

电子科技大学
UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

硕士学位论文

MASTER THESIS



论文题目 基于多传感器融合的移动机器人

SLAM 算法的研究与应用

学科专业 计算机应用技术

学 号 201521060426

作者姓名 潘志国

指导教师 罗光春 教授

分类号 _____

密级 _____

UDC ^{注1} _____

学 位 论 文

基于多传感器融合的移动机器人

SLAM 算法的研究与应用

(题名和副题名)

潘志国

(作者姓名)

指导教师

罗光春

教 授

电子科技大学

成 都

段琢华

教 授

电子科大中山学院

中 山

(姓名、职称、单位名称)

申请学位级别

硕士

学科专业

计算机应用技术

提交论文日期

2018.3.23

论文答辩日期

2018.5.18

学位授予单位和日期

电子科技大学

2018.6.28

答辩委员会主席

评阅人

注 1：注明《国际十进分类法 UDC》的类号。

Research and Application of SLAM Algorithm Based on Multi Sensor Fusion

A Master Thesis Submitted to

University of Electronic Science and Technology of China

Disicipline: **Computer Application Technology**

Author: **Zhiguo Pan**

Advisor: **Luo Guangchun**

School : **School of Computer Science & Engineering**

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

作者签名： 潘志国

日期： 2018 年 6 月 22 日

关于论文使用授权的说明

本学位论文作者完全了解电子科技大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权电子科技大学可以将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、汇编学位论文。

（保密的学位论文在解密后应遵守此规定）

作者签名： 潘志国

导师签名： 罗光春

日期： 2018 年 6 月 22 日

摘 要

随着传感器技术和人工智能技术的发展, 智能移动机器人逐渐进入到使用阶段, 代替人类进行重复劳动以及危险工作。而在机器人众多研究方向中有关移动机器人同时定位与建图(Simultaneous Localization And Mapping, SLAM)问题的解决是移动机器人实现完全自主工作的重要一环。SLAM 主要应用场景是在陌生环境中, 机器人最初没有周围环境的信息, 这时就需要机器人通过安装在本体上的传感器获得周围环境和自身状态的信息, 然后对信息进行分析加工, 从而得到陌生环境的地图以及自己所在地图的位置。但是单一的传感器误差较大, 由此本文进行了基于多传感器融合的 SLAM 算法的研究与应用。双目相机可以通过立体匹配恢复出空间路标点的深度信息, 相比于激光等其它传感器, 鲁棒性好获取信息更加丰富。惯性测量单元(Inertial Measurement Unit, IMU)在较短时间内精度较高。通过结合 IMU 和双目相机的优点, 本文提出了一种融合 IMU 和双目相机的改进 FastSLAM 算法。

首先, 研究了双目立体视觉工作原理。通过实验分析比较了立体匹配常用的特征算法, 决定选用 ORB(Oriented FAST and Rotated BRIEF)算法。本文对于 ORB 算法的提取和误匹配筛选做了改进。针对光线不好时图像噪声较多影响特征点的采集和匹配的问题, 在采集特征点之前先通过对比度被限制的直方图均衡化方法对图像进行预处理, 减少噪声的影响。针对 SLAM 系统对于特征点匹配精度要求较高的问题, 通过结合汉明距离设定阈值方法, 随机采样一致性算法以及匹配对间距离比和角度比误匹配筛选算法进行误匹配剔除, 明显减少了误匹配的数量。

然后, 分析了 SLAM 算法的运动过程模型和观测模型。针对现有的基于粒子群改进的 FastSLAM 算法可能会使采样粒子丧失多样性, 使粒子向局部最优解聚集的问题, 在已有的粒子群 FastSLAM 算法的基础上做出改进。通过引入免疫算法增加粒子的多样性, 改善了 FastSLAM 在迭代过程中粒子退化问题和由于多次重采样造成的粒子耗尽问题。

最后, 在深入研究改进的 SLAM 算法的基础上, 实现了双目相机和 IMU 融合的 SLAM 系统。该系统是在机器人操作系统(Robot Operating System, ROS)下开发的, 并且进行了系统测试, 对于改进的 FastSLAM 算法的有效性进行了验证。

关键词: ORB 算法, 多传感器融合, 免疫算法, 粒子群算法, FastSLAM

ABSTRACT

With the development of sensor technology and artificial intelligence technology, intelligent mobile Robots gradually replace humans for repeated work and dangerous work. In many research fields of robots, the problem of Simultaneous Localization And Mapping (SLAM) is the most important part of mobile robot's complete autonomous work. The SLAM problem can be described as: In an unknown environment, the mobile robot determines the environmental map and its position in the map by the sensors it carries. However, the single sensor has a large error. This thesis implements SLAM algorithm based on multi-sensor fusion. The binocular camera can recover the depth information of the spatial landmarks through stereo matching. Compared with other sensors such as laser, the binocular camera has better robustness and richer information. The inertial measurement unit (Inertial Measurement Unit, IMU) has a higher accuracy in a shorter time. By combining the advantages of IMU and binocular camera, an improved FastSLAM algorithm for fusion of IMU and binocular cameras is proposed in this thesis.

Firstly, This thesis studies the principle of binocular stereovision. This thesis compares the commonly used feature algorithms of stereo matching and finally the ORB (Oriented FAST and Rotated Brief) algorithm is chosen as the feature point extraction algorithm. This thesis improves the extraction of ORB algorithm and reduces false matching. This thesis improves the extraction of ORB algorithm and reduces false matching. Before the feature points are acquired, the image is preprocessed by the histogram equalization method with limited contrast, which reduces the influence of noise. This paper uses the Hamming distance setting threshold method, Random Sample Consensus (RANSAC) and matching pair distance ratio and angle ratio mis-matching screening algorithm to eliminate false matching, which obviously reduces the number of mis-match.

