



汕头大学
SHANTOU UNIVERSITY

硕士学位论文

题目：基于二维地图蒙特卡罗定位的三维室内
地图构建

英文题目：3D Indoor Map Building with Monte Carlo
Localization in 2D Map

姓名赵雷学号11409022

所在学院工学院导师姓名范衡

专业信息与通信工程

入学日期2014年9月答辩日期2017年5月

学位论文原创性声明

本论文是我个人在导师指导下进行的工作研究及取得的研究成果。论文中除了特别加以标注和致谢的地方外，不包含其他人或其他机构已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在论文中以明确方式标明。本人完全意识到本声明的法律责任由本人承担。

作者签名：_____ 日期：_____年____月____日

学位论文使用授权声明

本人授权汕头大学保存本学位论文的电子和纸质文档，允许论文被查阅和借阅；学校可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、缩印或其他复制手段保存和汇编论文；学校可以向国家有关部门或机构送交论文并授权其保存、借阅或上网公布本学位论文的全部或部分内容。对于保密的论文，按照保密的有关规定和程序处理。

作者签名：_____ 导师签名：_____

日期：_____年____月____日 日期：_____年____月____日

摘 要

移动机器人同步定位与地图构建（Simultaneous Localization and Mapping, SLAM）是实现机器人自主导航的关键技术之一。使用 SLAM 技术构建的二维地图已经在移动机器人自动导航中得到广泛的使用。机器人活动能力的不断增强和室内环境的日益复杂，使得机器人必须考虑环境的三维信息，三维地图无疑更接近于现实生活场景，机器人可以根据三维地图信息决定其移动策略，更好的实现自动导航。

随着图像处理技术和计算机视觉技术的发展，基于视觉定位的 SLAM 近几年开始广泛的应用于移动机器人三维室内地图构建中，然而视觉定位的实时性较差，同时在一些特征较少且高度对称的环境中，例如走廊，视觉定位会产生特征匹配失误，特征匹配失误会造成累积误差，累积误差导致机器人错误的定位，错误的定位会影响所构建三维地图的质量。二维地图下基于激光雷达的蒙特卡罗定位（Monte Carlo localization, MCL）有更高的定位精度，因此本文提出利用二维地图下基于激光雷达的蒙特卡罗定位来解决三维地图构建中的定位问题，同时利用 Kinect 传感器采集周围的环境信息来构建局部三维地图，通过二维地图下的定位我们能得到不同时刻机器人的位姿，之后求解出这些位姿间的转换矩阵，使用这些转换矩阵对局部三维地图进行拼接，最后得到全局三维地图。为了验证所提方法的性能，我们在机器人 TurtleBot2 和自制机器人上进行了实验，构建了三维室内走廊地图。同时在相同走廊场景下，使用 2014 年提出的 RGB-D SLAM 构建了三维地图，并与所提方法进行了实验对比，实验结果表明，本文所提方法在室内走廊环境中能构建出更好的三维地图。

关键词：激光雷达；Kinect；视觉定位；蒙特卡罗定位；三维室内地图

Abstract

Mobile robot simultaneous localization and mapping (SLAM) is one of the key technologies to realize autonomous navigation of the robot. Using SLAM technology to build the 2D map has been widely used in the autonomous navigation of mobile robots. As the mobility of a robot and the complexity of an indoor environment have been increasing, the robot must consider the 3D information of the environment. 3D map is undoubtedly closer to the real life scene. The robot can determine its mobile strategy according to the 3D map information, and can realize better autonomous navigation.

With the development of image processing and computer vision technology, the SLAM based on visual localization is widely applied in the robot indoor 3D map building. However, the real-time performance of visual localization is poor. Meanwhile, in feature-less and highly symmetrical environment, for example the corridor, visual localization will produce feature matching error. The feature matching error will bring cumulative error, which result in wrong localization. The wrong localization will affect the quality of the 3D map. The Monte Carlo localization(MCL) based on lidar in 2D map can achieve a higher localization accuracy. Therefore, we propose using MCL based on lidar in 2D map to solve the localization problem while using a kinect to collect 3D environment information. Localization in 2D map can afford us the robot poses at different times. Then the transformation matrix between these positions can be solved. By using these transformation matrix, the local 3D map can be merged one by one. At last, we can get the global 3D map. To study the performance of the proposed method, we make some experiments on robot TurtleBot2 and self-made robot and build 3D map of a corridor. And then comparing the experimental result with the 3D map that built by RGB-D SLAM provided in 2014. The experimental results demonstrate that our method has better effect in the indoor corridor environment.

Keywords: lidar; kinect; visual localization; MCL; 3D indoor map building

目 录

摘 要	I
Abstract.....	II
第一章 绪 论	1
1.1 课题研究背景与意义	1
1.2 国内外研究现状	2
1.3 主要研究内容	3
1.4 章节安排	4
第二章 Kinect 传感器和 RGB-D SLAM 算法	5
2.1 Kinect 体感传感器	5
2.1.1 Kinect 工作原理	5
2.1.2 摄像机模型	6
2.1.3 Kinect 深度数据数学模型	8
2.2 RGB-D SLAM 算法	10
2.2.1 特征提取与匹配	12
2.2.2 位姿估计	13
2.2.3 点云匹配	14
2.2.4 g2o 图优化	14
2.2.5 OctoMap 三维立体栅格地图创建	15
2.3 本章小结	16
第三章 SLAM 算法理论基础和蒙特卡罗定位	17
3.1 贝叶斯滤波理论	17
3.1.1 SLAM 中信度的定义	17
3.1.2 贝叶斯滤波公式推导	18
3.2 蒙特卡罗算法	20
3.2.1 蒙特卡罗定位算法思想	20
3.2.2 蒙特卡罗定位算法流程	21
3.3 本章小结	22
第四章 基于二维地图蒙特卡罗定位的三维室内地图构建方法原理	23
4.1 二维地图下的蒙特卡罗定位	23
4.1.1 机器人运动模型	24
4.1.2 测量模型	26
4.1.3 自适应蒙特卡罗定位	26
4.2 基于二维地图蒙特卡罗定位的三维地图构建方法	29
4.3 本章小结	31
第五章 基于二维地图蒙特卡罗定位的三维室内地图构建的设计与实现	32
5.1 实验平台硬件	32
5.1.1 TurtleBot 机器人	32
5.1.2 RPLidar 激光雷达测距传感器	34
5.1.3 Kobuki 控制底座运动学模型	35
5.1.4 硬件系统框图	36
5.2 实验软件平台	37
5.2.1 ROS 操作系统	37

5.2.2 实验软件系统流程图	39
5.3 实验结果与分析	40
5.3.1 不同运动模式下定位效果对比	41
5.3.2 使用机器人 TurtbItBot2 基于二维地图蒙特卡罗定位的三维地图构建实验	43
5.3.3 实验对比一基于 RGB-D SLAM 算法的三维地图构建实验	45
5.3.4 实验对比二使用自制机器人基于二维地图蒙特卡罗定位的三维地图构建实验	45
5.4 本章小结	48
第六章 总结与展望	49
6.1 总结	49
6.2 展望	49
参考文献	51
致 谢	55
攻读硕士学位期间主要的工作成果	57

第一章 绪 论

1.1 课题研究背景与意义

近年来,全球计算机科学、图像处理技术、人工智能和微电子技术的快速成长与发展,多学科交叉的移动机器人技术开始逐步应用并出现在人们的日常生活之中,例如家庭扫地机器人、餐厅送餐机器人、仓库运输智能移动机器人等等^{[1][2]}。

“智能移动机器人是一类能够通过传感器、感知环境和自身状态,实现在有障碍物的环境中向指定目标自动移动,从而完成一定功能的机器人系统”,其中同步定位与地图构建(Simultaneous Localization and Mapping, SLAM)是移动机器人实现自动导航的关键技术^[3]。

研究机器人自动导航问题的实质其实就是解决移动机器人的三个问题:1、我在哪里?2、我要去哪里?3、我如何过去?问题一和问题二对应于导航研究中的定位问题,问题三对应于自动导航研究中的路径规划问题^[4]。为了解决以上三个问题实现自动导航功能,机器人就要构建实时环境地图,并且用该地图进行自身定位与目标点定位,这就产生了另外一个问题:怎样构建环境地图?因此,“当处于未知环境中的机器人执行导航任务时,就面临着两个相互依赖、相互关联的问题:为了实现机器人的精确定位,机器人需要掌握精确的环境地图信息;为了构建准确的环境地图,机器人需要知道它在地图中各个时刻的精确位姿信息,SLAM问题的研究由此诞生”^[5]。

随着图像处理和计算机视觉技术的发展,基于视觉定位的SLAM近几年开始广泛的应用于移动机器人三维室内地图构建中,然而视觉定位的实时性较差,同时在一些特征较少的环境中,例如走廊,其结构高度对称且环境特征较少,此时视觉定位会产生特征匹配失误,特征匹配失误会造成累积误差,累积误差导致机器人错误的定位,错误的定位会影响所构建三维地图的质量,三维地图质量的好坏将会对机器人能否成功避开障碍物,准确抵达目标位置实现自动导航这一过程产生严重影响。由于机器人在二维地图下基于激光雷达的蒙特卡罗定位有更高的定位精度,因此,本文提出利用二维地图下基于激光雷达的蒙特卡罗定位来解决三维地图构建中的定位问题,同时利用能够获取颜色图像和深度图像的Kinect

传感器采集周围环境信息来构建三维地图。

传统的二维栅格地图只保存了环境中各种障碍物的二维平面位置、形状，大小等几何信息，而由 RGB-D 信息所创建的三维地图则包含了场景点云，点云中存储的各个点具有位置和颜色信息，所以能够用于描述三维空间中的物体特征。在三维点云地图中，机器人可以在坡面、楼梯等非平面的空间环境中进行定位，同时机器人还可以从三维地图信息中提取所要操作物体的位置和形状信息，从而使得机器人在三维空间中具备移动和作业能力，所以移动机器人三维地图构建的研究是一个具有现实应用意义的研究方向。

1.2 国内外研究现状

基于视觉定位的三维地图构建技术是近几年来移动机器人 SLAM 领域的热点研究问题。自从 Microsoft 在 2010 年下半年发布其首款 RGB-D 体感传感器 Kinect 之后，基于 Kinect 的三维地图构建技术获得快速发展。相比于利用激光传感器构建的二维地图，基于 RGB-D 传感器构建的三维地图不仅能够提供详细的环境描述，还可以实现物体识别、定位和抓取、制作语义地图等一系列激光传感器无法完成的功能。

P.Henry^[6]等人在 2010 年提出一种基于 RGB-D ICP 的方法，该方法使用 ICP 方法来对由颜色和深度图像生成的点云进行配准来构建三维室内环境地图，通过 Kinect 不断的重复观测，从而获得三维地图的收敛结果，最终降低定位过程中的累积误差。此方法需要进行反复观测，实用性不高，但是给移动机器人领域的研究人员提供了一种全新的思路。

Audras^[7]等人在 2011 年提出了一种 RGB-D 传感器的直接定位的方法，该方法省去特征提取过程，直接通过计算当前图像和参考帧的误差和雅克比矩阵，得到相对于参考帧的图像增量，更新地图。

F. Endres^[8]等人提出 RGB-D SLAM 算法，这种方法与 P.Henry 的方法类似，区别是前者采用稀疏束调整进行 2.5D 重映射误差优化，后者则采用 g2o 进行 3D 图优化，实时性方面后者优于前者。在后期的处理上，两种算法的地图表示方法不一样，前者将产生的点云转换成面元表示，后者利用体素表示，创建的地图均可直接用于定位与导航。

Stuckler^[9]等人提出一种基于点云匹配的方法,该方法结合颜色和深度信息到八叉树模型,利用特征描述子寻找两个八叉树模型的关联,并运行 ICP 算法进行匹配。

此外,近些年国内移动机器人领域在基于 RGB-D 传感器的三维地图构建技术方面也在快速成长与发展。

辛菁^[10]等人提出的机器人大视角 3D V-SLAM 方法,改进融合 BRIEF^[11]描述符和 FAST^[12]算子的 ORB^[13]特征提取与匹配流程,实现在视角大范围变化情况下相邻图像间的快速匹配,并使用 g2o 图优化算法对机器人的位姿进行优化,从而建立 3D V-SLAM 模型,最后构建 3D 地图并高效的估算出机器人的移动路径。

杨鸿^[14]等人提出一种利用帧之间的关系先使用 RANSAC 粗匹配,然后再进行 G-ICP 精确匹配的方法,来实现室内三维场景地图构建,并利用八叉树模型对三维地图进行多分辨率压缩存储,该方法有效降低了三维点云存储所需要的空间。

刘艳丽^[15]等人提出了一种基于视觉轨迹估计与扩展信息滤波的 VO-EIF SLAM 方法,同时将基于二进制描述符的特征关联方法与融合局部几何约束的多特征点快速匹配算法相结合,实现了快速闭环检测,最终完成三维地图的构建。

1.3 主要研究内容

本文主要工作在于研究基于二维地图蒙特卡罗定位的三维室内地图构建,针对视觉定位 RGB-D SLAM 过程中累积误差导致机器人错误定位的问题,提出了一种构建室内三维地图的方法:使用二维地图下基于激光雷达的蒙特卡罗定位来解决三维地图构建中的定位问题,同时将 Kinect 传感器采集的颜色图像和深度图像转换为局部三维地图,利用不同时刻位姿间的转换矩阵对局部三维地图进行拼接,更加准确的完成室内走廊三维地图的构建。主要研究内容如下:

(1) 研究 RGB-D 传感器 Kinect 相关技术和基于该传感器的视觉定位技术以及 RGB-D SLAM 算法。

(2) 研究 SLAM 的理论基础之一贝叶斯理论,然后介绍蒙特卡罗算法思想,讨论其算法流程和步骤。

(3) 介绍 ROS 机器人操作系统,在 Linux 环境下配置该操作系统,并利用 ROS 搭建室内三维地图构建系统。

(4) 研究在二维地图下基于激光雷达的蒙特卡罗定位所构建的三维地图相对于基于视觉定位 RGB-D SLAM 所构建三维地图的改善效果。

1.4 章节安排

论文章节安排如下：

第一章：绪论。叙述本文的研究背景、研究意义和移动机器人 SLAM 技术的发展概况及现状，并概括本文的研究内容。

第二章：RGB-D SLAM 算法。介绍能够同时获取颜色和深度信息的传感器 Kinect，讲解其工作原理、怎样获得点云以及摄像机模型。接着介绍了 RGB-D SLAM 的算法架构并对该架构中的各组成部分的具体实现进行了论述。

第三章：SLAM 算法理论基础和蒙特卡罗定位。研究贝叶斯理论，然后介绍蒙特卡罗算法思想，讨论其算法流程和步骤。

第四章：基于二维地图蒙特卡罗定位的三维室内地图构建方法原理。针对视觉定位 RGB-D SLAM 过程中累积误差引起机器人错误定位导致其所构建的三维地图产生扭曲的问题，提出了一种构建室内三维地图的方法：利用二维地图下基于激光雷达的蒙特卡罗定位来解决三维地图构建中的定位问题，同时结合能够获取颜色图像及深度图像的 Kinect 体感传感器收集周围环境信息来构建三维地图。

第五章：基于二维地图蒙特卡罗定位的三维室内地图构建的设计与实现。介绍了实验的硬件平台和软件平台，然后分析对比了使用本文方法利用机器人 TurtleBot2 所构建的三维室内走廊地图、使用 RGB-D SLAM 算法所构建的三维室内走廊地图和使用本文方法利用自制机器人所构建的三维室内走廊地图三个实验的实验结果。

第六章：总结与展望。该部分对本文的主要研究工作做出总结，同时也对以后的研究做设想及展望。

第二章 Kinect 传感器和 RGB-D SLAM 算法

Kinect 是 Microsoft 公司在 2010 年下半年开发的一款体感传感器^{[16][17]}，它能够同时提供颜色（RGB）和深度（Depth）图像实现三维深度图像的获取。Kinect 稳定的性能和适中的价格，使得它在移动机器人三维室内地图构建这一领域获得广泛应用与研究。本章首先介绍了 Kinect 的工作原理、怎样获得点云以及摄像机模型，接着介绍了 RGB-D SLAM 的算法架构并对该架构中的各组成部分的具体实现进行了论述。

2.1 Kinect 体感传感器

2.1.1 Kinect 工作原理

Kinect 传感器如图 2-1 所示。它有 3 个镜头，中部的为 RGB 相机，左边的为随机红外点云发射器，右边的为红外 CMOS 相机。

Kinect 相机得到的图像是 24 位深度 RGB 图像，分辨率为 640×480 ，最大帧率为 30HZ。左边的镜头和右边的镜头组成三维深度传感器，其水平视角为 57 度，垂直视角为 43 度。由深度传感器获得的深度图像的有效距离为 0.7m 到 6m，其中 1.2m 到 3.5m 的测量结果有较高的精度。利用 Kinect 获得的 RGB 图像和 depth 图像如图 2-2 所示。

