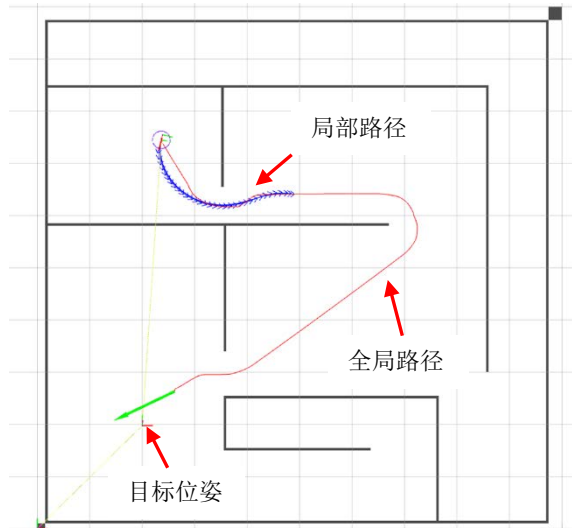


图4-7 路径规划仿真实验

从图 4-7 中可以看出全局路径紧贴障碍物，为最短路径；而局部路径规划算法考虑了机器人当前朝向，速度控制等动态约束，对机器人的轨迹进行实时地优化，

使机器人实现最快速的到达目标点，并在没有动态障碍物的情况下紧跟全局路径。

通过该仿真实验验证了整体路径规划算法的可行性与实用性：全局路径规划能够准确规划出一条安全无碰撞的最短路径，局部路径规划能够快速根据机器人的各个约束计算出实时地最快路径。

4.5 本章小结

本章首先对 AMCL 算法进行研究，并成功应用该算法解决机器人的全局定位与局部位姿跟踪问题。之后分别对全局与局部路径规划算法进行研究。应用了一种新型的局部路径规划算法，TEB 算法，实现了实时地轨迹优化；采用一种基于三维点云的避障策略，成功地实现三维避障；并对整体路径规划模块进行仿真分析与验证，验证了路径规划模块的有效性与可行性。

第5章 移动机器人导航系统的实验验证

5.1 引言

本文前面章节已经详细研究了导航系统中的各个组成模块的算法原理、实现过程以及整体架构，本章将对各个模块进行实验测试，验证各个模块的效果。分别进行主动 SLAM 实验；进行两种 SLAM 算法的建图效果对比，并对地图精度进行分析。之后为验证定位及路径规划模块的效果，进行自主导航实验，并对导航精度进行分析。最后，通过静态避障与动态避障实验验证导航系统静态与动态环境下的避障能力。

5.2 室内环境的地图构建实验

5.2.1 主动 SLAM 实验

为验证主动 SLAM 技术的可行性，应用实验平台进行了室内环境下的主动 SLAM 实验，实验场景为室内某房间环境，如图 5-1 所示。为了证明 SLAM 算法在复杂环境下的建图性能，选取了一个复杂、凌乱的环境，可以看到，整体环境虽然较为结构化，但是存在较多椅子、桌子、文件柜等不同形状尺寸的物品，使得整体环境十分复杂。



图5-1 某房间环境图片

本实验采用主动 SLAM 技术让机器人自主构建室内环境的二维栅格地图。图 5-2 a) 为主动 SLAM 实验的过程图。从中可以看出，在人为划定一个需要构建地图的区域后，主动 SLAM 技术使得机器人自主向未知区域前进，并且不断构建未知区域的环境地图。

主动 SLAM 技术通过边界探索模块检测已有地图边界并向路径规划模块发布

目标点，使得机器人从其它位置前往该区域。通过不断向着边界移动，SLAM 算法不断构建范围更大的地图，边界也将不断扩张，机器人最终将构建所规定区域的全部地图。

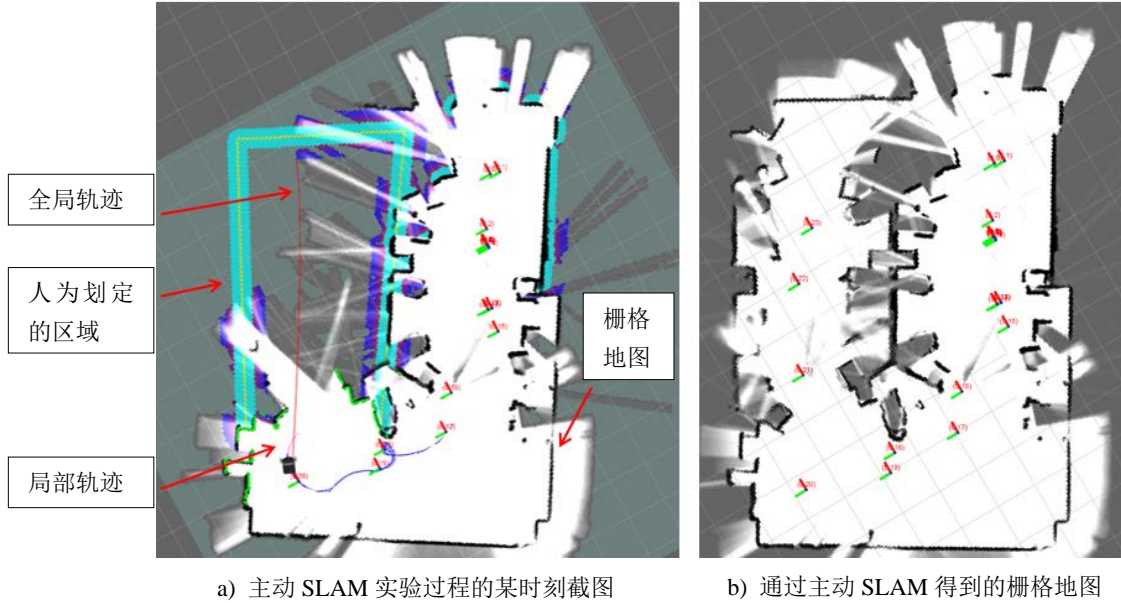


图5-2 主动 SLAM 技术实验

通过主动 SLAM 技术构建出来的地图如图 5-2 b) 所示。在整个地图构建过程中，人为影响因素仅仅是划定了一片区域，以告知机器人哪里需要建图，剩下的过程全都是由机器人自主完成。SLAM 模块负责构建地图，边界探索模块负责检测已有地图边界并发布目标点，路径规划模块负责计算到达该目标点的路径并进行避障与控制。而在机器人不断运动过程中，SLAM 算法又可以构建更远处的地图。三个模块相互作用，相互影响，形成了一个完整的闭环，使得机器人的自主建图成为了可能，由此验证了主动 SLAM 技术的可行性。