Secondly, the motion process model and the observation model of the SLAM algorithm are analyzed. The FastSLAM algorithm based on particle swarm optimization may make the sampling particles lose diversity and make the particle swarm to the local optimal solution. So this thesis makes improvements based on existing particle swarm FastSLAM algorithm. By introducing immune algorithm to increase mutation and

multiplication of particles and increase particle diversity, the problem of particle degradation during FastSLAM iteration and particle exhaustion caused by multiple resampling is improved.

Finally, based on the improved FastSLAM optimization algorithm, the SLAM system with binocular camera and IMU is realized in this thesis. And this system was developed under the Robot Operating System (ROS). This system was tested in a real-world scenario and the effectiveness of the improved FastSLAM algorithm was verified.

Keywords: ORB, Multi-sensor fusion, Immune algorithm, Particle Swarm Optimization, FastSLAM

目 录

第一章 绪论	1
1.1 研究背景及意义	1
1.2 国内外研究现状	2
1.3 研究内容	4
1.4 论文结构	5
第二章 相关理论与技术	6
2.1 SLAM 算法解决的问题	6
2.2 SLAM 算法的难点	7
2.3 SLAM 算法分类	8
2.3.1 概率化 SLAM 问题	9
2.3.2 基于扩展卡尔曼滤波的 SLAM 算法	9
2.3.3 基于粒子滤波的 SLAM 算法	11
2.4 SLAM 坐标系统	12
2.5 惯性导航系统	13
2.5.1 IMU 模型建立	14
2.5.2 IMU 姿态解算	15
2.5.3 位移计算	16
2.6 粒子优化算法	17
2.6.1 粒子群优化算法	17
2.6.2 免疫算法	18
2.7 多传感器融合的 SLAM 系统	19
2.8 本章小结	20
第三章 双目立体匹配关键技术的研究	21
3.1 引言	21
3.2 双目立体视觉	22
3.2.1 双目立体视觉模型	22
3.2.2 双目相机标定	24
3.3 图像特征提取算法和图像增强算法	27
3.3.1 ORB 算法	27
3.3.2 图像增强算法	29

3.4 改进的 ORB 算法	30
3.4.1 优化 ORB 算法的提取过程	30
3.4.2 优化 ORB 匹配的筛选过程	30
3.4.3 改进后的 ORB 算法流程	33
3.5 实验结果及分析	34
3.5.1 多种图像特征提取分析	35
3.5.2 多种图像特征匹配分析	37
3.5.3 改进的误匹配去除算法	40
3.6 本章小结	42
第四章 融合 IMU 和双目相机的 SLAM 算法的研究	44
4.1 引言	44
4.2 FastSLAM 算法概述	45
4.2.1 FastSLAM 算法分析	45
4.2.2 FastSLAM 算法的步骤	45
4.3 基于免疫算法和粒子群算法融合优化的 FastSLAM 算法	47
4.3.1 建立 IA-PSO-FASTSLAM 系统模型	47
4.3.2 融合粒子群算法和免疫算法优化粒子集	50
4.3.3 设定优化算法的阈值	53
4.3.4 IA-PSO-FastSLAM 算法流程	54
4.4 仿真结果及分析	56
4.5 本章小结	60
第五章 SLAM 系统搭建与实验	61
5.1 系统概述	61
5.2 实验环境介绍	61
5.2.1 ROS 系统简介	61
5.2.2 传感器介绍	62
5.3 系统详细设计	63
5.4 实验结果	64
5.4.1 基于数据集的实验	64
5.4.2 基于现实场景的实验	66
5.5 本章小结	69
第六章 总结与展望	70
6.1 总结	70

6.2 展望	70
致 谢	72
参考文献	73
攻硕期间取得的成果	78

第一章 绪论

1.1 研究背景及意义

随着计算机和人工智能等学科的发展，移动机器人相关技术得到了突飞猛进的发展，是目前科学技术最前沿的一个领域。

移动机器人应用场景非常广阔，主要包括卫生清洁，医疗辅助，安保消防等领域。而为了完成这些任务，就需要移动机器人具有感知周围环境信息和自身状态的能力。正如 Durrant-Whyte^[1]在论文所提到的，在陌生环境中，移动机器人在运行过程中需要面对三个问题。第一个是“我在哪里呢？”，这个问题的解决需要建立地图和获取机器人在地图中的位置。第二个是“我要去那里呢？”，这个问题是用户给机器人安排的任务所决定的。第三个是“我要怎么到那里去呢？”，这个问题需要通过路径规划的方法解决。第一个问题是解决后两个问题的前提。机器人确定了自身在地图中的位置，才能确定目的地以及如何前往。在已知环境下，机器人可以通过对路标点的感知，纠正自身的位姿信息。然而在周围环境信息未知时，移动机器人需要在没有先验信息的情况下，在自身运动的过程中一边建立周围环境的模型即建图，一边估计自身的运动状态即定位。在这个过程中机器人的建图和定位的精度是相互影响的，这就引出了机器人领域的热门研究同时定位与建图问题^[2]。

在陌生环境下，机器人通过本体携带的传感器获取信息实现同时定位与建图。对于机器人来说主要有内外两种传感器。内部传感器主要用来感知移动机器人自身的运动信息，主要包括 IMU、里程计和 GPS 等。外部传感器主要用来感知移动机器人运动过程中周边环境的信息，包括相机，激光雷达，超声波雷达等。