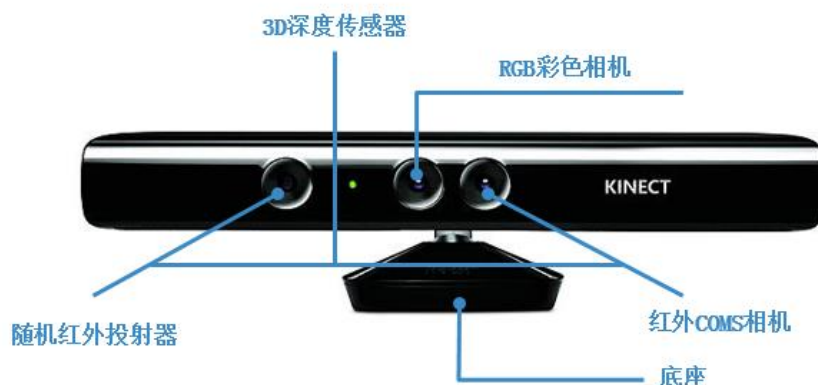


图 2-1 Kinect

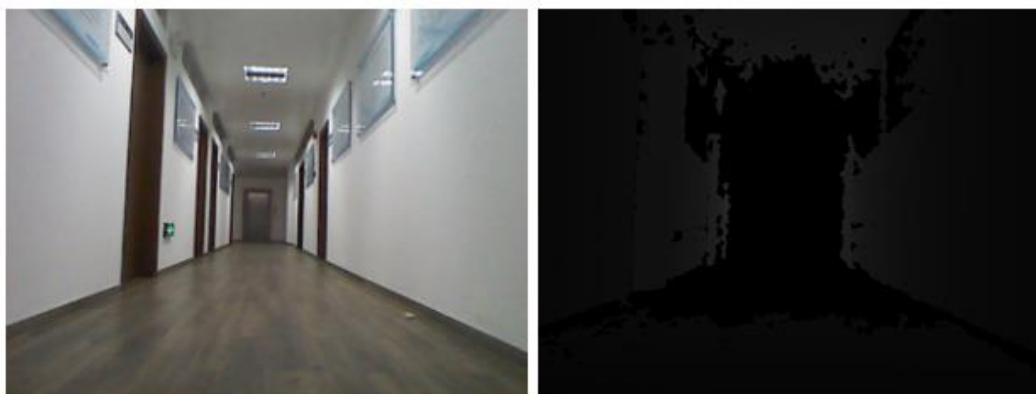


图 2-2 Kinect 获取的彩色图像和深度图像

使用一般的相机我们只能获取普通的 RGB 图像，如果我们想要得到周围环境的深度数据，通常要经过比较复杂的处理。例如利用从结构恢复运动（Structure from Motion, SFM）、视差图的方法，然而这些方法的获取深度数据的准确度难以保证。“Kinect 采用了以色列 primesense^[18]公司提供的 Light Coding 技术^[19]，这是使得 Kinect 摄像机能够在获取彩色图像的同时获取周围环境的深度信息。其 Light Coding 技术和传统 Light Coding 技术不同，传统的飞行时间（Time of Flight, TOF）技术采用的为连续脉冲方式，Kinect 采用的为连续照明方式。Kinect 使用是一般的 CMOS 感光芯片而非特制的，这也是其价格较低的原因。“Kinect 所使用的 Light Coding 技术的本质还是结构光技术，它使用激光散斑作为光源，照射需要测量的空间，并对该空间编码，其中激光散斑是激光照射到粗糙物体后形成的随机衍射斑点。这些斑点由于其高度随机性会随距离变换成不同图案，这种结构光打在物体上，测量物体上的散斑图案就可以标定物体在空间中的位置”^[20]。

2.1.2 摄像机模型

常用的摄像机模型有针孔相机模型、正交投影模型和拟透视投影模型，本文所使用的是针孔相机模型，该模型如图 2-3 所示。C 为三维空间坐标系原点，c 为二维成像平面坐标系原点，设相机的焦距是 f ，P 是三维空间中的任意一点，其空间坐标为 $p=[x,y,z]$ ，p 是 P 在相机二维成像平面上的投影，其坐标是 $p=[u,v,d]$

(u 、 v 为像素坐标, d 为深度数据)。

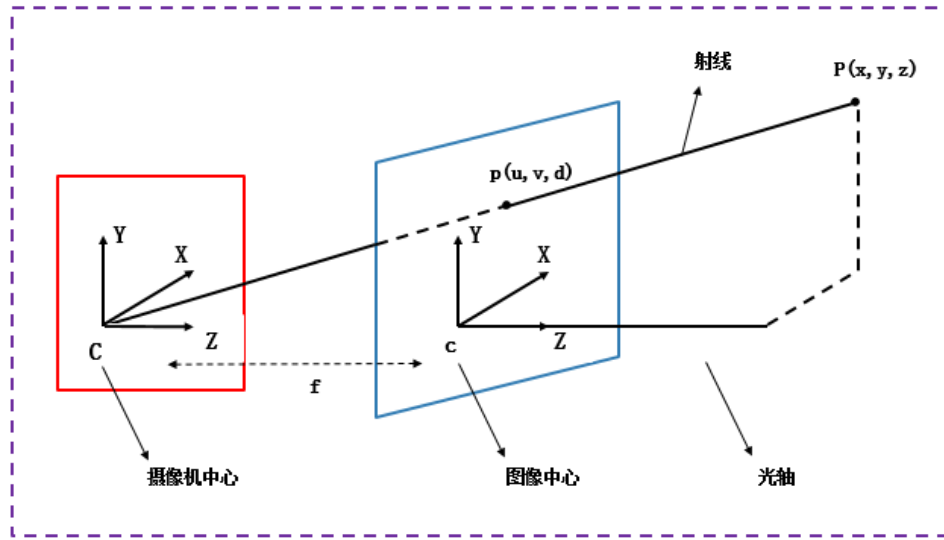


图 2-3 针孔相机模型

通过图 2-3 可知, 空间中任意一点坐标位置 $p=[x,y,z]$ 和它在图像中的像素坐标位置 $p=[u,v,d]$ 的对应关系如式 (2-1) (2-2) (2-3) 所示:

$$u = \frac{x \cdot f_x}{z} + c_x \quad (2-1)$$

$$v = \frac{y \cdot f_y}{z} + c_y \quad (2-2)$$

$$d = z \cdot s \quad (2-3)$$

其中, f_x 、 f_y 指相机在 X、Y 两个轴上的焦距, c_x 、 c_y 是指相机的光圈中心, s 是指深度图像的缩放因子, 在 Kinect 中其值为 1000, 对上面三式进行推导得到:

$$x = (u - c_x) \cdot z / f_x \quad (2-4)$$

$$y = (v - c_y) \cdot z / f_y \quad (2-5)$$

$$z = d / s \quad (2-6)$$

通常, 我们把 f_x 、 f_y 、 c_x 、 c_y 这四个参数定义为相机的内参矩阵 C 。Kinect

官方提供的内部参数为： $f_x = 518.0$ ； $f_y = 519.0$ ； $c_x = 325.5$ ； $c_y = 253.5$ ； $s = 1000$ ，但是为了更精确的定位，需要重新对 Kinect 进行标定。经典的标定方法是张正友标定^[21]，这种标定方法精度高，速度快，操作简单，易于实现，本文就是采用张正友标定法标定 Kinect 的内部参数。

在实际的情况中，真正的镜头通常有一些形变，主要的变形为径向形变，也会有轻微的切向形变，在这里主要考虑径向形变，假设 (u', v') 是实际投影的坐标，它和 (u, v) 之间满足：

$$u = u' + \delta_{u'}(u', v') \quad (2-7)$$

$$v = v' + \delta_{v'}(u', v') \quad (2-8)$$

考虑 δ 为径向形变量，则有：

$$\delta_{u'}(u', v') = k_1 u' r^2 + k_2 u' r^4 \quad (2-9)$$

$$\delta_{v'}(u', v') = k_1 v' r^2 + k_2 v' r^4 \quad (2-10)$$

其中 $r^2 = u'^2 + v'^2$ ， k_1 、 k_2 为径向形变参数，形变参数跟拍摄的场景无关，因此它们是内参数，而且与拍摄图像的分辨率无关。

2.1.3 Kinect 深度数据数学模型

利用 Kinect 的红外摄像机能够获得周围环境的深度信息，有了该信息我们就能够得到空间中某点的坐标位置。通常以 Kinect 摄像机的透视中心为原点来构建坐标系，取红外摄像机指向红外发射器的方向为 X 轴的正方向，用右手定则确定 Y 轴的正方向，Z 轴垂直于图像平面，其正方向为从图像平面指向场景中的物体。

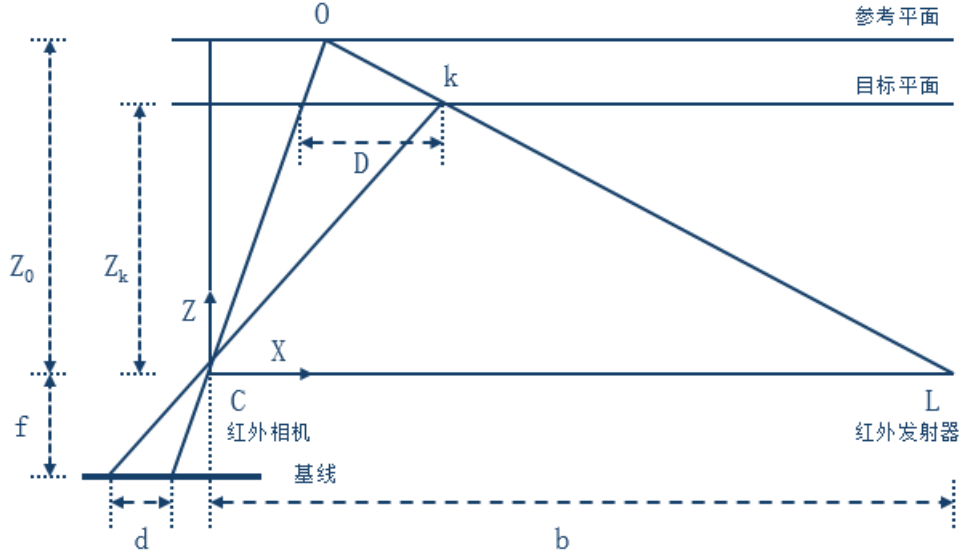


图 2-4 深度测量的数学模型

在图 2-4 中，深度三维坐标测量时有两个平面，一个参考平面和一个目标平面。当物体位置在参考平面时，记录下其散斑图案，当物体位于目标平面时，它的散斑图案会在 X 轴上会产生一个视差。由三角形相似可得：

$$\frac{D}{b} = \frac{Z_0 - Z_k}{Z_0} \quad (2-11)$$

$$\frac{d}{f} = \frac{D}{Z_k} \quad (2-12)$$

在 (2-11) 和 (2-12) 中， D 为物体在目标平面的移动的距离， b 为基线的长度， f 为 Kinect 红外摄像机的焦距， d 是散斑图案在图像平面的视差，联立上面两式，消去 D ，可得：

$$Z_k = \frac{Z_0}{1 + \frac{Z_0}{fb}d} \quad (2-13)$$

在 (2-13) 中常量 Z_0 、 f 和 b 可通过校准得到。相机的焦距 f 和环境中的某点的深度值（即该点的 Z 坐标）共同决定了该点的成像尺寸。由该点的图像坐标和尺度能够计算出 X、Y 轴的坐标，即式 (2-14)、(2-15)：

$$X_k = -\frac{Z_k}{f}(x_k - x_0 + \delta_x) \quad (2-14)$$

$$Y_k = -\frac{Z_k}{f}(y_k - y_0 + \delta_y) \quad (2-15)$$

其中, x_0 和 y_0 是二维成像平面坐标系的原点坐标, 原点的坐标是相机已经内定好的, x_k 和 y_k 是空间中一点在二维成像平面坐标系上投影的坐标, δ_x 和 δ_y 是镜头形变的校正项。通过式 (2-13) (2-14) (2-15) 可以得到空间中某点的坐标位置 (X_k, Y_k, Z_k) 。

2.2 RGB-D SLAM 算法

在 Microsoft 公司在 2010 年下半年推出 Kinect 之后, Microsoft 实验室和 Washington 大学、Carnegie Mellon 大学和德国 Freiburg 大学在 Kinect 上进行了很多的研究, 并应用在了移动机器人的定位、地图构建和自动导航中。Washington 大学使用基于图优化的 TORO 算法, 利用实时的视觉同步定位与构建地图系统构建三维地图。德国 Freiburg 大学采用类似的方法, 并对华盛顿大学的方法进行了改进, 他们使用 g2o^[22]图优化方法对相机位姿进行后端优化, 同时使用 SURF 特征点进行点云匹配, 之后 Carnegie Mellon 大学又对德国 Freiburg 大学的方法进行了改进, 逐渐演变出当前在移动机器人研究领域知名的 RGB-D SLAM^[23]算法, 该算法的框架图如图 2-5 所示。

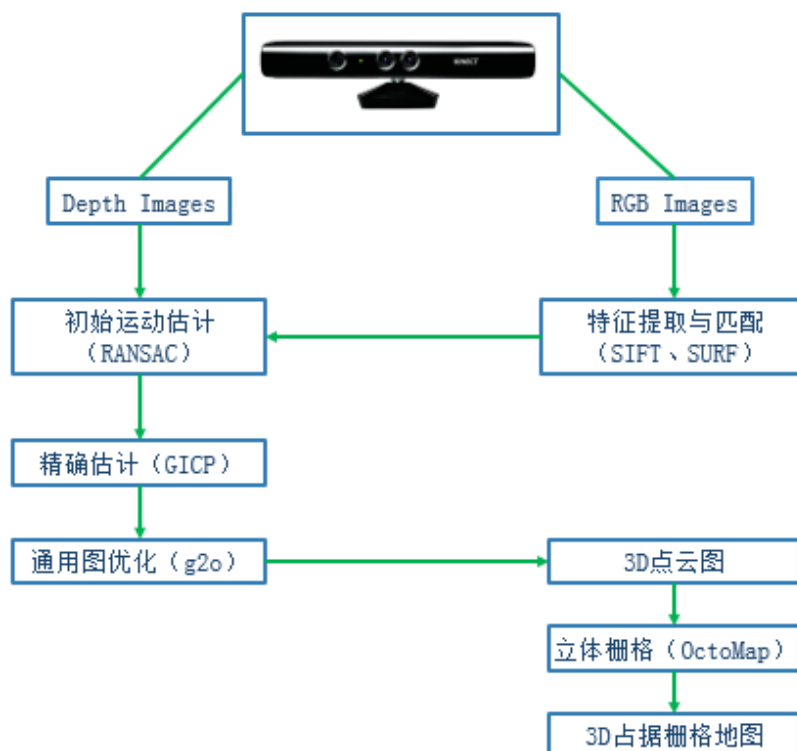


图 2-5 RGB-D SLAM 算法框图

RGB-D SLAM 算法主要包含以下五个流程：

(1) 特征提取与匹配：利用例如 SIFT、SURF 算法提取 Kinect 采集到的颜色图像中的特征点并对这些特征点进行匹配。

(2) 初始运动估计：第一步提取特征点会有一些离群点，利用 RANSAC 算法去除匹配之后的离群点对，进行相机初始运动估计。

(3) 初始位姿估计：将第二步得到的运动估计值作为初值，使用 ICP 方法对点云匹配，进行初始位姿的优化。

(4) 图优化：以不同时刻的位姿作为节点，帧与帧之间的位姿关系作为边，构造图，然后利用图优化算法，实现全局地图的一致性。

(5) 三维地图的构建：根据第四步的帧间关系，将不同时刻的位姿下获得点云融合，构造三维点云地图，接着利用 OctoMap 技术把三维点云地图压缩转变为基于体素的三维地图。

下面对 RGB-D SLAM 算法框架中各个部分所使用的方法进行描述。

2.2.1 特征提取与匹配

常用的图像特征点提取算法有 SIFT(Scale-invariant feature transform)^{[24][25][26]}算法、SURF(Speeded Up Robust Features)^{[27][28]}算法等。SIFT 算法对图像的尺度缩小、旋转、视角仿射变换具有不变性,即使改变旋转角度,图像亮度或拍摄视角,仍然能够得到好的特征点检测效果。它的缺点是每一个特征点都有一个 128 维的特征向量,这导致特征点匹配时速度较慢,同时 SIFT 算法对非线性的光照变换不够敏感,由于 SLAM 过程中实时性要求比较高,所以该算法不宜使用。SURF 算法是 SIFT 算法的改进算法,主要体现在计算速度上,它每个特征点有一个 64 维特征向量,所以特征点匹配时效率更高,同时对尺度和旋转具有鲁棒性。它利用 Hessian 矩阵所获得的图像局部最值是比较稳定的,但是其主方向要根据局部区域的梯度方向进行确定,这可能会导致所找到的主方向不够准确。考虑到移动机器人 3D SLAM 对实时性要求比较高,我们着重阐述提取特征效率更高的 SURF 算法。