5.2.2 SLAM 算法对比实验

通过与 GMapping-SLAM 算法进行构建地图对比实验，以对本文采用的 Cartographer-SLAM 算法所建地图的质量进行分析。

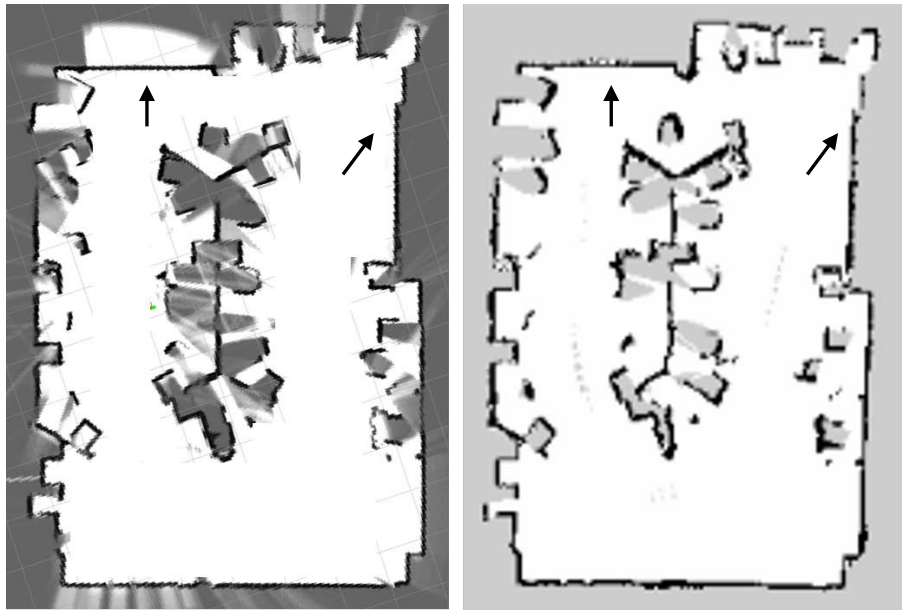
GMapping-SLAM 算法是开源的二维 SLAM 算法，该算法首先通过因式分解将定位与地图构建分解开来，应用粒子滤波理论进行位姿估计之后再行地图构建。由于其非常容易进行程序实现，一经发表就受到了非常大的关注，已经成为最受欢迎的 SLAM 之一。

为了控制变量，应用 ROS 中自带的录制工具 rosbag 进行激光雷达数据与里程计数据的录制，通过人为遥控机器人环绕图 5-1 所示的室内复杂环境运动一周

得到传感器数据集。之后分别应用 Cartographer-SLAM 算法与 GMapping-SLAM 算法对该数据集进行离线建图对比实验。由于算法接收的数据是相同的，所以通过两种算法得到地图的效果是两种算法效果的直接体现。

图 5-3 是分别用 Cartographer-SLAM 算法与 GMapping-SLAM 算法通过同样的数据集进行离线建图得到的二维栅格地图，两个地图的分辨率均为 5cm（图中细线组成的方格边长为 100cm，用于确定地图尺寸）。

通过对比两张地图可以看出：虽然实验环境十分复杂，二者都可以得到完整的栅格地图，但是显然 Cartographer-SLAM 算法得到的地图效果更好，所建地图的细节更加清晰，而 GMapping-SLAM 算法得到的地图较为粗糙。图 5-3 中的箭头指向了两算法效果差异较大的地方，可以看到图 5-3 a) 对于四周墙壁表达的更加接近直线，且相邻两边更加近似垂直，而图 5-3 b) 中的两条边出现了明显的偏差，使得本该垂直的两条边的夹角明显小于 90° ，并且直线表达的也不是平直。



a) Cartographer-SLAM 算法

b) GMapping-SLAM 算法

图5-3 室内环境下不同算法所建地图对比

实验场景是室内环境，属于典型的结构化场景，只不过多了椅子、柜子等物品使得整体结构不是那么明显。该房间形状属于矩形形状，相对的两条边很接近平行。从图 5-3 b) 中可以看出地图发生了整体的向右偏转，使得地图左右两边出现了明显的倾斜，而图 5-3 a) 中左右两边结构十分清晰，相邻两条边更加接近直角，两条相对的边也更加接近平行，与实际情况更加接近。

出现这种情况的原因是由于两算法在累计误差优化方面的差异。SLAM 算法在构建地图时由于传感器误差、系统建模误差等等的因素，在每次迭代中都会产生

一定的误差，误差会随着时间而累积，最终导致出现图 5-3 b) 中明显的整体倾斜。
Cartographer-SLAM 算法不断的通过前端的闭环优化与后端的全局优化消除累积误差，使得算法的鲁棒性更好，精度更高，从而得到了一致性更好的地图。

本节的对比实验证明了本文采用的 SLAM 算法的优越性，接下来将通过与实际的地图进行数据上的对比分析以确定该 SLAM 算法的精确程度。

5.2.3 地图精度分析

对实际环境与地图中的对应尺寸分别进行测量,通过二者的差值来确定 SLAM 算法构建的地图的精度。对真实环境的 6 个特征尺寸进行手动测量得到实际尺寸。地图中的对应尺寸由地图中的细线方格确定，细线方格为边长 1 米的正方形，根据对应比例可以确定对应尺寸。本次实验在实际环境中选取了 6 个具有代表特征的距离值，由于该房间短边的特征较为复杂且容易测量，在短边上选取 4 个特征尺寸，长边选取 2 个特征尺寸。该房间实际尺寸以及尺寸的分布如图 5-4 所示。