实现移动机器人的同时定位与建图，依赖于机器人自身的传感器对于外部环境的感知，但是由于激光雷达成本较高，体积较大和功耗大，很难推广普及。而通过相机来实现定位，相对来说安装更为方便，体积小成本低。现在主流使用的相机有单目相机，双目相机和深度相机。由于单目相机无法获取机器人运动的真实尺度信息^[3]，而深度相机在室外环境容易受到红外线的影响^[4]，所以双目相机相对其他两种相机即可以恢复出深度信息并且受到环境影响较小，作为优先考虑。在城市环境中 GPS 信号存在遮挡，缺失的问题，而 IMU 不存在该问题，并且实时性较好，短期内精度较高。是以进行结合双目相机和惯性测量单元的 SLAM 算法的研究。该融合算法有较高的研究价值和现实意义。

1.2 国内外研究现状

Smith 和 Cheesman^[5]在 1986 年的 ICRA 会议上首次提出用概率学的状态估计方法解决 SLAM 问题，用概率分布表示机器人运动的噪声和姿态，在这以后基于概率学的状态估计方法得到了快速发展。

SLAM 研究的是机器人在陌生环境下通过本体携带的传感器构建地图，同时确定自己所在地图的位置的问题。SLAM 系统主要有三个组成部分，并且是一个迭代的过程。首先通过当前机器人的位姿信息和运动方程估计下一时刻机器人的位姿信息。然后通过相机或者激光雷达等传感器获取路标信息，并且把路标信息和机器人的位姿信息相关联。等到下一时刻获取到了新的路标信息和机器人位姿信息之后，对于第一部分的信息进行更新。

通过贝叶斯滤波器解决 SLAM 问题的方法主要有两种。一种是基于扩展卡尔曼滤波(Extended Kalman Filter, EKF)的 SLAM 算法。一种是基于粒子滤波(Particle Filter, PF)的 SLAM 算法。在 1990 年的一篇论文中，Smith 和 Cheeseman^[6]通过采用 EKF 的方法解决了卡尔曼滤波(Kalman Filter, KF)仅限于处理线性系统的问题，将之扩展到了非线性系统问题的应用当中。S Holmes^[7]等人在传统的通过 EKF 来解决 SLAM 系统中非线性问题的基础上，引入了平方根无迹卡尔曼滤波器(Square Root Unscented Kalman Filter, SRUKF)降低了计算的复杂度，提高了系统的运行效率。EKF 需要保证噪声服从高斯分布，而粒子滤波(Particle Filter, PF)是非线性非高斯的，适用性更好。Dellaert^[8]最先将粒子滤波的思想适用到 SLAM 系统中。首先在机器人的初始位置周围按照某种先验分布随机产生一定数目的粒子，然后根据状态转移方程更新粒子的位姿信息。通过观测模型获取的路标点评估当前粒子的优劣，计算权重根据权值进行重采样，然后进行下一时刻的估计。但是为了提高粒子滤波的效果不得不增加粒子的数目，从而造成计算量的增大。Doucet 和 Murphy 等人^[9]提出了 Rao-Blackwellised 粒子滤波器(RBPF)。Montemerlo 等人^[13]在 Rao-Blackwellised 粒子滤波器基础上开发了 Fast-SLAM，并且得到了很好的效果。

在 SLAM 发展过程中由于相机具有体积小，功耗低，结构简单，成本低，图像信息丰富的优点。近年来视觉 SLAM 逐渐成为了研究的热点，并且逐步从实验室走向市场应用。而双目相机由于获取信息丰富，可以恢复出深度信息的优点，基于双目相机的 SLAM 算法发展迅速。在单目视觉 SLAM 的研究中，H Zhou^[10]等人提出通过一种将目标的空间目标函数进行迭代的 Sfm(Structure from motion)的方法，提高单目 SLAM 系统对于异常值的鲁棒性处理，同时使用扩展卡尔曼滤波和 Sfm 紧密结合的方法，在单目视觉的基础上可以方便的和其它传感器进行信息融合，并且在文章中对于算法收敛性的验证。双目立体视觉方面，F Bellavia^[11]等人提出了

一种新的立体视觉框架，将一种鲁棒的环链匹配方法应用于关键点的跟踪以及有效帧的选择，该方法依赖于观察到的姿态误差，并且作者将该方法应用到 KITTI 数据集中验证了该算法的有效性。在双目视觉 SLAM 中根据利用的图像信息不同可以分为两类，一类是使用图像特征的 SLAM 算法；一类是使用像素灰度信息的 SLAM 算法。G Zhang^[12]等人提出将相机获取到的图像中的特征线作为匹配的特征，该论文的 SLAM 系统使用双目相机作为传感器，通过运动估计，姿态优化等方法进行位姿估计，同时通过两种方法对于三维线进行初始化和投影，该算法表现出更好的重建性能。

单单只是使用相机作为传感器误差会比较大，现在很多研究都是通过惯性传感器和激光雷达来弥补视觉信息的误差。比如 Bouvrie 等人^[14]希望通过先验运动来提高 SLAM 算法的鲁棒性和精度，提出把初始姿态估计和通过 IMU 姿态解算的结果进行加权平均融合，得到移动机器人的新的姿态估计结果。Hjelmare 等人^[15]通过扩展卡尔曼滤波同时融合视觉传感器、轮式里程计以及陀螺仪。使用视觉传感器构建观测模型，使用轮式里程计构建运动模型。

