

基于 RGB-D 与激光雷达传感器的 SLAM 算法研究与实现

**RESEARCH AND REALIZATION OF SLAM
ALGORITHM BASED ON RGB - D AND LIDAR SENSOR**

学 院： 信息科学与技术学院

学科专业： 信息与通信工程

作者姓名： 赖秋玲

指导老师： 禹素萍

答辩日期： 2018 年 01 月 08 日

东华大学学位论文原创性声明

本人郑重声明：我恪守学术道德，崇尚严谨学风。所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已明确注明和引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品及成果的内容。论文为本人亲自撰写，我对所写的内容负责，并完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期： 年 月 日

东华大学学位论文版权使用授权书

学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校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，允许论文被查阅或借阅。本人授权东华大学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。

保密 ☐，在 _____ 年解密后适用本版权书。

本学位论文属于

不保密 ☐。

学位论文作者签名：

指导教师签名：

日期： 年 月 日

日期： 年 月 日

基于 RGB-D 与激光雷达传感器的 SLAM 算法研究与实现

摘要

同时定位与建图 (Simultaneous Locating And Mapping, SLAM) 是移动机器人领域的热门研究课题。SLAM 是移动机器人实现智能化的前提, 包含丰富环境信息的地图是移动机器人进行后续各类智能化任务不可或缺的。随着传感器技术的发展, RGB-D 相机、激光雷达等传感器减少了 SLAM 的实现难度, 但由于环境噪声和传感器自身的性能限制, 各类传感器收集到的环境信息可能存在不全面或者可信度较低的问题, 这使得 SLAM 所建的环境地图存在优化空间。因此研究基于 RGB-D 与激光雷达传感器的 SLAM 算法在理论和应用层面上均有意义。

本文以 SLAM 为研究目标, 重点研究基于 RGB-D 相机与激光雷达传感器的移动机器人 SLAM 实现算法, 并在此基础上实现了深度图像中缺失的有效值修复与传感器数据融合。本文设计并搭建移动机器人实验平台并在此平台上实现移动机器人路径规划系统。

首先, 本文研究现有基于滤波器的经典 SLAM 算法与视觉 SLAM 算法, 包括卡尔曼滤波 (Kalman Filter, KF)、扩展卡尔曼滤波 (Extended Kalman Filter, EKF)、粒子滤波 (Particle Filter, PF)、RGB-D SLAM、ORB-SLAM, 还研究了基于激光雷达传感器的 Gmapping 建图方法。

然后, 为了获取更丰富的环境信息, 弥补激光雷达传感器自身的性能限制, 如无法检测低于其安装平面的障碍物, 本文提出一种 RGB-D 相机与激光雷达传感器的数据融合方法, 并在此融合方法的基础上实现 Gmapping 算法。其中, 本文提出一种基于简单的线性迭代聚类 SLIC (Simple Linear Iterative Clustering, SLIC) 超像素分割的图像修复方法对 RGB-D 相机获取的深度图像进行修复。

最后，本文设计并搭建移动机器人实验平台，研究 SLAM 在移动机器人路径规划方面的应用，并在搭建的实验平台上实现移动机器人路径规划系统。

实验证明，本文提出的基于 SLIC 超像素分割的修复方法可以较好的修复深度图像中的缺失值，提出的 RGB-D 相机与激光雷达数据融合方法结合了两种传感器的优点，得到了更丰富的环境信息。

关键字：同时定位与建图；深度图像修复；数据融合；路径规划

RESEARCH AND REALIZATION OF SLAM ALGORITHM BASED ON RGB-D AND LIDAR SENSOR

ABSTRACT

Simultaneous Locating and Mapping (SLAM) is a hot research topic in the field of mobile robot. SLAM is a prerequisite for realizing intelligent mobile robot. The map containing rich environment information is indispensable for all kinds of intelligent tasks of mobile robot. With the development of sensor technology, sensors such as RGB-D cameras and Lidar reduce the difficulty of realizing SLAM. However, the environmental information collected by various sensors has certain uncertainties due to the environmental noise and the limited performance of sensors. This allows optimization space for the environment map built by the SLAM algorithm. Therefore, researching SLAM algorithm based on RGB-D and Lidar sensors has certain theoretical significance and application value.

In this dissertation, the SLAM algorithm for mobile robots based on RGB-D and Lidar sensors is studied, and on this basis, the refill of depth image missing values and fusion of sensor data are realized. A mobile robot experimental platform is designed and built on which the mobile robot path planning system is implemented.

First of all, some existing classical filter-based SLAM algorithm and visual SLAM algorithm are studied, including KF, EKF, PF, RGB-D SLAM and ORB-SLAM. Gmapping based on Lidar sensor is also studied.

Then, in order to obtain richer environment information and make up for the performance limitations of Lidar sensor, such as the inability to detect obstacles below its mounting plane, a data fusion method of RGB-D camera and Lidar sensor is proposed in this paper. The Gmapping algorithm is realized based on this data fusion method. In this paper, an image restoration method based on SLIC super-pixel segmentation is proposed to improve the depth image acquired by RGB-D camera.

Finally, a mobile robot experimental platform is designed and built. The application of SLAM in the path planning of mobile robots is studied and realized on the experimental platform.

Experiments show that the proposed method can well refill missing values of the depth image, and the proposed RGB-D camera and Lidar sensor data fusion method can combine the advantages of the two sensors to obtain richer environment information.

Author Name: Qiuling Lai (Information and Communication Engineering)

Supervised by Suping Yu

KEY WORDS: simultaneous locating and mapping, depth image restoration, data fusion, path planning

目 录

| | |
|----------------------------|-----|
| 摘 要..... | IV |
| ABSTRACT..... | VI |
| 目 录..... | VII |
| 1 绪论..... | 1 |
| 1.1 研究背景与意义..... | 1 |
| 1.1.1 研究背景..... | 1 |
| 1.1.2 研究意义..... | 3 |
| 1.2 国内外研究现状..... | 3 |
| 1.3 论文的主要工作和结构安排..... | 5 |
| 1.3.1 主要研究内容..... | 5 |
| 1.3.2 论文的结构安排..... | 5 |
| 1.4 本章小结..... | 6 |
| 2 SLAM 基本算法..... | 7 |
| 2.1 引言..... | 7 |
| 2.2 基于滤波器的 SLAM 实现算法..... | 7 |
| 2.2.1 卡尔曼滤波器 (KF) | 7 |
| 2.2.2 扩展卡尔曼滤波器 (EKF) | 8 |
| 2.2.3 粒子滤波器 (PF) | 9 |
| 2.2.4 FastSLAM..... | 9 |
| 2.3 视觉 SLAM 算法..... | 11 |
| 2.3.1 RGB-D SLAM 算法框架..... | 12 |
| 2.3.2 ORB-SLAM 算法框架..... | 13 |
| 2.4 激光雷达 Gmapping 建图..... | 14 |
| 2.5 本章小结..... | 16 |
| 3 RGB-D 相机与激光雷达数据融合 | 17 |
| 3.1 引言..... | 17 |
| 3.2 深度图像预处理..... | 18 |
| 3.2.1 深度图像修复方法..... | 18 |
| 3.2.2 改进的深度图像修复方法..... | 20 |
| 3.2.3 SLIC 超像素分割 | 22 |
| 3.2.4 分类修复深度图像缺失值..... | 24 |
| 3.2.5 交叉双边滤波..... | 24 |
| 3.3 雷达数据预处理..... | 25 |

| | |
|--|----|
| 3.4 传感器数据融合..... | 25 |
| 3.4.1 深度图像信息映射..... | 25 |
| 3.4.2 相机与激光雷达数据融合方法..... | 26 |
| 3.5 本章小结..... | 28 |
| 4 移动机器人路径规划系统设计..... | 29 |
| 4.1 引言..... | 29 |
| 4.2 路径搜索算法..... | 29 |
| 4.2.1 A*算法..... | 29 |
| 4.2.2 D*算法..... | 30 |
| 4.3 视觉 SLAM 算法框架分析..... | 32 |
| 4.4 路径规划系统设计..... | 33 |
| 4.5 本章小结..... | 34 |
| 5 实验平台与实验结果分析..... | 35 |
| 5.1 实验平台搭建..... | 35 |
| 5.2 Xtion Pro Live 传感器与 RPLIDAR 传感器精度测量与对比 | 38 |
| 5.3 实验结果与分析..... | 42 |
| 5.3.1 深度图像修复实验结果与分析..... | 42 |
| 5.3.2 深度相机和激光雷达数据融合实验结果与分析..... | 43 |
| 5.3.3 室内 Home Care Robot 机器人路径规划实现结果与分析.... | 44 |
| 5.4 本章小结..... | 46 |
| 6 总结与展望..... | 47 |
| 6.1 主要创新点..... | 47 |
| 6.2 工作总结与展望..... | 47 |
| 参考文献..... | 49 |
| 攻读硕士学位期间主要学术成果..... | 53 |
| 致谢..... | 54 |

1 绪论

1.1 研究背景与意义

1.1.1 研究背景

自 1956 年计算机科学家们在达特茅斯会议（Dartmouth Conference）提出人工智能^[1]（Artificial Intelligence, AI）这个概念以来，科研学者们不断地投入其中进行研究，并取得惊人的发展，尤其是近几年随着硬件处理器以及大数据的出现，科研学者们利用机器学习方法^[2]和深度学习技术^[3]让 AI 不断地从想象走进人类的现实生活。移动机器人^[4]是人工智能的一个重要载体，现如今，不论是在工业生产还是日常生活中，移动机器人都得到广泛的应用。

为了更好地协助或代替人类完成一些任务，移动机器人除了自身拥有的感知周围环境的能力外，还需要具备一些其他的能力，例如：定位、建图、路径规划以及导航等。图 1-1 描述了本文所研究的 SLAM^[5]，即同时定位与建图，它是定位和建图功能的同时实现。SLAM 作为移动机器人智能化的前提，其主要研究方向有视觉里程计、建图、闭环检测、后端优化、高级课题以及传感器技术六种方向，如图 1-2 所示。

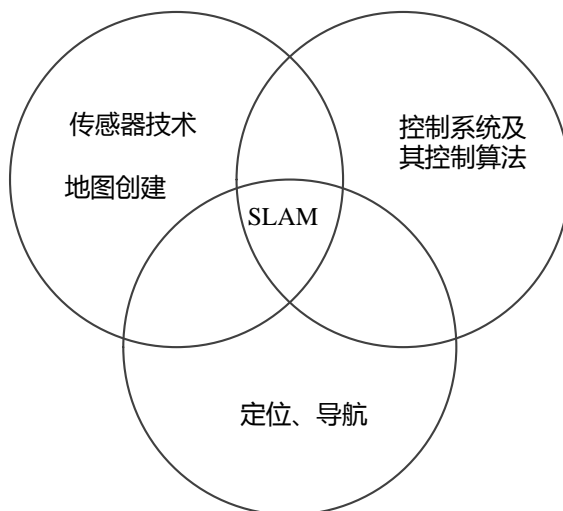


图 1-1 机器人的部分研究领域

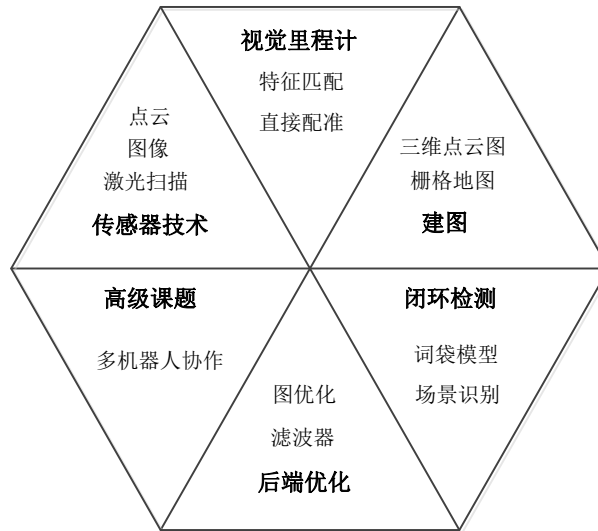


图 1-2 SLAM 的主要研究方向

SLAM 的基础理论指它采用的数学模型，不同的数学模型决定了数据处理的不同方法并且都有各自适用的场景和优缺点。起初，卡尔曼滤波器^[6]被众多研究者用于解决 SLAM 问题，而后在卡尔曼滤波器的基础上发展出多种滤波器方法，基于滤波器的方法也逐渐趋于成熟，但仍存在可改善的空间，如不适用动态场合以及运行资源消耗较大等缺陷。近几年，优化的方法逐渐引起人们关注，尤其是图优化^[7]。但是已有的优化方法也存在一定的缺点，如算法效率问题。解决 SLAM 问题的算法多种多样，且都有各自的优缺点，所以需要根据实际场景和条件来决定选用何种算法。

传感器数据是 SLAM 中唯一的输入数据。传感器的种类和其在机器人上的安装位置，都会反应在建模的方程上，从而影响 SLAM 求解过程的难易程度。传感器种类繁多，早期 SLAM 大都使用激光传感器，随着学者们对传感器技术研究的不断深入，深度相机^[8]、视觉相机、声呐（水下）等传感器降低了 SLAM 的实现难度。激光雷达传感器^[9]与 RGB-D 相机是两种比较受欢迎的传感器。激光雷达传感器结合 Gmapping 建图方法可以快速创建具有较高精度的二维平面环境地图，被广泛地应用在工业生产中，比如市场上的各类扫地机器人和服务型机器人。而 RGB-D 相机的出现，推动了视觉 SLAM 技术的发展，出现许多较优秀的视觉 SLAM 算法框架，比如 RGB-D SLAM 以及 ORB-SLAM。目前，虽然借助单种类传感器的移动机器人也可以较好的实现 SLAM，但由于各类传感器自身的制作原理，其性能受到一定的限制。所以学者们又研究了传感器数据融合技术^[10]，该技术可以结合不同传感器的优点，使 SLAM 获取包含更丰富环境信息的输入。

1.1.2 研究意义

SLAM 的实现过程围绕着三个问题：我在哪儿？我要去哪儿？我怎么过去？而这三个问题是实现智能化机器人的重要前提，许多高级任务比如导航和路径规划等均是基于 SLAM 的实现。SLAM 的目标在于建立一个全局一致的环境表示，创建或更新未知环境的映射问题，同时记录机器人的位置。虽然这似乎是一个先有鸡还是先有蛋的问题，但是学者们已经研究出在特定环境下可行的多种解决方法。

SLAM 实现算法是根据可用资源实现的，因此其目标并不是完美实现机器人的同时定位与建图，而是致力于实现可用可操作的应用方面。目前 SLAM 已经被应用于许多方面，比如自驾汽车、无人驾驶飞行器、自主水下航行器、行星式飞行器、新兴国产机器人、人体内部。由 Sebastian Thrun 领导的自驾 STANLEY 和 JUNIOR 汽车参加了 DARPA 大挑战赛，并在二十世纪九十年代获得了 DARPA 城市挑战赛的第二名，其中就包括了 SLAM 系统，使 SLAM 成为全球关注的焦点。大众市场的 SLAM 实现现在可以应用在诸如 Neato XV11 的消费者机器人真空吸尘器中。Google 和其他公司开发的自动驾驶汽车已经在美国一些州的公路上获得驾驶执照。

相比于依靠单个传感器实现 SLAM，传感器数据融合技术可以创建包含更丰富环境信息的地图，这有利于机器人更好的实现智能化功能。本文在研究了 SLAM 实现算法以及传感器数据融合技术的基础上，将激光雷达传感器与 RGB-D 相机获得的环境数据按所提出的融合方法进行融合，同时用一种基于 SLIC 超像素分割的图像修复方法对 RGB-D 相机获取的深度图像进行修复。在应用层面，在机器人操作系统^[11]（Robot Operating System, ROS）以及搭建的家用服务机器人 HCR（Home Care Robot, HCR）实验平台上实现移动机器人路径规划系统。

1.2 国内外研究现状

SLAM 问题的实质是如何处理由各类传感器（如激光雷达、立体相机或者 RGB-D 相机）获取的周围环境信息，来估计自身的位姿与状态。SLAM 的发展主要分为两个阶段：第一阶段为经典阶段^{[12][13]}（1986-2004），在这个阶段学者们为实现 SLAM 提出了基于贝叶斯的滤波器方法，包括卡尔曼滤波、扩展卡尔曼滤波^[14]、Rao-Blackwellized 粒子滤波^[15]等，该阶段第一次描述了高效率与鲁棒性数据关联的较大挑战性。第二阶段为算法分析阶段^[16]（2004-2016），该阶段主要对 SLAM 的基本性质进行研究，包括可观察性、收敛性和一致性。在这一时期，人们理解了稀疏性对高效地实现 SLAM 的关键作用，并且开源了较多的 SLAM 算法库。

在经典阶段，以卡尔曼滤波器作为建模的基础，分别建立线性运动、观测方程。运动方程描述了从状态 x_{k-1} 到状态 x_k 的变换过程，而观测方程则描述了从 x_k 是怎么获得观察数据 y_k 。但是在现实世界中的系统并非都是线性的，于是发展出扩展卡尔曼滤波，该算法对系统采取线性化的做法，在系统被近似成线性的情况下，噪声与状态在 EKF 中都被默认成高斯分布。但 EKF 方法存在一定的局限性与不足之处，即高斯分布作为输入经非线性系统处理之后的输出不会是高斯分布。EKF 只是用协方差和均值，并基于计算结果用一个新的高斯分布来代替这个非线性系统变换后的输出。为了弥补 EKF 的不足之处，学者们提出无迹卡尔曼滤波^[17]（Unscented Kalman Filter ,UKF），UKF 致力于解决一个高斯分布输入到非线性系统中如何用一个新的高斯分布很好地近似输出结果。而 UKF 的不足之处是它仍假设输出的结果是服从高斯分布的，然而实验表明即使在这种情况下，高斯分布的非线性变换结果并不是高斯分布，所以粒子滤波^[18]采用一种不同的方法来描述这个分布，即用大量的采样点来描述，实质是一个不断采样，计算权重，然后重采样的循环过程。粒子权重值的大小跟观测方程的符合程度有关，符合程度大的粒子其相应的权重值就大，而且在重采样过程中那些权重值大的粒子容易被采到。但是粒子滤波方法也有不足之处，即采样的粒子数量随着程序的运行其所需数目呈指数增长，所以 PF 适用于解决低维度空间问题。FastSLAM 算法^[19]通过求解移动机器人位姿和地图的后验概率来实现 SLAM，该算法较之前的实现算法减少了一定的计算量。