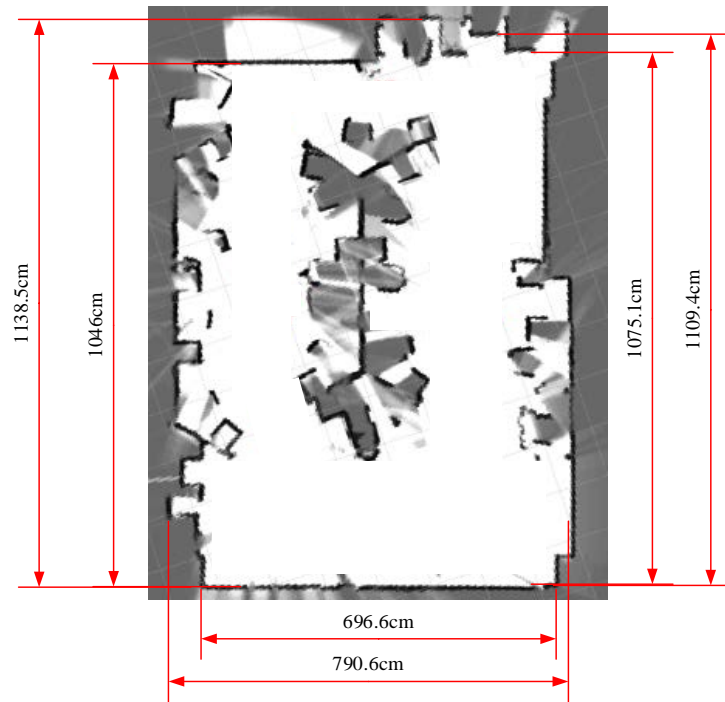


图5-4 室内环境某房间的实际尺寸

通过对比测量得到的数据如表 5-1 所示，该表展示了 6 组环境实际尺寸与地图对应尺寸偏差的大小。实际尺寸是通过测量两个墙面之间的距离得到的；由于栅格地图上的障碍物有厚度这一属性，本次实验中将障碍物厚度的中间值当做测量点，以得到地图尺寸偏差的平均值。

由表中数据可以看出，地图尺寸与实际尺寸最大偏差为 7.53cm，最小偏差为 2.36cm，平均偏差为 4.78cm，表中的偏差值十分接近，不存在偏差很大的情况，证明了该算法的稳定性。

实验房间整体面积为 80 平方米左右。Cartographer-SLAM 算法在此面积下的地图平均误差能够达到 5cm 左右，证明了该地图的精度较高。这个精度已经达到了当今 SLAM 领域的领先水平，可以达到导航的要求。

表5-1 地图精度分析

次数	实际尺寸/cm	地图尺寸/cm	绝对误差/cm
1	696.60	701.47	-4.87
2	790.60	796.76	-6.16
3	1046.00	1053.53	-7.53
4	1138.50	1135.59	2.91
5	1109.40	1111.76	-2.36
6	1075.10	1080.00	-4.90
	平均偏差		4.78

闭环优化以及始终在后端进行的全局优化不断地消除累积误差，全局优化将总体误差平均分布到位姿图中的每个位姿中以达到误差最小化的目的，使得地图的整体误差能够保持在一定范围内，使得该算法具有较高的准确性。

本节首先通过主动 SLAM 实验验证了主动 SLAM 技术的可行性，展示了其主动建图过程以及最终得到的地图。之后使用相同的数据集对两种 SLAM 算法进行离线建图实验，通过对比二者地图可知，本文采用的 Cartographer-SLAM 算法建图的质量好于 GMapping-SLAM 算法。最后，对比地图中的尺寸与实际尺寸间的偏差，确定了所建地图的误差与精度，满足导航任务的要求。

5.3 移动机器人的定位及自主导航实验

完成了室内环境的地图构建之后，就可以进行已知地图下的定位及自主导航任务。本节首先进行室内环境下的自主导航实验，以验证定位模块与路径规划模块的性能。其次，通过机器人到达目标点的实际位置与给定位置之间的偏差对导航精度进行分析。

5.3.1 已知地图的定位及自主导航实验

本节将通过机器人的自主导航实验对定位算法的位姿跟踪能力以及路径规划模块的路径计算能力进行分析。

自主导航实验通过 ROS 中的可视化仿真软件 Rviz 进行控制，实验步骤如下：首先加载已知地图，应用 AMCL 定位算法进行全局定位以确定机器人在地图中的位置，在定位算法的粒子收敛之后，给定机器人导航的目标位姿，验证在运动过程中定位的效果以及计算路径的效果。

室内环境下的自主导航实验如图 5-5 所示，该图展示的是经过全局定位之后的步骤，可以看到此时定位算法的粒子已经收敛，聚集在一起。在设定目标位姿之后，可以很清楚的看到路径规划模块计算的全局路径与局部路径，证明了路径规划模块的路径计算能力。路径规划模块将根据机器人位姿以及目标点位姿计算出全局路径以及局部路径并控制机器人沿着局部路径前进，使得机器人最终安全抵达目标点。

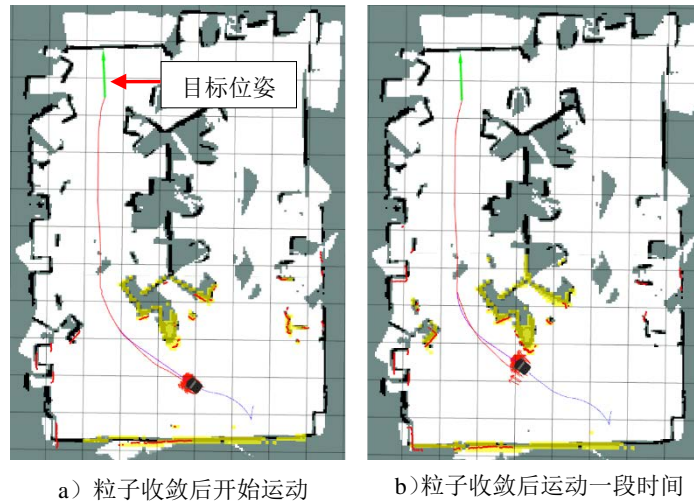


图5-5 路径规划实验

比较图 5-5 中的两张图片，发现在运动过程中，代表机器人位姿的粒子始终聚集在一起，保持在一个较小的范围内，说明定位算法能够在运动过程中较为准确地对确定机器人真实位姿，证明了 AMCL 定位算法具有准确的跟踪位姿的能力。