相比于国外这三十年在 SLAM 领域的快速发展，国内在机器人同时定位与建图方面也获得了极大的进步。但是由于研究起步时间相对较晚，相对国外研究还有一定差距。宋宇等人^[16]为了减少 SLAM 非线性模型系统的误差，通过容积率来求解 SLAM 的非线性函数高斯权重积分，提高了 SLAM 算法的精度。杜钊君等人^[17]通过融合双目相机和激光测距仪来获取机器人的精准定位，把从双目相机和激光测距仪获取到的两种信息进行融合，机器人的定位准确性和鲁棒性都得到了改善。陈炜楠等人^[18]将元胞自动机应用到同时定位与建图当中，提高了定位结果的准确度。此外很多大学和研究机构的学者都开展了大量有关移动机器人 SLAM 问题的研究，取得了很多成果。吕丹娅^[19]等人将改进的 ORB 算法引入到单目相机作为传感器的 SLAM 系统的特征匹配中来，提高了 SLAM 系统的匹配精度，同时通过引入尺度空间的概念明显提高了系统对于尺度变化的适应性，通过 ORB 特征有效减少了 SLAM 系统的运行时间提高了系统的运行效率。熊斯睿^[20]在全景视觉的基础上研究了机器人 SLAM 算法的实现，首先构建了基于全景视觉的相机成像模型，然后改进了角点的检测方法，提高了焦点的检测精度和稳定性，而在 SLAM 算法中数据关联问题上，作者采用最近邻算法进行解决，最后将该算法在机器人平台上进行了有效性验证。白云汉^[21]通过将卷积神经网络目标检测算法引入到 SLAM 系统当中，以此来构建语义地图，保存空间中较为丰富的语义信息，该方法充分利用了神经网络的优点提高了语义提取的速度，同时保证了 SLAM 系统的准确性。

1.3 研究内容

本文主要研究基于双目相机和 IMU 融合的移动机器人 SLAM 问题。主要研究移动机器人在同时定位与建图过程中涉及到的基本理论要点和技术难点以及整个实现过程。搭建基于双目视觉和 IMU 融合算法框架,并且提出相应的改进方法。最后系统实验表明通过改进的方法可以使 SLAM 系统在定位精度和鲁棒性上有一定提升。本文研究内容如下所示:

(1)研究了 SLAM 发展现状,学习了最新的研究成果。同时研究了 SLAM 相关理论基础,分析了 SLAM 系统主要流程,以及 SLAM 算法需要解决的问题,同时对常用的 SLAM 算法进行分析。

(2)研究立体匹配需要用到到的相关理论,建立双目相机模型,分析研究了双目相机的标定和矫正方法。对于立体视觉最为重要的图像特征匹配问题,通过对常用图像特征进行对比研究,最终决定选用 ORB^[22]作为匹配特征。本文充分考虑了天气和光照对于双目立体视觉的影响,研究提出了相应的改进方法。在进行 ORB 特征提取之前,先使用在对比度上被限制的自适应直方图均衡化(Contrast Limited Adaptive Histogram Equalization, CLAHE)^[23]对图像进行图像增强。然后构造金字塔模型提高 ORB 特征的尺度不变性,提高图像提取的效率。通过 ORB 算法匹配获取到的匹配对数目通常很多,能够充分满足观测方程对路标点的需求,但是在这些匹配对中有一部分是误匹配对需要剔除,同时考虑到空间路标点坐标的准确性对于 SLAM 系统精度的影响。本文改进了 ORB 算法的误匹配剔除策略,通过结合汉明距离设定阈值的方法和优化的随机采样一致性^[24]算法以及考虑到匹配特征点之间的距离比和角度差的误匹配剔除方法进行误匹配剔除。由于通过更为严格的误匹配剔除策略进行误匹配剔除,可以得到更加精确的匹配对,满足 SLAM 系统的需求。

(3)研究基于滤波的 SLAM 系统模型,由于本文使用 IMU 和双目相机进行多传感器融合实现同时定位与建图,所以提出了基于 IMU 的运动模型以及基于双目立体视觉的观测模型。研究分析了本文使用的 FastSLAM^[25-27]算法的具体实现过程。FastSLAM 将 SLAM 过程分为了定位和建图两个部分。通过粒子滤波进行机器人定位,通过扩展卡尔曼滤波进行地图构建。在定位阶段通过分散的粒子来模拟机器人的位姿,在不断的迭代运动过程中,会造成粒子的退化,使粒子的权重逐渐聚集到一部分粒子上,如果使用重采样复制权值高的粒子,删除权值低的粒子又会造成粒子耗尽,本文通过引入免疫算法和粒子群算法的思想,使粒子向高似然区域运动的同时增加粒子的多样性,较好地解决了粒子退化问题。

(4)最后在 ROS 系统下获取 IMU 和双目相机数据,对获取的数据进行处理分

析,作为输入传递给观测方程和运动方程,然后使用改进的 FastSLAM 算法实现机器人同时定位与建图系统,并且通过实验证明了改进算法的有效性。

1.4 论文结构

本文主要分为六个章节,每个章节具体结构如下:

第一章为绪论。首先对论文所要研究的 SLAM 问题的意义和价值进行了阐述。然后介绍了当前国内外同时定位与建图领域的研究现状。最后介绍了论文的研究内容和具体结构。

第二章为解决 SLAM 问题使用的相关理论,以及在实现 SLAM 系统时用到的技术。首先介绍了 SLAM 的概念,以及实现 SLAM 系统遇到的难点问题。然后介绍了主流的 SLAM 算法,包括基于扩展卡尔曼滤波^[28-30]和基于粒子滤波^[31-32]的方法。再然后分析了 IMU 的模型以及 IMU 的姿态解算方法,通过 IMU 进行航迹推算的过程。最后介绍了本文使用的 SLAM 系统框架。

第三章为本文的重点章节。在这一章中主要介绍了作为观测方程输入的双目立体视觉系统模型,通过双目相机获取空间坐标点深度的方法。然后介绍了双目相机的标定和矫正方法,并且对于 SLAM 系统使用的双目相机进行标定和矫正。对于获取双目立体视觉需要使用的匹配点对的常用特征点提取和匹配算法进行了对比分析,并且提出了改进方法,提高了匹配效率和误匹配剔除效率。