经过经典阶段之后，基于图模型的图优化被用于解决 SLAM 问题。在这算法分析阶段，解决 SLAM 的算法一般分成前端与后端两部分，前端通过各类传感器来获取环境信息并进行处理，在约束方程的约束下创建图模型。最后将构建的图模型交给后端，结合观测信息，通过优化算法进行优化，并在优化之后得到地图。图优化的代表算法有 RGB-D SLAM^[20]、ORB-SLAM^[21]和 ORB-SLAM2^[22]等。

国外学者们在移动机器人 SLAM 研究领域取得了较为突出的成果，Georg Klein 和 David Murray 于 2007 提出一种估计未知场景中相机姿态的方法，即并行跟踪和映射 PTAM^[23]（Parallel Tracking and Mapping, PTAM），该方法将跟踪与建图作为两个独立的线程，在双核计算机的并行线程中进行处理，具有较好的准确性与鲁棒性。Christian Forster 等人提出 SVO^[24]（Fast Semi-Direct Monocular Visual Odometry）算法，该算法舍弃用于运动估计的特征提取与匹配步骤，而是直接对像素强度进行操作，该算法不仅提高了运行速率而且实现了高频率下的亚像素精度。Thomas Whelan 等人发表 Elastic Fusion 算法^[25]，该算法实现了实时稠密的视觉 SLAM，利用密集的帧模型摄像机跟踪和 surfel 表示模型进行实现，并采用 GPU（Graphics Processing Unit, GPU）进行硬件加速算法运行。Felix Endres 等人^[20]于 2014 年提出基于 RGB-D 相机的 RGB-D SLAM

开源算法, Ra´ul Mur-Artal^[21]于 2015 年提出 ORB-SLAM 单目 SLAM 系统, 这些算法的稳健性与鲁棒性较以往算法均有较大的提高。国外研究者为实现移动机器人 SLAM 提出了较多的实现算法, 为 SLAM 的研究发展做出了贡献。

国内对移动机器人 SLAM 的研究开始得也较早, 特别是近几年随着虚拟现实^[26] (Virtual Reality, VR) 以及增强现实^[27] (Augmented Reality, AR) 等技术的出现, SLAM 引起了大家的广泛关注。哈尔滨工业大学计算机科学与计算学院的蔡则苏等人^[28]在 2005 年的全国光学与光电子学学术研讨会上, 提出使用激光测距传感器完成对家用机器人的同时定位与建图。国内学者对 SLAM 技术的发展也十分关注, 彭胜军等人^[29]阐述了对 SLAM 的理解, 展望该技术的未来发展方向。武汉科技大学的闵华松等人^[30]对视觉 SLAM 算法框架 RGB-D SLAM 框架进行改进, 把机器人惯性传感器信息和体感传感器信息合并互补, 从而改善其定位效果。一些 SLAM 算法具有一定的局限性, 比如只适用于室内的较小环境, 在大规模环境中的定位与建图效果则不佳, 王忠立等人^[31]研究了在环境规模较大场景下的 SLAM 实现方法。罗元^[15]等人改进了基于 Rao-Blackwellized 的粒子滤波器, 完成较高精度地实时栅格地图创建, 为移动机器人在导航方面做出一定贡献。国内学者们不仅仅将 SLAM 应用于移动机器人, 也将其应用到一些新型飞行器上, 王国胜等人^[32]研究了无人飞行器视觉 SLAM 中的视觉特征检测以及特征匹配问题。赵辉等人^[33]将 SLAM 技术应用于自主式水下航行器 (Autonomous Underwater Vehicle, AUV) 的自主导航中, 基于扩展卡尔曼滤波完成导航的仿真实现, 为 AUV 的导航指出新的研究方向。

1.3 论文的主要工作和结构安排

1.3.1 主要研究内容

研究 SLAM 实现算法, 主要是基于滤波器的经典 SLAM 实现算法以及视觉 SLAM 算法; 对由 RGB-D 相机获得的深度图像中存在缺失值现象, 提出一种利用 SLIC 超像素分割的修复方法对深度图像中的缺失值进行有效修复; 为了获得包含更丰富环境信息的地图, 提出一种融合方法, 将 RGB-D 相机与激光雷达传感器数据按所提方法进行融合; 设计并搭建移动机器人实验平台, 分别实现 RGB-D SLAM 和 ORB-SLAM2 算法, 分析对比这两种算法, 并在搭建的实验平台上实现机器人室内路径规划系统。

1.3.2 论文的结构安排

第 1 章: 绪论, 介绍 SLAM 研究课题的含义, 阐述其研究背景、研究意义和国内外研究现状, 并说明本文的主要工作和章节安排。

第 2 章: SLAM 基本算法, 研究了 SLAM 实现的主要两大类方法, 即基于滤波器的经典算法和基于图优化方法的视觉 SLAM 实现算法。基于滤波器方法, 本文重点研究了几种经典实现算法: 卡尔曼滤波器算法, 粒子滤波与

FastSLAM。对基于图优化 SLAM 实现算法，本文主要研究了近几年较受关注的 RGB-D SLAM 和 ORB-SLAM 算法，最后研究了基于激光雷达传感器的 Gmapping 方法。

第 3 章：RGB-D 相机与激光雷达数据融合，研究了由 RGB-D 相机获取的深度图像中的有效值缺失修复方法，并提出一种基于 SLIC 超像素分割技术的修复方法对缺失的有效值进行修复。研究 RGB-D 相机（Xtion Pro Live）和激光雷达传感器（RPLIDAR）的各自优缺点，提出一种融合方法，将相机和雷达传感器获取的环境信息进行融合，改善了激光雷达感知范围存在缺陷的不足之处。

第 4 章：移动机器人路径规划系统设计，研究经典路径搜索算法，并分析对比 RGB-D SLAM 与 ORB-SLAM2 算法框架，设计移动机器人路径规划系统。

第 5 章：实验平台与实验结果分析，设计并搭建移动机器人实验平台，然后介绍了本文用到的传感器，即激光雷达和 RGB-D 相机，最后分析本文的实验结果。

第 6 章：总结与展望，总结本文的主要创新点和对 SLAM 研究课题未来的展望。

1.4 本章小结

主要介绍 SLAM 这个课题的含义及其研究背景、研究意义与国内外研究现状，阐述了国内外 SLAM 实现算法的发展现状。最后介绍论文的主要研究内容和章节安排。

2 SLAM 基本算法

2.1 引言

SLAM 实现算法依据其建立模型基础的不同，大致归为两种，一种是基于滤波器模型的算法，另一种是基于图优化模型的算法。基于滤波器的算法理论基础与框架均已比较完善，主要包括以下三种：扩展卡尔曼滤波、粒子滤波器和 FastSLAM，而优化方法近几年使用较多，尤其是基于图理论的图优化方法。

2.2 基于滤波器的 SLAM 实现算法

2.2.1 卡尔曼滤波器 (KF)

卡尔曼滤波^[34]，也被称为线性二次估计 (Linear Quadratic Estimation, LQE)，是一种使用一系列包含统计噪声和其他不确定因素的观测值对未知变量进行估计的算法。卡尔曼滤波利用贝叶斯估计变量的联合概率分布，这比仅仅基于单个测量值的估计更准确。卡尔曼滤波常用于线性系统，也就是系统的噪声是高斯分布，其运动、观测方程都是线性的。卡尔曼滤波实质上是利用贝叶斯法则计算状态变量 x 的后验概率分布。公式 (2-1) 为系统状态方程，公式 (2-2) 为系统观测方程，其中 Q_k ， R_k 是两个噪声项的协方差矩阵。 A ， B 是系统参数， C 是观测系统的参数。卡尔曼滤波采用递推方式，即从 x_{k-1} 估计 x_k 的无偏最优估计。

$$x_k = Ax_{k-1} + Bu_{k-1} + \omega_{k-1} \quad (2-1)$$

$$y_k = Cx_k + n_k \quad (2-2)$$

$$\omega_k \sim N(0, Q_k) \quad (2-3)$$

$$n_k \sim N(0, R_k) \quad (2-4)$$

用 \hat{x} 表示状态变量 x 的后验概率，用 \tilde{x} 表示它的先验概率。如果假设系统是线性系统且噪声服从高斯分布，那么状态变量 x 服从高斯分布，进而求解 x 的期望与协方差矩阵。记第 k 时刻的状态变量服从： $x_k \sim N(\tilde{x}_k, P_k)$ ，为了得到状态变量 x 的最大后验估计，于是计算：

$$\hat{x} = \arg \max p(x|y, u) = \arg \max \frac{p(y|x, u)p(x|u)}{p(y|u)} = \arg \max p(y|x)p(x|u) \quad (2-5)$$

使公式 (2-5) 的后验概率 \hat{x} 最大，经过推导数学公式，得出卡尔曼滤波的求解过程：

$$\tilde{P}_k = A\hat{P}_{k-1}A^T + Q_k \quad (2-6)$$

$$\tilde{x}_k = A\hat{x}_{k-1} + Bu_{k-1} \quad (2-7)$$

$$K_k = \tilde{P}_k C^T (C\tilde{P}_k C^T + R_k)^{-1} \quad (2-8)$$

$$\hat{P}_k = (I - K_k C)\tilde{P}_k \quad (2-9)$$

$$\hat{x}_k = \tilde{x}_k + K_k(y_k - C\tilde{x}_k) \quad (2-10)$$

定义 \tilde{x}_k 为 k 时刻的先验状态变量估计, \hat{x}_k 为已知测量变量 y_k 后 k 时刻的后验状态估计, 公式 (2-6) 与公式 (2-7) 为预测阶段, 更新阶段用到的公式为公式 (2-8)、公式 (2-9)、公式 (2-10), 其中 \tilde{P}_k 和 \hat{P}_k 分别为先验估计和后验估计的协方差。

在预测阶段中, 卡尔曼滤波器产生当前状态变量的估计以及它们的不确定性。一旦观察到下一次测量的结果 (必然受到包括随机噪声在内的一定量误差的影响), 则使用加权平均来更新这些估计, 并以更高的确定性给予更多的权重。该算法是递归而且可以实时运行, 只使用当前的输入测量结果和过去计算的状态及其不确定性矩阵, 不需要额外的过去信息。

卡尔曼滤波的前提假设是系统是线性的而且噪声分布是服从高斯分布, 但实际 SLAM 实现场景中通常不能符合这些前提假设。在实际的移动机器人运动场景中, 机器人的系统状态方程与更新方程均为非线性, 此时卡尔曼滤波对实现 SLAM 表现出不足之处, 针对卡尔曼滤波的局限性, 学者们提出了扩展卡尔曼滤波方法。

2.2.2 扩展卡尔曼滤波器 (EKF)

扩展卡尔曼滤波^[35]是卡尔曼滤波器的非线性版本, 适用于包含各种非线性行为的系统, EKF 算法以一种迭代的方式在当前估计工作点下将非线性的模型线性化。扩展卡尔曼滤波算法的核心思想是首先用高斯分布去近似一个经过非线性系统变换的高斯分布, 随后在求解 k 时刻的状态变量时, 将系统近似化为线性系统。公式 (2-11) 是系统状态方程, 公式 (2-12) 是测量方程。

$$x_k = f(x_{k-1}, v_k, \omega_k) \quad (2-11)$$

$$z_k = g(x_k, n_k) \quad (2-12)$$

$$f(x_{k-1}, v_k, \omega_k) \approx f(\hat{x}_{k-1}, v_k, 0) + \frac{\partial f}{\partial x_{k-1}} (x_{k-1} - \hat{x}_{k-1}) + \frac{\partial f}{\partial \omega_k} \omega_k \quad (2-13)$$

$$g(x_k, n_k) \approx g(\tilde{x}_k, 0) + \frac{\partial g}{\partial x_k} n_k \quad (2-14)$$

$$F_{k-1} = \frac{\partial f}{\partial x_{k-1}} |_{\hat{x}_{k-1}} \quad (2-15)$$

$$G_k = \frac{\partial g}{\partial x_k} |_{\tilde{x}_k} \quad (2-16)$$

$$\tilde{P}_k = F_{k-1} \hat{P}_{k-1} F_{k-1}^T + Q_k' \quad (2-17)$$

$$\tilde{x}_k = f(\hat{x}_{k-1}, v_k, 0) \quad (2-18)$$

$$K_k = \tilde{P}_k G_k^T (G_k \tilde{P}_k G_k^T + R_k')^{-1} \quad (2-19)$$

$$\hat{P}_k = (I - K_k G_k) \tilde{P}_k \quad (2-20)$$

$$\hat{x}_k = \tilde{x}_k + K_k (y_k - g(\tilde{x}_k, 0)) \quad (2-21)$$

在 \hat{x}_{k-1} , \tilde{x}_k 点附近对系统进行线性近似, 如公式 (2-13)、公式 (2-14) 所示, 在完成非线性系统线性化操作之后, 将噪声和状态变量均假设为服从高斯分布, 随后估计噪声与状态变量的期望与协方差。扩展卡尔曼滤波的形式与

卡尔曼滤波是相同的，只是将卡尔曼滤波中的系统参数 A 和观测参数 C 用线性化之后的矩阵替换。

扩展卡尔曼滤波比卡尔曼滤波更好的适用于非线性 SLAM 实现场景。但是扩展卡尔曼滤波未解答如何构造一个新的高斯分布去代替经过非线性系统的高斯分布。扩展卡尔曼滤波存在以下不足之处：一个高斯分布输入非线性系统其输出并不一定是服从高斯分布；扩展卡尔曼滤波计算均值和协方差，实质是用高斯分布来近似变换后的结果，在实际情况下，这个近似效果可能很差；在线性化非线性系统步骤时，直接丢掉高阶项；线性化的工作点即 \hat{x}_{k-1} ， \tilde{x}_k 输入的是状态变量的估计值。于是，在这个工作点计算的 F 和 G ，也不是最好的。在对计算非线性系统输出的均值步骤时，EKF 计算的是 $\mu_y = f(\mu_x)$ 的形式。其求解的只是输出分布的估计值，并非真实均值，协方差的计算结果也是如此。EKF 对相关高斯噪声的假设，严重损害了 EKF SLAM 处理不确定性的能力。在后验的不确定性较大情况下，EKF 中的线性化会失败。

2.2.3 粒子滤波器 (PF)

粒子滤波器舍弃了卡尔曼滤波的方法，因为高斯分布的非线性变换依然不是高斯分布，卡尔曼滤波的近似思想有一定局限性。粒子滤波采用大量采样点来描述这个分布，且是无参数的。粒子滤波算法流程图如图 2-1，实质是一个不断采样---计算权重---重采样的过程。粒子滤波算法中的状态空间模型可以是非线性的，对于初始状态与噪声分布形式没有任何要求。但是，在高维度的系统中，粒子滤波的实时性不佳，所以较适用于低维度空间。

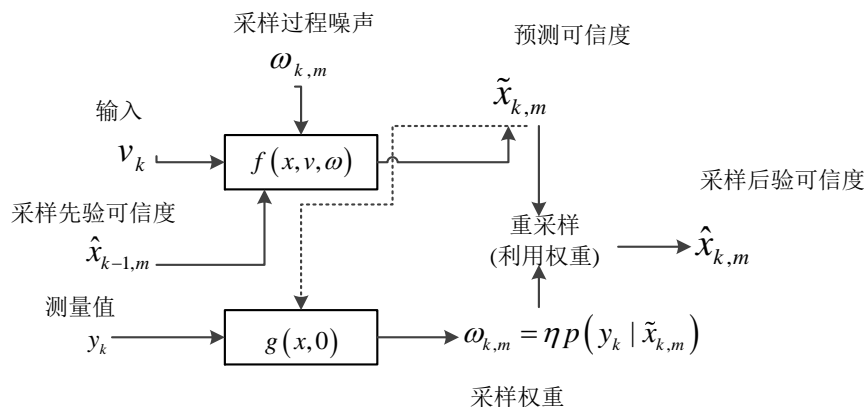


图 2-1 粒子滤波算法流程图

2.2.4 FastSLAM

FastSLAM 算法^[36]较好地改善了随着移动机器人不断运动，环境信息不断增加，粒子滤波算法效率下降的问题。FastSLAM 是基于扩展卡尔曼滤波和粒子滤波的混合算法，它通过后验 SLAM 解决了维度问题。在 FastSLAM 算法中每个粒子维护自己的地图，地图中的每个特征由低维度的 EKF 表示。

FastSLAM 算法是一种有效的同时定位与建图实现方案，利用 Rao-Blackwellized 粒子来表示后验概率，同时结合粒子滤波与卡尔曼滤波表示方式。在 Murphy 等学者所做的工作基础上，Montemerlo 等人对 Rao-Blackwellized 进行扩展，提出 FastSLAM 算法，如图 2-2 所示。

SLAM 问题被 FastSLAM 算法分解为定位问题与一系列以机器人姿态估计为前提的地标估计问题。FastSLAM 基于改进的粒子滤波器来估算机器人路径，粒子均用于卡尔曼滤波器在机器人路径估算的条件下进行地标位置估计。FastSLAM 算法使用基于树的数据存储方式，算法复杂度是 $O(M \log K)$ ， M 是粒子的数目， K 是地标的个数，即环境特征的个数，优于基于扩展卡尔曼滤波器的其他算法。FastSLAM 虽然在复杂度上比基于扩展卡尔曼滤波的 SLAM 低，但是随着地图规模的增大，粒子数量会增加很快，而且粒子的质量会下降，从而影响算法的精确度。对于 FastSLAM 的不足之处，学者们提出 FastSLAM2.0 算法^[37]，如图 2-3 所示，在估计机器人位姿时，加入一个传感器观测值和一个重要性分布函数，该方法在运动噪声比环境噪声较大的情况下可以减少粒子的浪费，进而提高算法的精度。