5.3.2 移动机器人的导航精度分析

上节通过定位及自主导航实验验证了定位与路径规划模块的性能，本节将对自主导航的精度进行实验分析。通过让机器人进行同一起始点与目标点的导航，并重复多次实验以确定自主导航的精确程度。

设定机器人的起始点位姿为 $(0,0,0)^T$ ，终点位姿为 $(1,0,0)^T$ ，该向量中前两项代表地图坐标系中的 X 值与 Y 值，第三项为机器人的位姿。

实验过程如下：第一步：给机器人发布地图坐标系下的目标点位姿 $(0,0,0)^T$ ，让其导航到起始点位置；第二步：给机器人发布地图坐标系下的目标点位姿 $(1,0,0)^T$ ，

让其导航到终点位置，如图 5-6 a) 所示；第三步：记录机器人两主动轮中心的位置与机器人的朝向，计算机器人真实位姿与目标位姿在 X、Y 方向与机器人朝向上的偏差；再次让机器人回到起始点以进行下一次实验，如图 5-6 b) 所示。

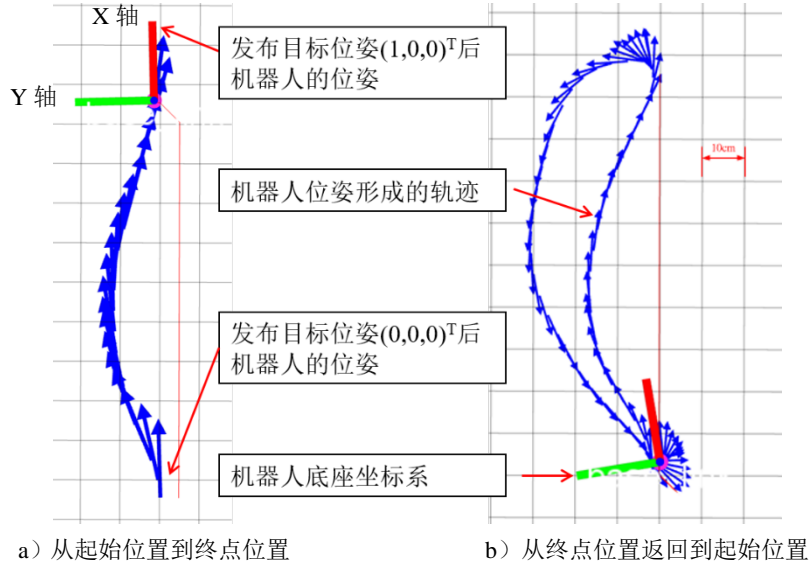


图5-6 实验过程示意图

表 5-2 展示了机器人十次导航实验记录到的偏差值，从表中可以看出，X 方向的最大偏差为 0.1 米，Y 方向的最大误差为 0.07 米，角度误差最大为 12.4 度。通过实验可以得出，机器人自主导航的整体位置偏差保持 ± 0.01 米的范围内。此实验结果说明在室内环境下，导航系统可以达到较好的导航精度，可以满足室内的服务机器人的精度要求，但是达不到工业级别的精度要求。

表5-2 导航实验测试结果

次数	X 方向偏差/m	Y 方向偏差/m	角度偏差/度
1	-0.10	-0.06	6.0
2	0.04	-0.04	12.4
3	0.05	-0.02	10.6
4	-0.05	0.02	7.7
5	-0.05	0.05	7.3
6	0.06	-0.03	12.3
7	0.03	-0.04	4.5
8	0.06	0.07	7.9
9	-0.07	0.03	6.3
10	-0.06	0.02	5.6

导致精度不是特别高的可能原因如下：

(1) 定位算法的精度不足。由前述章节可知，基于粒子滤波理论的定位算法

粒子的始终收敛在直径 20cm 圆形范围内,由算法本身性质导致的定位精度不会特别高。定位的不精确导致机器人以为已经到达目标点从而停止移动,使得停止位姿与目标位姿偏差较大。导航精度很大程度取决于定位精度,在理论上,只要定位绝对准确,机器人可以到达想去的任何位置;

(2) 实验平台的运动精度问题,导致机器人的运动始终存在一定偏差,而这种偏差在从运动到停止时的影响特别大,使得机器人的停止位置出现偏差;

(3) 导航系统参数配置的原因,有两个参数控制着机器人到达终点时在位置与角度的允许偏差值,将允许偏差设定的稍大以防止机器人在接近目标点时由于距目标点的偏差达不到要求而不停地进行原地旋转,从而导致偏差较大。

5.4 移动机器人的避障实验

室内环境十分复杂,不仅充满着尺寸不同的静态障碍物,还存在着行动的人、位置可能变动的椅子等动态障碍物,因此,导航系统不仅仅需要拥有路径规划的能力,还需要拥有躲避小尺寸障碍物以及动态障碍物的能力。本节将通过实验来验证路径规划模块的三维避障能力与动态避障能力。

为了单独验证路径规划模块的效果,将两组实验的预制地图均设置为空白地图。由于地图是空白的,机器人的导航将不存在定位问题,机器人的运动也将变为相对运动。在导航过程中,机器人可以仅通过传感器的观测进行路径的计算,而不必考虑地图中的障碍物。这使得避障实验的变量完全控制在路径规划模块上。

第一个实验为走廊环境下的三维静态避障实验,在机器人与目标点之间放置两个高度较小的障碍物,以验证路径规划模块的三维避障能力;第二个实验为动态障碍物躲避实验,在机器人导航过程中,让行人突然出现在全局路径上以验证路径规划模块的实时性及其动态避障能力。

5.4.1 移动机器人的静态避障实验

通过如下实验验证路径规划模块三维避障的能力:在机器人与目标点之间放置两个小尺寸障碍物,白色障碍物高 4.0 cm,黄色障碍物高 3.5cm,一前一后放置,两障碍物前后相距 80cm,左右相距 60cm,机器人预置在白色障碍物一侧。该实验实景如图 5-7 所示。