基于 SURF 算法的特征提取与匹配框架图如图 2-6 所示。

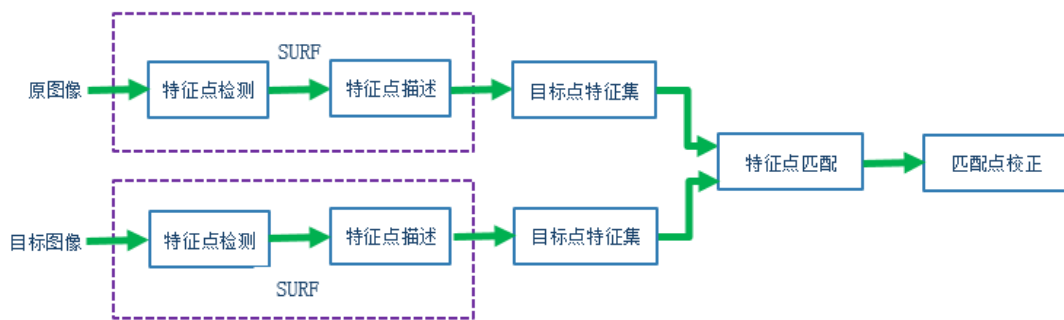


图 2-6 SURF 算法基本模型

上述过程主要分为两个步骤:

- (1) 特征点提取: 首先利用 Hessian 矩阵, 计算原图像与目标图像的特征值; 然后根据该特征值是否为领域极大值, 来确定该点是否为特征点。
- (2) 特征点匹配: 首先计算特征点的主方向和提取特征描述符, 之后提取特征点的特征向量, 利用两个特征向量的相似程度计算两个点是否为两幅图像中相

对应的特征点；然后采用两向量内积的最大值为最匹配的点，设定一个阈值，只有当这个最大值大于该阈值时才认为这两个特征点是相匹配的。

2.2.2 位姿估计

随机采样一致性方法 RANSAC (Random Sample Consensus)^[29]是从一组包含异常值的观测数据集中估计其数学模型参数的迭代方法，用于去除图像特征中点中的一些离群点，从而提高匹配精度。

RANSAC 算法基于这样一种假设：数据中包含大量的内围点，这些内围点符合同一个模型，外围点不符合这个模型。RANSAC 算法的输入一组观测数据集、拟合参数和拟合数据的方法。首先随机选取观测数据集中的若干数据，并假设这些数据是内围点，然后计算这些点的模型作为初始拟合模型；用这个模型去测试观测数据集中的剩余数据，若有数据在误差阈值范围内满足初始拟合模型，则这个数据是内围点，否则不是；测试完全部数据后，若得到的内围点足够多，则算法结束，否则重新选点重复上述算法过程。RANSAC 算法的具体步骤如图 2-7 所示。

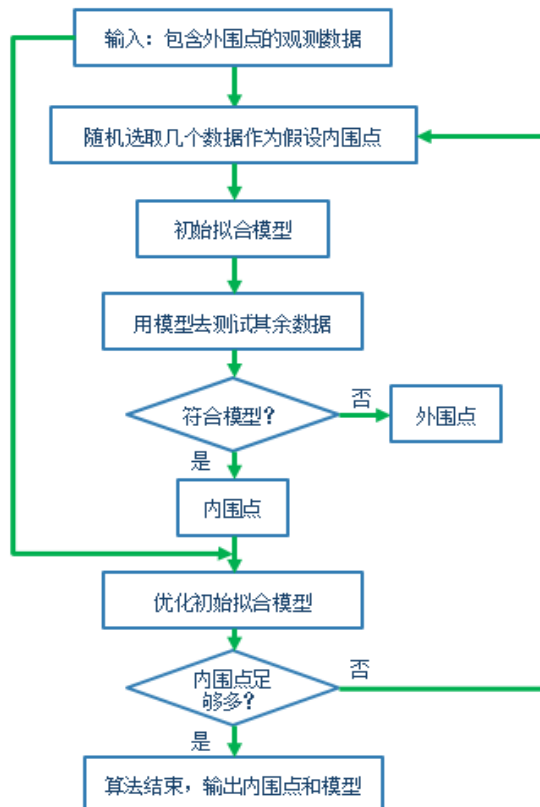


图 2-7 RANSAC 算法流程图

2.2.3 点云匹配

迭代最近点算法 (Iterative Closest Point, ICP) ^[30] 是计算机视觉领域中解决 3D 点云匹配最经典的算法。ICP 算法是由 Besl 和 Mckay 在 1992 年首次提出, 是一种基于形态的点集对点集的匹配算法, 用于求解两个点集之间的变换关系。 P_R 和 P_L 是两组对应的三维点集, 其中 $P_R = \{P_{r1}, P_{r2}, \dots, P_{rm}\}$, $P_L = \{P_{l1}, P_{l2}, \dots, P_{ln}\}$ 。我们要使用 ICP 算法求出两组点集的变换关系, 使 (2-16) 成立:

$$P_R = RP_L + t \quad (2-16)$$

式 (2-16) 中 R 是 3×3 的旋转矩阵, $t = [t_x, t_y, t_z]^T$ 是 3×1 的平移矩阵。ICP 算法的目的是使计算得到的 R 和 t 满足:

$$\min d = \sum_{i=1}^n \|p_{ri} - (Rp_{li} + t)\|^2 \quad (2-17)$$

ICP 算法的步骤如下:

- (1) 计算三维点集 P_R 和 P_L 的重心坐标, 进行点集进行中心化, 变换成新的点集坐标;
- (2) 从 P_R 和 P_L 中随机选取三组对应点, 计算旋转矩阵 R 和平移矩阵 t ;
- (3) 利用 R 和 t 将 P_L 变换为 P_L' , 计算 P_L' 中的点到 P_R 中对应点的距离的平方和 d ;
- (4) 若 d 大于设定距离阈值, 重复以上步骤, 直到满足 d 小于给定距离阈值或循环次数达到最大值时, 循环停止。

2.2.4 g2o 图优化

g2o(General Graph Optimization), 即通用图优化算法。在计算机视觉和移动机器人领域, 有许多问题的求解都能够转化到图中, 从而把问题转化为求取非线性误差函数的最小值问题, 在图优化中, 移动机器人在连续时间内求取的位姿被看做是图中的节点, 节点之间的边 (约束) 则代表测量的位姿变换。

由于估计误差的存在, 位姿变换组成的边不能形成全局一致的轨迹, 为了计

算全局一致的轨迹，可以使用 g2o 进行位姿图的优化。g2o 是一个开源图优化器，可以计算位姿图中非线性误差函数的最小值，该函数可由下式表示：

$$F(X) = \sum_{\langle i, j \rangle \in \mathcal{C}} e(x_i, x_j, z_{ij})^T \Omega_{ij} e(x_i, x_j, z_{ij}) \quad (2-18)$$

我们要目标是找到 $\arg \min F(X)$ ，式 (2-18) 中 $F(X)$ 为要优化的轨迹函数， X 代表位姿向量， Z_{ij} 和 Ω_{ij} 代表节点 x_i 与 x_j 之间的均值和信息矩阵，也就是前端计算的位姿变换， $e(x_i, x_j, z_{ij})$ 是测量位姿 x_i 与 x_j 满足约束 Z_{ij} 程度大小的误差函数，若 $e = 0$ ，则说明 x_i 和 x_j 完全满足 Z_{ij} 。

其实上述问题本质上是一个 BA(Bundle Adjustment)问题，即假设空间中某一点 P 在两帧图像中的投影点是 p_1 ， p_2 ，相机从第一帧的位置到第二帧的位置变换矩阵是 R 、 t ，摄像机的内参数矩阵是 K ，则有：

$$\begin{cases} \lambda_1 p_1 = KP \\ \lambda_2 p_2 = K(RP + t) \end{cases} \quad (2-19)$$

其中 λ 是尺度化因子，即现在的问题变为已知 p_1 、 p_2 、 K ，求 R 、 t 、 p ，可以用下式来表示，其中 i 表示第 i 个特征点：

$$\min_{X, R, t} \sum_{i=1}^N \left\| \lambda_1^i p_1^i - KP_i \right\|^2 + \left\| \lambda_2^i p_2^i - K(RP_i + t) \right\|^2 \quad (2-20)$$

BA 问题可以通过 g2o 求解，因此对于上述的图优化问题，只要将定义的节点和边添加到位姿图，就可以利用 g2o 进行位姿图的优化。在优化结束后，提取出优化的位姿，并用优化的位姿对环境进行地图构建。

2.2.5 OctoMap 三维立体栅格地图创建

OctoMap^[31]是一个基于八叉树（Octree）的绘图系统，它可以表现环境的三维结构，利用 OctoMap 我们能够优雅的压缩、更新三维地图，并且分辨率可调。相比点云，它能够省下大量的空间。

Octree 是一种用于描述三维空间的树状数据结构，它的每一个节点代表一个

立体元素。八叉树的构建是一个迭代过程，首先设定最小立体元素的体积，将地图中的每个立体元素分成 8 个小的立体元素，它们形成了 8 个子节点，如果子节点的体积大于设定值，则继续划分，直到元素体积小于或等于设定值时结束，这个最小的立体元素的体积大小代表了八叉树的分辨率。在三维地图的构建过程中，多数情况下，我们可以用布尔属性来描述八叉树中立体元素的被占据性。如果八叉树中某一个立体元素被认定为已被占据，那么它在八叉树中的相关节点将被初始化。在布尔属性设置中，每一个未被初始化的节点都可以被认定为是未知的或空闲的。为了解决这种模糊性，没有被初始化过的底层节点将被模型化为未知区域，而空闲的立体元素在树种可以被明确地表示为空闲节点，运用这种布尔占据状态可以对八叉树进行压缩存储。如果一个节点所有子节点都为未知的或空闲的，那么这个节点便可以修剪掉，这大大减少了树中需要被保存的节点的数目。由八叉树建立的地图如图 2-8 所示，从左到右使用不同的分辨率。

利用 OctoMap 方法创建三维地图的过程可以总结为：构建八叉树，传感器融合、树压缩、内存有效使用，地图文件压缩。



图 2-8 使用八叉树建立的地图

2.3 本章小结

本章介绍了能够同时获取颜色和深度信息的传感器 Kinect，讲解其工作原理、怎样获得点云以及摄像机模型。接着讲解了 RGB-D SLAM 的算法架构并对该架构中的各组成部分的具体实现进行了论述，包括图像特征与匹配、位姿估计、点云匹配、g2o 图优化方法、OctoMap 三维立体体素地图创建。

第三章 SLAM 算法理论基础和蒙特卡罗定位

SLAM 使用机器人携带的传感器所获取的控制信息和测量信息,来进行自身定位和构建周围环境地图,从而实现自动导航等任务。然而由于传感器本身性能的限制,以及环境中多种不确定性因素的影响,使得移动机器人很难准确的自主定位。为了解决机器人在环境中面临的各种不确定性,越来越多研究人员采用基于概率的定位方法,将概率理论应用到机器人定位方法中,即机器人的位姿信息通过概率的形式来表示,所有基于概率定位的方法都是以贝叶斯滤波理论(Bayes Filter, BF)为基础的。

3.1 贝叶斯滤波理论

贝叶斯理论是英国著名的数学家贝叶斯提出的一种概率理论,后来逐渐发展并被运用到统计推理方法中。BF 将 Bayes 公式与统计的方法相结合来解决感知信息的不确定性。最早应用 BF 的是文献^[32],该文献从贝叶斯决策理论的观点解决了一般的控制与随机估计问题^[33]。

贝叶斯滤波利用控制输入数据和传感器的测量数据来估计机器人的位姿,其核心思想是:以到当前时刻 t 为止所收集的数据为条件,递归地的估计位姿状态空间的后验信度。在 SLAM 问题中,机器人周围环境和自身描述为动态系统,机器人的位姿(二维笛卡尔坐标位置)和航向角描述为状态 x 。机器人定位时,它在 t 时刻对自己所处的状态有一个信度估计,该信度描述了机器人处于状态空间所有状态的概率分布情况。

3.1.1 SLAM 中信度的定义

从概率的角度来看,移动机器人的位姿可以通过一种概率分布来描述,称为信度:

$$Bel(x_t) = p(x_t | D_t), x \in X \quad (3-1)$$

信度代表了在 t 时刻,在给定所有的数据 D_t 条件下,移动机器人在状态 x_t 的概率, X 是所有位姿的集合。

在移动机器人同步定位与地图构建领域中，数据可以划分为控制数据和观测数据。控制数据代表系统的输入，由机器人电机编码器测得。测量数据代表机器人周围的环境信息，如通过激光雷达测的距离数据。

因此，先验信度定义为：

$$\overline{Bel}(x_t) = p(x_t | u_{1:t}, z_{1:t-1}) \quad (3-2)$$

其中 $u_{1:t} = \{u_1, u_2, \dots, u_t\}$ ， $z_{1:t-1} = \{z_1, z_2, \dots, z_{t-1}\}$ ， $\overline{Bel}(x_t)$ 是机器人在 t 时刻融合控制数据 u_t ，但还未融合测量数据 z_t 时的信度。

后验信度定义为：

$$Bel(x_t) = p(x_t | u_{1:t}, z_{1:t}) \quad (3-3)$$

它是机器人在 t 时刻融合了测量数据 z_t 和控制数据 u_t 所获得的信度。

3.1.2 贝叶斯滤波公式推导

在初始时刻 $t = 0$ 时，先验信度 $\overline{Bel}(x_t)$ 定义为初始信度 $Bel(x_0)$ 。如果机器人初始状态未知，则先验信度 $Bel(x_0)$ 的状态空间表现为一均匀分布函数，构成了机器人全局定位问题。如果机器人初始状态已知，则先验信度 $Bel(x_0)$ 的状态空间表现为一高斯分布函数，构成了机器人的位姿跟踪定位问题。

在 t 时刻，机器人在进行运动 u_t 后，先验信度为：

$$\overline{Bel}(x_t) = p(x_t | u_{1:t}, z_{1:t-1}) \quad (3-4)$$

当机器人接收到测量数据 z_t 后，其后验信息为：

$$Bel(x_t) = p(x_t | u_{1:t}, z_{1:t}) \quad (3-5)$$

利用全概率公式，我们可以推导出先验信度为：

$$\overline{Bel}(x_t) = \int p(x_t | x_{t-1}, u_{1:t}, z_{1:t-1}) \times p(x_{t-1} | u_{1:t}, z_{1:t-1}) dx_{t-1} \quad (3-6)$$

根据一阶马尔科夫假设：给定当前状态的知识，过去和未来的状态之前相互独立，即如果我们在已知位姿 x_{t-1} 的条件下，过去的测量数据和控制数据将不能再给求取 x_t 的概率提供额外的信息，则有：

$$p(x_t | x_{t-1}, u_{1:t}, z_{1:t-1}) = p(x_t | x_{t-1}, u_t) \quad (3-7)$$

在式 (3-7) 中，我们保留了控制数据 u_t ，因为它的出现的时刻在 x_{t-1} 之后。

同时，由于我们的控制都是随机选择的，所以控制数据 u_t 可以安全的从

$p(x_{t-1} | u_{1:t}, z_{1:t-1})$ 中消去。最终我们得到了先验信度的递推更新公式：

$$\begin{aligned} \overline{Bel}(x_t) &= \int p(x_t | x_{t-1}, u_t) \times p(x_{t-1} | u_{1:t-1}, z_{1:t-1}) dx_{t-1} \\ &= \int p(x_t | x_{t-1}, u_t) \times Bel(x_{t-1}) dx_{t-1} \end{aligned} \quad (3-8)$$

式 (3-8) 中， $p(x_t | x_{t-1}, u_t)$ 为机器人的运动模型。

根据贝叶斯法则 $p(x | y, z) = \frac{p(y | x, z)p(x | z)}{p(y | z)}$ ，后验信度可写为：

$$Bel(x_t) = \frac{p(z_t | x_t, u_{1:t}, z_{1:t-1}) \times p(x_t | u_{1:t}, z_{1:t-1})}{p(z_t | u_{1:t}, z_{1:t-1})} \quad (3-9)$$

再次根据一阶马尔科夫假设，我们在状态 x_t 已知的条件下，想要得到测量 z_t 的概率，由于过去的测量数据和控制数据不能给我们提供更多的信息，所以它们可以省去，则有：

$$p(z_t | x_t, u_{1:t-1}, z_{1:t-1}) = p(z_t | x_t) \quad (3-10)$$

将 (3-10) 代入 (3-9) 得：

$$Bel(x_t) = \frac{p(z_t | x_t) \times p(x_t | u_{1:t}, z_{1:t-1})}{p(z_t | u_{1:t}, z_{1:t-1})} \quad (3-11)$$