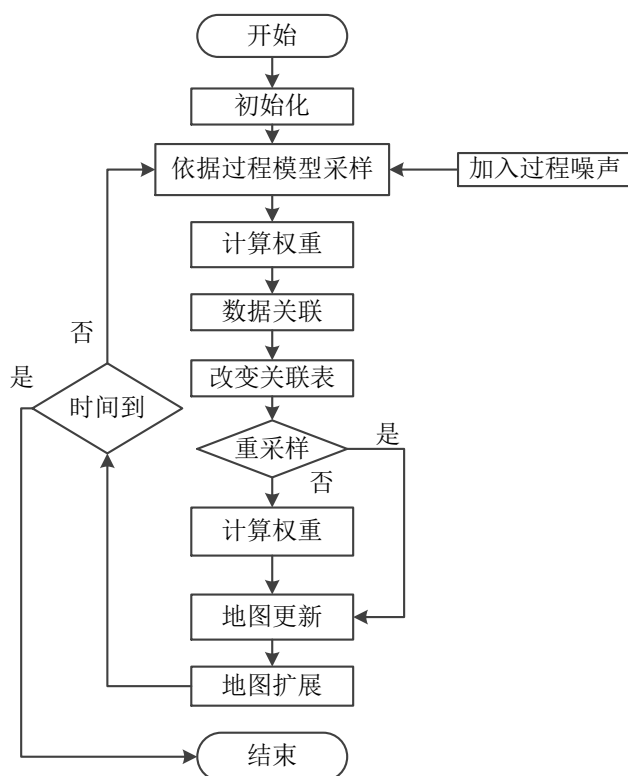


图 2-2 FastSLAM1.0 算法

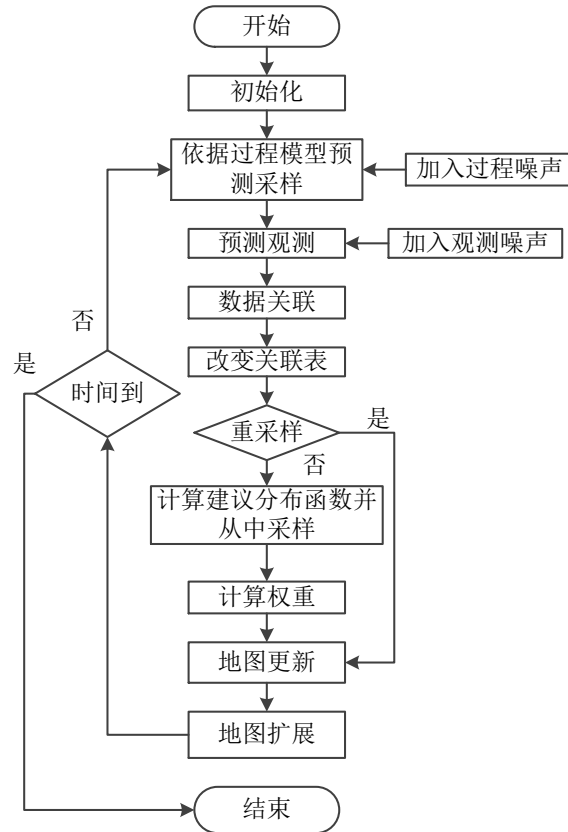


图 2-3 FastSLAM2.0 算法

2.3 视觉 SLAM 算法

视觉 SLAM 方法^{[38][39][40]}也被称为“结构和运动估计”^[41]，它使用相机传感器作为数据输入设备，从而计算机器人的位姿和创建地图。立体相机通常用于从相应图像的纹理区域视差中获得稀疏距离信息。与基于激光传感器的 SLAM 算法相比，视觉 SLAM 系统通常从相机图像中提取稀疏的关键点。视觉特征点具有比典型几何结构更加独特的优点，它简化了数据关联过程。学者们较常用的关键点检测器和描述子主要有尺度不变特征变换 SIFT (Scale-Invariant Feature Transform, SIFT) 与 ORB (Oriented Brief, ORB)，描述子可以很容易地与不同的关键点检测器相结合。最近推出的诸如 Microsoft Kinect 或 Asus Xtion Pro Live 之类的 RGB-D 相机为激光扫描仪提供了一个有价值的替代品，因为它们有低廉的价格，较小的尺寸和较轻的重量而且能够提供密集深度信息。深度传感器投射红外光谱中的结构光，该结构光由具有小基线的红外相机感知。由于结构光传感器对光照敏感，所以它们通常不适用于阳光直射的场景下使用。基于飞行时间原理的摄像机对光照的敏感性较差，但分辨率较低，噪声较大，难以校准，而且价格昂贵。Henry 等人发表了第一个 RGB-D SLAM 系统，他们将视觉特征与广义迭代最近点算法 ICP^[42] (Iterative Closest Point, ICP) 进行结合起来创建和优化姿态图。Zeng 等人表明使用八叉树数据结构^[43]来保存距离值，

能够大大减少信息存储的内存需求。Hu 等人提出了一种 SLAM 系统，该系统在具备和不具备可用物体深度信息的两种情况下交叉使用 Bundle Adjustment 方法^[44]，这使得由于距离限制和阳光等因素导致缺乏深度信息的 SLAM 实现结果更为可靠。

2.3.1 RGB-D SLAM 算法框架

RGB-D SLAM 算法框架由 Felix Endres^[20]等人于 2014 年提出。RGB-D SLAM 算法框架不需要其他传感器或测距仪，凭借低成本和轻量级的 RGB-D 传感器（如 Microsoft Kinect），RGB-D SLAM 算法框架可用于小型家用机器人，如真空吸尘器，以及各式各样的飞行器。此外，该系统还可以用于详细的三维模型的自由重建。除了系统本身之外，RGB-D SLAM 算法框架还对公开的基准数据集进行了彻底的实验评估。研究者还分析和讨论了几个参数的影响，如特征描述符的选择，视觉特征的数量和验证方法。实验结果表明，RGB-D SLAM 算法框架可以有效地处理具有挑战性的场景，例如快速摄像机运动和功能差的环境，同时具备足够快的在线操作，而且该系统是完全可用的开源，已经被学者们广泛地应用于相应的研究领域。

RGB-D SLAM 算法框架是首批较好利用 RGB-D 相机的彩色与深度图像的视觉 SLAM 算法框架之一。与之前的视觉 SLAM 算法框架相比，该算法框架引入了几个扩展，旨在进一步的提高系统的鲁棒性与准确性。具体表现在，RGB-D SLAM 算法框架提出使用环境测量模型 EMM（Environment Measurement Model, EMM）来验证通过特征对应和 ICP 算法估计的变换。

如图 2-4 所示，RGB-D SLAM 算法框架总体可以分为前端模块、后端模块和建图模块。前端模块负责实时处理传感器数据从而提取不同时间点下的机器人与地标之间的几何关系。前端模块是特定于所使用的传感器类型，传感器类型不同，前端模块也有相应的区别。除了测量运动本身的传感器之外，机器人的运动需要从一系列观察结果中计算出来。根据传感器类型的不同，可以使用不同的方法来计算两个观察值之间的运动。当输入传感器设备是 RGB-D 相机时，输入分别为彩色图像 I_{RGB} 和深度图像 I_D ，使用从彩色图像 I_{RGB} 中计算出的高维度描述子向量 $d \in R^{64}$ 来确定地标，将这些高维度描述子和 $y \in R^3$ 存储在一起，它们相对于观测者的位姿 $x \in R^6$ 。为了处理例如由传感器噪声引入的固有不确定性，SLAM 系统的后端对构建的具有不确定的图进行优化。图结构的优化可以用于获得所表示的机器人轨迹的最大似然解。利用已知轨迹，可以把传感器获取的环境信息投影到共同的坐标系中。然而，在大多数应用中，需要特定于任务的地图表示，因为直接使用传感器数据进行计算效率非常低。从 RGB-D 数据创建一个三维概率占用图，可以有效地用于导航和操纵任务。

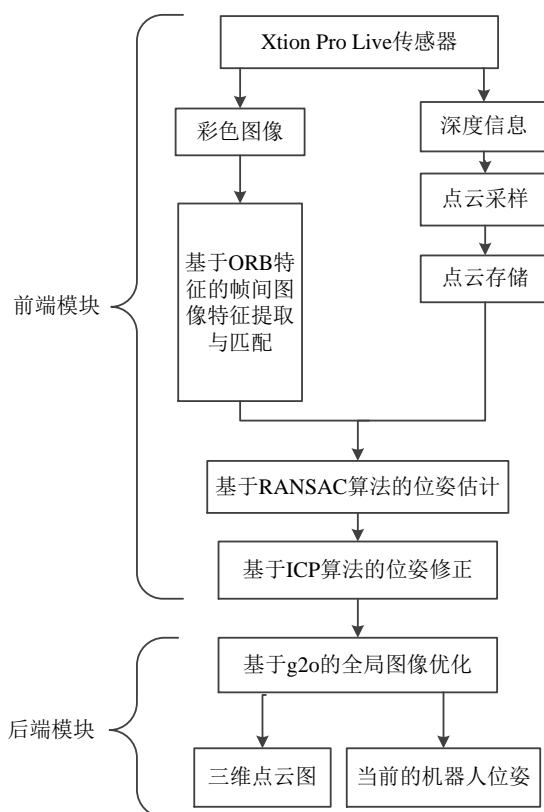


图 2-4 RGB-D SLAM 算法框架

2.3.2 ORB-SLAM 算法框架

ORB-SLAM^[21]是另一种近几年较受关注的单目视觉 SLAM 算法框架，可以在不同规模的环境中运行。该算法框架鲁棒性较好，可以进行全局自动初始化。

由于提供了强大的匹配网络和良好的初步预测，BA（Bundle Adjustment, BA）可以实现相机定位的准确估计以及稀疏几何重建。很长时间以来，这种方法被认为是实时应用程序。视觉 SLAM 的目标是在重建环境信息的同时估计摄像机的轨迹。BA 的第一个实时应用是 Mouragon 等人^[45]的视觉测距工作。其次是 Klein 和 Murray 开创的 SLAM 工作，称为 PTAM^[23]。该算法虽然局限于小规模运算，分为关键帧选择，特征匹配，点三角三部分，但却为每帧相机定位以及跟踪失败后的重定位提供了简单而有效的方法。但是 PTAM 存在局限性，比如缺少闭环检测，对闭塞情况处理不够，创建地图需要人为干预等。ORB-SLAM 算法框架是基于 PTAM 的思想，分别结合了 G'alvez-L'opez 等人^[46]的场景识别，Strasdat 等人^[47]的闭环检测。ORB-SLAM 算法框架的优点是可以在大规模环境中进行实时运行，并实时进行闭环检测，在光线变化较大的情况下也能实时对相机进行重定位。ORB-SLAM 算法框架如图 2-5 所示，主要由相机位姿跟踪、局部建图和闭环检测三个线程组成。在相机位姿跟踪线程中，主要是完成特征提取、初始化位姿估计以及关键帧的筛选；在局部建图线程中，完成

对冗余地图点与关键帧的删除，从而提高算法的运行效率；在闭环检测线程中，完成闭环检测与融合并实现全局优化。

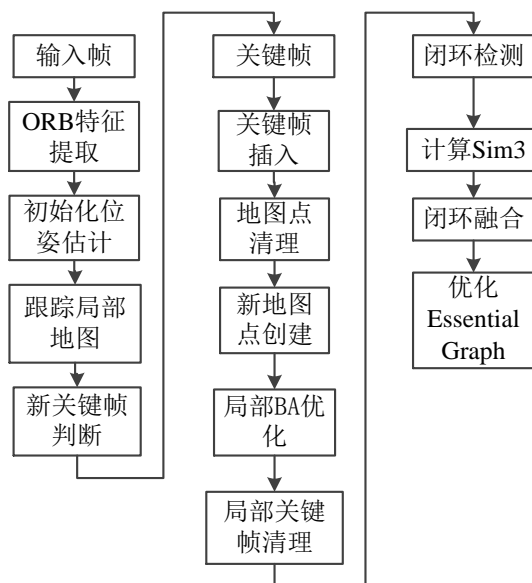


图 2-5 ORB-SLAM 算法系统框架

2.4 激光雷达 Gmapping 建图

粒子滤波器自 20 世纪 90 年代以来一直用于目标跟踪和移动智能设备的自定位。这些滤波器可以有效地处理非高斯噪声的非线性系统，为概率机器人中的复杂问题提供了解决方案。粒子滤波器在过去二十年中发展较快，出现了较多高效的方案，其特征不在于提高精度和降低计算需求。粒子滤波的核心思想在于具有潜在的非线性系统动力学和测量模型的状态空间上潜在的非高斯多模态后验概率分布的数值计算。后验近似为随机样本，也称为“粒子”。每个粒子与权重系数相关联，表明其后验概率密度的重要性。点估计值（后期的预期值或最大值）可以很容易地从蒙特卡罗表示法及其相关的不确定性度量中计算出来。以下两个问题与粒子滤波器生成质量密切相关：如何构建有效的分布方案；如何选择粒子数，这两者都影响了粒子滤波器应用的结果。

在设计粒子的分布方案时，目标是 minimized 粒子权重的方差。最佳方案指的是通过结合最新的接收测量来实现该目标，导致了分布密度和后验分布之间的最大重叠。然而，最佳方案只是一个概念性的解决方案。在实践中，仍然不清楚如何设计它。使用机器人运动模型作为分布方案是不够的，因为它在不可能状态下浪费了大量的粒子，这样可能导致样本量减少，粒子滤波器收敛于不正确的假设。学者们提出了几个近似最佳分布的方案，如辅助型粒子滤波、局部线性化滤波器等。文献^[48]提出一种用于粒子滤波的分布，该粒子分布方案受当前激光雷达测量值所影响，通常比里程计获取的数据正确，这种技术称为 Gmapping。Gmapping 是十分有效的，因为它不仅提高了粒子评估质量，而且

由于每个粒子被更好地分配，从而减少了所需的粒子数目。Gmapping 是一个很好的分布方案，分两个阶段进行。在第一阶段，在由机器人运动模型获得的粒子附近计算最大似然估计。在第二阶段，最优的分布被构造为高斯密度，其平均值和方差计算是第一阶段的最大似然估计计算的结果。Gmapping 是基于 Rao-Blackwellized 粒子滤波实现的，该算法致力于解决在给定观测值 $Z_{1:\tau} = Z_1, \dots, Z_\tau$ 和里程计测量值 $U_{2:\tau} = U_1, \dots, U_\tau$ 的条件下，如何计算出后验概率 $p(X_{1:\tau}, m | Z_{1:\tau}, U_{2:\tau})$ 并输出栅格地图，该算法首先对每一个样本进行扫描匹配，匹配成功之后，进行采样并计算高斯过程，随后进行新位姿采样并更新重要权值，然后更新地图与样本集，最后如果误差在阈值范围内则算法结束，具体算法如表 2-1 所示。

表 2-1 Gmapping 算法

| Gmapping 算法 | |
|---|---|
| 输入： $t-1$ 时刻的样本集 S_{t-1} 最新雷达扫描数据 z_t 最新里程计测量值 u_{t-1} | for 每一个 $x_j \in \{x_1, \dots, x_K\}$ do $\mu_t^{(i)} = \mu_t^{(i)} + x_j \cdot p(z_t m_{t-1}^{(i)}, x_j) \cdot$ $p(x_t x_{t-1}^{(i)}, \mu_{t-1}^{(i)})$ $\eta_t^{(i)} = \eta_t^{(i)} + p(z_t m_{t-1}^{(i)}, x_j) \cdot$ $p(x_t x_{t-1}^{(i)}, \mu_{t-1}^{(i)})$ end for $\mu_t^{(i)} = \mu_t^{(i)} / \eta^{(i)}$ $\Sigma_t^{(i)} = 0$ for 每一个 $x_j \in \{x_1, \dots, x_K\}$ do $\Sigma_t^{(i)} = \Sigma_t^{(i)} + (x_j - \mu^{(i)}) (x_j -$ $\mu^{(i)})^T \cdot p(z_t m_{t-1}^{(i)}, x_j) \cdot p(x_j x_{t-1}^{(i)}, \mu_{t-1}^{(i)})$ end for $\Sigma_t^{(i)} = \Sigma_t^{(i)} / \eta^{(i)}$ $x_t^{(i)} \sim \mathcal{N}(\mu_t^{(i)}, \Sigma_t^{(i)})$ $w_t^{(i)} = w_{t-1}^{(i)} \cdot \eta^{(i)}$ end if $m_t^{(i)} = \text{integrateScan}(m_{t-1}^{(i)}, x_t^{(i)}, z_t)$ $S_t = S_t \cup \{ \langle x_t^{(i)}, w_t^{(i)}, m_t^{(i)} \rangle \}$ end for $N_{eff} = \frac{1}{\sum_{i=1}^N (\tilde{w}^{(i)})^2}$ if $N_{eff} < T$ then $S_t = \text{resample}(S_t)$ end if |
| $S_t = \{ \}$ for 每一个 $s_{t-1}^{(i)} \in S_{t-1}$ do $\langle x_{t-1}^{(i)}, w_{t-1}^{(i)}, m_{t-1}^{(i)} \rangle = s_{t-1}^{(i)}$ $x_t'^{(i)} = x_{t-1}^{(i)} \oplus u_{t-1}$ $\hat{x}_t^{(i)} = \text{argmax}_x p(x m_{t-1}^{(i)}, z_t, x_t'^{(i)})$ if $\hat{x}_t^{(i)} = \text{失败}$ then $x_t^{(i)} \sim p(x_t x_{t-1}^{(i)}, u_{t-1})$ $w_t^{(i)} = w_{t-1}^{(i)} \cdot p(z_t m_{t-1}^{(i)}, x_t^{(i)})$ else for $k = 1, \dots, K$ do $x_k \sim \{x_j x_j - \hat{x}^{(i)} < \Delta\}$ end for $\mu_t^{(i)} = (0, 0, 0)^T$ $\eta^{(i)} = 0$ | |

2.5 本章小结

本章研究了 SLAM 实现算法的几种典型主流框架，包括基于滤波技术的卡尔曼滤波、扩展卡尔曼滤波、FastSLAM、粒子滤波，以及近几年较受关注且快速发展的基于视觉传感器的视觉 SLAM 实现算法框架，包括 RGB-D SLAM 算法框架和 ORB-SLAM 算法框架。最后介绍了被广泛应用于室内场景的二维激光雷达建图方法，即 Gmapping 方法。

3 RGB-D 相机与激光雷达数据融合

3.1 引言

正如人们在感知周围环境时会同时借助于视觉、听觉、触觉等多种信息，多传感器数据融合与信息互补的本质就是将不同种类与数量传感器获取的数据进行较好的利用。多传感器数据融合比依靠单个传感器技术有更多的优点，丰富的观察数据有利于提高可靠性，而多种类信息之间的融合可以改善单一传感器的性能，在时间和空间上，各种类传感器之间进行配合可以扩宽收集环境信息的范围。同时，在某些特定条件下，利用多个高性价比的传感器取代价格高的传感器，以获得相同或者误差允许范围内的测量精度，从而降低系统总成本。概况地说，多传感器数据融合和补偿主要有如下特性：

- (1) 扩展了时空覆盖范围；
- (2) 提升了数据的可信度；
- (3) 增强了系统的分辨能力；
- (4) 提高了系统的可靠性和鲁棒性；
- (5) 降低了系统成本；
- (6) 增强了系统的生存能力；
- (7) 减少了信息的模糊性。