第四章为本文的重点章节。实现双目相机和 IMU 融合的改进 FastSLAM 算法。这一章首先提出了 SLAM 的系统模型,包括运动模型和观测模型,然后提出了基于 FastSLAM 的改进算法,并且通过实验对于改进算法的正确性加以证明。

第五章为系统实现章节。在 ROS 系统下对 IMU 和双目相机数据进行获取,同时对获取的数据进行处理分析,作为输入传递给观测方程和运动方程,然后使用改进的 FastSLAM 算法构建 SLAM 系统。

第六章为总结和展望。在这一章中,首先对本文前五个章节的内容进行一个总结,同时对于本文提出的创新点进行阐述。然后基于本文在 SLAM 方向上研究的不足之处,提出改进方向,为以后的研究打下基础。

第二章 相关理论与技术

在这一章中将要介绍本文涉及到的 SLAM 的相关概念。同时介绍了基于贝叶斯滤波器的 SLAM 方法，主要包括基于扩展卡尔曼滤波器的 SLAM(EKF-SLAM)算法和基于粒子滤波的 SLAM(PF-SLAM)算法。然后介绍了 IMU 的系统模型。最后分析了双目相机和 IMU 融合的 SLAM 系统框架。

2.1 SLAM 算法解决的问题

SLAM 算法是用来实现移动机器在未知环境下，通过本体携带的传感器，感知周围未知环境的一系列路标，以确定机器人自身所在的位置，并且构建路标位置地图。该过程如图 2-1 所示：

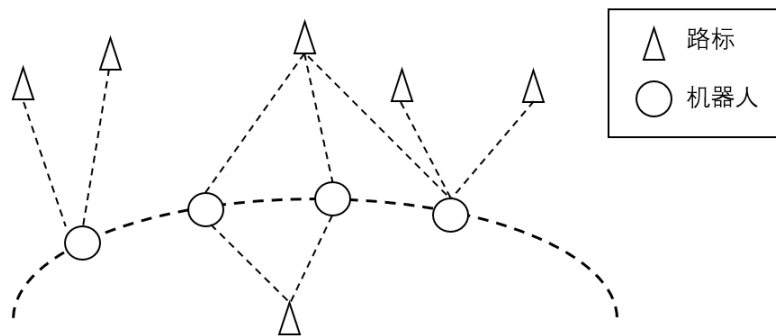


图 2-1 SLAM 问题简单模型

但是由于在运动过程中对于路标点和机器人自身位姿估计存在误差，所以在实际计算中 SLAM 问题处理过程如图 2-2 所示。

在图 2-2 中白颜色圆圈表示机器人的真实位姿信息，白颜色三角表示机器人观测到的真实路标点。连接白颜色圆圈的实线表示移动机器人真实的运动轨迹。灰色圆圈表示估计所得的机器人位姿状态，灰色三角表示估计所得的路标点位置，虚线是估计的运动轨迹。通过图 2-2 可以看出 SLAM 问题实际上是一个概率估计问题，通过不同的估计方法使得最终获取到的机器人位姿信息和轨迹信息尽可能的接近真实的结果。当 SLAM 系统运行时，因为机器人观测模型和运动模型的误差以及周围环境的干扰等原因，会造成 SLAM 系统的先验概率不能够准确地计算出机器人自身的位姿信息和路标点的位置。所以需要通过使用本体携带的传感器观测到

的数据来计算路标点在空间中的位置，然后计算后验概率分布，对 SLAM 系统的误差进行校正。整个机器人位姿计算的过程就是一个不断地估计然后校正的过程。

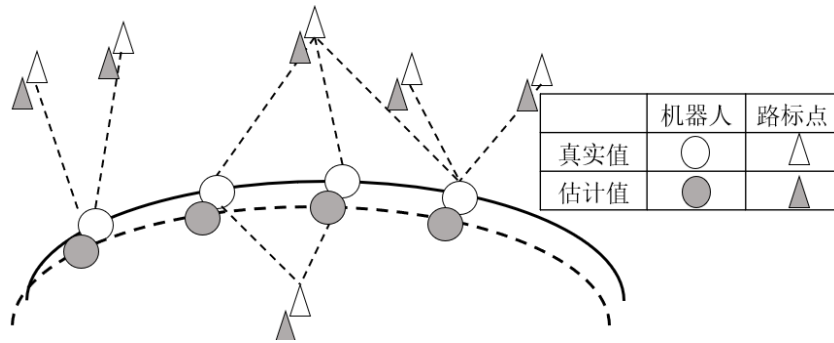


图 2-2 SLAM 问题实际模型

2.2 SLAM 算法的难点

SLAM 问题从出现到现在已经有二十多年的历史，研究者众多并且已经取得了不错的成果，但是想要把 SLAM 应用于室外大规模非结构化的环境下，仍然存在许多理论和实际问题需要解决。这些问题主要有数据关联^[33]，环境的不确定性影响^[34]，多传感器融合^[35]以及计算复杂度高^[36]等问题。具体来说主要有：

(1) 数据关联问题

在 SLAM 系统中需要对不同的地点和不同时间传感器观测到的数据和已经保存在数据库中的地图信息数据进行关联，确定是否为同一数据，这个就是数据关联问题。数据关联问题是 SLAM 系统中非常关键的问题，数据关联的过程是通过传感器对周围环境进行分析提取新的特征作为路标，把获取到的新路标和地图库中的路标进行匹配，如果没有匹配的路标就把新的路标添加到地图库中，如果有匹配就把路标的特征信息加入到系统中提高路标估计的准确性。随着 SLAM 系统的运行，由于匹配的复杂度会不断增加以及特征提取存在误差，会影响 SLAM 系统数据关联的成败。目前处理数据关联问题的方法主要有基于概率数据关联的算法、KNN 算法等。正确的数据关联是实现 SLAM 算法的关键，针对虚假关联，JCHoi^[37]提出了将两两相容性概念应用于基于概率的数据关联方法，并且通过 EKF-SLAM 做了仿真实验证明了该理论的正确性。