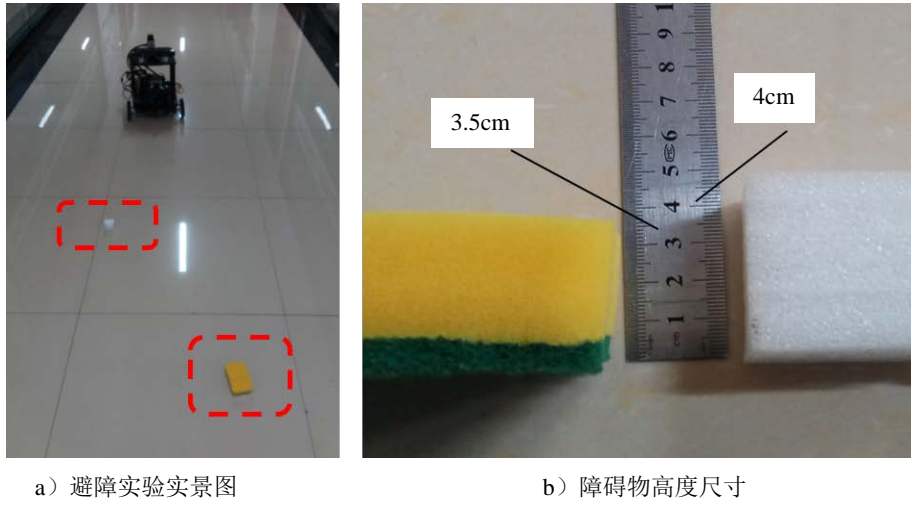


图5-7 静态避障实验实景图

图 5-8 为静态避障实验的 4 个不同时刻的截图。从控制界面可以清楚的看到机器人与障碍物在各个时刻的状态以及它躲避障碍物的动作。机器人最初直线行驶，在接近障碍物时改变方向以绕过障碍物，最终安全地到达了指定目标点。图 5-8 d) 中展示了机器人导航过程中的全部轨迹。由于两个障碍物之间的距离较远，足够机器人从两者中间通过，所以最终路径规划算法计算的路径是从两个障碍物中间穿过去。

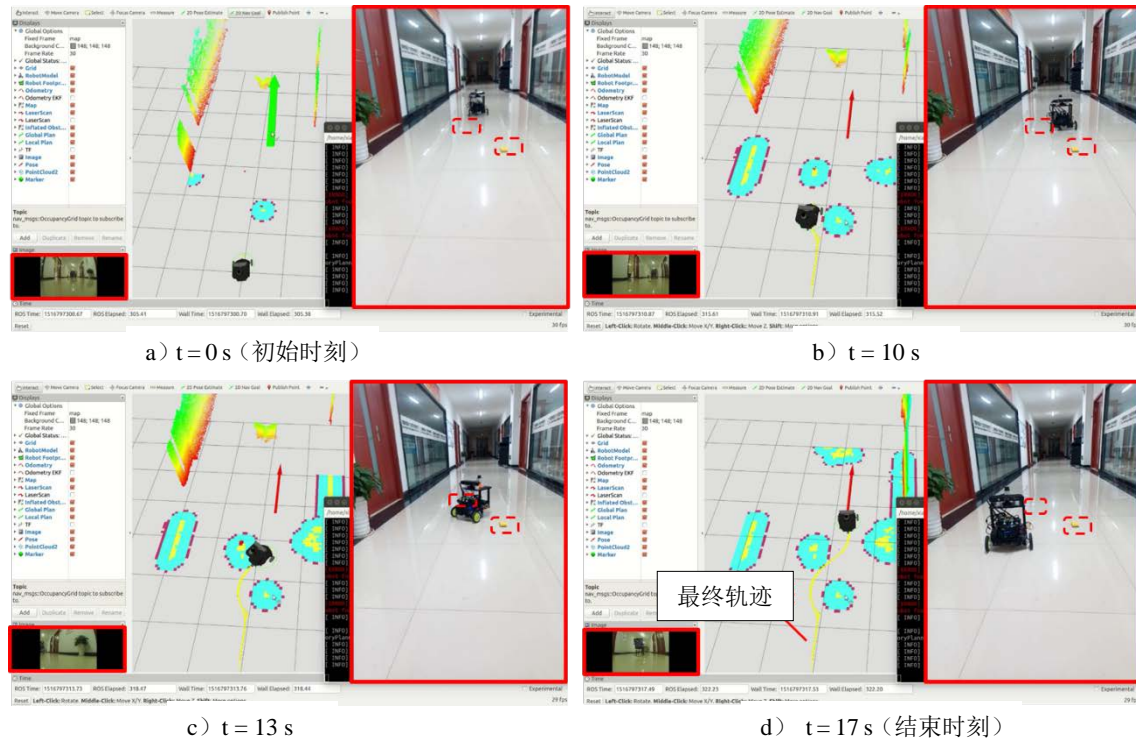


图5-8 静态避障实验过程图

本实验证明了路径规划模块对于尺寸较小的障碍物的检测能力。两个障碍物一个高度为 3.5cm，一个高度为 4cm，机器人可以检测到它们并进行有效的避障。由于 Kinect 相机的精度限制，目前路径规划模块能够识别的障碍物最小高度可以到达 3cm，如此的避障能力已经可以适用于大部分室内复杂环境下的安全导航。

本实验对路径规划模块三维避障的能力进行了验证。相比于目前大多数学者应用的二维激光雷达避障策略，本文应用的三维避障策略明显使得室内导航系统更加安全与实用。

5.4.2 移动机器人的动态避障实验

通过让行人突然出现在机器人已经计算好的全局路径上，以验证路径规划模块的动态避障能力。实验过程如下，事先给定机器人空白地图，使得机器人的导航变成相对运动；在机器人全局路径计算完毕后，让行人突然出现在该路径上以阻挡其运动，观察机器人之后的动作。