传感器数据融合是一个较新的研究课题，传感器数据融合与传感器优缺点互补技术还未发展成熟。首先因为融合方法的理论基础与广义的融合方法还没有形成，许多传感器数据融合方法只适用于特定的情况，不具备普遍适用性；其次，每一种传感器都会不可避免地受自身和环境的约束，这样可能会造成测量结果的差别较大甚至是矛盾的；最后为了实现较好地各类传感器数据融合需要较复杂的融合步骤，加大了开发难度，同时这些算法有可能无法使用于一些对实时性要求较高的系统。

激光雷达测量数据存在较明显噪声，摄像头需对内部参数标定与校正。为此，选择多传感器融合与补偿拓扑结构实现融合与补偿。其特点在于两类传感器根据各自特性完成局部信息处理，然后向上层处理机构返回局部信息处理结果，由上层信息融合中心作下一步处理。通常的传感器数据融合处理步骤如图 3-1 所示。

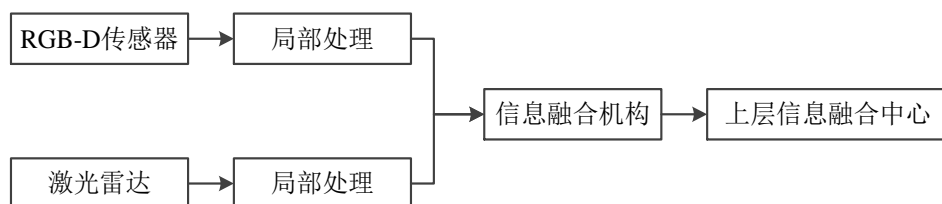


图 3-1 信息融合处理

事实上，单一的激光雷达数据或者 RGBD 摄像头数据，都难以单独对目标对象进行准确和稳定地识别，所以采用数据层的信息融合，保留足够的冗余来互补信息，为进一步的环境信息提取识别与融合提供了可能性。

3.2 深度图像预处理

3.2.1 深度图像修复方法

随着 RGB-D 相机在游戏行业的推出和基于 RGB-D 相机的应用开发套件的推出，与 RGB-D 相机相关的产业已经发展到游戏应用、控制设备的手势识别、3-D 通信等。RGB-D 传感器的普及也许能够归因于其低成本和实时深度图的生成能力。然而，由深度相机的制作原理以及环境中的反射光被遮挡或者物体边缘反射率急剧变化导致生成的深度图存在伪像，如生成的对象边界不足，缺少深度值和深度图与彩色图像之间的对准，深度图像中会出现没有深度值与某些像素关联的现象，即“黑洞”现象。深度图像中有效值缺失会影响 SLAM 生成的地图质量。因此对深度图像进行修复是有必要的。

过去几年，对由深度相机获取的深度图像的修复问题，学者们做了一些尝试。文献^[49]根据先前的 $2m+1$ 帧，先对帧与帧之间进行运动估计与补偿，然后再用加权中值滤波来填充“黑洞”。文献^[50]将深度图像区分出前景区域与背景区域，然后对前景区域和背景区域分布使用不同的方法来填充“黑洞”，而且该文献还提出了两个遮挡类别的存在，即由于前景物体的边缘而引起的遮挡以及由于发光物体和其他因素引起的遮挡。文献^[51]用双边联合滤波来对“黑洞”进行填充，滤波处理只包含那些可靠的像素，并利用高斯平均模型对深度值进行更新。本文选了三种修复方法进行介绍。

Sergey Matyunin 等人于^[49]2011 年提出一种利用由 RGB-D 相机得到的彩色图像与运动信息来提高深度图像质量。他们利用相机视频序列和与之相应的深度图像，对每一帧 i ， $I_i(x, y)$ 表示像素的密度或者是彩色信息，对彩色图像来说 $I_i(x, y)$ 是三维向量，对灰度图来说是灰度标量。 n 表示当前帧数， $D_i(x, y)$ 表示第 i 帧的 (x, y) 位置的深度值， $B_i(x, y)$ 表示第 i 帧的闭塞图。 $B_i(x, y)$ 值由公式 (3-1) 给定。

$$B_i(x, y) = \begin{cases} 1, & \text{如果像素}(x, y) \in \text{闭塞区域} \\ 0, & \text{其他情况} \end{cases} \quad (3-1)$$

算法总共包含五个步骤：第一步计算运动场矢量，计算 I_n 帧与相邻 $n-m$ 帧到 $n+m$ 帧之间的运动估计，运动场矢量为 $MV_i(x, y) = (u_i(x, y), v_i(x, y))$ ，定义 $MV_n(x, y) \equiv 0$ ；第二步计算运动场矢量的置信度量 $C_i(x, y)$ ， $C_i(x, y) \in [0, 1]$ ；第三步对深度图和每一帧图像进行运动补偿， D_i^{MC} 表示对深度图的运动补偿， I_i^{MC} 表示对每一帧图像的运动补偿， D_i^{MC} 和 I_i^{MC} 均是利用运动矢量 $MV_n(x, y)$ 按照公式 (3-2) 和公式 (3-3) 进行计算；

$$I_i^{MC}(x, y) = I_i(x + u_i(x, y), y + v_i(x, y)) \quad (3-2)$$

$$D_i^{MC}(x, y) = D_i(x + u_i(x, y), y + v_i(x, y)) \quad (3-3)$$

第四步用计算出的 D_i^{MC} 和 I_i^{MC} 以及 C_i 对深度图像像素过滤筛选；第五步用筛选出来的邻近像素对“黑洞”进行填充修复。这种方法主要使用颜色信息填充深度孔。在某些颜色信息不能反映有效深度信息的地区，该方法会产生不正确的深度值。

Na-Eun Yang 等人^[52]于 2012 年提出用“黑洞”邻近深度信息区域分布对深度图像的“黑洞”进行填充。他们首先检测出深度图像中的“黑洞”区域，因为由 RGB-D 获取的深度图像中的“黑洞”的灰度值为 0，所以很容易将黑洞检测出来，并把深度图进行二值化。然后用八连通对区域进行标记，随后用对标记区域进行膨胀处理，为了获得有效的深度值分布，他们对每一个标记的区域进行直方图分布计算，量化深度值到 0-255，然后按照公式 (3-4) 进行统计。

$$h^l(d_k) = n_k^l \quad 0 \leq k \leq 255 \quad (3-4)$$

其中， d_k 表示第 k 深度量化值， n_k^l 是在标记的区域 l 里 d_k 出现的次数。随后在给定区域标记的有效区域的直方图中有一些主峰，这些主峰是指像素总量大于该区域总像素的百分之十的值。最后，他们确定一个阈值作为主要深度值的平均值，取大于阈值的深度值的中值对“黑洞”进行填充。

文献^[53]针对文献^{[49][50][51]}所提方法均未考虑彩色图像与深度图像之间像素的对齐，而且在填充“黑洞”之后没有对深度图像边缘和深度图像进行修正，提出一种修复方法，该方法包含两步。首先是使用高斯加权移动窗口函数从相对应的彩色图像中计算出高斯权重，从而区分物体。“黑洞”可以通过一个 $N \times N$ 的窗口用“黑洞”周围像素的灰度平均值来进行填充，其中 N 是窗口的宽度。为了能够区分彩色图像中的物体，他们使用类似双边滤波中的高斯权重函数。在计算高斯权重之后用均值滤波 MF (Mean Filter, MF) 进行“黑洞”填充。 d_p 表示需要修复的像素，窗口的宽度为 w ，窗口的中心在 d_p 。在窗口内的像素表示为 d_q 其中 $q = \{1, 2, \dots, w^2\}$ 。 c_p 表示 d_p 对应彩色图像中的像素点，同样大小窗口函数也以 c_p 为中心，在该窗口内的像素为 c_q 。为了区分彩色图像中的不同物体，首先计算 c_p 与 c_q 的灰度值差。然后将这个差值用公式 (3-5) 转换为高斯权重，其中 σ_r 表示高斯函数的标准差。

$$w_q = \exp\left(\frac{-(\|c_p - c_q\|)^2}{2\sigma_r^2}\right) \quad (3-5)$$

w_q 将根据先前决定的阈值 τ 进行比较，转化为 1 或 0。然后根据公式 (3-6) 将 w_q 为 1 的值进行取均值计算，将 d_p 的值用 d'_p 进行替换。

$$d'_p = \frac{1}{N} \sum_{i=1}^{w^2} w_i d_i \quad (3-6)$$

N' 是权值 w_q 为 1 的像素数量。通过计算像素之间的差异并使用高斯加权内核中的差异，确保只有那些颜色相似的像素被用于后续的深度图像“黑洞”修复过程中。该过程通常需要几次迭代才能完全填满所有的“黑洞”。然后该文献还对修复后的深度图像进行了增强处理，目的是平滑深度图像，同时保留那些对基于深度图像进行三维重建比较重要的边缘信息。

在过去的时间内，学者们对深度图像的增强技术已经做了一些研究，学者们围绕联合双边滤波的概念提出了几种增强的方法。文献引入的联合双边滤波是双边滤波的扩展。联合双边滤波与双边滤波是依赖相同范围和域的过滤器，但两者的不同之处在于，联合双边滤波增加了彩色图像中的滤波范围。他们提出基于结构感知各向异性 AD (Anisotropic Diffusion, AD) 扩散的多尺度深度图滤波技术，各向异性扩散是一种非线性，空间变异的扩散技术。AD 是一种非线性且空间变化的扩散技术，用于图像处理，采用偏微分方程去除处理过程中的噪声。关于各向异性技术，文献^[53]描述了各向异性扩散过程，并引入了“边缘停止”功能来防止边缘扩散。在结构感知各向异性扩散的多尺度深度图滤波技术，首先将深度图像与彩色图像进行 8 因子下采样，在下采样之后，从下采样的彩色图像中利用结构感知各向异性扩散计算出高斯权重，随后以 2 因子使用各向异性扩散将深度图像向上采样，不断执行采样过程直到深度图像变回原来的分辨率。为了防止一个物体的深度渗透到另一个物体，需要区分各向异性扩散过程中的物体，引入结构相似性指数 SSIM (Structural Similarity, SSIM)，SSIM 指数将结构和颜色合并到一个度量中，它本质上是一个结构感知的图像失真指标。如果考虑以两个像素为中心的两个窗口 x 和 y 进行比较， μ_x 和 μ_y 是均值， σ_x 和 σ_y 是标准差， σ_{xy} 是两个窗口的协方差，则两个像素之间的 SSIM 按公式 (3-7) 进行计算，其中 C_1 和 C_2 是常数。文献^[53]不但使用各向异性技术，而且计算两个窗口之间的结构相似指数。他们将这种技术取名为 AD-SSIM 滤波技术。这有助于只允许具有相似邻域（颜色和结构）的像素并入到滤波处理中。

$$SSIM = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (3-7)$$

3.2.2 改进的深度图像修复方法

深度图的获取方法大致可以归为两类，一类是基于彩色图像的估计，另一类是通过深度传感器的直接测量。立体匹配是一种基于多个彩色图像获取深度图的视觉计算方法，但是这种方法计算复杂度高，深度的估计精度在无纹理和遮挡的区域将失败。成本较小的 RGB-D 相机已被用于获得 RGB 图像与深度图像。因为许多基于深度图像的应用性能严重依赖于深度信息的质量而且获得密集的深度图像对于图像处理和与计算机视觉相关的应用也同样重要，因此对深度图像进行有效地修复是具有一定实际应用意义。

RGB-D 相机传感器捕捉的深度图像，可以获取每个像素深度信息。但是这些深度图像经常会出现像素缺失的区域，即空洞现象如图 3-2 的 (b) 所示，这些空洞没有像素的深度信息。为了改善 RGB-D 相机采集到的深度图像质量，应该解决以下问题。首先，在深度图像中会出现部分像素没有深度信息的“黑洞”现象。其次，由于红外光线的多次反射或散射，深度图像受到光学噪声的影响。“黑洞”现象产生的原因主要是物体被遮挡或存在凹面物体，从而导致红外光线可能被物体吸收或遮挡引起的。深度图像的噪声也可能是由于传感器自身设备获取图像失真或者立体匹配估计所引起。在由遮挡引起的深度图像的“黑洞”周围，遮挡物与被遮挡物之间的深度值是相关的。由非反射物体引起，即红外光线被吸收的情况下，“黑洞”会在物体内部以孤立的孔洞形式出现。

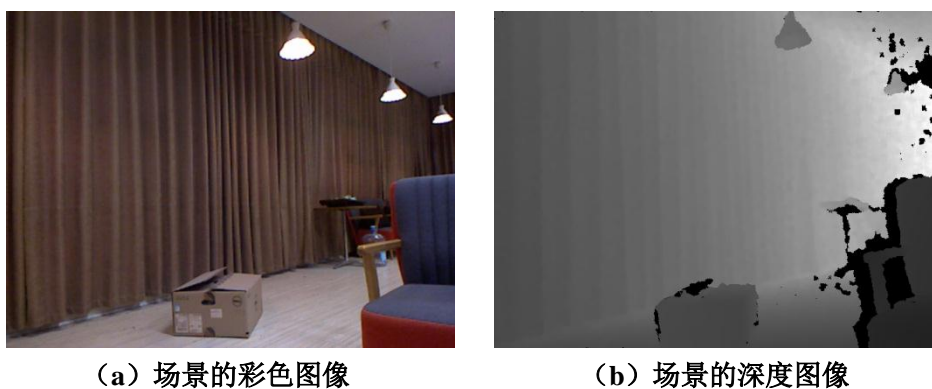


图 3-2 Xtion Pro Live 获取的图像

本文通过对深度图像中像素深度信息修复方法的研究，提出一种基于 SLIC 超像素分割的深度图像修复方法。首先分别将彩色图像进行超像素分割，将深度图像按照公式 (3-8) 进行二值化处理从而检测出“黑洞”，如图 3-3 所示。然后用八连通方法对“黑洞”进行标记，标记后如图 3-4 所示，随后将标记的“黑洞”区域对应到已经完成超像素分割的彩色图像中，根据“黑洞”在分割块中位置不同，采取不同的修复方法进行修复，最后利用交叉双边滤波对改善后的深度图像实现增强。

$$D_p = \begin{cases} 1 & 255 \geq d_p \geq 1 \\ 0 & d_p = 0 \end{cases} \quad (3-8)$$

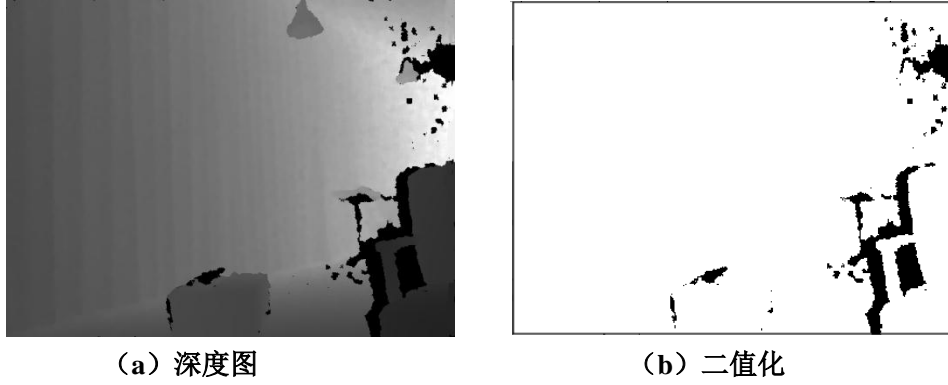


图 3-3 二值化结果图

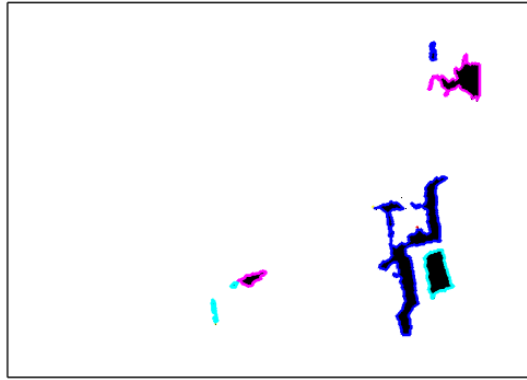


图 3-4 八连通区域检测

3.2.3 SLIC 超像素分割

超像素将像素分组为可感知且有意义的子区域，可用于替换像素网格的刚性结构。超像素减少了图像的冗余度，为后续图像处理过程减少了复杂度。SLIC 超像素分割是依据图像平面中的像素色彩信息的相似性与像素之间的距离是否接近来划分像素块的。该算法是在五维空间 $[labxy]$ 完成的， $[lab]$ 是像素颜色信息在 CIELAB 空间中的表示向量， xy 则表示像素在该空间中的位置信息。为了实现在五维空间中对像素进行聚类，该算法提出了一个全新的距离度量概念，该度量考虑到了超像素的尺寸，用这个距离度量可以使得聚类的大小和它们在空间的大小大致相等。

第一步假设 K 为预期超像素数量， N 是总的像素数量，可以计算出超像素的尺寸为 N/K ，相邻的两个超像素区域的中心之间距离约等于 $S = \sqrt{N/K}$ 。在像素开始聚类之前，以 S 作为间隔对像素进行采样以达到初始化目的，设定 K 超像素的聚类中心为 $C_k = [l_k, a_k, b_k, x_k, y_k]^T$ ，其中 $k = [1, K]$ 。超像素的空间范围近似为 S^2 ，那么超像素的聚类范围为 $2S * 2S$ ，为了在五维空间更好的描述颜色距离信息，定义新的距离度量 D_s 如公式（3-9）、（3-10）、（3-11）所示，其中通过设定 m 的值可以调节超像素的紧密度。

$$d_{lab} = \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2} \quad (3-9)$$

$$d_{xy} = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2} \quad (3-10)$$

$$D_s = d_{lab} + \frac{m}{s} d_{xy} \quad (3-11)$$

相关步骤如表 3-1 所示，在初始化的聚类中心的 $n \times n$ 的范围内，通过公式（3-12）计算该范围内的梯度值，其中 $I(x, y)$ 表示 xy 位置像素的 $[lab]$ 空间的向量，随后把梯度最小位置选为新的聚类中心。

$$G(x, y) = \|I(x + 1, y) - I(x - 1, y)\|^2 + \|I(x, y + 1) - I(x, y - 1)\|^2 \quad (3-12)$$