其中， $p(z_t | x_t)$ 为测量模型，它代表机器人在 t 时刻通过传感器由位姿状态 x_t 观测到 z_t 的概率密度。

将 (3-8) 代入 (3-11) 得到后验信度的递推更新公式：

$$Bel(x_t) = \eta p(z_t | x_t) \overline{Bel}(x_t) \quad (3-12)$$

其中， $\eta = p(z_t | u_{1:t}, z_{1:t-1})^{-1}$ ，作为标准化因子，又称归一化常数。

综上所述，利用贝叶斯滤波理论来解决机器人的定位问题，大致分为两个步骤：一是运动更新，初始时刻，根据机器人的运动模型和控制数据 u_t ，实现

$Bel(x_{t-1})$ 到 $\overline{Bel}(x_{t-1})$ 的推导,; 二是测量更新, 由机器人的测量模型和测量数据 z_t , 实现从 $\overline{Bel}(x_{t-1})$ 到 $Bel(x_t)$ 推导过程。

3.2 蒙特卡罗算法

蒙特卡罗 (Monte Carlo) 算法又可以称为随机抽样法, 它是一种采用概率统计理论近似求解数学、工程问题及金融经济的算法^[34]。该算法起源于1777年法国数学家蒲丰 (Buffon) 随机投针推算圆周率 π 的实验, 到 20 世纪 40 年代由乌拉姆 (Ulam) 和冯·诺依曼 (Von Neumann) 在美国第二次世界大战研制原子弹“曼哈顿计划”中提出, 并使用世界赌城摩罗哥的 Monte Carlo 作为该方法的名称。

3.2.1 蒙特卡罗定位算法思想

Monte Carlo 算法的核心思想和掷骰子决定事物结果的原理是一样的, 通过随机样本来模拟随机过程, 再根据统计方法来估算数字特征, 从而求出实际问题的解。它能够客观的描述随机性事件, 真实的反应物理实验过程, 并能有效的解决现实生活中的多种随机性事件。

蒙特卡罗粒子滤波被成功的应用于解决移动机器人的定位问题, 我将该算法称之为蒙特卡罗定位 (Monte Carlo Localization, MCL)^[35]。MCL 算法是 BF 理论的一种表现形式, 对应于 BF 的过程, MCL 首先根据机器人的运动模型进行预测, 然后当获取了激光测量值后, 再根据测量模型进行更新, 实现从上一个时刻信度 $Bel(x_{t-1})$ 到下一个时刻信度 $Bel(x_t)$ 的推导过程。

对于移动机器人的定位问题, MCL 的思路是采用带有权重的粒子表示假设的机器人位姿, 其中权重因子以概率的形式描述了该假设位姿是机器人真实位姿的可能性。基本的蒙特卡罗定位其实质是贝叶斯滤波算法的变形, 是对动态系统后验概率密度的递归估计:

$$Bel(x_t) = \eta p(z_t | x_t) \int p(x_t | x_{t-1}, u_t) Bel(x_{t-1}) dx_{t-1} \quad (3-13)$$

其中, x_t 代表机器人在 t 时刻位姿, u_t 代表机器人在 t 时刻的运动数据, z_t 代表在 t 时刻传感器的测量数据, $p(x_t | x_{t-1}, u_t)$ 代表机器人的运动模型,

$p(z_t | x_t)$ 代表机器人的测量模型， η 为归一化常数。

3.2.2 蒙特卡罗定位算法流程

蒙特卡罗定位算法主要由五部分构成，分别是初始化、运动更新和测量更新、更新粒子集合、根据权重大小重采样、时间更新。其算法流程图如图 3-1 所示：

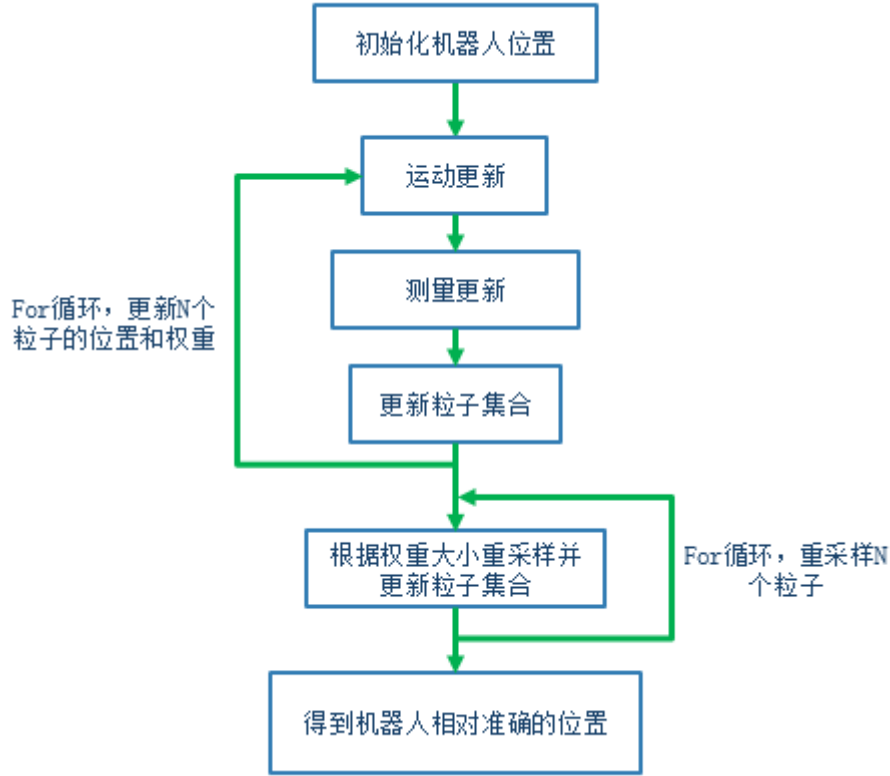


图 3-1 MCL 算法流程图

MCL 算法通过一组带有权重的离散粒子集合来描述信度 $Bel(x_t)$ ：

$$\chi_t = \{(x_t^{(i)}, \omega_t^{(i)}) | i = 1, \dots, N\} \quad (3-14)$$

其中， N 代表样本容量， $\omega_t^{(i)}$ 代表在 t 时刻假设位姿 $x_t^{(i)}$ 为机器人真实位姿的概率，且 $\sum_{i=1}^N \omega_t^{(i)} = 1$ 。

(1) 初始化

从先验概率分布 $Bel(x_0)$ 中抽取 N 个独立同分布的随机样本集合，设为

$\chi_0 = \{(x_0^{(i)}, \omega_0^{(i)}) | i = 1, \dots, N\}$ ，其中 $x_0^{(i)}$ 为第 i 个样本的位姿， $\omega_0^{(i)}$ 表示第 i 个

样本的权重，初始阶段每个样本的权重都是 $\omega_0^{(i)} = \frac{1}{N}$ 。

(2) 运动更新

运动更新阶段，针对每一个粒子，基于机器人的运动模型，结合里程计的控制信息，对每一个粒子的位姿进行更新：

$$x_t^{[m]} = \text{sample_motion_model}(u_t, x_{t-1}^{[m]}) \quad (3-15)$$

(3) 测量更新

测量更新阶段，针对每一个粒子，基于机器人的测量模型，结合激光传感器测量信息，对每一个粒子的权重进行更新。

$$\omega_t^{[m]} = \text{measurement_model}(z_t, x_t^{[m]}, m) \quad (3-16)$$

(4) 更新粒子集合

将每一个粒子运动更新和测量更新的结果保存到粒子集合 $\bar{\chi}_t$ ， $\bar{\chi}_t$ 为未进行重采样之前的粒子集合。

$$\bar{\chi}_t = \bar{\chi}_t + \langle x_t^{[m]}, \omega_t^{[m]} \rangle \quad (3-17)$$

(5) 重采样并更新粒子集合

根据每个粒子的权重大小进行重采样，权重越大的粒子被复制的次数越多，相反，权重很小的粒子可能被舍弃，之后更新粒子集合。

$$\text{draw } i \text{ with probability } \propto \omega_t^{[i]} \quad (3-18)$$

$$\text{add } x_t^{[i]} \text{ to } \chi_t \quad (3-19)$$

3.3 本章小结

本章介绍了 SLAM 理论，探讨了贝叶斯滤波及其公式推导，讨论了贝叶斯滤波在移动机器人中的应用，同时介绍了以贝叶斯滤波为基础的蒙特卡罗算法及其算法思想和算法流程。

第四章 基于二维地图蒙特卡罗定位的三维室内地图构建方法原理

为了提高所构建三维地图的质量，其中最关键就是提高机器人定位的精度，对于机器人的定位问题，国内外学者进行很多研究，各种不同的定位算法取决于所采用的传感器，传感器已经从传统的里程计、超声波、激光雷达测距仪逐渐过渡到现在的视觉传感器。文献^[36]分别综述了目前三种常见的机器人定位算法：两种扩展卡尔曼滤波（Extended Kalman Filter, EKF）算法、粒子滤波（Particle Filter, PF）算法以及基于视觉定位的 SLAM 算法，并且分析了上述几种算法的特点，然而在实际使用中，以上三种算法在定位精度方面都有所欠缺。文献^[37]使用激光雷达传感器获取周围环境的距离信息，利用线段特征匹配的扩展卡尔曼滤波进行机器人的定位，该方法在周围环境中的线段特征较少时，其定位精度会变的比较差。文献^[38]构造激光雷达的测量模型和里程计的误差模型，使用扩展卡曼滤波实现了机器人的定位，但在室内走廊等空间结构化的环境中，定位精度不高，所构建的二维地图准确度也比较差。文献^[39]使用 Kinect 相机，提取 Kinect 所拍摄图像每一帧的 SURF 特征，然后结合视觉里程计和通用图优化的方法进行机器人的定位，这种方法在环境中的视觉特征点比较少时，定位准确度会大幅度下降，导致所构建的三维地图不够理想。

因此，本文提出基于二维地图蒙特卡罗定位的三维地图构建方法，以里程计、Kinect、激光雷达为主要传感器，以机器人 TurtleBot2 为实验平台，一方面建立机器人的里程计运动模型和激光雷达测量模型，使用蒙特卡罗算法将机器人的运动模型和测量模型进行融合，在事先构建好的二维地图中实现机器人的定位；另一方面使用 RGB-D 相机 Kinect 采集周围环境的颜色和深度信息来构建局部三维地图，通过机器人定位得到不同时刻的位姿，求出这些位姿间的转换矩阵，利用该转换矩阵将局部三维地图进行拼接，最终构建出环境整体的三维地图。

4.1 二维地图下的蒙特卡罗定位

使用蒙特卡罗算法，我们能够在二维地图下对机器人进行定位。蒙特卡罗定

位（MCL）是一种基于粒子滤波的算法，它能够用于解决机器人的局部定位问题和全局定位问题，该算法比较容易由计算机实现。由于机器人的运动模型

$p(x_t | x_{t-1}, u_t)$ 和测量模型 $p(z_t | x_t)$ 被应用与 MCL 的运动预测阶段和测量更新阶段，所以我们首先来介绍这两个模型^[40]。

4.1.1 机器人运动模型

机器人的运动模型使用机器人内部的里程计的测量值作为机器人相关运动信息 u_t ，通过读取电机编码器的测量值，底层控制芯片收集机器人的运动距离信息，利用底层控制芯片中的程序处理这些运动信息，我们就能够得到机器人的里程计信息 u_t 。机器人里程计模型如图 4-1 所示。

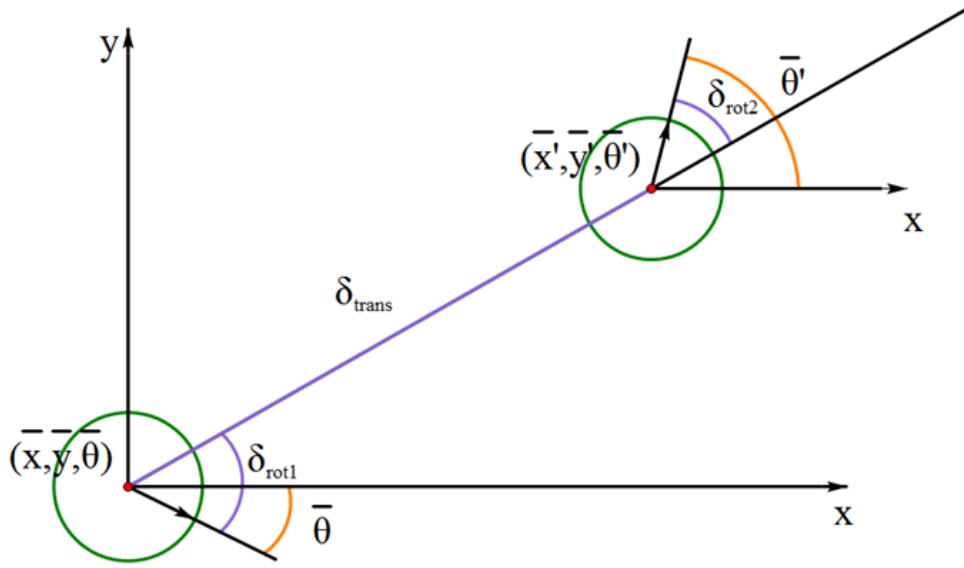


图 4-1 机器人里程计模型

在图 4-1 中，在时间间隔 $(t, t-1]$ 中，机器人从位姿 x_{t-1} 移动到位姿 x_t ，同时里程计给我们反馈了从 $\bar{x}_{t-1} = (\bar{x} \ \bar{y} \ \bar{\theta})^T$ 运动到 $\bar{x}_t = (\bar{x}' \ \bar{y}' \ \bar{\theta}')^T$ 相关运动信息。这里字母上的横线代表该变量为里程计的测量值，该测量值属于机器人的内部坐标系，机器人内部坐标系与全局坐标系的关系是未知的。机器人在时间间隔 $(t, t-1]$ 的运动被近似为一次旋转 δ_{rot1} ，加一次平移 δ_{trans} ，然后是第二次旋转

δ_{rot2} 。下面公式描述了如何由里程计的反馈值 $u_t = (\bar{x}_{t-1} \ \bar{x}_t)^T$ 计算两次旋转和一次平移的值：

$$\delta_{rot1} = \text{atan2}(\bar{y}' - \bar{y}, \bar{x}' - \bar{x}) - \bar{\theta} \quad (4-1)$$

$$\delta_{trans} = \sqrt{(\bar{x} - \bar{x}')^2 + (\bar{y} - \bar{y}')^2} \quad (4-2)$$

$$\delta_{rot2} = \bar{\theta}' - \bar{\theta} - \delta_{rot1} \quad (4-3)$$

其中， atan2 是 atan 的扩展：

$$\text{atan2}(y, x) = \begin{cases} \text{atan}(y/x) & \text{if } x > 0 \\ \text{sign}(y)(\pi - \text{atan}(|y/x|)) & \text{if } x < 0 \\ 0 & \text{if } x = y = 0 \\ \text{sign}(y) \pi/2 & \text{if } x = 0, y \neq 0 \end{cases} \quad (4-4)$$

为了给运动误差建模，假设两次旋转和一次平移的真实值为里程计的测量值减去一个独立的噪声 ε_{b^2} ：

$$\hat{\delta}_{rot1} = \delta_{rot1} - \varepsilon_{\alpha_1 \delta_{rot1}^2 + \alpha_2 \delta_{trans}^2} \quad (4-5)$$

$$\hat{\delta}_{trans} = \delta_{trans} - \varepsilon_{\alpha_3 \delta_{trans}^2 + \alpha_4 \delta_{rot1}^2 + \alpha_4 \delta_{rot2}^2} \quad (4-6)$$

$$\hat{\delta}_{rot2} = \delta_{rot2} - \varepsilon_{\alpha_1 \delta_{rot2}^2 + \alpha_2 \delta_{trans}^2} \quad (4-7)$$

其中 ε_{b^2} 为一个均值为 0，方差为 b^2 的噪声变量，它通常服从高斯分布或三角分布，本文选取的是高斯分布，参数 α_1 ， α_2 ， α_3 ， α_4 是机器人具体的误差参数。

最终，上一个时刻的位姿 x_{t-1} 经过两次旋转 $\hat{\delta}_{rot1}$ 、 $\hat{\delta}_{rot2}$ ，和一次平移 $\hat{\delta}_{trans}$ ，得到 t 时刻机器人真实的位姿 $x_t = (x' \ y' \ \theta')^T$ ：

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} \hat{\delta}_{trans} \cos(\theta + \hat{\delta}_{rot1}) \\ \hat{\delta}_{trans} \sin(\theta + \hat{\delta}_{rot1}) \\ \hat{\delta}_{rot1} + \hat{\delta}_{rot2} \end{pmatrix} \quad (4-8)$$