(2) 环境不确定性问题

这一问题是由于移动机器人在未知环境下，只能够通过自身的传感器来获取周围环境的信息。但是由于周围环境的干扰以及传感器自身的噪声，会在 SLAM 系统计算过程中引入误差，并且各种测量误差之间有一定的相关性，从而给系统的输出带来不确定性。除了周围环境的影响和传感器误差，系统的误差还来源于数学

建模引入的误差。这些误差会在系统的运行过程中不断累积影响机器人定位的精度。所以环境的不确定性对于 SLAM 系统精度有着重要的影响。PF Alcantarilla^[38]针对动态环境下的 SLAM 算法的问题,提出了将密集场景流的概念引入 SLAM 中检测运动的对象,在不增加测量到估计的错误的基礎上产生更好的定位效果,并且对于室外动态环境下 SLAM 算法的性能进行了验证。

(3) 融合问题

SLAM 算法的应用场景是基于未知环境的,所以需要使用自身携带的传感器获取周围环境的信息,进行定位与建图。常用的传感器包括相机,激光传感器,微波传感器以及 IMU,但是多个传感器获取到的信息数据是相互独立的,需要对这些数据进行时间序列上的控制,同时还要通过一定的基于概率的方法将多个传感器信息进行融合。T Qin^[39]等人将单目相机和 IMU 融合实现移动机器人的同时定位与建图,并且通过数据集验证了算法的有效性。

(4) 计算复杂度问题

SLAM 系统对于实时性要求非常高,但是随着运动过程中观测的路标点越来越多,需要保存的信息也会随之增长,当需要处理数据关联时,就需要耗费大量的时间进行匹配更新,造成计算的复杂度也会非常大,影响系统的实时性。C Chen^[40]等人提出一种基于线性回归的卡尔曼滤波的改进的 CDKF-SLAM,通过重构状态估计方程和预测方程改进采样策略,算法的复杂度降到了 $O(N^2)$ 。

2.3 SLAM 算法分类

移动机器人的 SLAM 问题用来解决机器人在陌生环境下,通过本体携带的传感器获取陌生环境信息和自身位姿估计,实现同时定位与建图。当移动机器人在陌生环境下运动时没有任何先验知识以及人类辅助,在这种情况下 SLAM 是实现机器人完全自主行动的关键。移动机器人在运动过程中由于周围环境和噪声的不确定性,引入了概率的方法解决 SLAM 问题。

SLAM 问题中观测模型和运动模型都是非线性的。所以需要使用一种非线性的滤波器解决 SLAM 问题。扩展卡尔曼滤波器作为一种非线性的滤波器从而被广泛的应用于 SLAM 问题当中,是早期处理 SLAM 问题的主流算法。但是由于扩展卡尔曼滤波的使用仅限于高斯噪声模型,在非线性场景下效果不好,基于粒子滤波的 SLAM 算法逐渐得到应用。PF-SLAM 算法通过粒子集代替概率分布,通过样本集合逼近概率分布,PF-SLAM 算法能够解决多峰分布问题,减少内存占用,不受高斯分布的影响,提供更加准确的定位。在这一节中首先介绍 SLAM 问题如何用概率来表示,然后介绍了 EKF-SLAM 和 PF-SLAM 两种通过概率解决 SLAM 问题

的算法。

2.3.1 概率化 SLAM 问题

通过状态向量 s^t 和地图 m 来表示在 t 时刻机器人的位置和路标点的位置。用 u^t 表示 t 时刻的控制量，用 z^t 表示 t 时刻的观测量。先验概率函数 $p(s_t | s_{t-1}, u_t)$ 表示机器人在 t 时刻的运动模型。观测概率函数 $p(z_t | s_t, m)$ 表示机器人在 t 时刻的观测模型。SLAM 问题可以转化成求解后验概率函数 $p(s^t, m | z^t, u^t)$ 的问题。 t 时刻状态向量 s^t 和地图 m 的后验概率函数 $p(s^t, m | z^t, u^t)$ 可以通过 Markov 假设和贝叶斯公式表示为式 2-1:

$$p(s^t, m | z^{1:t}, u^{1:t}) \propto p(z^t | s^t, m) \int p(s^t | s^{t-1}, u^t) p(s^{t-1}, m | z^{1:t-1}, u^{1:t-1}) ds^{t-1} \quad (2-1)$$

2.3.2 基于扩展卡尔曼滤波的 SLAM 算法

作为一种效率较高的递归滤波器，EKF 被广泛的应用到 SLAM 领域。最为原始的卡尔曼滤波只能应用在服从高斯分布的线性系统中，EKF 是对卡尔曼滤波的改进算法，EKF 算法可以应用于非线性的系统。EKF-SLAM 算法首先要构建机器人的运动模型和观测模型。运动过程模型如式 2-2 所示:

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

其中 x_t 是 t 时刻的位姿信息， f 是运动过程函数， u_t 表示系统的控制量。考虑到通过运动过程函数求解位姿会存在误差，所以在公式最后加上协方差为 Q 的零均值高斯白噪声 v_t 。观测方程如式 2-3 所示:

$$z_t = h(x_t) + w_t \quad (2-3)$$

其中 z_t 是 t 时刻的观测信息， h 是观测函数。因为观测方程存在误差，所以引入协方差为 R 的零均值高斯白噪声 w_t 。EKF-SLAM 是三个步骤的迭代过程。首先进行预测，然后观测路标点，最后更新预测信息，不停迭代。在 EKF-SLAM 问题中机器人运动状态的先验估计使用 \hat{x}^- 来表示，并且 $\hat{x}^- = \hat{x}_{t|t-1}$ 。状态的后验概率使用 \hat{x}^+ 来表示，并且 $\hat{x}^+ = \hat{x}_{t|t}$ 。算法具体实现的过程如下所示:

(1) 预测

在预测阶段， t 时刻机器人的状态 \hat{x}^- ，以及 t 时刻的协方差矩阵 P_t^- 和观测量 z_t^- 通过式 2-4 进行计算:

$$\begin{cases} \hat{x}_t^- = f(\hat{x}_{t-1}^+, u_t) \\ \hat{z}_t^- = h(\hat{x}_t^-) \\ P_t^- = \nabla f_{x_{t-1}} P_{t-1}^+ \nabla f_{x_{t-1}}^T + Q_t \end{cases} \quad (2-4)$$

其中雅可比矩阵 $\nabla f_{x_{t-1}}$ 表示对运动过程函数 f 在 \hat{x}_{t-1}^+ 进行泰勒展开后进行线性化得到的，如式 2-5 所示：

$$\nabla f_{x_{t-1}} \triangleq \frac{\partial f(\hat{x}_{t-1}^+, u_t)}{\partial x} \quad (2-5)$$

(2) 观测

在观测阶段，机器人在状态 x_t 对于周围环境进行观测，当观测到 z_t 后可以算出新息 v_t ，如式 2-6 所示：

$$v_t = z_t - \hat{z}_t^- \quad (2-6)$$

当认为数据关联的过程是理想状态时可以得到新息的协方差矩阵，如式 2-7 所示：

$$S_t = \nabla h_{x_t} P_t^- \nabla h_{x_t}^T + R_t \quad (2-7)$$

其中 ∇h_{x_t} 表示观测函数对于观测方程 h 的线性化近似。

(3) 更新

在更新阶段首先计算卡尔曼增益为式 2-8：

$$W_t = P_t^- \nabla h_{x_t}^T S_t^{-1} \quad (2-8)$$

然后计算机器人的更新后的运动状态估计 \hat{x}_t^+ 和协方差矩阵 P_t^+ ，计算过程如式 2-9 所示：

$$\begin{cases} \hat{x}_t^+ = \hat{x}_t^- + W_t v_t \\ P_t^+ = P_t^- - W_t S_t W_t^T \end{cases} \quad (2-9)$$

EKF-SLAM 算法对于 SLAM 算法运行过程中的收敛性，不确定性以及地图增长都有了系统的解决方案。但是，EKF-SLAM 仍然有许多尚未解决的问题。EKF-SLAM 对于运动模型以及观测模型都有很强的依赖性，当不能构建两个模型的时候通常不会使用 EKF-SLAM 算法。针对 EKF-SLAM 存在的问题有很多改进方法，

X Xie^[41]等人提出了一种考虑到偏置的估计和补偿问题的改进 EKF 算法，提高了位置的估计精度，并且通过实验验证了算法的有效性。PP Wang^[42]等人提出了一种改进的 EKF-SLAM 算法，机器人可以在生成的地图环境中定位和构建最佳的路径。该算法将强跟踪滤波器与 EKF-SLAM 算法相结合，引入了随时间变化的权重调整因子，该因子可根据运动状态引起的改变而改变，从而增加或者降低滤波器增益，进一步提高状态估计精度。同时通过仿真结果表明该算法具有更好的鲁棒性，且误差小于传统算法。

2.3.3 基于粒子滤波的 SLAM 算法

粒子滤波算法是一种蒙特卡罗方法，通过样本来估计概率分布，粒子滤波适应于几乎所有的状态空间模型，通过使样本的数量不断增加来逼近概率密度分布。粒子滤波可以很好的反映基于控制函数和观测函数的后验概率分布，所以认为粒子滤波可以应用到 SLAM 问题的解决中来。粒子滤波器相对于传统的滤波方法更加简单易用，对于非线性非高斯环境也能很好的适应。

基于粒子滤波的 SLAM 算法使用一个带有随机的加权粒子的集合来估计后验概率分布，可以适用于所有的噪声分布，不需要一定是高斯噪声，它的精度可以接近最优估计。PF-SLAM 过程如下：

(1) 初始化

在 $k=0$ 时刻，按照概率 P 抽取 M 个样本点得到粒子集合 $\{s_t^i, i=1, 2, 3 \dots M\}$ 。

(2) 进行重要性采样

根据提议分布 $q(s_t | s_{0:t-1}^i, z_{1:t})$ 对于粒子进行重要性采样。

(3) 计算粒子的权值

通过式 2-10 计算粒子的权值。

$$w_t^i = w_{t-1}^i \frac{p(z_t | s_t^i) p(s_t^i | s_{t-1}^i)}{q(s_t | s_{0:t-1}^i, z_{1:t})} \quad (2-10)$$