在分配步骤，每个像素都与其距离最近的聚类中心建立关系，该聚类中心的搜索区域需要包括该像素，这样就可以减少计算量，因为限制搜索区域的大小减少了需要计算距离度量的像素数量。在分配步骤结束之后，更新步骤会将每块搜索区域的聚类中心更新为该搜索区域中所有像素向量 $[lab_{xy}]$ 的平均值。分配步骤与更新步骤被不断地重复执行直到误差收敛到可接受的范围。本文由 RGB-D 相机获取的彩色经 SLIC 超像素分割的效果图如图 3-5 所示。

表 3-1 SLIC 超像素分割算法

SLIC 超像素分割算法

以 S 为采样间隔将像素进行初始化，聚类中心初始化为 $C_k = [l_k, a_k, b_k, x_k, y_k]^T$

计算 $n \times n$ 范围的梯度值，将聚类中心移至梯度值最小的位置

重复

for 每一个聚类中心 C_k do

 根据公式（3-11）的距离度量，从 $2S * 2S$ 的领域内分配最佳匹配像素

end for

更新聚类中心以及计算新聚类中心与旧聚类中心之间的残差 E

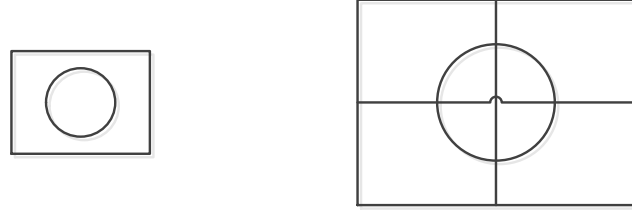
直到 $E \ll$ 阈值



图 3-5 SLIC 超像素分割结果图

3.2.4 分类修复深度图像缺失值

深度图像中的“黑洞”位置对应到经过 SLIC 超像素分割的彩色图像上，其与分割块的位置关系分为两种情况，如图 3-6 所示。



(a) “黑洞”在分割块内

(b) “黑洞”在分割块交界处

图 3-6 “黑洞”与分割块的位置关系

当“黑洞”在分割块内时，对分割块内的所有像素按照公式 (3-13) 统计该分割块内出现次数最多的灰度值，因为深度图像中的灰度值即表示像素的深度信息，所以用该灰度值填充“黑洞”。

$$Sum(d_k) = n \quad 0 \leq k \leq 255 \quad (3-13)$$

当“黑洞”位置是在几个分割块的相交处时，首先统计各个重叠部分的像素数目 s_l ，其中 l 表示与“黑洞”有重叠的分割块，定义参数 $\alpha_l = s_l/S$ ，其中 S 为“黑洞”的像素总数。随后同样的按照公式 (3-14) 统计该分割块内 l 出现次数最多的灰度值 d_l ，最后根据公式计算出的灰度值 d 填充“黑洞”。

$$d = \alpha_l d_l \quad (3-14)$$

3.2.5 交叉双边滤波

在“黑洞”修复之后，对深度图像采用交叉双边滤波进行滤波以达到平滑和保留边缘信息的目的。公式 (3-15) 和公式 (3-16) 表示的是对输入的图像 D' 在像素 p 进行双边滤波输出 D 的过程， Ω 表示 p 的周围像素集合， $\omega(p)$ 是一个归一化参数。双边滤波器的权重是依据空间滤波核 s 计算，而 s 对 p 与 q 进行空间距离即 $\|p - q\|$ 进行估计，范围滤波核 r 用于估计 $|D'(q) - D'(p)|$ 。而交叉双边滤波^[54]是双边滤波器的变体，在交叉双边滤波器中输入的要平滑的图像数据是与有边缘信息的图像数据是分离的，这样就可以在平滑的同时也很好的保存了边缘信息。因此，交叉双边滤波中范围滤波核估算的对象不是整个地图 D' ，而是 I 。交叉双边滤波器可以在双边滤波器中增加一个内核 c 来实现，用于评估待过滤的视差估计的可靠性。交叉双边过滤器由公式 (3-17) 和公式 (3-18) 定义。

$$D(p) = \frac{1}{\omega(p)} \sum_{q \in \Omega} s(\|p - q\|) r(|D'(q) - D'(p)|) D'(q) \quad (3-15)$$

$$\omega(p) = \sum_{q \in \Omega} s(\|p - q\|) r(|D'(q) - D'(p)|) \quad (3-16)$$

$$D(p) = \frac{1}{\omega(p)} \sum_{q \in \Omega} s(\|p - q\|) r(|I(q) - I(p)|) c(q) D'(q) \quad (3-17)$$

$$\omega(p) = \sum_{q \in \Omega} s(\|p - q\|) r(|I(q) - I(p)|) c(q) \quad (3-18)$$

3.3 雷达数据预处理

传感器在工作时，通常会将收集到的环境信息转换变为电信号，目前为了便于系统对数据进行处理，传感器通常把这些电信号经模数处理变成数字信号，随后将数据传送到上位机进行后续处理，或者传送到数据存储单元保存。传感器在对信息进行采集、传输及一系列后续处理过程中，会受到多方面的影响如外界环境、系统性能与人为因素等，无法避免地受到噪声干扰，影响数据质量，甚至淹没了数据的特征，给后续的分析处理带来了一定的难度。因此对传感器获取的数据进行预处理从而提高数据质量是必要的。

移动车载激光雷达不可避免地存在抖动，而且外界光照条件不可能保持恒定不变，加上被测物体存在镜面反射或透射而使测量不准，所获得的距离角度信息通常包含了很多噪声，在使用之前要进行必要的滤波处理。激光雷达采集的数据属于序列数据，对于序列数据的预处理通常采取滤波处理，常用的有中值滤波、低通滤波、一维模板滤波等。作为一种精度较高的传感器，激光雷达采集的数据在滤波过程中应尽量保持数据的原始性，考虑到这一点，低通滤波和模板滤波由于会大量改变原始数据而并不适用，所以本文采用中值滤波，如公式（3-19）所示。

$$y(i) = \text{median}\{x(i - N), \dots, x(i), \dots, x(i + N)\} \quad (3-19)$$

3.4 传感器数据融合

3.4.1 深度图像信息映射

为了将激光雷达传感器与 RGB-D 相机的优点进行融合，需要将由 RGB-D 相机获取的图像深度信息转换成激光雷达扫描形式的数据。首先按公式（3-20）和公式（3-21）计算出 RGB-D 相机的“扫描”范围。

$$\theta_{min} = \arccos\left(\frac{r_R r_C}{|r_R| |r_C|}\right) \quad (3-20)$$

$$\theta_{max} = \arccos\left(\frac{r_L r_C}{|r_L| |r_C|}\right) \quad (3-21)$$

$$\theta_{inc} = \frac{(\theta_{max} - \theta_{min})}{(w - 1)} \quad (3-22)$$

其中 r_R , r_C , r_L 均为三维向量，分别表示深度图像中间一行的最右边，中间和最左边像素点在空间中的三维坐标。如图 3-7 所示， θ_{max} , θ_{min} 分别表示由 RGB-D 相机转换成的虚拟扫描最大，最小角度。 θ_{inc} 为虚拟扫描增量角， w 为深度图像的宽度。

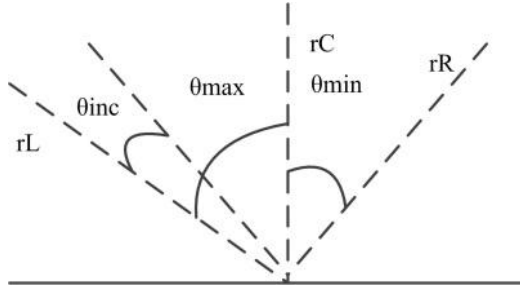


图 3-7 深度图像转换成“扫描”数据

由图 3-8 所述的小孔成像模型，可由公式（3-23）计算出 X 。

$$X = (x \cdot Z) / f \quad (3-23)$$

其中 f 为相机焦距， x 为光学原点到像素点的距离， Z 为物体到相机成像板的距离。因此，由公式（3-24）可在每个像素点计算出虚拟扫描原点到物体的距离。

$$r_{ij} = \sqrt{(X^2 + Z^2)} \quad (3-24)$$

公式（3-25）表示选出每列中的最小值，其中 i 与 j 是图像的行与列的下标，图像大小为 320×240 像素。这样不仅可以压缩数据，减少计算量，而且模拟激光雷达获得与物体的垂直距离。

$$r_j = \min(r_{1j}, r_{2j}, \dots, r_{640j}) \quad (3-25)$$

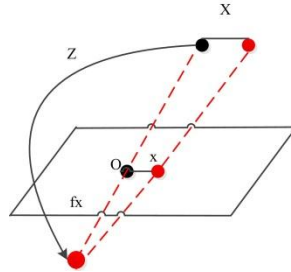


图 3-8 小孔成像原理

3.4.2 相机与激光雷达数据融合方法

Rao-Blackwellized 粒子的粒子滤波 RBPF（Rao-Blackwellized Particle Filter, RBPF）是一类蒙特卡洛方法，使用粒子集来近似后验概率的 SLAM 问题。根据文献^[55]，其核心思想是估算地图 m 的联合后验概率分布 $p(x_{1:t}, m | z_{1:t}, u_{1:t-1})$ 和机器人运动轨迹 $x_{1:t} = x_1, \dots, x_t$ 。其中， $z_{1:t} = z_1, \dots, z_t$ 为移动平台的观测值， $u_{1:t-1} = u_1, \dots, u_{t-1}$ 为里程计测距值。公式（3-26）是 Rao-Blackwellized 粒子滤波解决 SLAM 问题的计算公式：

$$p(x_{1:t}, m | z_{1:t}, u_{1:t-1}) = p(m | x_{1:t}, z_{1:t}) \cdot p(x_{1:t} | z_{1:t}, u_{1:t-1}) \quad (3-26)$$

这种方法使我们能够首先估计轨迹然后根据轨迹来计算地图。由于地图非常依赖机器人，这种方法提供了一种有效的姿态估计计算方式。

本文应用 Gmapping 方案来建立二维网格地图，该方法自适应开源算法，减少了 Rao-Blackwellized 方法中的粒子数目，较好地实现机器人 SLAM。通过解耦状态空间的马尔可夫链，简化了 RBPF 计算估计概率所需的步骤，并大大减少数量粒子。Gmapping 通过使用近似采样权重来确定粒子高斯分布，减少了样本的数量，优化了计算效率。利用 Gmapping 建立了室内环境的二维网格图之后，移动平台轨迹也会被计算出来。

本文提出的数据融合方法是在 Gmapping 建图进程运行过程中，实时地调用 RGBD 摄像头发布的 RGB 图像和深度图像节点，通过 3.4.1 小节中介绍的数据转换方式，得到障碍物的深度信息，再进行数据融合。具体融合算法如式 (3-27) 所示。

$$G(\text{depth}_{rgb}, \text{map}_{lidar}) = \sum_{i=1}^n \sum_{j=1}^m r(g_{rgb}(i, j), g_{lidar}(i, j)) \quad (3-27)$$

其中用 map_{lidar} 来表示雷达地图， depth_{rgb} 表示摄像头深度信息， $r(g_{rgb}(i, j), g_{lidar}(i, j))$ 代表了具体融合规则，详见表 3-2。

表 3-2 信息融合规则

| | | Xtion Pro Live | | |
|---------|----|----------------|----|----|
| | | 占据 | 空闲 | 未知 |
| RPLIDAR | 占据 | 占据 | 占据 | 占据 |
| | 空闲 | 占据 | 空闲 | 空闲 |
| | 未知 | 占据 | 空闲 | 未知 |

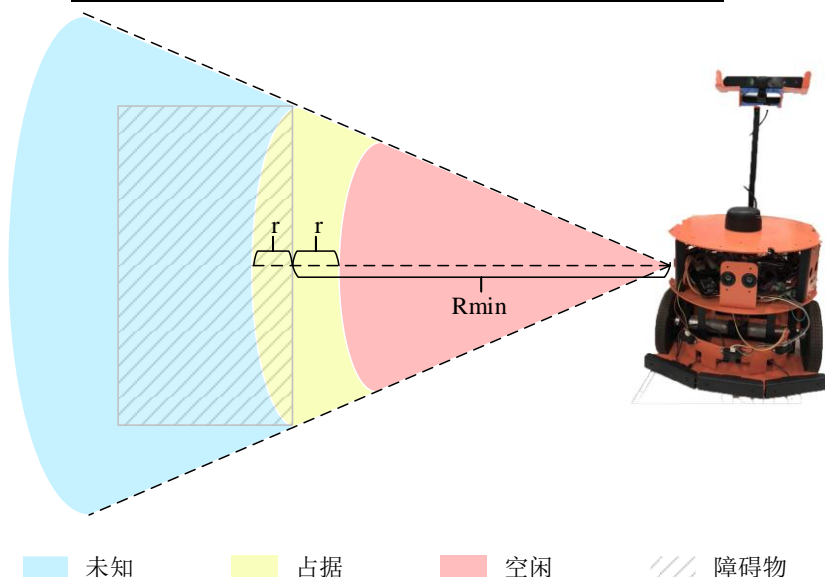


图 3-9 地图状态定义

如图 3-9 所示，传感器获得的数据可以分为三种状态：空闲、占据和未知。 R_{min} 代表机器人到障碍物的最短距离， r 代表机器人最小转弯半径。为了留给机器人充足的避障区域，以 $R_{min} \pm r$ 为半径所得到的扇区被定义为占据，占据区

域与机器人之间的扇形区域被定义为空闲，占据区域之外的区域被定义为未知，因为传感器无法获取该区域的数据。

3.5 本章小结

本章研究了针对深度图像中的像素深度信息缺失而提出的修复方法，主要研究通过“黑洞”周边深度信息分布进行修复、利用彩色图像和运动信息进行修复、使用高斯权重与取均值这三种修复方法。并在研究与分析深度图像修复方法的基础上，提出一种利用 SLIC 超像素分割的深度图像“黑洞”填充方法。并提出一种将激光雷达与深度相机获取的信息进行融合的方法。

4 移动机器人路径规划系统设计

4.1 引言

随着机器人技术和智能家电的发展，移动机器人如何较快且准确地实现这些智能任务近来备受关注，其中导航能力是评估智能机器人性能的重要指标。如我们所知，导航是环境感知和路径规划的结合。对于环境感知，目前在移动机器人最常使用的是 SLAM。与视觉 SLAM 相比，基于激光雷达的 SLAM 因其较高的精度和较低的计算复杂度被广泛地应用于移动机器人。利用 SLAM 提供的地图，路径规划方法可以帮助机器人平稳准确地移动而不会发生碰撞。路径规划作为移动机器人必不可少的能力，对其研究具有重要意义。

4.2 路径搜索算法

4.2.1 A*算法

PETER E.等人^[56]在 1968 年提出 A*算法，A*算法是用于搜索最短路径的算法。A*算法基本思想是结合 Dijkstra 算法所用的离起始点最近的顶点信息和 Greedy best-first search 算法采用的离目标点最近的顶点信息。一般用公式 (4-1) 来说明 A*算法，其中 n 是路径上的最后一个节点， $g(n)$ 表示从起始点到 n 的路径代价（例如步数）， $h(n)$ 是一个启发式，预测从顶点 n 到目标点的最划算的代价。 $h(n)$ 依据不同的问题具体形式是不同的。每一次循环，算法找到一个 n 使得 $f(n)$ 最小，从而搜索到目标点的最小消耗，即最短路径。A*算法的实现一般是用一个优先级队列按一定的规则来重复地对节点进行存储与删除，该优先级队列一般称为开集合。在算法进行的每一步， $f(n)$ 值最小的节点从开集合中被移除，同时计算更新该节点周边节点的 $f(n)$ 和 $g(n)$ 的值，并将这些节点加入开集合。算法一直重复直到目标节点的 $f(n)$ 值低于开集合中的任何节点的 $f(n)$ 值。目标节点的 $f(n)$ 值即为最短路径的长度。目前为止描述的算法只给出了最短路径的长度。为了找到实际的步骤序列，可以很容易地修改该算法，使得路径上的每个节点跟踪其前任。在算法运行之后，结束节点将指向其前任，依此类推，直到某个节点的前任是开始节点。图 4-1 是 A*算法的示意图。

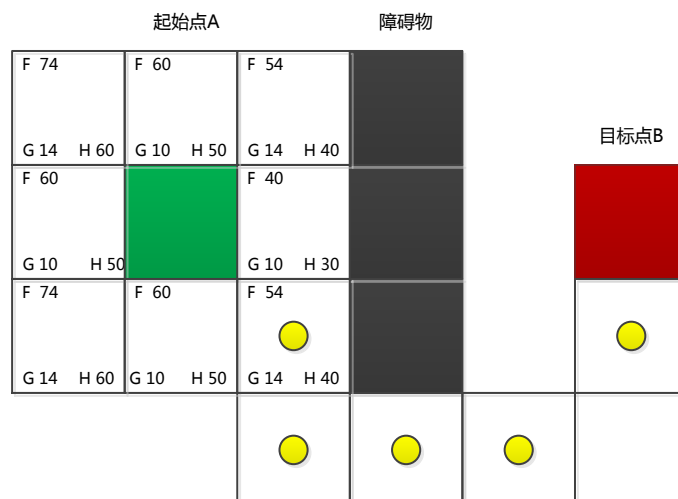


图 4-1 A*算法示意图

$$f(n) = g(n) + h(n) \quad (4-1)$$

A*算法因其灵活性和易于使用特性，在路径搜索方法中备受欢迎。但是经典的A*算法存在一些缺陷，比如不能感知机器人自身的大小。学者们对A*算法进行改进，陈刚等人^[57]对A*算法进行改进，并应用于游戏地图中的路径搜索，钱红昇等人^[58]针对经典A*算法效率不高的不足之处，在城市道路网络分层的应用背景下对A*算法进行了改进，提高了算法在实际应用场景的搜索效率。

4.2.2 D*算法

如何更快且准确地实现移动机器人路径规划任务问题一直是学者们的研究课题。大多数的研究者都假定机器人在路径规划开始之前就已经有一个完整而且准确的环境模型，而对部分已知环境模型情况，关注较少。部分已知环境模型的应用情况一般是探索机器人在没有完整的地图情况下，必须移动到目标位置。现有的方法大都基于已知信息先初始化路径规划，然后当传感器发现障碍物的时候，进行局部修改路径或者重新规划整个路径，这种方法放弃了优化且计算效率较低。

D*算法能够以一种高效、最优和完整的方式在未知的、部分已知的和不断变化的环境中实现路径规划。研究文献已经解决了一个或多个机器人通过障碍物到达目标的运动规划问题。大部分工作都假定在机器人开始移动之前，环境是完全已知的。运动规划算法使用距离变换或启发式搜索状态空间（例如，可见性图，网格单元）以找到从机器人的开始位置到目标位置的最低成本路径，成本可以定义为旅行距离，能量消耗，暴露于危险的时间等。