其中 $x_{t-1} = (x \ y \ \theta)^T$ 代表上一个时刻的位姿。根据上式，每一个经过里程计

模型变换的粒子将服从运动模型 $p(x_t | x_{t-1}, u_t)$ 的概率分布。

4.1.2 测量模型

在得到运动模型概率 $p(x_t | x_{t-1}, u_t)$ 之后，我们就可以计算测量概率 $p(z_t | x_t)$ 了。假设 m 为已知的地图信息（我们可以利用 gmapping 构建二维环境地图得到地图信息 m ）， $z_t^i = (i = 1, 2, 3, \dots, K)$ 代表传感器的单个测量。则我们可以通过下式来计算 $p(z_t | x_t, m)$ ：

$$p(z_t | x_t, m) = \prod_{k=1}^K p(z_t^k | x_t, m) \quad (4-9)$$

如果 $z \neq z_{\max}$ ，则单个测量的似然 $p(z_t^k | x_t, m)$ 用以下公式计算：

$$x_{z_t^k} = x + x_{k,sens} \cos \theta - y_{k,sens} \sin \theta + z_t^k \cos(\theta + \theta_{k,sens}) \quad (4-10)$$

$$y_{z_t^k} = y + y_{k,sens} \cos \theta + x_{k,sens} \sin \theta + z_t^k \sin(\theta + \theta_{k,sens}) \quad (4-11)$$

$$dist = \min_{x', y'} \left\{ \sqrt{(x_{z_t^k} - x')^2 + (y_{z_t^k} - y')^2} \mid \langle x', y' \rangle \text{ occupied in } m \right\} \quad (4-12)$$

$$p(z_t^k | x_t, m) = z_{\text{hit}} \cdot \text{prob}(dist, \sigma_{\text{hit}}) + \frac{z_{\text{random}}}{z_{\max}} \quad (4-13)$$

其中， $\text{prob}(dist, \sigma_{\text{hit}})$ 代表在均值为 0，标准差为 σ_{hit} 高斯分布条件下，计算当值为 $dist$ 的概率。 $x_{k,sens}$ 和 $y_{k,sens}$ 代表激光雷达传感器相对于机器人中心的位置， $\theta_{k,sens}$ 表示激光雷达传感器光束方向相对于机器人朝向的角度， z_{hit} ， z_{random} ， $z_{\max}$ 代表测量模型的内部参数。

4.1.3 自适应蒙特卡罗定位

蒙特卡罗定位（MCL）有很多不同的版本。本文中我们使用的是自适应蒙特卡罗定位(Adaptive Monte Carlo Localization, AMCL)，AMCL 是 MCL 的升级版，它使用 KLD 采样技术来更新粒子。KLD 采样可以随着时间对粒子个数进行调整

[40]。KLD 采样由 KL 散度 (Kullback-Leibler divergence) [41] 推导得到, 其中 KL 散度也称为 KL 距离, 它能够测出两个概率分布之间的差异大小。在 MCL 的每一次迭代中, KLD 采样通过概率 $1-\delta$ 来确定样本的个数。为了得到一个逼近的误差界, 我们假设真实的后验概率密度是由一个离散的, 分段的连续分布给出的。KLD 采样时确定样本数目, 使样本的最大似然估计 (Max Likelihood Estimation, MLE) 和真实的后验概率密度之间的距离小于事先指定的阈值 ε 。因为真实的后验概率密度和 MLE 之间的距离是通过 KL 散度来测量的, 因此称为 KLD 采样方法。KL 散度是两个概率分布 p 和 q 之间差异大小的度量, 定义为:

$$K(p, q) = \sum_x p(x) \log \frac{p(x)}{q(x)} \quad (4-14)$$

KL 距离的值都为非负, 当且仅当两个分布的相同时 KL 距离取值才为 0, 它是概率分布之间差异的衡量标准[42]。假设 n 个样本是从一个带有 K 个不同区间的离散分布中采样得到的, 向量 $\underline{X} = (X_1, \dots, X_k)$ 代表从每个区间获得的样本数量, 且 $\underline{X} \sim M_k(n, \underline{p})$, 其中 $\underline{p} = p_1, \dots, p_k$ 表示每个区间的真实概率, $\underline{p}$ 的 MLE 采用的 n 个样本, 通过 $\hat{\underline{p}} = n^{-1} \underline{X}$ 给出。利用似然比统计量 λ_n 来检验 $\underline{p}$:

$$\log \lambda_n = \sum_j^k X_j \log \left(\frac{\hat{p}_j}{p_j} \right) \quad (4-15)$$

因为 $X_j = n\hat{p}_j$, 所以有:

$$\log \lambda_n = n \sum_j^k \hat{p}_j \log \left(\frac{\hat{p}_j}{p_j} \right) \quad (4-16)$$

由于似然比统计量是 MLE 和真实概率分布之间 KL 散度的 n 倍, 即:

$$\log \lambda_n = nK(\hat{\underline{p}}, \underline{p}) \quad (4-17)$$

可以证明, 似然比收敛于一个自由度为 $k-1$ 的 χ^2 分布:

$$2 \log \lambda_n \rightarrow_d \chi_{k-1}^2, n \rightarrow \infty \quad (4-18)$$

MLE 和真实分布之间的 KL 距离小于或等于 ε 的概率使用 $p_p(K(\underline{\hat{p}}, p) \leq \varepsilon)$ 来表示, 样本数量和这个概率之间的关系为:

$$\begin{aligned} P_p(K(\underline{\hat{p}}, p) \leq \varepsilon) &= P_p(2nK(\underline{\hat{p}}, p) \leq 2n\varepsilon) \\ &= P_p(2\log \lambda_n \leq 2n\varepsilon) \\ &\approx P(\chi_{k-1}^2 \leq 2n\varepsilon) \end{aligned} \quad (4-19)$$

其中通过下式来确定 χ^2 分布的分位数:

$$p(\chi_{k-1}^2 \leq \chi_{k-1,1-\delta}^2) = 1 - \delta \quad (4-20)$$

如果 $2n\varepsilon = \chi_{k-1,1-\delta}^2$, 通过 (4-19) 和 (4-20) 可得:

$$P_p(K(\underline{\hat{p}}, p) \leq \varepsilon) \approx 1 - \delta \quad (4-21)$$

由此可以得出样本数量和近似特性之间的关系, 即样本数量为:

$$n = \frac{1}{2\varepsilon} \chi_{k-1,1-\delta}^2 \quad (4-22)$$

我们根据 (4-22) 来确定 n , 这需要计算 χ^2 分布的分位数。由 Wilson-Hilferty 变换近似可以得到:

$$n = \frac{1}{2\varepsilon} \chi_{k-1,1-\delta}^2 \approx \frac{k-1}{2\varepsilon} \left\{ 1 - \frac{2}{9(k-1)} + \sqrt{\frac{2}{9(k-1)}} z_{1-\delta} \right\}^3 \quad (4-23)$$

(4-23) 表明, 当样本数量 $n \approx \frac{k-1}{2\varepsilon} \left\{ 1 - \frac{2}{9(k-1)} + \sqrt{\frac{2}{9(k-1)}} z_{1-\delta} \right\}^3$ 时, MLE 和真

实分布之间的 KL 距离将小于或等于误差界限 ε 。其中, $z_{1-\delta}$ 可以在标准的统计表中得到, 为标准正态分布的 $1-\delta$ 分位数。由 (4-23) 可以看出, 所需样本数量与引进的区间数量 K 成线性关系, 同时与误差界限 ε 成反比。

基于 KLD 采样的 AMCL 算法流程如下:

输入: $t-1$ 时刻的粒子集合 $S_{t-1} = \{(\mathbf{x}_{t-1}^{(i)}, \omega_{t-1}^{(i)} \mid i=1, \dots, n)\}$ 和控制量 u_{t-1} , t 时刻的测量值 z_t , 界限 ε 和 δ , 区间 b 尺寸 Δ , 最小样本容量 $n_{\chi_{\min}}$, 二维地图信息 m , $S_t = \emptyset$, $n=0$, $n_s=0$, $k=0$, $\alpha=0$;

- Step1: 对粒子进行运动更新 $x_t^{(i)} = \text{sample_motion_model } p(x_t^{(i)} | x_{t-1}^{(i)}, u_t)$,
并计算该粒子相应的权重 $\omega_t^{(i)} = p(z_t | x_t^{(i)}, m)$;
- Step2: 权重求和 $\alpha = \alpha + \omega_t^{(i)}$;
- Step3: 更新粒子样本集合 $S_t = S_t + \langle x_t^{(i)}, \omega_t^{(i)} \rangle$;
- Step4: 如果 $x_t^{(i)}$ 落入空区间 b , 更新区间数量 $k = k + 1$, $b = \text{non-empty}$;
- Step5: 如果 $k > 1$, 则有 $n_s = \frac{k-1}{2\varepsilon} \left\{ 1 - \frac{2}{9(k-1)} + \sqrt{\frac{2}{9(k-1)}} z_{1-\delta} \right\}^3$;
- Step6: 更新样本容量 $n = n + 1$;
- Step7: 若 $n < n_s$ 或 $n < n_{s\min}$, 返回到 Step1;
- Step8: *For* $i = 1, \dots, n$, 权重归一化 $\omega_t^{(i)} = \omega_t^{(i)} / \alpha$;
- 输出: S_t ;

4.2 基于二维地图蒙特卡罗定位的三维地图构建方法

基于视觉定位三维地图构建方法的缺点:

(1) 在基于视觉定位的 SLAM 中使用了视觉里程计方法来解决定位问题, 视觉里程计方法所包含的图像特征提取与匹配过程将消耗大量的计算资源, 导致基于视觉定位的 SLAM 在实际使用中实时性不高。

(2) 由于视觉定位依赖于图像信息, 在类似于走廊和过道等高度对称的室内环境场景中, 从不同时刻图像中提取的特征匹配时将会产生误匹配, 误匹配的存在将导致定位时累积误差的产生。

(3) 同时, 因为视觉定位只依赖一个例如 Kinect 的视觉传感器, 所以由定位产生的累积误差比较难校正。

针对以上问题, 本文提出了一种基于二维地图蒙特卡罗定位的三维地图构建方法。该方法使用基于激光雷达和里程计的 AMCL 在二维地图下对机器人进行定位, 省去了图像特征提取与匹配过程, 这一方面避免了图像特征误匹配的产生, 另一方面还增强了构建三维地图的实时性。同时该方法融合激光雷达、机器人里

程计等多个传感器的信息，来解决定位时累积误差校正的问题。

基于二维地图蒙特卡罗定位的三维地图构建流程图如图 4-2 所示。首先，利用 gmapping 算法我们先构建一副环境的二维地图。然后在已经构建好的二维地图中，我们使用游戏手柄远程控制机器人运动，机器人在运动的过程中，通过里程计和激光雷达传感器，不断的获取机器人的运动距离信息 u_t 和周围环境的距离信息 z_t ，机器人的运动距离 u_t 和周围环境的距离信息 z_t 送给基于 KLD 采样的 AMCL 算法处理，经过 AMCL 算法处理后，我们就能对机器人在事先构建好的二维地图中进行定位，得到机器人位姿。随着机器人不断的运动，我们可以得到机器人在二维地图中不同时刻的位姿 $x_1, x_2, \dots, x_t$ ，下面的公式描述了机器人上一个时刻位姿和下一个时刻位姿之间的转换关系：

$$x_t = R \cdot x_{t-1} + t \quad (4-24)$$

其中， R 代表旋转矩阵， t 代表平移向量。由于 Kinect 放置在机器人上，它和机器人之间的位置关系是固定，所以 R 和 t 也代表机器人和 Kinect 的旋转矩阵和平移矩阵。利用旋转矩阵 R 和平移 t ，我们能够得到转换矩阵 T ：

$$T = \begin{bmatrix} R_{3 \times 3} & t_{3 \times 1} \\ O_{1 \times 3} & 1 \end{bmatrix} \in R^{4 \times 4} \quad (4-25)$$

每当机器人在二维地图中通过定位获得一个新位姿 x_t ，Kinect 就会在该位置拍摄一幅二维颜色图像和一副深度图像。利用颜色图像和深度图像我们就可以构建一幅局部的三维地图，其中三维地图由点云形式呈现。下面的公式描述了如何通过二维 RGB 图像和 depth 图像来构建三维点云地图：

$$z = d / s \quad (4-26)$$

$$x = (u - c_x) \cdot z / f_x \quad (4-27)$$

$$y = (v - c_y) \cdot z / f_y \quad (4-28)$$

其中， c_x 、 c_y 、 f_x 、 f_y 代表 Kinect 的内部参数， d 代表深度信息， s 代表缩放尺度因子， u 和 v 是空间一点在二维图像中的像素位置， x 、 y 、 z 为三

维点云的空间位置。

假设机器人在位姿 x_{t-1} 处获取局部三维点云地图 p_{t-1} ，在位姿 x_t 处获取局部三维点云地图 p_t ，那么利用转换矩阵 T 我们就能对 p_{t-1} 和 p_t 进行点云融合得到一幅新的地图^[43]。将此新地图与下一个时刻的三维局部地图一个接一个的融合，最终我们能够得到一幅全局三维地图。

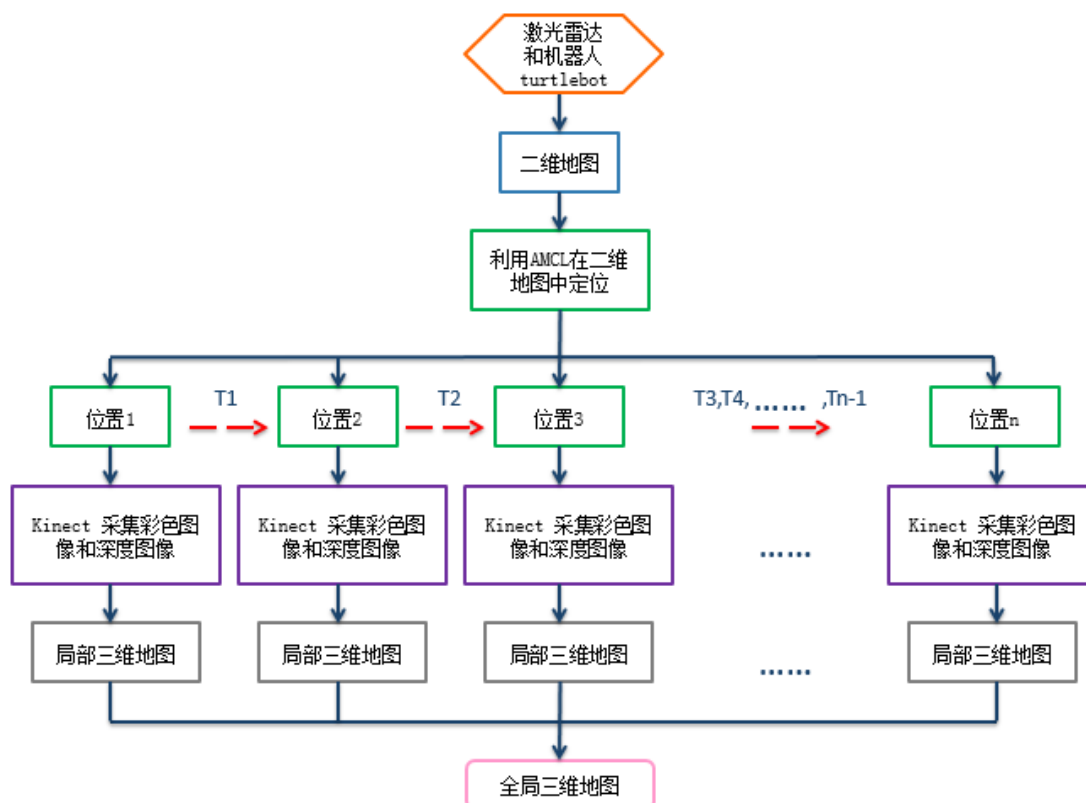


图 4-2 基于二维地图蒙特卡罗定位的三维地图构建流程图

4.3 本章小结

本章介绍了二维地图下基于激光雷达的蒙特卡罗定位融合 Kinect 传感器采集的环境信息的三维地图构建。针对视觉定位 3D SLAM 过程中图像特征点误匹配产生的累积误差导致机器人错误定位的问题，提出了一种构建室内三维地图的方法：利用二维地图下基于激光雷达的蒙特卡罗定位替代视觉定位，同时结合能够获取颜色图像和深度图像的 Kinect 传感器采集周围环境信息来构建三维地图。