图 5-9 为动态避障的实验的过程图，整体动态避障实验对机器人进行了 3 次阻碍，图 5-9 为第一次避障动作的 4 个不同时刻的截图。

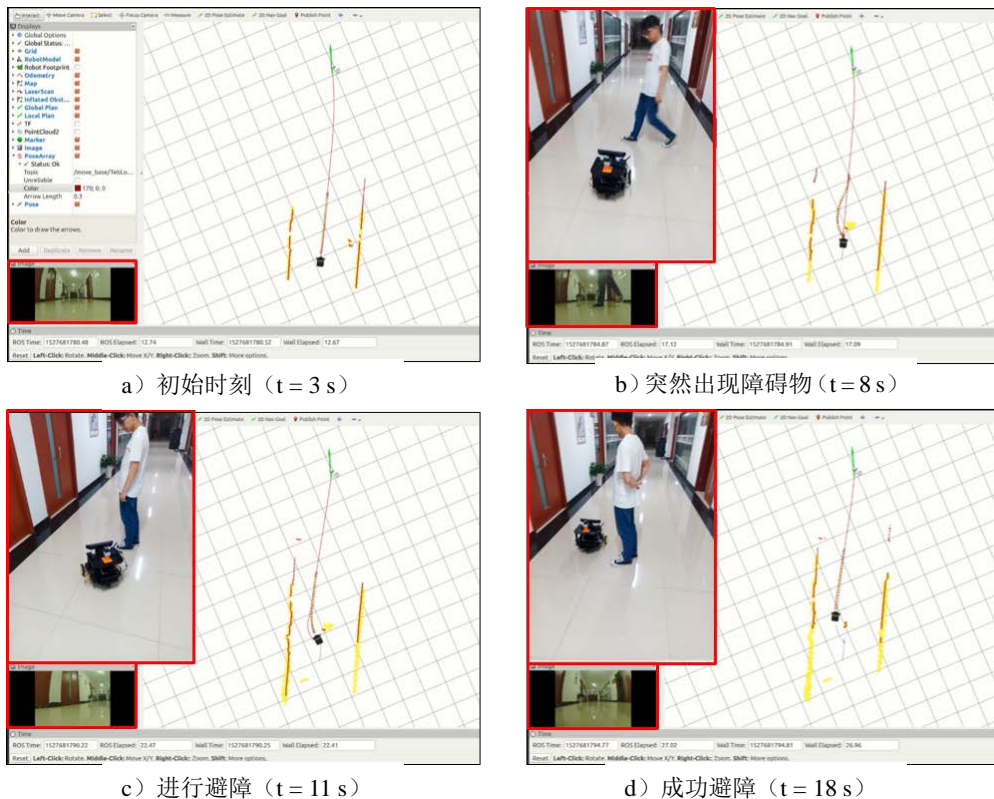


图5-9 动态避障实验过程图

从图 5-9 a) 可以看出，在机器人的前进路径上没有障碍物存在是，机器人规

划的路径是一条直接到达目标点的直线。图 5-9 b) 为行人突然出现在全局路径上，局部路径规划是实时进行的所以能够在行人刚出现时就重新规划路径，以对行人进行躲避。图 5-9 c) 为机器人跟随局部路径移动正在对行人进行躲避。图 5-9 d) 为成功避开行人之后再次为目标点进行路径规划，由于没有障碍物所以计算出的仍然是一条直线。

机器人能够对突然出现的行人进行有效躲避，证明局部路径规划算法的实时性与鲁棒性满足动态避障的要求。同时，机器人能够成功的避开障碍物并到达目标点证明了全局与局部路径规划算法的有效性。

本节通过对静态避障与动态避障实验验证了路径规划模块的避障能力。能够对小尺寸静态障碍物进行躲避，证明了路径规划算法计算的路径是安全有效的，同时也验证了路径规划模块具有三维避障的能力，性能远优于基于二维激光雷达的二维避障；能够对动态的行人进行有效的避障，证明了路径规划模块的实时性与鲁棒性达到了对动态物体避障的要求。

5.5 本章小结

本章通过三组实验验证了系统各个模块的效果与性质。通过主动 SLAM 实验证明了主动 SLAM 技术的可行性；通过两种 SLAM 算法建图效果的对比，证明了本文选用的 SLAM 算法的优越性，并对其建图精度分析，证明了该算法所建地图的准确性。对定位及路径规划模块进行自主导航实验，并对导航精度进行了分析。最后对导航系统的避障能力进行实验测试，证明了导航系统具有三维避障能力以及对动态物体避障的能力。

结 论

移动机器人导航技术处于当前科学研究领域的前沿，是高新技术的发展重要方向，而基于 SLAM 的导航更是机器人领域的研究热点。为了让室内导航技术更加自主化与智能化，本文对基于 SLAM 的室内导航技术进行了研究，对其中的 SLAM、定位以及路径规划三个方面进行研究。本文所取得的主要研究成果如下：

(1) 为了使机器人能够在未知环境下进行导航，设计了一种主动 SLAM 技术，应用基于边界的探索模型将 SLAM 与路径规划结合在一起，使得机器人能够自主地进行未知环境的地图构建；应用基于图优化理论的 SLAM 算法实现了高效、高精度的室内环境地图构建。

(2) 为了实现已知地图情况下的自主导航，对定位算法及路径规划算法进行了研究。应用自适应的蒙特卡洛定位算法实现了全局定位与局部位姿跟踪，并通过仿真实验对定位算法的收敛过程进行分析；应用 A* 算法实现了已知地图的全局路径规划，并通过仿真对比证明了 A* 算法的优越性；应用了新颖的局部路径规划算法进行实时地轨迹优化，实现了更稳定的避障与控制，实现了动态环境下的导航；为了实现室内复杂环境下的小尺寸物体的避障，采用了一种基于三维点云的避障策略，实现了室内复杂环境下三维避障能力的提升。

(3) 设计并完成了室内导航系统各模块的性能与精度的验证实验。主动 SLAM 实验验证了主动 SLAM 技术的可行性。对 SLAM 算法所建地图进行定性与定量分析，得到地图与实际环境对应尺寸间的平均偏差为 5cm，满足室内导航的要求。自主导航实验表明了定位及路径规划算法的可靠性，证明导航系统的精度为 $\pm 0.01\text{m}$ ；避障实验验证了三维避障策略的可行性，表明路径规划算法的鲁棒性与实时性能够达到动态避障的要求。