在机器人开始移动之前，机器人可能具备部分或者没有关于环境的信息，但是配备有能够在移动时测量环境的传感器。这种情况下的一种路径规划方法是使用已知的信息生成全局路径，然后尝试绕过那些传感器检测到的障碍物，当出现当前路径被完全堵塞，则重新规划一条路径。Lumelsky 等人^[59]最初假设

环境中没有障碍物，直接将机器人移向目标。如果障碍物妨碍路径，则机器人围绕周边运动直到找到离目标最近的障碍物的点，机器人随后继续直接运动到目标。Pirzadeh^[60]采取一种新的策略，机器人在环境中游荡直到发现目标。Korf 等人^[61]使用初始化地图信息来估算每个状态的目标成本，并在机器人在环境中移动时有效更新回溯成本。

虽然这些方法是完整的，但是它们也并不是最理想的，因为在获取传感器信息的情况下它们不产生最低成本路径，并假设所有已知的先验信息是正确的。通过从已知的地图信息中计算最佳路径，沿着路径移动机器人直到其达到目标或者其传感器检测到地图和环境之间的差异，对地图进行更新，随后重新从当前状态规划出到目标状态的最佳路径。虽然这种暴力的重新规划方法是最优的，但它可能是非常低效的，特别是在目标遥远而且地图信息很少的广阔环境中。Zelinsky 等人通过使用四叉树表示自由和障碍空间来提高效率，从而减少了在规划空间中搜索的状态数量。然而，对于自然地形，地图可以在连续区域上的每个位置处对机器人可遍历性进行编码，从而让四叉树的使用不适当或不理想。

D*算法是一种使用传感器和环境地图来生成最优路径规划的方法，环境地图可以是完整的、未知的或包含有关环境的部分信息。对于未知的环境区域，地图可能包含近似信息，占用随机模型，甚至启发式估计。D*算法是根据有向图中的最佳发现路径问题来制定的，因此该算法可以用于任何规划表示，包括可见性图和网格单元结构。

路径规划者的目标是将机器人从世界上的某个位置移动到目标位置，从而避免所有的障碍物并使正的成本度量（例如，横越的长度）最小化。机器人从一个特定的状态开始，跨过弧线（引起遍历的代价）到达其他状态，直到达到目标状态，用 G 表示。除 G 之外每个状态 X 都有一个反向指向下一个状态 Y 的指针 $b(X) = Y$ 。D*使用反向指示器来表示目标的路径。从状态 Y 到状态 X 的成本是由成本函数 $c(X, Y)$ 给出的正数。如果 Y 没有 X 的弧，那么 $c(X, Y)$ 是不确定的。如果定义了 $c(X, Y)$ 或 $c(Y, X)$ ，则两个状态 X 和 Y 是空间中的邻居。

像 A*一样，D*也有一个用于保存状态的开列表。该列表用于记录成本函数的变化信息以及计算路径成本。对每一个状态 X 都有一个标志 $t(X)$ ，如果 X 从未在开列表中，则 $t(X) = new$ ，如果 X 目前在开列表中 $t(X) = OPEN$ ，如果 X 已经不在开列表中，则 $t(X) = CLOSED$ 。对每一个状态 X ，D*算法通过成本函数 $h(G, X)$ 来计算 X 到 G 的成本。在一定的条件下，这个估计等价于由隐式函数 $o(G, X)$ 给出的从状态 X 到 G 的最优（最小）成本。对于在开列表的每个状态 X ，即 $t(X) = OPEN$ ，关键函数 $k(G, X)$ 等于 $h(G, X)$ 修改之前的最小值，从状态 X 存进开列表开始，所有值估计均通过 $h(G, X)$ 完成。关键函数将开列表中的状态分为两种类型：如果 $k(G, X) < h(G, X)$ ，则称为“升状态”，如果 $k(G, X) = h(G, X)$ 则称为“降状态”。D*算法用开列表的“升状态”来记录关于路径成本

增加的信息，用“降状态”来记录路径成本减少的信息。上述两种信息的变化通过从开列表中反复移除状态而发生。每次从列表中删除一个状态，它都会被扩展，以将成本变化传递给它的邻居。这些邻居依次被放置在开列表中以继续这个过程。开列表中的状态由它们的关键函数的值进行排序。参数 $k_{min} = \min(k(X))$ 是在开列表中的全部 X 的参数，即 $t(X) = OPEN$ 。参数 k_{min} 是 D* 算法中的一个重要参数：路径成本小于或等于 k_{min} 是最佳的路径，大于 k_{min} 则是不佳的路径。参数 k_{old} 的值等于从开列表中最新移除的状态的 k_{min} 值。如果没有状态从开列表中被移除，则 k_{old} 是未定义状态。

对于所有涉及目标状态的两态函数都用 $f(X) \equiv f(G, X)$ 函数表示。用 $f(o)$ 表示独立于作用域的函数。D* 算法主要有两个函数组成，状态处理函数和成本修改函数。状态处理函数用于计算到达目的地的最佳路径成本，成本修改函数是用于改变成本函数 $c(o)$ 以及并将受影响的状态放入开列表中。初始化所有状态的 $t(o) = new$, $h(G) = 0$, G 是开列表中的状态。状态处理函数一直被循环执行，直到机器人的状态 X 已经从开列表中移除，即 $t(X) = CLOSED$ 或者函数返回值等于负 1 时，此时序列 $\{X\}$ 已被计算或者不存在。然后机器人按照序列 $\{X\}$ 进行前进，直到机器人到达目的地或者公式 $c(o)$ 出现错误（出现障碍物）。成本修改函数用于校正 $c(o)$ ，并将受影响的状态放入开列表中。用 Y 表示 $c(o)$ 出现错误的状态。调用状态处理函数，直到其返回 $k_{min} \geq h(Y)$ ，成本变化被传到状态 Y , $h(Y) = O(Y)$ 。这样就生成了一个新的序列 $\{Y\}$ ，机器人将沿着新序列移动到目标任务。

4.3 视觉 SLAM 算法框架分析

传感器技术不断提高的同时，各类 SLAM 算法框架也应运而生。在各类传感器中，RGB-D 相机以较低价格和在场景识别中能丰富环境信息的特点，被广泛地应用于 SLAM，即视觉 SLAM。ORB-SLAM2 和 RGB-D SLAM 是近几年较受关注的 SLAM 算法框架。

ORB-SLAM2 是目前较优秀的 SLAM 算法系统框架，该系统主要由三个平行线程组成，如图 4-2 所示。

(1) 相机位姿跟踪线程：首先将输入帧进行预处理，如图 4-3 所示。然后相机位姿跟踪，若不插入关键帧则不启动局部地图线程。局部地图跟踪是把当前帧与部分地图进行匹配，利用最小化投影误差来优化位姿。最后依据是否很长时间没有插入关键帧或者跟踪失败等因素来判断是否生成关键帧。

(2) 局部地图线程：该线程主要是负责处理新生成的关键帧，剔除地图中新添加的质量不高的地图点，然后将通过筛选的关键帧进行三角化恢复部分地图点。而后进行地图点的融合并进行局部 BA(Bundle Adjustment) 优化，最后将那些关键帧根据一定的规则进行筛选，以减少重复率，节省存储空间。

(3) 闭环检测及矫正线程：利用基于 DBoW2 的场景识别进行闭环检测，而后通过优化位姿图来实现闭环矫正。

在闭环检测及矫正线程结束之后，触发第四个线程进行全局的 BA 优化，并更新地图。该算法利用 ORB 特征点完成跟踪、建图、以及场景识别的任务。这些 ORB 特征点对于旋转和尺度变换有较强鲁棒性，不受光线变换的影响并且提取速度较快，可以满足实时性的需求。

而 RGB-D SLAM 主要由前端与后端两部分组成，前端部分处理深度相机获取的环境信息，解算出环境信息的空间几何位置，后端用 g2o 库将前端部分得到的机器人位姿进行优化。最终得到环境三维点云图和机器人位姿。

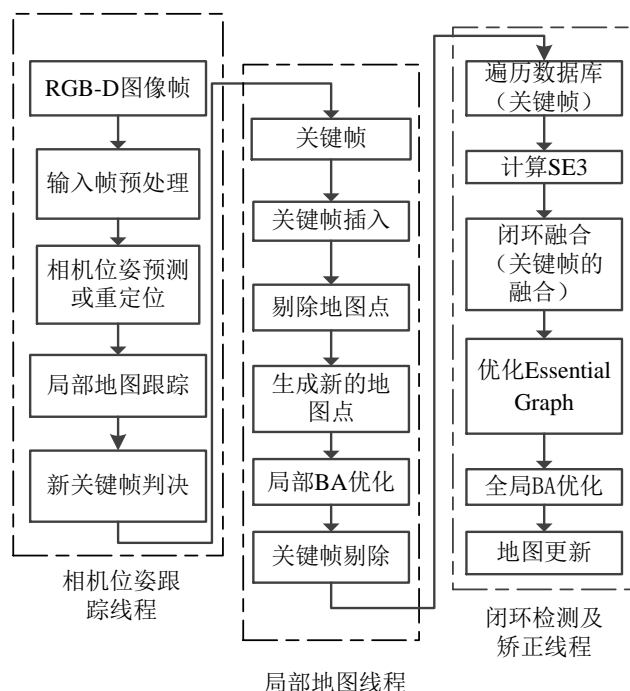


图 4-2 ORB-SALM2 算法框架

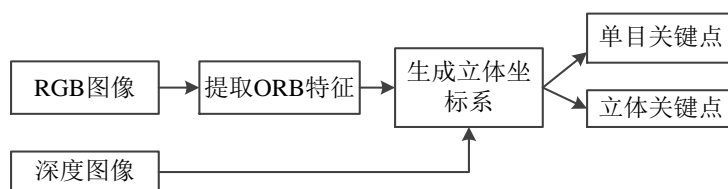


图 4-3 图像预处理

4.4 路径规划系统设计

该系统首先通过 HCR 机器人搭载激光雷达和 RGB-D 传感器，在 ROS 的运行环境下，分别实现基于 ORB-SLAM2 和 RGB-D SLAM 算法框架的同时定位与建图，并对比两种算法生成的三维点云图效果。然后针对激光雷达只能获取

特定高度环境信息的不足之处，将较好的三维点云图与激光雷达生成的二维平面图进行融合，得到含有较丰富环境信息的环境二维栅格地图，并在此融合的栅格地图上利用 A*算法路径搜索算法，对机器人进行路径规划。室内 HCR 机器人路径规划系统的实现方案如图 4-4 所示。

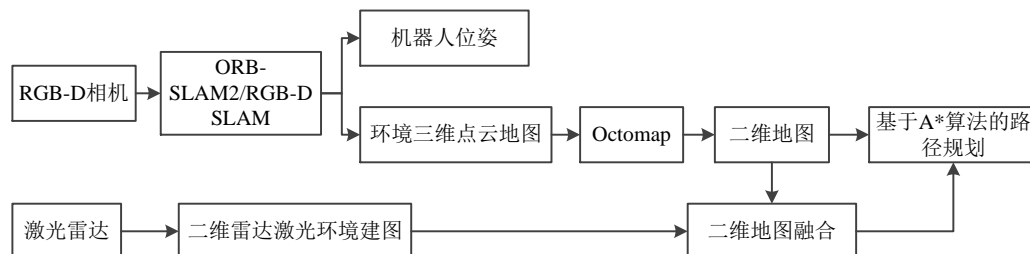


图 4-4 路径规划系统的实现方案

实现方案由解算机器人位姿、获得环境地图与实现路径规划三部分组成：

(1) 解算机器人位姿：由 RGB-D 相机采集环境的彩色和深度信息，经由 ORB-SLAM2 算法或者 RGB-D SLAM 算法，解算出机器人的当前位姿 $[X_m \ Y_m \ \theta_m]$ 。

(2) 获得环境地图：通过（添加了三维点云建图模块）ORB-SLAM2 或者 RGB-D SLAM 实现算法框架，拼接融合关键帧生成环境三维点云图，通过 ROS 的第三方库 Octomap 得到三维网格地图 Octomap 图，该地图像是由很多小方块组成，分辨率高时，方块较小。Octomap 用 0 到 1 的概率值来区分方块为三种状态：空闲（概率值 < 0.5 ）、占据（概率值 > 0.5 ）与未知（概率值 $= 0.5$ ）。Octomap 用八叉树来存储环境信息，遍历八叉树，将概率大于 0.5 的全部方块投影得到二维平面地图。然后将此平面地图与激光雷达生成的地图进行或操作，得到融合后的二维栅格地图。

(3) 实现路径规划：在融合了环境点云信息的二维栅格地图上，将机器人的位姿和机器人模型导入，确定一个目标点，通过 A*算法，搜索出一条路径，并生成机器人控制命令。

4.5 本章小结

本章首先研究了用于路径规划的经典路径搜索方法，分别是 A*算法与 D*算法，并对比两种视觉 SLAM 算法框架，即 RGB-D SLAM 算法框架与 ORB-SLAM2 算法框架，随后完成对室内移动机器人路径规划系统的设计。

5 实验平台与实验结果分析

5.1 实验平台搭建

本实验借助的机器人是 HCR，该机器人是一个开源移动机器人平台，有丰富的传感器接口，配置较新的减速电机，整合了 Arduino Mega 和 Nano 等控制器，适合算法开发。机器人实验平台搭建分为三个步骤，首先是 HCR 机械部件的整机组装，然后对底层传感器分别进行调试，调试电机驱动器的控制和跌落传感器、碰撞传感器的刹车机制。随后是设计机器人的运动控制层，并建立运动控制层和上位机之间的通信，通过在上位机中发布主题来发送电机控制消息、订阅电机编码器的计数值等运动控制消息。最后在嵌入式开发板中集成激光雷达传感器与 RGB-D 传感器，完成 SLAM 算法的实现。

下面介绍本实验中主要使用的组件。考虑到机器人本体的负重，用 NVIDIA 嵌入式 Linux 开发板代替笨重的计算机是一个合适的方案。在嵌入式开发板上安装 Ubuntu14.04 操作系统。采用思岚科技公司的 RPLIDAR 激光雷达传感器，如图 5-1 所示；华硕公司所产的 RGB-D 相机（Xtion Pro Live），如图 5-2 所示。搭建完毕的机器人平台如图 5-3 所示。



图 5-1 RPLIDAR 传感器



图 5-2 Xtion Pro Live 传感器

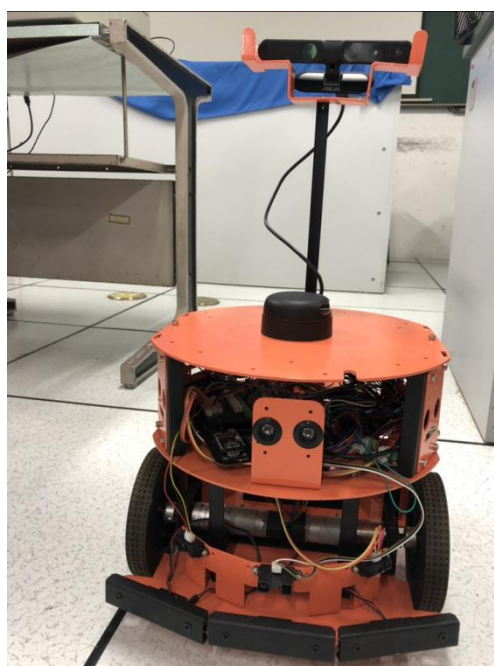


图 5-3 Home Care Robot 机器人平台

在传感器层中，本实验主要用到了红外传感器、跌落传感器、超声波传感器和碰撞传感器。在机器人较低的位置安装了 7 个 sharp 红外传感器，其感应的范围为 10-80cm，可以根据周围环境的变换进行及时的应对。在机器人第二层位置安装 5 个超声波传感器，用于弥补激光雷达传感器无法检测到透明的障碍物的缺陷。在机器人底盘处装有防跌落传感器，规避机器人在台阶附近跌落的风险。机器人正前方的三个方向分别使用 3 个碰撞传感器，其作用是在机器人已经撞上障碍物，系统对此状态下发出快速回应，即停止或后退。底层传感器通过 pin 接口与运动控制层连接，返回采集的数据。

机器人的运动控制是一个复杂的系统工程。首先需要控制电机并且通过采集编码器上的信息了解电机的运动状况；其次需要收集处理传感器发回的信息保证机器人运动的安全可靠；最后还需要通讯模块用来调试进而实现与上位机

之间的信息交换。本实验选用集成板作为机器人的运动控制板，该集成板有 Arduino Mega 2560 和 2 块 Arduino Nano 以及电机驱动板。Arduino 是一个集合了硬件与软件的开源电子平台。开发者通过 Arduino 编程语言与 Arduino 软件可以向 Arduino 平台发送一组指令来告诉平台该实现什么功能。由于其简单易用的用户体验，Arduino 已经被用于数千个不同的项目和应用程序中。本实验中 Arduino Mega 2560 作为主控板，采集传输数据以及其他扩展模块，一块 Arduino Nano 作为电机控制以及数据采集，另一块用来驱动超声波传感器，整个集成板如图 5-4 所示。

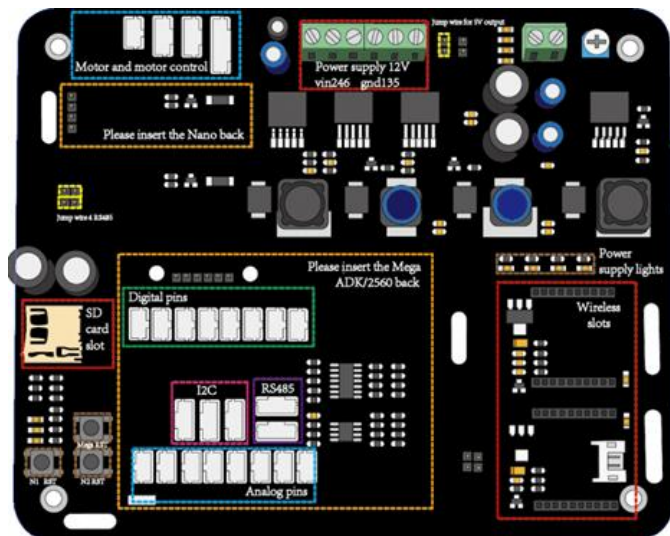


图 5-4 集成板

在算法开发层，选用 NVIDIA 嵌入式开发板，搭载 linux-arm 操作系统，并装有机机器人操作系统^[62]（Robot Operating System, ROS）软件平台。ROS 是编写机器人相关程序的灵活开源框架。它是一个集合，该集合包括许多工具、库等。因为创建真正强大通用机器人软件具有较大的挑战性，ROS 致力于简化在各种机器人平台上实现复杂且功能强大的机器人行为。例如，一个实验室中可能有室内环境创建地图方面的专家，可以为创建地图提供一流的技术支持，而另一个实验室或团体可能已经发现一种计算机视觉方法，能够很好地从一堆物体中准确且快速地标识出指定的目标物体。ROS 就是专门针对上述的团体进行设计的，在彼此的研究基础上进行协作开发，极大地推动了机器人领域研究发展。具体的 SLAM 算法的通过 C++ 工程实现。整体的系统架构如图 5-5 所示。