第五章 基于二维地图蒙特卡罗定位的三维室内地图构建的设计与实现

5.1 实验平台硬件

5.1.1 TurtleBot 机器人

Turtlebot^[44]是美国 Willow Garage 公司开发的开源移动机器人平台，具有开源软件、价格低廉等优点，受到机器人爱好者的广泛欢迎。目前为止，TurtleBot 机器人已有两代产品 TurtleBot1 和 TurtleBot2。硬件方面，第一代产品以 iCreate Robot 为控制底座，第二代产品以 Kobuki 为控制底座，如图 5-1 所示。软件方面，两代产品均使用机器人操作系统（Robot Operating System, ROS）^[45]为软件平台进行开发。



图 5-1 机器人 TurtleBot2

本章实验使用的是 TurtleBot2，其主要硬件如下：

（1）Microsoft Kinect 1.0 传感器 Kinect 是微软在 2010 年推出的一款体感传感器，如图 5-2 所示。它由 RGB 相机、深度传感器、麦克风阵列和机动倾斜马达等组成。Kinect 具有成本低、精度高、操作简单、软件库丰富等优点，它可以同时采集 RGB 图像和 depth 图像，能够装配在智能机器人上用来感知周围环

境^{[46][47]}。随着 Kinect 的软件开发包 Kinect for Windows SDK 和 OpenNI 的代码资源逐渐开源共享,越来越多的爱好者和研究人员对 Kinect 进行开发和研究,进一步扩展了其应用领域。



图 5-2 Microsoft Kinect1.0 传感器

(2) Kobuki 控制底座 Kobuki 是韩国 Yujin 公司开发的移动机器人平台,如图 5-3 所示。机器人接口控制板有 32 个内置传感器,两个驱动轮,2 个车轱共四个可移动小轮,110 度/秒单轴陀螺仪,一个预排程序的行为编码器,一个可以扩大的输入输出端口和一个后挡板。开放式接口可以直接实现对机器人的移动、声音以及显示、输入传感器的操作。同时 Kobuki 包含一个完整的机器人开发包,使得用户能直接编程实现上层的机器人行为而无需关心底层控制代码和机械装配。用户还能够在该移动平台上加装额外的电子设备,例如激光雷达传感器、机械手、显示器,惯性测量单元 IMU 等,来进行的创造性开发。



图 5-3 Kobuki 控制底座

5.1.2 RPLidar 激光雷达测距传感器

激光雷达测距传感器用于二维地图的构建和在二维地图中定位，机器人在从工业环境走向家庭环境的过程中面临着许多问题，其中最重要的就是价格太贵。作为移动机器人用来定位和构建地图的重要传感器的激光雷达，其价格成本往往在数万元，近年来国内团队 RoboPeak 推出低成本激光雷达 RPLidar（如图 5-4 所示），售价仅为普通工业级激光雷达的十分之一，传统的激光雷达传感器使用高精度的光学部件和机械部件来确保性能，而 RPLidar 使用了相对普通、价格低廉的光学部件和机械部件，但采用更加高效先进的算法处理来弥补了其性能。虽然此款激光雷达传感器扫描频率低、采样角度没有工业级激光雷达传感器稳定，但是其扫描出来的二维地图效果非常好。所以对于一些特定的使用场景很适合，譬如做 SLAM，最终经过性价比的权衡，本文最终选择 RPLidar 激光雷达作为测距传感器，它可以对周围环境进行 360 度的扫描，利用其获取的数据结合 AMCL 算法在二维地图中进行定位，能很好的完成本实验的任务要求。



图 5-4 RPLidar 激光雷达

5.1.3 Kobuki 控制底座运动学模型

Kobuki 控制底座可以进行平移、旋转和弧线运动，其差速驱动运动学模型如图 5-5 所示。

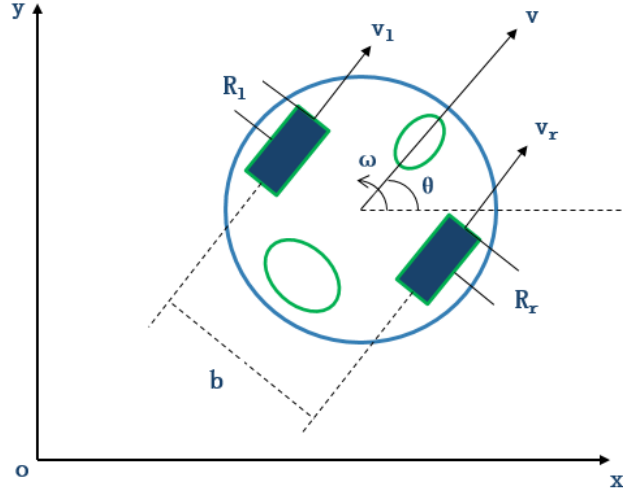


图 5-5 Kobuki 控制底座运动学模型

根据以上运动学模型，控制底座的线速度和角速度公式为：

$$v = \frac{v_l + v_r}{2} \quad (5-1)$$

$$\omega = \frac{v_r - v_l}{b} \quad (5-2)$$

式 (5-1) 和 (5-2) 中， v_l 和 v_r 为控制底座左轮和右轮的速度， b 为左右轮之间的间距。那么从 $t-1$ 时刻到 t 时刻，时间间隔为 Δt ，控制底座的动力学过程模型的方程为：

$$x_t = x_{t-1} + \begin{pmatrix} v\Delta t \cos \theta \\ v\Delta t \sin \theta \\ \omega\Delta t \end{pmatrix} \quad (5-3)$$

其中 $x_{t-1} = (x \ y \ \theta)^T$ 为 $t-1$ 时刻的位姿， $x_t = (x' \ y' \ \theta')^T$ 为 t 时刻的位姿。

5.1.4 硬件系统框图

机器人的硬件系统框图如图 5-6 所示,其硬件系统主要由由激光雷达、Kinect、笔记本电脑、底层控制芯片、电机驱动模块、电机和电源模块组成。实验中所使用的笔记本电脑的 CPU 为 Intel Core i7 系列,工作频率达 3.5GHz,同时配有 12G 内存以及 256G 固态存储器,可满足机器人对运算性能的需求。

上层硬件系统:激光雷达传感器收集的环境距离数据同时底层控制芯片收集的里程计数据送给笔记本电脑中基于 KLD 采样的 AMCL 算法处理,其中 AMCL 算法在 Linux 环境下的 ROS 操作系统中实现,距离数据和里程计数据经过 AMCL 算法处理后我们就能够对机器人在二维地图下进行定位,得到不同时刻的位姿。

利用这些位姿,我们可以求解出位姿间的转换矩阵 T 。在不同时刻的位姿处,利用 Kinect 采集深度图像和颜色图像,这些图像被送往笔记本电脑,通过程序被转换成局部三维点云地图。

底层硬件系统:计算机向底层交互控制指令,通过串口与底层芯片进行通信。一方面底层控制芯片将计算机传来的指令转化为直流电机的控制信号,并通过电机驱动模块驱动电机运动;另一方面底层控制芯片将实时采集得到的电机编码器读数转化为机器人里程计数据送往计算机。电源模块负责向 Kinect 和电机驱动模块提供电源。

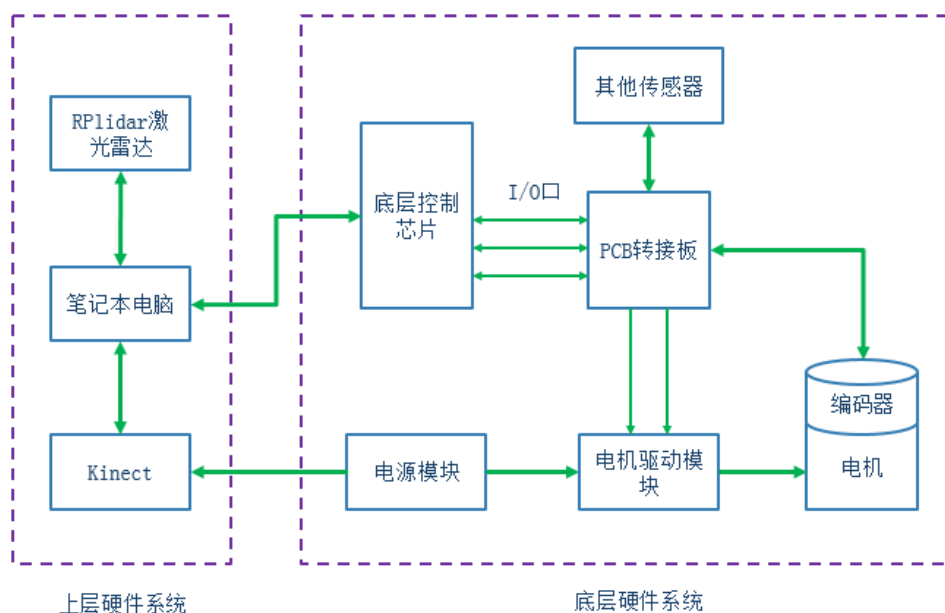


图 5-6 硬件系统框图

5.2 实验软件平台

5.2.1 ROS 操作系统

本文采用的软件系统是 ROS (Robot Operation System) 机器人操作系统。为了解决机器人各种复杂进程的管理问题、推动机器人程序代码的复用与模块化, 机器人软件架构体系 ROS 就此诞生。ROS 是由 Stanford 大学 AI 实验室的 STAIR 项目与美国 Willow Garage 公司共同研发的一款类似于操作系统的专用机器人开源系统, ROS 提供了一个标准的操作系统环境, 包括硬件抽象、底层设备控制、通用功能的实现、进程间消息转发和功能包管理等。

ROS 系统的架构被划分为三个部分: 文件系统级、计算图级、社区级。

(1) 文件系统级

与其他操作系统相似, ROS 中不同的文件要被放在不同的文件夹下, 根据功能的不同, ROS 的文件分为以下几类:

功能包 (Package): Package 是 ROS 软件组织的基本形式, 它包括 ROS 中实现各种功能的节点和配置文件等。

功能包清单 (Package manifest): 用于描述 Package 的许可信息、依赖关系、编译标志等信息。

功能包集 (Stack): Package 的集合, Package 经过组合形成 Stack, Stack 能够实现更加高级的功能, 例如 Navigation Stack。

消息类型 (Message type): Message 的描述, 定义了 ROS 中传送消息和响应的数据结构。

服务类型 (Service type): Service 的描述, 定义了 ROS 中由每个节点提供的关于 Service 请求和响应的数据结构。

(2) 计算图级

如图 5-7 所示, ROS 会建立一个连接到所有进程的网络, 在 ROS 中的所有节点都能访问这个网络, 并通过该网络相互传递信息, 获得其他节点发布的信息, 并将本节点的信息发送到网络上。

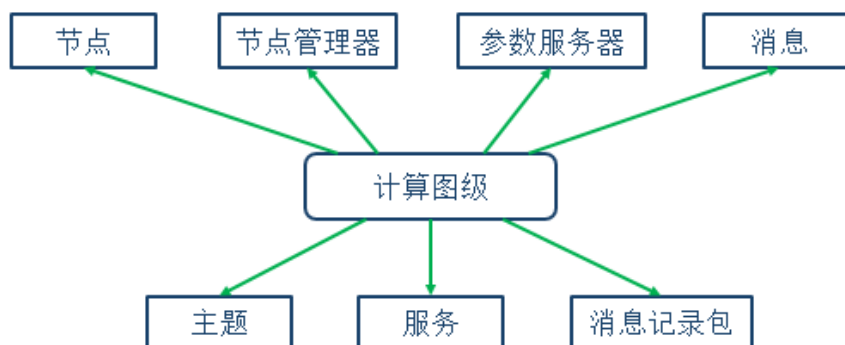


图 5-7 ROS 计算图级

节点 (Node): Node 是主要的计算机执行进程，一个机器人系统由很多节点组成，每一个节点都可以实现不同的功能。

节点管理器 (Master): 利用 Master 可以对 Node 的名称进行注册和查找等。

参数服务器 (Parameter Server): Parameter Server 能够使数据通过关键词存储在系统，通过使用 Parameter，能够在运行时对 Node 进行配置或者改变 Node 的任务。

消息 (Message): 节点间通过消息完成彼此的沟通。

主题 (Topic): ROS 中消息通过 Topic 发送。一个 Node 通过把消息传递到一个给定的 Topic 来发送消息，其他的 Node 可以订阅该 Topic 获取消息。

服务 (Service): Service 能够使 Node 绕开主题，直接与某个 Node 进行交互。

消息记录包 (Bag): Bag 是一种文件格式，它能够保存和回放机器人操作系统 ROS 中的消息数据。

(3) 社区级

ROS 社区级指的主要是 ROS 中的资源，在 ROS 社区中人们可以使用专用的网络社区分享机器人软件代码以及相互交流经验。

5.2.2 实验软件系统流程图

实验软件系统流程如图 5-8 所示。首先，通过游戏手柄远程控制机器人在环境中走动，在移动过程中激光雷达采集环境距离信息同时里程计采集机器人运动距离信息，我们使用 Gmapping 功能包处理这些信息构建出二维地图。然后在已经构建好的二维地图中开启 AMCL 节点，使用游戏手柄远程控制机器人运动，机器人在运动的过程中，通过里程计和激光雷达传感器，不断的获取机器人的运动距离信息和周围环境的距离信息，这些信息被送给基于 KLD 采样的 AMCL 算法处理，经过 AMCL 算法处理后，我们就能对机器人在事先构建好的二维地图中进行定位，得到机器人位姿。

随着机器人不断的运动，我们可以得到机器人在二维地图中不同时刻的位姿，求解这些位姿间的转换关系，我们就能得到转换矩阵。每当机器人在二维地图中通过定位获得一个新位姿，Kinect 就会在该位置拍摄一幅二维颜色图像和一副二维深度图像。得到颜色图像和深度图像之后，使用摄像机模型我们就可以构建一幅局部的三维地图，其中三维地图由点云形式呈现。然后利用转换矩阵对不同时刻的局部点云地图进行逐一的拼接，我们就能得到最终的全局三维地图。

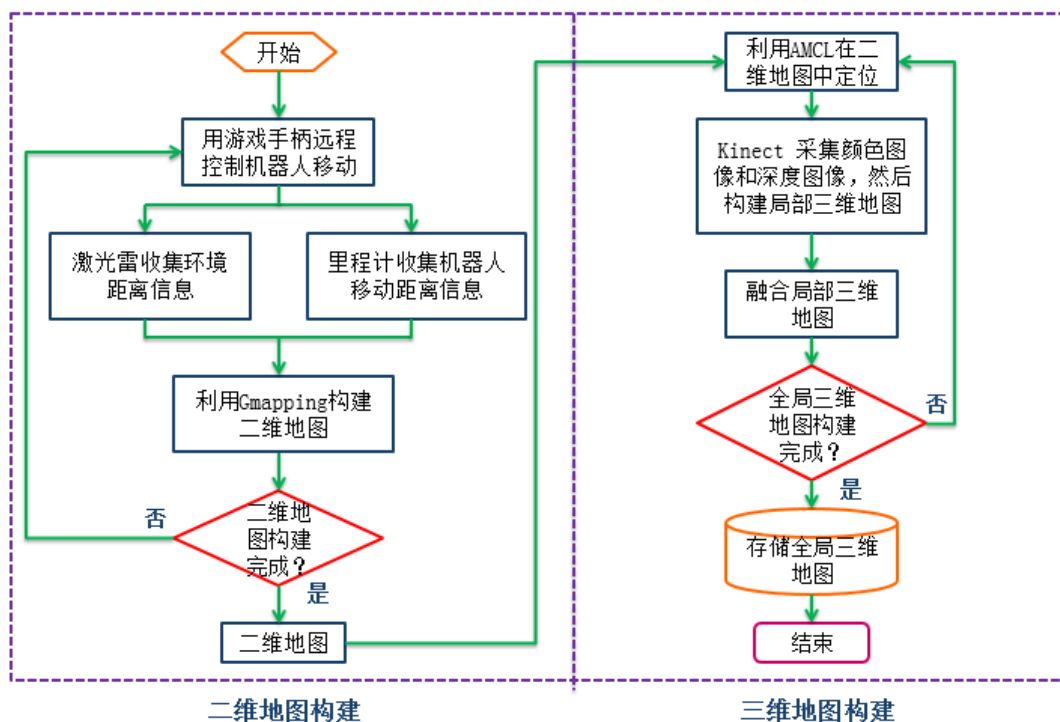


图 5-8 软件系统流程图

5.3 实验结果与分析

为了验证本文方法的有效性和优越性,分别以机器人 TurtlBot2 和自制机器人为移动平台,以 RPLidar 激光雷达和 Kinect 为主要传感器,在一个 30*2.5 米的室内走廊场景下进行实验,该走廊两边为棕色木门和白色墙壁,墙顶部分有白色荧光灯,走廊的尽头是电梯,实验场景如图 5-9 (a) (b) (c) (d) 所示。实验分为四部分,第一部分为不同运动模式下基于视觉定位的 RGB-D 算法和本文所提方法定位性能对比。第二部分为基于二维地图蒙特卡罗定位的三维地图构建。第三部分和第四部分为实验对比。