根据本文所取得的成果，还可以在以下几个方面进行进一步的研究：

(1) 大范围的建图。由于本文中使用的二维激光雷达的最大有效距离仅为 4 米，在大范围建图时产生的地图会出现偏差。在实验条件允许的情况下，应该选用有效距离更远的激光雷达以进行大范围地图构建的研究。

(2) 提高定位算法精度。本文选用的定位算法虽然很好的解决了全局定位与位姿跟踪问题，但是其定位精度并不是很高，达不到工业级别的精度要求。在后续的研究中可以改变定位算法，使定位精度进一步提升，达到工业级别的要求。

参考文献

- [1] 马家辰, 张琦, 马立勇, 等. 基于多行为融合的移动机器人路径规划方法[J]. 北京理工大学学报, 2014, 34(6):576-581.
- [2] 徐曙. 基于 SLAM 的移动机器人导航系统研究[D]. 华中科技大学, 2014.
- [3] Ohnishi N, Imiya A. Appearance-based navigation and homing for autonomous mobile robot[J]. Image & Vision Computing, 2013, 31(6-7):511-532.
- [4] Durrant - Whyte H. Where am I? A tutorial on mobile vehicle localization [J]. Industrial Robot, 1994, 21(2):11-16.
- [5] 辛冠希. 基于 RGB-D 摄像机的同步定位与建图研究[D]. 哈尔滨工业大学, 2016.
- [6] 谭雪, 戴更新. 应用于物流 AGV 磁导航传感器的研究[J]. 物流科技, 2017, 40(8):37-39.
- [7] 张毅, 杜凡宇, 罗元, 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, 33(10): 2970-2972.
- [8] 孙奇. AGV 系统路径规划技术研究[D]. 浙江大学, 2012.
- [9] Al-Mutib K N, Mattar E A, Alsulaiman M M, et al. Stereo vision SLAM based indoor autonomous mobile robot navigation[C]//Robotics and Biomimetics (ROBIO), 2014 IEEE International Conference on. IEEE, 2014: 1584-1589.
- [10] 曹智荀, 王宝华. 激光导引叉车在国内的应用[J]. 物流技术与应用, 2006, 11(7):98-99.
- [11] Durrantwhyte H, Bailey T. Simultaneous localization and mapping: part I[J]. IEEE Robotics & Amp Amp Automation Magazine, 2017, 13(2):99 - 110.
- [12] Thrun S, Burgard W, Fox D. Probabilistic robotics[M]. MIT press, 2005.
- [13] 赵钰. 基于激光传感器室内机器人自主导航技术研究[D]. 天津理工大学, 2017.
- [14] Li J. A new efficient pose estimation and tracking method for personal devices: application to interaction in smart spaces[D]. Universidad Complutense de Madrid, 2016.
- [15] Haute T V, Poorter E D, Lemic F, et al. Platform for benchmarking of rf-based indoor localization solutions[J]. IEEE Communications Magazine, 2015, 53(9): 126-133.
- [16] Rosmann C, Feiten W, Wosch T, et al. Efficient trajectory optimization using a sparse model[C]//Mobile Robots (ECMR), 2013 European Conference on. IEEE, 2013: 138-143.

- [17]顾冬雷, 李晓格, 王硕. 移动机器人路径规划方法[J]. 机器人技术与应用, 2014(1):28-30.
- [18]Quinlan S, Khatib O. Elastic bands: Connecting path planning and control[C]// Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on. IEEE, 1993: 802-807.
- [19]Hu T C, Kahng A B, Robins G. Optimal robust path planning in general environments[J]. IEEE Transactions on Robotics and Automation, 1993, 9(6): 775-784.
- [20]刘旭红, 张国英, 刘玉树, 等. 基于多目标遗传算法的路径规划[J]. 北京理工大学学报, 2005, 25(7):613-616.
- [21]Deng X, Mirzaian A. Competitive robot mapping with homogeneous markers[J]. IEEE Transactions on Robotics and Automation, 1996, 12(4): 532-542.
- [22]Smith R, Self M, Cheeseman P. A stochastic map for uncertain spatial relationships[C]// International Symposium on Robotics Research. MIT Press, 1988:467-474.
- [23]Kalman R E. A new approach to linear filtering and prediction problems[J]. Journal of basic Engineering, 1960, 82 (1) : 35-45.
- [24]陈白帆. 动态环境下移动机器人同时定位与建图研究[D]. 中南大学, 2009.
- [25]Wan E A, Van Der Merwe R. The unscented kalman filter for nonlinear estimation[C]//Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000. Ieee, 2000: 153-158.
- [26]Doucet A, De Freitas N, Murphy K, et al. Rao-Blackwellised particle filtering for dynamic Bayesian networks[C]//Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc., 2000: 176-183.
- [27]Montemerlo M, Thrun S, Koller D, et al. FastSLAM: a factored solution to the simultaneous localization and mapping problem[C]// Proc. of Theaaai National Conference on Artificial Intelligence. 2002:593-598.
- [28]庄严, 徐晓东, 王伟. 移动机器人几何-拓扑混合地图的构建及自定位研究[J]. 控制与决策, 2005, 20(7):815-818.
- [29]季秀才. 机器人同步定位与建图中数据关联问题研究[D]. 国防科学技术大学, 2008.
- [30]李昀泽. 基于激光雷达的室内机器人 SLAM 研究[D]. 华南理工大学, 2016.
- [31]韩明瑞, 周波, 钱堃, 等. 基于激光雷达的室外移动机器人三维定位和建图[J].