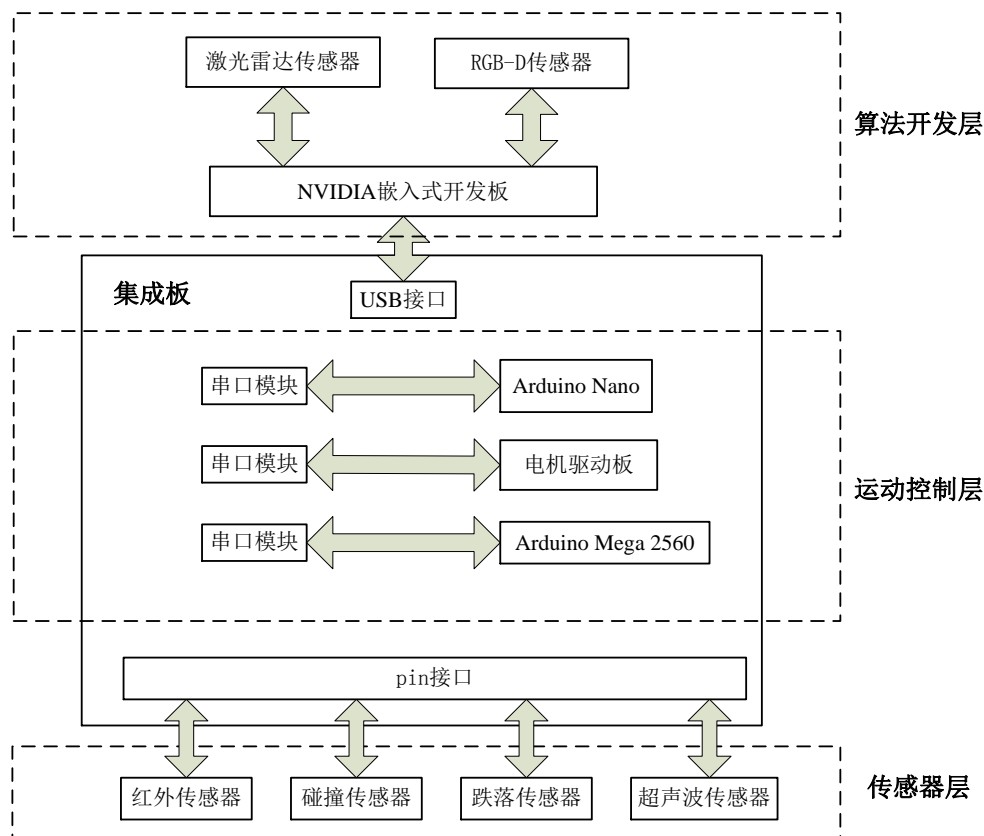


图 5-5 系统架构图

5.2 Xtion Pro Live 传感器与 RPLIDAR 传感器精度测量与对比

为了更好地了解各类传感器性能，设计了实验，并验证和比较 RGB-D 相机（Xtion Pro Live）和激光雷达传感器（RPLIDAR）测量的有效范围与测量精度。RGB-D 相机由 RGB 传感器与深度传感器组成，分别捕获彩色图像（RGB）与每个像素深度信息（深度图像），通过结构化红外光或飞行时间（Time Of Flight, TOF）技术对深度信息进行估计。RGB-D 相机因其较高的性价比，被广泛的应用于各类场景中，如 SLAM、3-D 场景解析、语义映射和远程呈现、立体增强现实、3-D 形状扫描、构建密集型三维地图、人体跟踪、远程康复的补胎检测以及对象和手势识别。但相机被用于各类场景之前，均要完成校准工作。本实验相机校准过程如图 5-6 所示。激光雷达采用的是 RPLIDAR，该传感器为 360 度全方位扫描，10 赫兹自适应扫描频率，每秒 4000 次激光测距，不低于 8 米/16 米的测距范围。



图 5-6 RGB-D 相机校准

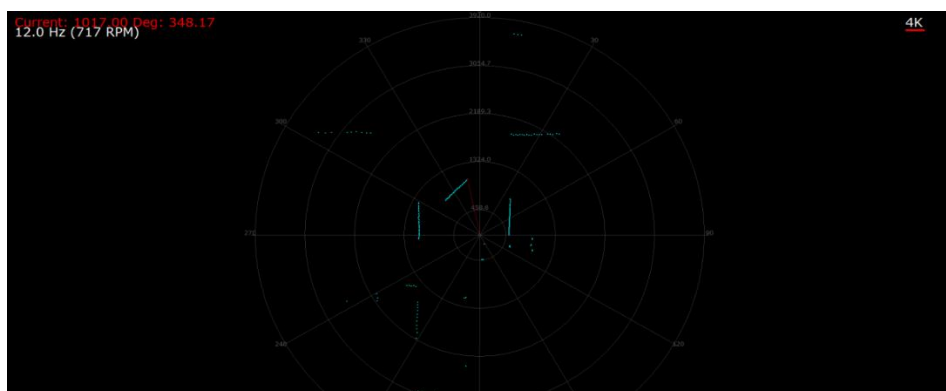
实验分别测量了两种传感器垂直和偏角度运作时的精度和范围。图 5-7 (a)、图 5-7 (b)、图 5-7 (c) 分别表示实验场景 RGB 图像、深度图像、激光雷达扫描图。



(a) 实验场景图



(b) 深度图像



(c) 激光雷达扫描图

图 5-7 实验图

Xtion Pro Live 官方推荐的使用距离为 0.8 米到 4.0 米之间，测量精度随着距离的增加呈下降趋势。相机的深度图是通过结合 OpenCV 和 OpenNI 在 Ubuntu14.04 系统下获取，OpenNI 中表示深度像素的格式为 `XnDepthPixel`，实际上是单一通道的 16 位正整数，因此为了能显示出深度图像，需要进行数据格式

的转换和归一化处理。表 5-1 是激光雷达与 RGB-D 相机在垂直测量时的实验数据。

表 5-1 雷达与 RGB-D 相机垂直距离测量数据 (mm)

| | | | | | | |
|----------|--------|--------|--------|--------|--------|--------|
| 实际距离 | 500 | 1000 | 1500 | 2000 | 2500 | 3000 |
| 雷达 | 509 | 1008 | 1515 | 2005 | 2532 | 3025 |
| RGB-D 相机 | 533.5 | 1007.7 | 1511.6 | 1985.9 | 2489.7 | 2993.6 |
| 实际距离 | 3500 | 4000 | 4500 | 5000 | 5500 | 6000 |
| 雷达 | 3552 | 4053 | 4557 | 5024 | 5462 | 6021 |
| RGB-D 相机 | 3467.9 | 4001.4 | 4475.6 | 4979.5 | 5483.4 | 6076.2 |

首先，为了比较相机和雷达的测量距离精确度，用 matlab 拟合出了相机深度信息和实际距离之间的映射关系，实验表明其关系符合线性关系，相关系数为 0.9997。

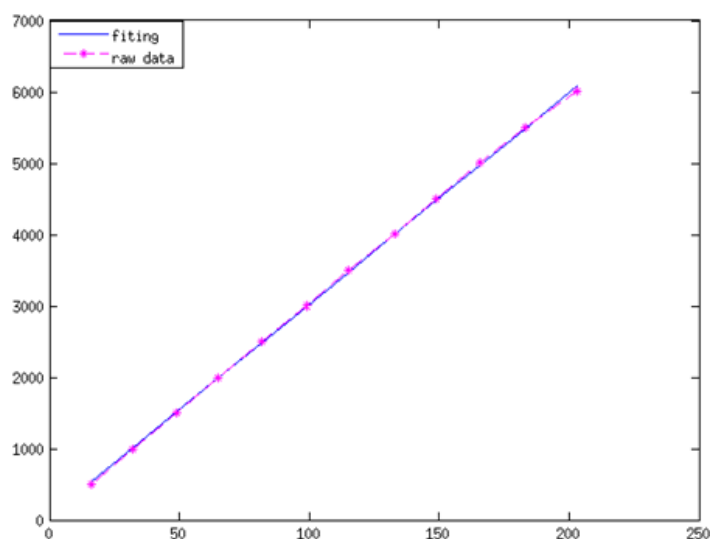


图 5-8 Xtion Pro Live 深度像素值与实际距离的映射关系

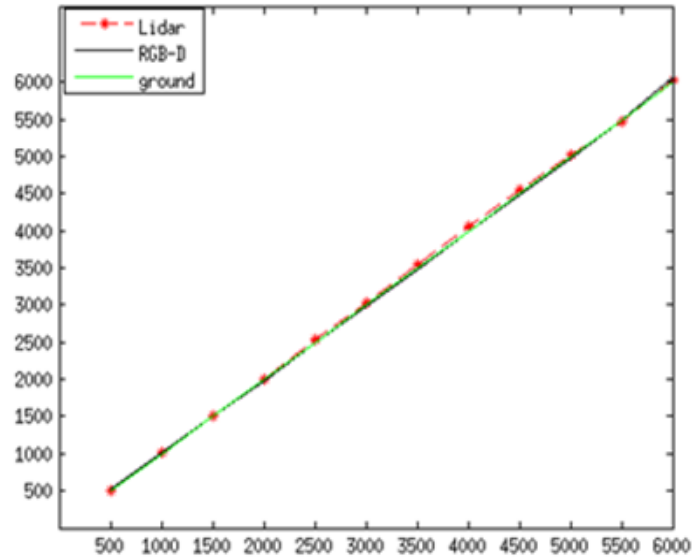


图 5-9 RPLIDAR 与 Xtion Pro Live 对物体的垂直测距对比

从图 5-8、图 5-9 可看出在测量垂直距离时，雷达与相机的精确度相近。相机的均方误差 $MSE=3.325031e+01$ ，雷达的均方误差 $MSE=2.856605e+01$ 。表 5-2 是两种传感器相对物体存在一定角度下的测距数据。

表 5-2 雷达与 RGB-D 相机斜角距离测量数据 (mm)

| | | | | | | |
|----|-----------|-----------|-----------|-----------|-----------|-----------|
| 实距 | 854~102.5 | 1268~1350 | 1880~2065 | 2460~2650 | 3075~3153 | 3460~3550 |
| 雷达 | 857~1017 | 1273~1349 | 1907~2059 | 2493~2682 | 3096~3149 | 3496~3647 |
| 相机 | 889~1037 | 1185~1363 | 1719~1808 | 2400~2549 | 2875~2993 | 3438~3497 |

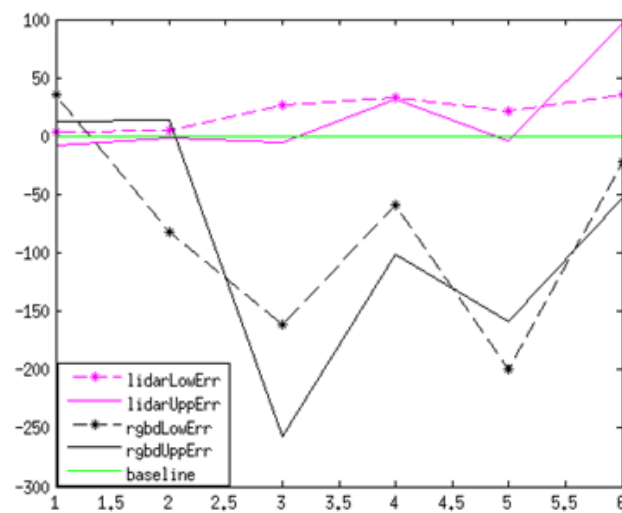


图 5-10 Xtion Pro Live 与 RPLIDAR 在偏角度下测距绝对误差

从图 5-10 以及实验数据可以看出，当测量有一定角度的物体距离时，雷达的精确度要比相机的高，但是雷达的测量性能和物体材质有一定的关系。在上述的实验中可以得出一个结论，当物体正对传感器时，相机的数据更精确，当有一定角度物体时，雷达的数据更精确，在传感器数据融合的实验中，可将相机与雷达安装在机器人的正前方。

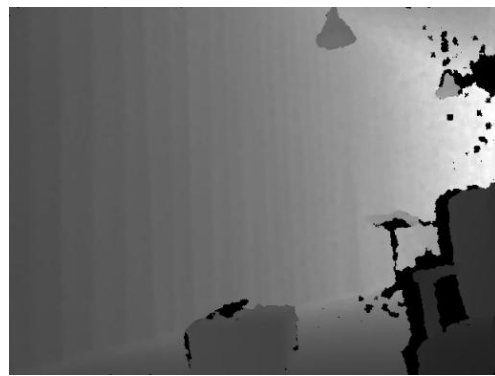
5.3 实验结果与分析

5.3.1 深度图像修复实验结果与分析

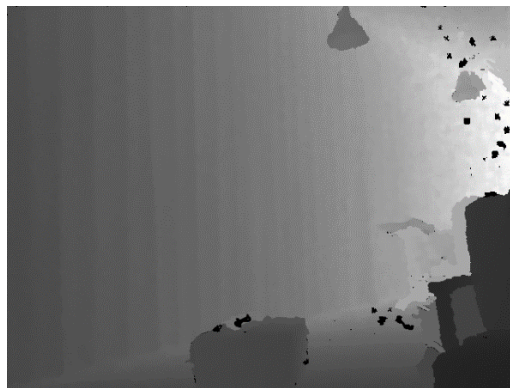
为了改善深度图像的质量，文献^[63]提出一种自适应双边滤波的修复方法，该方法对于噪声去除与锐化增强都具有理想的效果，而且滤波器的参数是经过参数训练方法训练出来的。尽管这种方法修复效果理想，但是其需要一个较复杂的训练过程。而文献^[52]通过“黑洞”周边像素的形态学操作，得到一个像素修复的有效区域，并根据这个有效区域内像素的分布来对缺失像素信息进行填充。该方法实现简单，但是在确定有效区域时，形态学的膨胀操作容易将与“黑洞”无关的像素归并进到有效填充区域。图 5-11 的 (a) 是实验场景图的彩色图像，图 5-11 的 (b) 是实验场景的深度图像，图 5-11 的 (c) 是利用“黑洞”周边有效区域修复方法的结果图，图 5-11 的 (d) 为本文提出的深度图像修复方法的结果图，图 5-11 的 (e) 为交叉双边滤波结果图，从实验结果图可证明，本文提出的“黑洞”填充方法较好地修复了深度图像中的缺失深度信息。



(a) 场景的彩色图像



(b) 场景的深度图像



(c) 利用“黑洞”周边的有效区域修复结果图



(d) 本文提出的“黑洞”填充方法结果图



(e) 交叉双边滤波结果图

图 5-11 深度图像修复实验结果图

5.3.2 深度相机和激光雷达数据融合实验结果与分析



(a) Gmapping 算法实现结果



(b) 基于数据融合的 Gmapping 算法实现结果

图 5-12 数据融合实验对比图

在移动机器人平台上分别运行 Gmapping 算法和基于数据融合的 Gmapping 算法，所建立的地图如图 5-12 的 (a) 和 (b) 所示。融合算法的最大优点在于能够检测出低于激光雷达所在平面的障碍物，图中红圈标出的部分是一个平放的纸箱，传统的 Gmapping 算法受雷达位置的限制并未检测到，而融合算法利用了 RGBD 摄像头的信息能够将其识别出来。此外，传感器信息的冗余有助于算法的收敛。表 5-3 中列出了两种算法的运行时间及数据点个数，基于数据融合的 Gmapping 算法在不增加时间复杂度的情况下，提升了 27% 的精度。

表 5-3 算法运行参数对比表

| 算法 | 运行时间(s) | 数据点 (个) |
|------------------|---------|---------|
| Gmapping | 117 | 502 |
| 基于数据融合的 Gmapping | 125 | 636 |

5.3.3 室内 Home Care Robot 机器人路径规划实现结果与分析

该系统将 Home Care Robot 机器人放置在室内，机器人有电机驱动，并有障碍识别功能，可以在室内自行行走，RGB-D 相机获取环境信息，通过 ORB-SLAM2 算法框架和 RGB-D SLAM 算法框架实现同时定位与建图，实验结果如图 5-13 所示，(a) 为实验场景图，(b) 分别为 ORB-SLAM2 生成的三维点云图，(c) 为 RGB-D SLAM 生成的三维点云图，(d) 为激光雷达 Gmapping 生成的二维平面图。由实验结果可看出，ORB-SLAM2 算法框架的闭环检测效果优于 RGB-D SLAM 算法。RGB-D SLAM 算法通过计算关键帧中的特征点描述子来进行回环检测，而 ORB-SLAM2 是通过基于 DBoW2 的场景识别来进行闭环检测，而且具有闭环修正模块，较好地修正了地图。而且在实验过程中发现，ORB-SLAM2 算法的实时性也较 RGB-D SLAM 好。ORB-SLAM2 算法采用的是多线程方式，而 RGB-D SLAM 是单线程，在同样的硬件设备条件下，ORB-SLAM2 运行速度较快，可以满足实际应用中实时性的要求。ORB-SLAM2 算法的基于 DBoW2 的场景识别模型具有全局重定位能力，在跟踪丢失之后，可以进行重新定位，在算法鲁棒性方面也较优。