(a) 走廊整体场景



(b) 走廊一侧棕色木门



(c) 走廊一侧白色墙壁



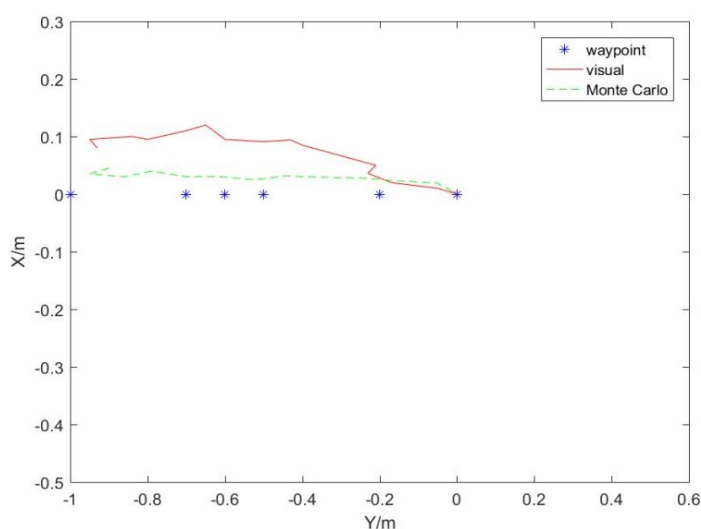
(d) 走廊尽头电梯

图 5-9 实验场景

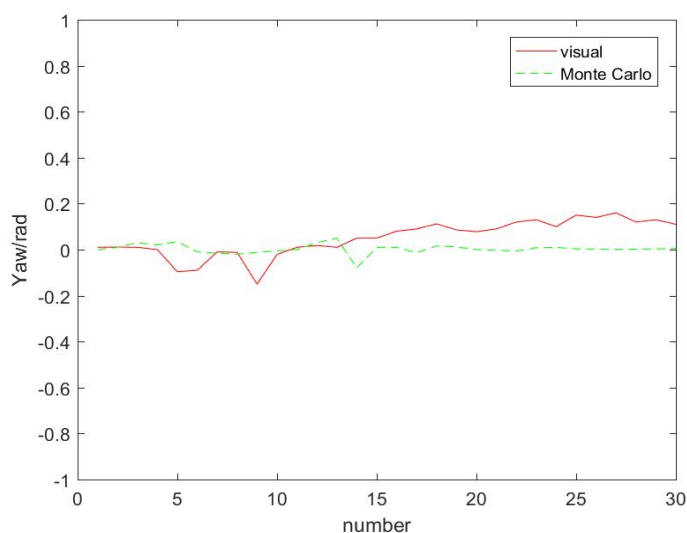
5.3.1 不同运动模式下定位效果对比

为了验证本文所采用方法的性能，进行了机器人直线运动与椭圆运动两组实验，测试系统的定位性能，并与基于视觉定位的 RGB-D SLAM 系统的定位性能做了比较。在实验之前，我们先设置好路径关键点 **waypoint**，然后根据移动机器人的初始位姿，测量出 **waypoint** 的二维坐标，并以这些点为基准，对比定位的效果。

(1) 直线运动



(a) 位移比较

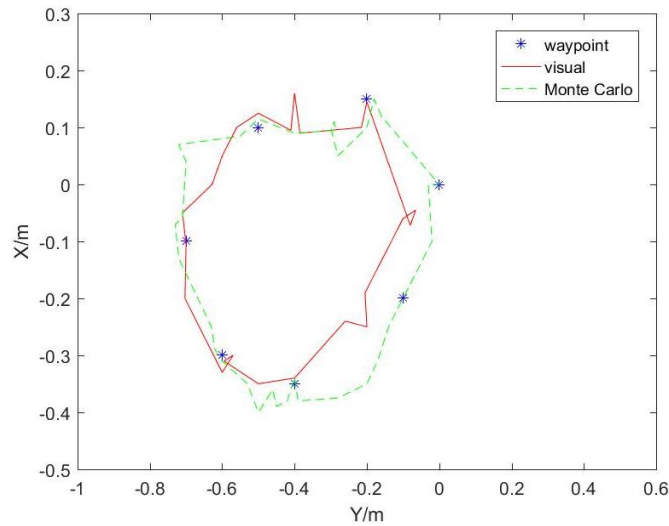


(b) 姿态角比较

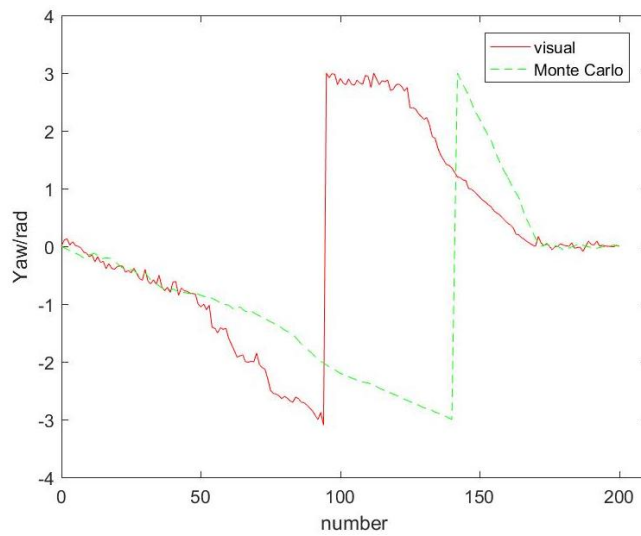
图 5-10 直线运动下 RGB-D SLAM 方法与本文所采用方法定位结果的比较

图 5-10 为直线运动下基于视觉定位的 RGB-D SLAM 与本文所采用方法定位结果的比较, 在该实验中机器人的运动轨迹为沿着 Y 轴运动一段距离。图 (a) 为两种方法的位移比较, 图 (b) 为两种方法的姿态角比较。从图 (a) 中能够看出两种方法都可以追踪机器人的运动轨迹, 但是本文所采用方法在 X 轴方向的偏移量更贴近于 waypoint, 位移误差更小。从 (b) 中可以看出, 机器人在沿着 Y 轴运动过程中, 其方向保持不变, 两种方法估计的姿态角都保持在 0rad 左右, 但是本文所采用方法更贴近于 0rad, 姿态角误差更小。

(2) 椭圆运动



(a) 位移比较



(b) 姿态角比较

图 5-11 椭圆运动下 RGB-D SLAM 方法与本文所采用方法定位结果的比较

图 5-11 为椭圆运动下基于视觉定位的 RGB-D SLAM 与本文所采用方法定位结果比较。在图 (a) 位移比较中机器人的起点为 (0,0) 点, 可以看出两种方法都能够回到起点, 基于视觉定位的方法比本文所采用方法所获得的轨迹更加平滑, 但本文所采用的方法更贴近于 waypoint。在图 (b) 姿态角比较中, 机器人的起始姿态角为 0 度, 随着机器人旋转慢慢运动到 -180 度, 最后到 -360 度, 两种方法都可以追踪机器人角度的改变, 具有收敛性。

5.3.2 使用机器人 TurtbltBot2 基于二维地图蒙特卡罗定位的三维地图构建实验

(1) 二维地图构建

为了构建图 5-9 场景的三维地图, 我们首先要构建其二维地图, 在本文中, 我们选择 Gmapping 来构建二维地图。Gmapping^{[48][49][50]}是一个 ROS 功能包, 它使用高效的 RBPF (Rao-Blackwellized particle filter)^{[51][52]}算法结合激光雷达传感器获取的环境距离信息和机器人的里程计信息来构建二维地图。利用 Gmapping 构建好的二维地图如图 5-12 所示。



图 5-12 利用 Gmapping 所构建的实验场景的二维地图

图 5-12 中黑色的线条和点代表环境中的障碍物, 例如走廊环境下的墙和门, 这些地方对于机器人来说都是不可行区域, 机器人要避免与发生碰撞, 白色和灰色的地方分别表示可行区域和未知区域。

(2) 利用 AMCL 算法在二维地图中对机器人进行定位

二维地图构建完成后, 我们就可以利用 AMCL 算法在二维地图中对机器人进行定位。图 5-13 展示了机器人 TurtlBot2 初始定位的情况。开始时, 机器人 TurtlBot2 不知道它在环境中所处的位置, 因此机器人在二维地图中任何一个位置出现的概

率是相同的。

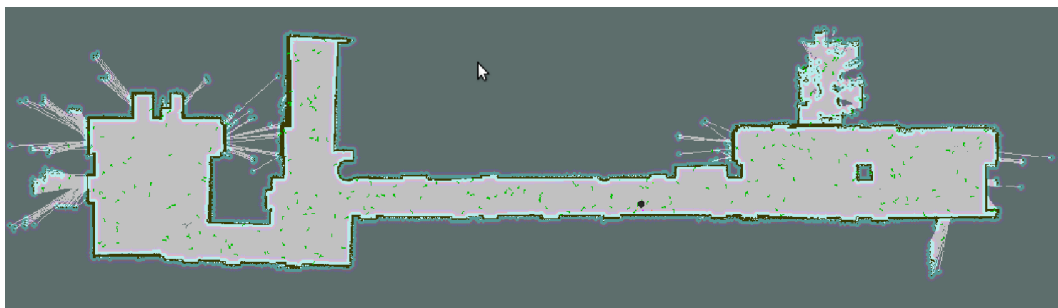


图 5-13 机器人在二维地图下的初始定位情况

图 5-13 中，散乱分布在地图中带有箭头的绿色点代表粒子，每一个粒子的位置都表示机器人 TurtlBot2 在二维地图中可能出现的位置。图中部黑色的大圆点代表机器人模型，它的位置代表定位结果，该定位结果由 AMCL 算法处理过的粒子估计得到。经过几次运动之后，带有箭头绿色粒子的位置开始聚集，这表示机器人 TurtlBot2 利用 AMCL 算法正在逐渐确定其在二维地图中所处的位置。图 5-14 显示了定位结果，从该图中可以看出绿色的粒子聚集在估计位置的附近。

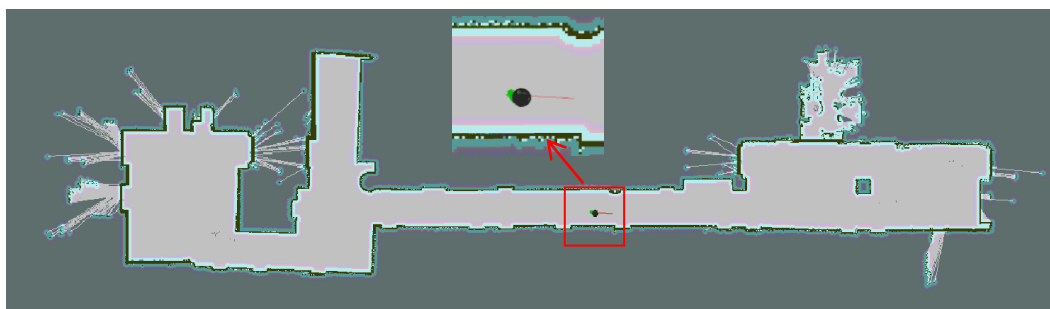


图 5-14 几次移动之后机器人的定位情况

(3) 三维地图构建

三维室内地图的实景为图 5-9 中所示的走廊，该走廊长约 30 米，宽约 2.5 米。在三维地图构建实验中，在 Lniux 操作系统下运行本文所提的方法。我们利用游戏手柄来远程控制机器人 TurtlBot2，让机器人从走廊一端开始运动，然后沿着走廊的中轴线向前移动，直到抵达走廊另一端的电梯。每当机器人 TurtleBot2 移动 0.2 米，它就会向 ROS 主题发布一次位姿信息。与此同时，在该位置的局部三维地图会被构建，利用转换矩阵 T 将该局部三维地图将与上一个位置获取的局部三维地图进行融合。所构建的全局三维室内走廊地图如图 5-15 所示，从该图中我们可以清晰的辨别出白色的墙体、灰色的电梯、走廊两边的棕色的木门和墙顶的白色荧光灯，实验结果表明本文所提的方法能够较好的构建出走廊的三维环境地图。

5.3.3 实验对比一基于 RGB-D SLAM 算法的三维地图构建实验

为了验证本文方法的有效性和优越性,我们与 Felix Endres^[23]等人提出的基于视觉定位的 RGB-D SLAM 方法做了实验对比。在对比实验中,我们在 Linux 操作系统下运行 RGB-D SLAM 程序。我们使用游戏手柄来远程控制机器人,由于误差的存在,两次实验中机器人的运动轨迹不可能完全相同,所以机器人的运动轨迹只能与上一个实验中的轨迹相接近。

对比实验的三维室内地图结果如图 5-16 所示。在图 5-16 中部,黄色线条的交点代表机器人 t 时刻的位姿 x_t ,黄色的线条代表这些位姿之间的转换关系。这些位姿和位姿之间的转换关系用 RGB-D SLAM 后端的通用图优化方法 (General Graph Optimization, g2o) 不停的进行优化。

由 RGB-D SLAM 系统构建的三维地图如图 5-16 所示。由于实验场景缺乏特征,主要由白墙和棕色的门构成,这导致基于视觉定位的 RGB-D SLAM 算法在视觉定位过程中产生许多由图像特征点误匹配引起的累积误差。随着机器人移动不断增加,这些误差累积值越来越大,即使后端有 g2o 优化也无法减少这些误差,这导致最终构建出来的室内三维走廊地图效果比较差。

5.3.4 实验对比二使用自制机器人基于二维地图蒙特卡罗定位的三维地图构建实验

为了进一步验证本文所提方法的有效性,我们在团队自制的机器人上也进行了实验,自制机器人如图 5-17 所示。该自制机器人的硬件系统与 TurtleBot2 类似,主要区别在底层控制芯片的不同,自制机器人选择 STM32 作为其底层控制芯片。

在此实验中,我们在同样在 Linux 操作系统下运行基于二维地图蒙特卡罗定位的三维地图构建程序。我们使用游戏手柄来远程控制机器人,由于误差的存在,此实验中机器人的运动轨迹不可能与上述两个实验完全相同,所以机器人的运动轨迹只能与之前实验中的轨迹相接近。实验步骤与 5.3.2 中类似,首先构建二维地图,然后利用基于 KLD 采样的 AMCL 算法对机器人在二维地图中进行定位,最后使用基于二维地图蒙特卡罗定位的三维地图构建方法构建走廊的三维地图。



图 5-17 自制移动机器人

图 5-18 中的实验结果表明，使用自制机器人构建的走廊三维地图出现明显的走样，而用 TurtleBot2 构建的走廊的三维地图的形状与实际的走廊基本符合。导致该情况的原因主要有两点：

(1) 一方面自制移动机器人由于其机械结构不精确产生的机械误差，尤其是自制机器人的车轮的半径和车轮之间的轮基距不一致以及车轮位置安放不对称引起的噪声，这些噪声会使各个车轮产生速度差，导致机器人直线行走时，实际里程计的轨迹偏向一边。而基于 KLD 采样的 AMCL 算法中，机器人运动模型所使用的高斯噪声其均值为零，因此，在这种情况下，使用自制机器人不能保证构建出良好的三维走廊地图。

(2) 另一方面由于自制机器人控制系统不够稳定，实验中控制机器人移动时，机器人有时会突然的加速、突然的减速或者快速旋转某个一较小的角度，这些运动会使 AMCL 算法定位的位置产生跳跃，这些不规则的运动就会导致图 5-18 中的三维地图中重影和不平整部分的产生。