- 华中科技大学学报(自然科学版), 2015(s1):315-318.
- [32] 邹谦. 基于图优化 SLAM 的移动机器人导航方法研究[D]. 哈尔滨工业大学, 2017.
- [33] Yuan X, Martínez-Ortega J F, Fernández J A S, et al. Aekf-slam: a new algorithm for robotic underwater navigation[J]. *Sensors*, 2017, 17(5): 1174-1204.
- [34] Bloss R. Ease of programming and sophisticated sensors see robots advancing in transport logistics, palletizing, order picking and assembly[J]. *Industrial Robot*, 2013, 40(5): 420-424.
- [35] Lu D V, Allan D B, Smart W D. Tuning Cost Functions for Social Navigation[M]// *Social Robotics*. Springer International Publishing, 2013:442-451.
- [36] 聂鑫. 基于激光雷达的移动机器人自主导航研究[D]. 湖南大学, 2011.
- [37] 奚婉. 基于激光扫描的电力巡检机器人导航系统研制[D]. 浙江大学, 2017.
- [38] 王超. 室内全向移动机器人系统设计及导航方法研究[D]. 哈尔滨工业大学, 2017.
- [39] Chong T J, Tang X J, Leng C H, et al. Sensor technologies and simultaneous localization and mapping (SLAM)[J]. *Procedia Computer Science*, 2015, 76: 174-179.
- [40] 王丽佳. 基于图优化的单目视觉 SLAM 技术研究[D]. 华中科技大学, 2016.
- [41] Quigley M, Conley K, Gerkey B P, et al. ROS: an open-source robot operating system[C]// *ICRA Workshop on Open Source Software*. 2009: 5-10.
- [42] Grisetti G, Kummerle R, Stachniss C, et al. A tutorial on graph-based SLAM[J]. *IEEE Intelligent Transportation Systems Magazine*, 2011, 2(4):31-43.
- [43] Hess W, Kohler D, Rapp H, et al. Real-time loop closure in 2D LIDAR SLAM[C]// *Robotics and Automation (ICRA)*, 2016 IEEE International Conference on. IEEE, 2016: 1271-1278.
- [44] 袁源强. 智能轮椅室内建图及自主导航技术研究[D]. 哈尔滨工业大学, 2017.
- [45] Konolige K, Grisetti G, Kümmerle R, et al. Efficient sparse pose adjustment for 2D mapping[C]// *Intelligent Robots and Systems (IROS)*, 2010 IEEE/RSJ International Conference on. IEEE, 2010: 22-29.
- [46] 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[D]. 哈尔滨工业大学, 2016.
- [47] Yamauchi B. A frontier-based approach for autonomous exploration[C]// *IEEE International Symposium on Computational Intelligence in Robotics and Automation*,

1997. Cira'97. Proceedings. IEEE, 2002:146-151.
- [48]沈俊. 基于 ROS 的自主移动机器人系统设计与实现[D]. 西南科技大学, 2016.
- [49]张振东. 基于激光雷达的服务机器人室内动态导航方法研究[D]. 哈尔滨工业大学, 2014.
- [50]程新景. 煤矿救援机器人地图构建与路径规划研究[D]. 中国矿业大学, 2016.
- [51]张明明. 基于 Kinect2 的光伏清洗机器人实时环境重建与自主导航技术研究[D]. 哈尔滨工业大学, 2016.
- [52]Rösmann C, Feiten W, Wösch T, et al. Trajectory modification considering dynamic constraints of autonomous robots[C]//Robotics; Proceedings of ROBOTIK 2012; 7th German Conference on. VDE, 2012: 1-6.

哈尔滨工业大学学位论文原创性声明和使用权限

学位论文原创性声明

本人郑重声明：此处所提交的学位论文《基于 SLAM 的室内移动机器人导航技术研究》，是本人在导师指导下，在哈尔滨工业大学攻读学位期间独立进行研究工作所取得的成果，且学位论文中除已标注引用文献的部分外不包含他人完成或已发表的研究成果。对本学位论文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明。

作者签名：  日期：2018 年 6 月 27 日

学位论文使用权限

学位论文是研究生在哈尔滨工业大学攻读学位期间完成的成果，知识产权归属哈尔滨工业大学。学位论文的使用权限如下：

（1）学校可以采用影印、缩印或其它复制手段保存研究生上交的学位论文，并向国家图书馆报送学位论文；（2）学校可以将学位论文部分或全部内容编入有关数据库进行检索和提供相应阅览服务；（3）研究生毕业后发表与此学位论文研究成果相关的学术论文和其它成果时，应征得导师同意，且第一署名单位为哈尔滨工业大学。

保密论文在保密期内遵守有关保密规定，解密后适用于此使用权限规定。

本人知悉学位论文的使用权限，并将遵守有关规定。

作者签名：  日期：2018 年 6 月 27 日

导师签名：  日期：2018 年 6 月 27 日

致 谢

本文是在导师白清顺副教授的亲切关怀和耐心指导下完成的，十分感谢白老师不辞辛苦地为我一遍一遍的改论文。白老师严谨的科研态度与兢兢业业的工作作风深深的影响了我，让我学习到了科研的精神和做事的态度。同时，白老师在生活方面给予的关怀和帮助，让我倍感温暖，难以忘怀。在此，谨向恩师表示衷心的感谢和深深的敬意！

感谢芜湖哈特机器人产业技术研究院有限公司为我提供的实习机会，为我课题的顺利进行提供了实验平台以及各方面的支持。感谢服研中心的各位同事对我的帮助与照顾，与他们之间的技术交流使我对课题有了更深刻的认识。感谢研究院的高云峰老师与李超博士对我的指导与帮助，为我论文的撰写与课题的方向提供了很大的指导。

感谢实验室的郭辉、张凯、史小全、陶明方同学，以及其它同学和师兄们，他们帮助身在外地的我做了很多事情，如交材料、下载论文等等太多的事情，尤其感谢我的同班同学郭辉，基本班级里发的材料都是他帮我领的，真的十分感谢大家。感谢一同在研究院实习的方贤根、尹乐、江璇、熊一帆同学，感谢他们的陪伴与对我的帮助。感谢我的室友于杰明与夏龙光同学，那些一同住宿的日子让我十分怀念。

感谢我的家人们在这两年里给我的鼓励与支持，他们一直默默的为我付出，关心着我，他们是我心中永远的牵挂。

最后衷心感谢所有关心和帮助过我的老师、同学、亲人和朋友们！