(a) 场景图



(b) ORB-SLAM2 点云图



(c) RGB-D SLAM 点云图



(d) 激光雷达平面图

图 5-13 ORB-SLAM2, RGB-D SLAM 与激光雷达建图结果

从以上的实验结果图分析，该系统选用 ORB-SLAM2 生成的三维点云图即图 5-13 的 (b) 图，经过 Octomap 处理，得到如图 5-14 所示的三维网格环境地图，即 Octomap 图。Octomap 使用八叉树的数据结构存储地图，用八叉树编码的对象中包含的信息与在空间枚举中包含的信息相同，例如占用网格。从存储的角度来看，数据被存储在分层树结构中，在此结构中节点是一个个不相交的立方体，每层节点的数量呈指数下降。树中的每个节点均表示一个区域，一个区域具有一个或多个定义该区域的节点值，如果一个节点的值完全描述了该区域，则它是终端节点或叶子节点。八叉树数据结构有以下几个优点，第一，任意对象可以表示为最小立方体的精度。第二，所有对象只需要一组操作和分析算法。第三，所有对象都在空间上保持预先排序。第四，算法只需要在生成输出树的同时依次遍历输入树即可实现两个或更多八叉树之间执行布尔运算。此外，每个级别上的节点代表整个对象在该级别上的解析，因此算法可以根据任务在适合的级别上操作。本方法首先遍历八叉树节点，删除值小于 0.5 的节点（大于 0.5 表示该区域被占概率较大），遍历雷达平面高度的那层数据，然后将其投影得到二维平面地图，为了便于使用 A* 算法，将障碍物方格化，最后将投影得到的平面图与激光雷达平面图进行或运算，获得融合了环境三维点云信息和激光雷达平面扫描信息的二维栅格地图。



图 5-14 Octomap 图

在融合三维环境点云信息的栅格图上，通过经典路径搜索算法 A*，对 HCR 机器人实现路径规划。实验结果如图 5-15 所示。由路径规划结果看出，在融合了点云信息的环境地图上，中间四把椅子（障碍物）的信息更加的明显，有利于机器人更好地进行路径规划。

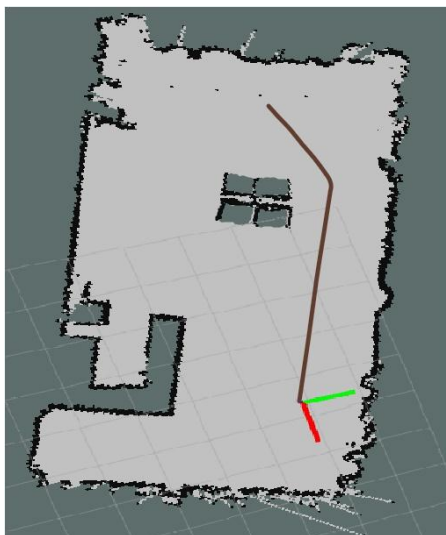


图 5-15 路径规划结果图

5.4 本章小结

本章对本文设计并搭建的移动机器人实验平台进行介绍，对用到的主要传感器 Xtion Pro Live 和 RPLIDAR 激光雷达设计实验，对两种传感器的性能，主要对测量精度和测量范围进行检测和对比。对深度图像修复结果与其他算法的修复结果进行对比分析，分析了传感器数据融合实验结果，对室内移动机器人路径规划进行了实现，并分析对比了 RGB-D SLAM 与 ORB-SLAM2 两种经典视觉算法框架。

6 总结与展望

6.1 主要创新点

在阅读与研究前辈们在深度图像修复方面提出的方法基础上，本文提出一种基于 SLIC 超像素分割的图像修复方法，首先对深度图像进行二值化以检测出需要修复的区域，将彩色图像进行 SLIC 超像素分割处理，随后根据“黑洞”位置与分割块之间位置关系不同采取不同的方法对深度图像进行修复，最后用交叉双边滤波对修复后的深度图像进行平滑滤波。实验结果表明提出的方法较好地修复了由物体边缘反射光变化剧烈或者物体被遮挡等原因引起的深度图像“黑洞”问题。

因为不同的传感器有各自的优缺点，本文分析了较常用的激光雷达传感器与深度相机 Xtion Pro Live 的优缺点，提出一种将激光雷达传感器与深度相机获取的环境信息进行融合的方法，并实现两种传感器数据的融合，得到了更丰富的环境信息。

本文设计并搭建移动机器人实验平台，并在该平台上完成 HCR 机器人路径规划系统的设计与实现。

6.2 工作总结与展望

本文所做的工作主要如下：

1. 了解国内外 SLAM 的发展现状及其研究背景与意义，研究与 SLAM 相关的基本算法实现框架。

2. 针对 RGB-D 相机获取的深度图像由于遮挡以及反射光被吸收等原因引起的深度信息缺失问题，提出了一种基于 SLIC 超像素分割的图像修复方法，有效地填充了深度图像中的缺失值。

3. 设计实验并验证了 RGB-D 相机（Xtion Pro Live）与激光雷达传感器（RPLIDAR）的测量精度与测量范围，分析研究两种传感器的优缺点，提出一种融合两种传感器数据的方法，用 RGB-D 相机弥补了激光雷达传感器测量范围的不足。

4. 设计并搭建移动机器人实验平台，并在搭建的实验平台上设计并实现 HCR 移动机器人路径规划系统，并分析对比了两种视觉 SLAM 算法框架，即 RGB-D SLAM 与 ORB-SLAM2。

SLAM 实现算法框架在过去的几年里得到了较快的发展，但是仍然存在几个有待解决的问题。第一个问题是关于可用结果的一般性问题，许多在姿态图优化前提下所得出的全局解是否可以推广到任意的因子图。第二个问题是算法的鲁棒性问题，对于异常值存在的情况下，算法是否能依然保持稳定。第三个

问题，在机器人位姿跟踪失败情形下全局重定位精度是否可以再提高。未来 SLAM 实现算法的发展可能会朝着实用性更好，鲁棒性更强的方向发展。传感器作为 SLAM 算法框架的唯一输出，传感器技术的发展也将对 SLAM 起着很大的影响。各类传感器均有各自的优缺点，所以传感器融合技术的发展也会推动 SLAM 实现算法的发展。虽然基于 SLAM 的应用产品还未全面地进入消费市场，目前 SLAM 已经被应用于多类服务型机器人、游戏市场、虚拟现实与增强现实等方面，但是像智能手机一样普及还未实现，而实现这一目标需要克服一些难题与障碍。

参考文献

- [1] Schliferstein A. Principles of Artificial Intelligence[J]. Pattern Analysis & Machine Intelligence IEEE Transactions on, 1981, PAMI-3(1):112.
- [2] Murphy K P. Machine Learning: A Probabilistic Perspective[M]. MIT Press, 2012.
- [3] 尹宝才, 王文通, 王立春. 深度学习研究综述[J]. 北京工业大学学报, 2015(1):48-59.
- [4] 曹凯. 移动机器人技术研究现状与未来[J]. 信息系统工程, 2013(5):140-140.
- [5] Cadena C, Carlone L, Carrillo H, et al. Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age [J]. IEEE Transactions on Robotics, 2016, 32(6): 1309-1332.
- [6] 秦永元, 张洪钺, 汪叔华. 卡尔曼滤波与组合导航原理[M]. 西北工业大学出版社, 2015.
- [7] 梁明杰, 闵华清, 罗荣华. 基于图优化的同时定位与地图创建综述[J]. 机器人, 2013, 35(4):500-512.
- [8] 向学勤, 潘志庚, 童晶. 深度相机在计算机视觉与图形学上的应用研究[J]. 计算机科学与探索, 2011, 05(6):481-492.
- [9] 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.
- [10] Kubelka V, Oswald L, Pomerleau F, et al. Robust Data Fusion of Multimodal Sensory Information for Mobile Robots[J]. Journal of Field Robotics, 2015, 32(4):447-473.
- [11] Quigley M, Conley K, Gerkey B P, et al. ROS: an open-source Robot Operating System[C]// ICRA Workshop on Open Source Software. 2009.
- [12] Bailey T, Durrantwhyte H. Simultaneous localisation and mapping (slam) part 2: State of the art[J]. IEEE Robotics & Amp Amp Automation Magazine, 2006, 13(3):108-117.
- [13] Durrantwhyte H, Bailey T. Simultaneous localization and mapping: part I[J]. IEEE Robotics & Amp Amp Automation Magazine, 2006, 13(2):99 - 110.
- [14] 时也, 吴怀宇, 徐文霞,等. 基于扩展卡尔曼滤波器的移动机器人 SLAM 研究 [J]. 电子设计工程, 2012, 20(1):104-106.
- [15] 罗元, 傅有力, 程铁凤. 基于改进 Rao-Blackwellized 粒子滤波器的同时定位与地图构建[J]. 控制理论与应用, 2015, 32(2):267-272.
- [16] Dissanayake G, Huang S, Wang Z, et al. A review of recent developments in Simultaneous Localization and Mapping[C]// IEEE International Conference on Industrial and Information Systems. IEEE, 2011:477-482.

- [17] 陈晨, 程荫杭. 基于迭代无迹卡尔曼滤波的 SLAM 算法仿真研究[J]. 系统仿真学报, 2012, 24(8):1643-1650.
- [18] 王法胜, 鲁明羽, 赵清杰, 等. 粒子滤波算法[J]. 计算机学报, 2014, 37(8):1679-1694.
- [19] Hfinel D, Burgard W, Fox D, et al. An Efficient FastSLAM Algorithm for Generating Maps[J]. 2003, 1:206 - 211.
- [20] Endres F, Hess J, Sturm J, et al. 3-D Mapping With an RGB-D Camera[J]. IEEE Transactions on Robotics, 2014, 30(1):177-187.
- [21] Mur-Artal R, Montiel J M M, Tardós J D. ORB-SLAM: A Versatile and Accurate Monocular SLAM System[J]. IEEE Transactions on Robotics, 2015, 31(5):1147-1163.
- [22] Mur-Artal R, Tardos J D. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras[J]. IEEE Transactions on Robotics, 2017, 33(5):1255-1262.
- [23] Klein G, Murray D. Parallel Tracking and Mapping for Small AR Workspaces[C]// IEEE and ACM International Symposium on Mixed and Augmented Reality. IEEE Computer Society, 2007:1-10.
- [24] Forster C, Pizzoli M, Scaramuzza D. SVO: Fast semi-direct monocular visual odometry[C]// IEEE International Conference on Robotics and Automation. IEEE, 2014:15-22.
- [25] Whelan T, Salas-Moreno R F, Glocker B, et al. ElasticFusion[J]. International Journal of Robotics Research, 2016, 35(14):1697-1716.
- [26] 赵沁平, 周彬, 李甲, 等. 虚拟现实技术研究进展[J]. 科技导报, 2016, 34(14):71-75.
- [27] 丁群, 刘键薇, 洪娜佳, 等. 增强现实技术[J]. 黑龙江科技信息, 2015(30).
- [28] 蔡则苏, 洪炳镨, 魏振华. 基于激光测距传感器的家庭机器人的位置跟踪[C]// 2004 全国光学与光电子学学术研讨会、2005 全国光学与光电子学学术研讨会、广西光学学会成立 20 周年年会论文集. 2005.
- [29] 彭胜军, 马宏绪. 移动机器人导航空间表示及 SLAM 问题研究[C]// 全国自动化新技术学术交流会会议论文集(一). 2005:1-4.
- [30] 闵华松, 杨杰. 融合 IMU 的 RGBD-SLAM 算法改进研究[J]. 计算机工程与设计, 2015, 36(1):120-126.
- [31] 王忠立, 赵杰, 蔡鹤皋. 大规模环境下基于图优化 SLAM 的后端优化方法[J]. 哈尔滨工业大学学报, 2015, 47(7):20-25.
- [32] 王国胜, 郭峰, 刘峰. 无人飞行器视觉 SLAM 特征检测和匹配算法研究[J]. 计算机测量与控制, 2015, 23(7):2453-2455.
- [33] 赵辉, 杜航原, 张虎. 基于 SLAM 算法的 AUV 自主导航仿真研究[J]. 山西电子技术, 2015(4):86-89.

- [34] 敬喜. 卡尔曼滤波器及其应用基础[M]. 国防工业出版社, 1973.
- [35] 李国新, 何英杰, 许海涛. 扩展卡尔曼滤波器[J]. 电子设计工程.
- [36] Sahu M, Kamble C. Comparison of FastSLAM1.0 and FastSLAM2.0 for Relatively High Motion Noise Environment[J]. Hcrl Open International Journal of Technology Innovations & Research, 2015, 14.
- [37] Armesto L, Ippoliti G, Longhi S, et al. FastSLAM 2.0: Least-Squares Approach[C]// Ieee/rsj International Conference on Intelligent Robots and Systems. IEEE, 2006:5013-5018.
- [38] Davison A J. Real-Time Simultaneous Localisation and Mapping with a Single Camera[C]// IEEE International Conference on Computer Vision, 2003. Proceedings. IEEE, 2008:1403.
- [39] Kameda Y. Parallel Tracking and Mapping for Small AR Workspaces (PTAM) Augmented Reality[J]. Journal of the Institute of Image Information & Television Engineers, 2012, 66(1):45-51.
- [40] Engel J, Schöps T, Cremers D. LSD-SLAM: Large-Scale Direct Monocular SLAM[M]// Computer Vision – ECCV 2014. Springer International Publishing, 2014:834-849.
- [41] Jin H, Favaro P, Soatto S. Real-time 3D motion and structure of point features: a front-end system for vision-based control and interaction[C]// Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on. IEEE, 2000:778-779 vol.2.
- [42] 吴友光. 迭代最近点算法研究[J]. 电脑知识与技术:学术交流, 2012(11):7568-7570.
- [43] 马晓晨, 孔小利. 基于深度八叉树的三维数据场 LOD 可视化[J]. 计算机应用, 2010, 30(1):47-49.
- [44] Triggs B, Mclauchlan P F, Hartley R I, et al. Bundle Adjustment — A Modern Synthesis[M]// Vision Algorithms: Theory and Practice. Springer Berlin Heidelberg, 1999:298-372.
- [45] Mouragnon E, Lhuillier M, Dhome M, et al. Real Time Localization and 3D Reconstruction[C]// IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2006:363-370.
- [46] Galvez-López D, Tardos J D. Bags of Binary Words for Fast Place Recognition in Image Sequences[J]. IEEE Transactions on Robotics, 2012, 28(5):1188-1197.
- [47] Strasdat H, Davison A J, Montiel J M M, et al. Double window optimization for constant time visual SLAM[C]// International Conference on Computer Vision. IEEE Computer Society, 2011:2352-2359.
- [48] Balasuriya B L E A, Chathuranga B A H, Jayasundara B H, et al. Outdoor robot navigation using Gmapping based SLAM algorithm[C]// Moratuwa Engineering Research Conference. IEEE, 2016:403-408.

- [49] Matyunin S, Vatolin D, Berdnikov Y, et al. Temporal filtering for depth maps generated by Kinect depth camera[C]// 3dTV Conference: the True Vision - Capture, Transmission and Display of 3d Video. IEEE, 2011:1-4.
- [50] Berdnikov Y, Vatolin D. Real-time Depth Map Occlusion Filling and Scene Background Restoration for Projected-Pattern-based Depth Camera[C]// 2011.
- [51] Salgado L. Efficient spatio-temporal hole filling strategy for Kinect depth maps[J]. Proceedings of SPIE - The International Society for Optical Engineering, 2012, 8290(9):13.
- [52] Yang N E, Kim Y G, Park R H. Depth hole filling using the depth distribution of neighboring regions of depth holes in the Kinect sensor[C]// IEEE International Conference on Signal Processing, Communication and Computing. IEEE, 2012:658-661.
- [53] Vijayanagar K R, Loghman M, Kim J. Refinement of depth maps generated by low-cost depth sensors[C]// Soc Design Conference. IEEE, 2013:355-358.
- [54] Jachalsky J, Schlosser M, Gandolph D. Confidence evaluation for robust, fast-converging disparity map refinement[C]// IEEE International Conference on Multimedia and Expo. IEEE, 2010:1399-1404.
- [55] Murphy K P. Bayesian Map Learning in Dynamic Environments[C]// 2008:1015--1021.
- [56] Hart P E, Nilsson N J, Raphael B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths[J]. IEEE Transactions on Systems Science & Cybernetics, 2007, 4(2):100-107.
- [57] 陈刚, 付少锋, 周利华. A*算法在游戏地图寻径中的几种改进策略研究[J]. 科学技术与工程, 2007, 7(15):3731-3736.
- [58] 钱红昇, 葛文锋, 钟鸣, 等. 基于分层的改进 A*算法在路径规划中的应用, 2013 年 9 月 2 日[J]. 计算机工程与应用, 2013.
- [59] Lumelsky V J, Stepanov A A. Dynamic path planning for a mobile automaton with limited information on the environment[J]. Automatic Control IEEE Transactions on, 1986, 31(11):1058-1063.
- [60] Pirzadeh A, Snyder W. A unified solution to coverage and search in explored and unexplored terrains using indirect control[C]// IEEE International Conference on Robotics and Automation, 1990. Proceedings. IEEE, 1990:2113-2119 vol.3.
- [61] Korf R E. Real-Time Heuristic Search: First Results.[C]// National Conference on Artificial Intelligence. AAAI Press, 1987:133-138.
- [62] 张建伟, 张立伟, 胡颖, 等. 开源机器人操作系统——ROS[M]. 科学出版社, 2012.
- [63] Kopf J, Cohen M F, Lischinski D, et al. Joint bilateral upsampling[C]// ACM SIGGRAPH. ACM, 2007:96.

攻读硕士学位期间主要学术成果

第一作者. 基于数据融合的 SLAM 系统研究与路径规划实现 [J]. 《电脑知识与技术》. 2017,13(33): 24-26

致谢

自本课题研究开始，至本论文初稿结束，已经一年有余。感谢导师禹素萍副教授在研究生阶段的谆谆教诲，不论在科研还是在生活阶段，都给了我莫大的关心帮助。在论文的每个阶段，禹老师都认真把关，悉心指导，对于我的每一个问题，都耐心解答。禹老师诲人不倦，一丝不苟的学术态度也深深影响着我。从禹老师身上，我不仅学到了如何以严谨的态度做学术，还学到了如何平衡学术研究与日常生活之间的关系。在为人处事方面，禹老师也是我学习的榜样，明白什么才是真正重要和值得珍惜的。

感谢实验室的小伙伴在生活和学业上的诸多帮助，正因为有你们，使我在实验室的每一天都特别充实，我们一起探讨学术，一起用心生活，互相鼓励，共同进步。感谢丁绅一同学在课题方方面面的耐心讲解，感谢他在机器人硬件方面给我提供的极大帮助。感谢顾晓雯学姐在论文撰写方面给予的指导，生活上的关心。

感谢家人一如既往的支持，支持我做的每一个决定，对我生活的照顾，在外求学的十年时光，你们是我最大的动力。

同样感谢森浦资讯给我提供了宝贵的实习机会，让我学习到了如何以合作的方式完成一个难题，感谢公司里优秀小伙伴的帮助与指导。

最后，感谢在百忙之中参与论文审阅、评议和参加答辩的各位老师！