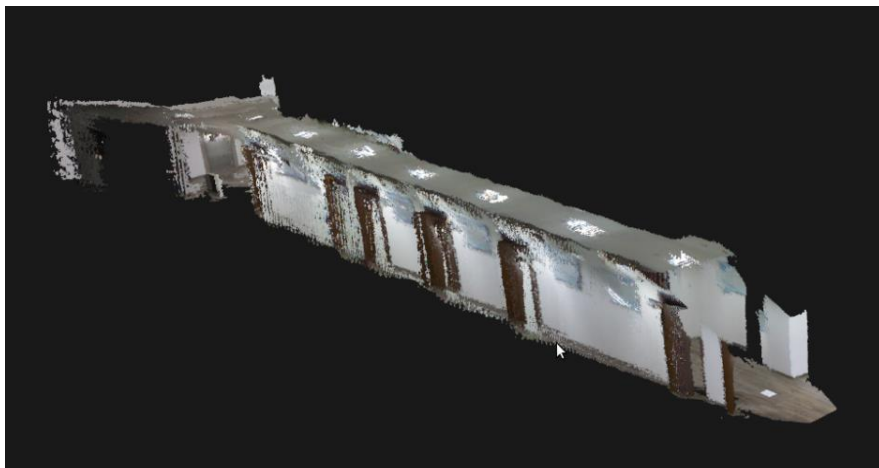


图 5-15 使用本文方法利用 TurtleBot2 构建的室内走廊三维地图

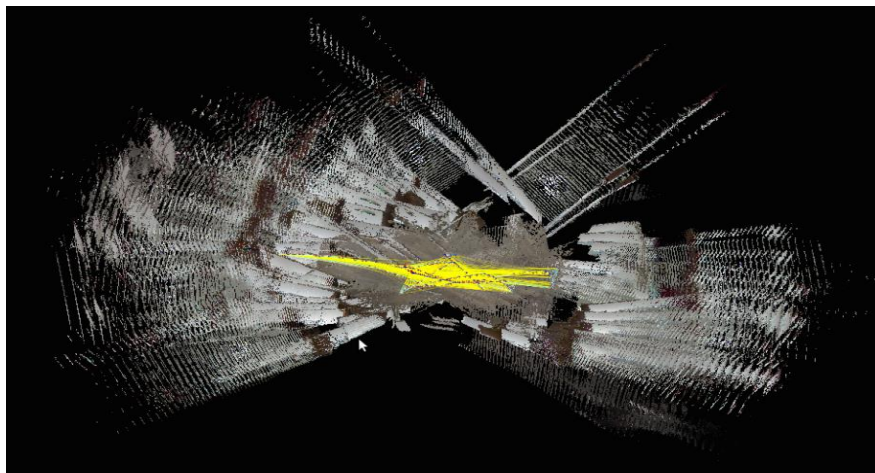


图 5-16 利用 RGB-D SLAM 构建的室内走廊三维地图

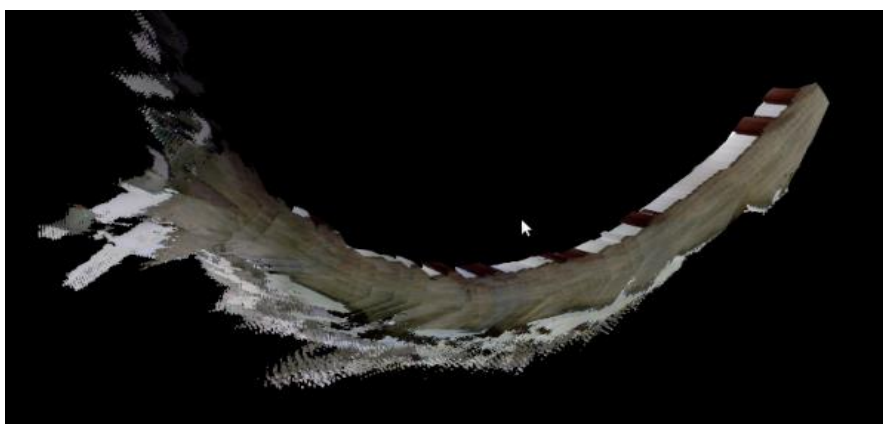


图 5-18 使用本文方法利用自制机器人构建的室内走廊三维地图

5.4 本章小结

本章介绍了实验所用的硬件平台和软件平台，然后分析对比了基于二维地图蒙特卡罗定位的三维室内地图构建和基于 RGB-D SLAM 算法的三维室内地图构建的实验结果。

第六章 总结与展望

6.1 总结

本文针对 RGB-D SLAM 算法在基于视觉定位的三维地图构建时存在的误匹配与定位误差难以修正的问题，提出一种基于二维地图蒙特卡罗定位的三维地图构建方法，并搭建了实际的硬件平台对算法进行验证与分析。本论文的工作总结为以下几点：

(1) 介绍了能够同时获取颜色和深度信息的传感器 Kinect，研究其工作原理、怎样获取点云以及摄像机模型，深入研究了 RGB-D SLAM 的算法架构并对该架构中的各组成部分的具体实现进行了阐述。

(2) 研究了贝叶斯理论，介绍了蒙特卡罗算法思想，详细讨论了其算法流程和步骤。

(3) 针对基于视觉定位的 RGB-D SLAM 过程中累积误差导致机器人错误定位的问题，提出了一种构建室内三维地图的方法：利用二维地图下基于激光雷达的蒙特卡罗定位解决定位问题，同时结合能够获取颜色图像和深度图像的 Kinect 传感器采集周围环境信息来构建三维地图。

(4) 搭建本实验所用的机器人硬件平台和软件平台，实现了对基于二维地图下蒙特卡罗定位三维地图构建方法的实验设计和验证，通过实验证明所提方法能在一定程度上提高系统的定位准确性，同时对特定场景下原始算法定位效果很差的问题有所改善。

6.2 展望

本文提出的方法虽然在一定程度上提高了三维室内地图构建的质量，但仍存在一些不足之处，需要进一步的改进与完善：

(1) 机械部分是影响机器人运动精度的重要因素，理想模型与机械结构存在的差异必然导致机器人定位精度的误差，所以在下一步研究中，要将机器人的机械误差考虑在内，这样对机器人定位精度有一定帮助。

(2) 本文中所提方法的验证是在相对理想的环境下进行的，没有考虑其他复杂环境因素，例如，有时移动机器人需要高速运动、地面不平整等情况，同

时算法的稳定性和自适应性方面考虑的比较少，所以算法的实用性是下一步研究的重点。

（3）在实际实验中，机器人携带的里程计的数据有时会有比较大的飘移，所以在下一步研究中，考虑融合惯性传感器的信息来为机器人提供更加准确的定位。

参考文献

- [1] 张亮. 移动机器人同步定位与地图重建算法研究[D] 浙江大学博士学位论. 2009, 1-4.
- [2] 李人厚, 宋青松. 自主移动机器人导论[M]. 西安: 西安交通大学出版社. 2013.
- [3] S. Thrun, J. J .Leonard, “Simultaneous localization and mapping,” in Springer handbook of robotics, pp.871-889, Springer Berlin Heidelberg, 2008.
- [4] H. Durrant-Whyte. Where am I? A tutorial on mobile vehicle localization[J]. Industrial Robot: An International Journal, 1994, 21(2), 11-16.
- [5] S. Thrun. Robotic mapping: A survey[J]. Exploring artificial intelligence in the new millennium. 2002, 1-35.
- [6] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments[C]. I. J.Robotic Res. 31(5): 647-663, (2012).
- [7] C. Audras, A. I. Comport, M. Meilland, and P. Rives. Real-time dense appearance-based SLAM for RGB-D sensors[C]. In Proc. Australian Conf. Robotics and Automation, Dec. 2011.
- [8] F. Endres, J.Hess, D. Cremers et al. An Evaluation of the RGB-D SLAM System[C].In Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA), 2012.
- [9] J. Stuckler and S. Behnke. Model learning and real-time tracking using multi-resolution surfel maps[C]. in AAAI Conf. on Artificial Intelligence (AAAI), 2012.
- [10] 辛菁, 苟蛟龙, 马晓敏等. 基于 Kinect 的移动化器人大视角 3 维 V-SLAM[J]. 机器人, 2014, 36 (5) : 560-568.
- [11] Calonder M, Lepetit V, Strecha C, et al. BRIEF: binary robust independent elementary features[C]. Proceedings of the 11th European conference on Computer vision, Springer-Verlag, 2010:778-792.
- [12] Rosten E, Porter R, Drummond T. Faster and better: a machine learning approach to corner detection[J]. IEEE Transactions on Software Engineering, 2010, 32(32):105-19.
- [13] E.Rublee, V.Rabaud, K,Konolige, et al. ORB: an efficient alternative to SIFT or SURF[C]. IEEE International Conference on Computer Vision, 2011: 2564-2571.
- [14] 杨鸿. 基于深度视觉的服务机器人感知技术与应用研究[D]: [硕士学位论文]. 南

- 京：东南大学, 2014.
- [15] 刘艳丽. 融合颜色和深度信息的三维同步定位与地图构建研究[D]: [博士学位论文]. 长沙：中南大学, 2014.
- [16] J. Smisek, M. Jancosek and T.Pajdla. 3D with Kinect [C]. IEEE International Conference on Computer Vision Workshops (ICCV Workshops), November 2011, pp:1154-1160.
- [17] K. Khoshelham and S. O. Elberink. Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications [J]. Sensors: Journal on the Science and Technology of Sensors and Biosensors, 2012, 12: 1437-1454.
- [18] Prime Sense. <http://www.primesense.com>[EB/OL]
- [19] 余涛应用开发实战：用最自然的方式与机器对话机械工业出版社, 2012.
- [20] 杨杰. 融合 IMU 和 Kinect 的机器人定位算法研究[D]: [硕士学位论文]. 武汉：武汉科技大学, 2014.
- [21] Z. Zhang, "A flexible new technique for camera calibration," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 11, pp. 1330-1334, 2000.
- [22] R. Kümmerle, G. Grisetti, H. Strasdat, et al., "g 2 o: A general framework for graph optimization," IEEE International Conference on Robotics and Automation (ICRA), pp. 3607-3613, 2011.
- [23] F. Endres, J. Hess, J. Sturm, et al., "3-D mapping with an RGB-D camera," IEEE Transactions on Robotics, vol. 30, no. 1, pp.177-187, 2014.
- [24] C. Harris and M. Stephens. A combined corner and edge detector[C]. In Alvey vision conference, pages 147–151, 1988.
- [25] K. Konolige and M. Agrawal. Frame SLAM: From bundle adjustment to real-time visual mapping[C]. IEEE Transactions on Robotics, 25(5), 2008.
- [26] Juan and O. Gwun. A comparison of sift, pca-sift and surf[C]. International Journal of Image Processing, 2009.3(4):143–152.
- [27] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features[C].Computer Vision ECCV 2006: 404–417.
- [28] 赵毅力, 徐丹, 潘志庚. 基于 SURF 特征点的图像配准系统[J]. 计算机应用. Vol, 31, Suppl.1, June 2011.
- [29] Martin A. Fischler and Robert C. Bolles, Random Sample Consensus: A Paradigm for Model

- Fitting with Applications to Image Analysis and Automated Catography[J].Graphics and Image Processing, 1981:381-395.
- [30] A. Johnson and S. B. Kang. Registration and integration of textured 3-d data[C]. In International Conference on Recent Advances in 3-D Digital Imaging and Modeling (3DIM '97), pages 234-241, May 1997.
- [31] K. M. Wurm, A. Hornung, M. Bennewitz et al. Octomap :A probabilistic, flexible, and compact 3d map representation for robotic systems[C]. In Proc. of the ICRA 2010 workshop on best practice in 3D perception and modeling for mobile manipulation, volume 2, 2010. Cited on pages 3 and 75.
- [32] Ho Y.C, Lee R.C.K. A Bayesian approach to problems in stochastic estimation and control[C]. IEEE Trans. Automat. Contr.,1964, 9(4): 333-339.
- [33] Russell S.J, Norvig P. Artificial Intelligence: A Modern Approach[M]. 2nd ed. Prentice Hall, 2003.
- [34] 徐钟济. 蒙特卡罗方法[M]. 上海科学技术出版社, 上海, 1985.
- [35] F. Dellaert, W. Burgard, D. Fox, et al., "Monte Carlo localization for mobile robots," Proceedings of the IEEE International Conference on Robotics and Automation, pp.1322-1328, Detroit, USA, 1999.
- [36] 张国良, 汤文俊, 敬斌等. 基于线段特征匹配的 EKF-SLAM 算法[J]. 控制工程, 2012, 19(006): 1019-1024.
- [37] 安雷, 张国良, 张维平等. 移动机器人扩展卡尔曼滤波定位与传感器误差建模[J]. 信息与控制, 2012, 41(4): 406-412.
- [38] 李海洋, 李洪波, 林颖等. 基于 Kinect 的 SLAM 方法[C] 2013 年中国智能自动化学术会议论文集(第五分册). 2013.
- [39] 柯翔, 陈小平, 靳国强等. 一种适用于室内服务机器人的实时物体识别系统[J]. 2013.
- [40] S. Thrun, W. Burgard, D. Fox, "Probabistic Robotics," The MIT Press, September, 2005.
- [41] J. M. Joyce, "Kullback-leibler divergence," International Encyclopedia of Statistical Science, pp.720-722, Springer Berlin Heidelberg, 2011.
- [42] RoboCup 人形足球机器人的蒙特卡罗自主定位算法研究[D]: [硕士学位论文]. 吉林: 吉林大学, 2012.
- [43] http://docs.pointclouds.org/trunk/group__registration.html

- [44] <http://www.turtlebot.com>
- [45] <http://www.ros.org>
- [46] R. A. El-Laithy, Jidong Huang, M. Yeh. Study on the use of Microsoft Kinect for robotics applications[C]. Position Location and Navigation Symposium(PLANS), 2012 IEEE/ION. IEEE, 2012: 1280-1288.
- [47] 朱涛, 金国栋, 芦利斌. Kinect 应用概述及发展前景[J].现代计算机, 2013. 02 下:8-11.
- [48] G. Grisetti, C. Stachniss, W. Burgard, “Improving grid-based SLAM with Rao-Blackwellized particle filters by adaptive proposals and selective resampling,” in Proceedings of the 2005 IEEE International Conference on Robotics and Automation , pp. 2443–2448, Barcelona, Spain, 2005.
- [49] G. Grisetti, C. Stachniss, W. Burgard, “Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters,” IEEE Transactions on Robotics, vol. 23, no. 1, pp. 34-46, 2007.
- [50] <http://www.ros.org/gmapping>
- [51] Murphy, P. Kevin, “Bayesian map learning in dynamic environments,” in Proc. Conf. Neural Inf. Process. Syst, pp.1015–1021, 1999.
- [52] A. Doucet, J. de Freitas, K. Murphy, S. Russel, “Rao-Blackwellized particle filtering for dynamic Bayesian networks,” in Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence, pp. 176–183, 2000.

致 谢

光阴如梭，时光荏苒，又到了一年木棉盛开之际，今年已迎来我在汕头大学的第三个年头。校门前的凤凰花即将开放，预示着我的硕士学习生涯即将结束，即将要告别这敬爱的母校以及亲爱的老师和同学们，每念及此，心里总是充满着无数的感恩之情。

难忘母校的培育之恩，汕大坚持“有志、有识、有恒、有为”的育人目标，注重对学生个人能力的培养，为莘莘学子提供了广阔的平台。丰富的图书馆馆藏资源让同学们沉浸在知识的海洋中，优秀的师资为学子“传道、授业、解惑”，还有安静优美的校园环境总是让人流连忘返。

难忘老师的栽培之恩，特别感谢我的导师范衡教授，是他引领我走向了学术的殿堂，接受更多的专业知识。在范衡老师的谆谆教诲之下，我丰富了知识，深邃了思想，开阔了视野。范衡老师不仅是我学术上的导师，更是我人生的导师，他那严谨踏实的治学态度、求真务实的工作作风、顶天立地的学术思想及宽容大度的长者风范在我心中留下了深刻的印象，成为了激励我今后人生前进的动力。

感谢与我相伴的研究小组成员，他们对研究充满了热情，在研究上兢兢业业、勤勤恳恳地钻研，在每周的小组讨论，他们让我学到了更多的知识，拓宽了我的思维，并形成了团结合作的良好学风，更在研究上给我极大的灵感与帮助，在此我衷心地表示感谢！

同样要感谢的是电子系的全体同学，在这个大家庭里，大家互帮互助，相处非常地融洽，为我的生活充满了欢乐。还记得初来乍到懵懂无知，对研究不知所措时，是上一届的师兄给予我开导的话语，教给我做研究的经验，使我才得以走出初时的迷茫。如今师兄师姐们已经步入社会，祝你们工作顺利，事业有成！还有那一群同样面临毕业的同班同学们，我们在一起相处三年，互相帮助、互相陪伴，结下了深厚的友谊。如今就要分开，却留下了珍贵的回忆，让我怎能不感谢你们的相伴！

最后，深深感谢父母的养育之恩，谢谢你们含辛茹苦把我养大，让我接受教育。虽然你们不了解我的学术研究，但当我遇到困难向你们倾诉时，你们总是给我极大的鼓舞，并在生活上给予我无私的帮助。

我怀揣着深深的感激之情，感谢所有关心与帮助过我的人，谢谢你们！

赵 雷

2017 年 5 月 汕头

攻读硕士学位期间主要的工作成果

1、 Lei Zhao, Zhun Fan, Wenji Li, Honghui Xie, Yang Xiao. 3D Indoor Map Building with Monte Carlo Localization in 2D Map . Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII), 2016 International Conference on. IEEE.(EI accepted)

2、 Yang Xiao, Zhun Fan, Wenji Li, Shen Chen, Lei Zhao, Honghui Xie. A Manipulator Design Optimization Based on Constrained Multi-objective Evolutionary Algorithms. Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII), 2016 International Conference on. IEEE.(EI accepted)

3、 Honghui Xie, Zhun Fan, Wenji Li, Yibiao Rong, Yang Xiao, Lei Zhao. Tobacco Plant Recognizing and Counting Based on SVM. Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII), 2016 International Conference on. IEEE.(EI accepted)