(4) 进行权值归一化

把计算完权重的粒子的权重进行归一化如式 2-11 所示：

$$w_t^i = w_t^i / \sum_{j=1}^n w_t^j \quad (2-11)$$

(5) 对于粒子进行重采样

当粒子集的有效粒子数少于阈值时进行重采样。重采样的过程就是复制权值

大的粒子，删除一些权值较小的粒子，得到 M 个服从于分布 $p(s_{0:k}^i | z_{1:k})$ 的样本集合。

(6) 通过粒子估计机器人的真实状态

通过步骤 5 得到重采样后的粒子集。然后通过粒子的状态和以及每一个粒子的权重值来估计机器人的位置。计算过程如式 2-12 所示：

$$\hat{s}_t = \sum_{i=1}^M w_t s_t^i \quad (2-12)$$

(7) 判断是否需要继续迭代计算

判断 PF-SLAM 系统是否已经执行完毕如果没有执行完毕跳转回步骤(2)继续迭代运算。

通过上述 7 个步骤可以有效的解决 SLAM 问题，但是基于粒子滤波 SLAM 方法仍然存在一些问题，机器人的位姿是通过粒子的分布来近似估计的，当机器人在运动的过程中周围环境非常复杂时就需要采样大量的样本才能更好的近似于 SLAM 系统的后验概率密度，这样就造成了算法的复杂度增加。所以需要通过自适应的策略对样本数量进行增减。通过重采样会造成粒子退化和粒子耗尽的现象，需要通过改进的方法保持粒子的多样性克服粒子耗尽的问题。对于 PF-SLAM 存在的问题，已经有很多的改进算法。Sjia^[43]等人提出了一种基于 OpenMP 的多线程粒子滤波算法，减少粒子滤波的时间从而提高 SLAM 系统的执行效率。朱福利^[44]等人通过 GPU 对于粒子滤波过程进行并行优化加速，明显减少了由于粒子增加造成的定位延迟以及定位的误差。

2.4 SLAM 坐标系统

本文需要用到的坐标系统主要有三个，一个是世界坐标系 W ，它是整个系统的全局坐标系，使用的是地理坐标系。一个是 IMU 坐标系 L ，一个是相机坐标系 C ，如图 2-3 所示。

世界坐标系的 Y 轴是重力加速度的方向，和地理的垂线重合，在机器人运动的初始状态时需要保证 Y 轴和重力加速度方向是相同的，所以需要在运动初始时计算加速度计和重力加速度间的旋转，然后进行调整。

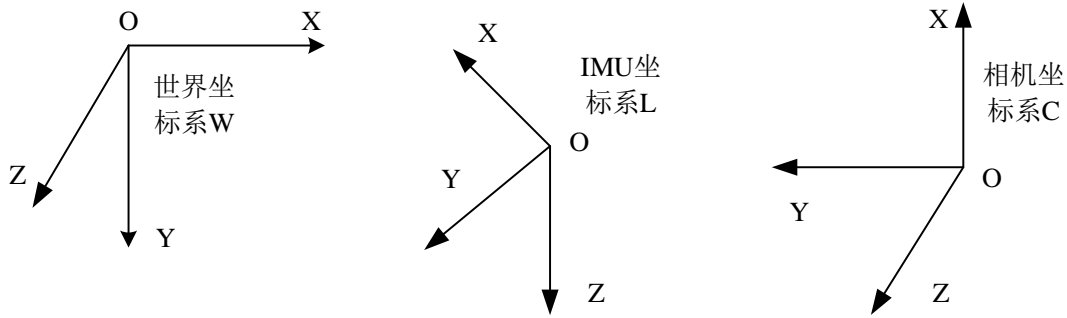


图 2-3 坐标系统

IMU 坐标系和相机坐标系与世界坐标系之间的运动可以使用旋转和平移来表示。旋转量通常使用旋转矩阵 R 或者四元数 q 来表示。旋转矩阵是一个 3×3 的矩阵，同时在计算过程中还会用到四元数来表示旋转，四元数是一个三维的复数如式 2-13 所示，把四元数转化为旋转矩阵的公式如式 2-14 所示：

$$q = w + xi + yj + zk \quad (2-13)$$

$$R(q) = \begin{bmatrix} 1 - 2(y^2 + z^2) & 2(xy - zw) & 2(xz + yw) \\ 2(xy + zw) & 1 - 2(x^2 + z^2) & 2(yz - xw) \\ 2(xz - yw) & 2(yz + xw) & 1 - 2(x^2 + y^2) \end{bmatrix} \quad (2-14)$$

同时考虑双目相机采集的特征点到相机坐标的变化，需要使用变换矩阵，平移 t 使用笛卡尔坐标描述，是一个 3 维向量。通过引入变换矩阵表示特征点坐标系之间的变化，变换矩阵包括了旋转和平移两个变量，通过引入齐次坐标一次计算就可以获取坐标系之间的变换，变换矩阵 T 如式 2-15 所示：

$$T = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} \quad (2-15)$$

2.5 惯性导航系统

目前惯性导航系统根据是否把传感器固定安装在机器人本体上，可以大体分为捷联式和平台式两种。最早出现的惯导系统是平台式惯导系统，它由计算机、加速度计和陀螺仪组成，通过把加速度计以及陀螺仪直接固定安装在载体上计算载体运动时的位姿信息。而捷联惯导系统把加速度计和陀螺仪分离出来，没有固定安

装在载体上。相对于平台式惯导系统更加简单成本更加低廉，应用场景更多，但是由于体积小更容易被振动和冲击影响造成动态误差，所以捷联惯导系统需要补偿角速度和加速度引起的动态误差。在 SLAM 系统中使用的 IMU 属于捷联式惯导系统。

IMU 由陀螺仪和加速度计组成，是一种不依赖于外部信息的导航系统，相对其它导航系统来说更加隐蔽安全。IMU 的应用场景主要是运动的设备上，例如移动机器人、汽车和无人机等。陀螺仪输出的数据是机器人的当前时刻的角速度，仅代表这一时刻的状态。角速度不能直接应用到角度计算上，需要通过对角速度进行积分得到角度相对于前一时刻转动了多少。将计算结果

加上机器人上一时刻的角度，从而得到这一时刻机器人的角度信息。在角度的计算过程中，积分的间隔时间 dt 越小，所得到的角度精度越高。由于在计算角度的过程中，计算的基准是自身，没有外部参照系进行校准，所以会有累计误差，并且累计误差会逐渐增加，使得结果偏离真实角度，所以陀螺仪大部分用来做短时间内的角度计算。加速度计用来计算机器人运动时三个方向上的加速度，通过对加速度进行二次积分可以得到机器人的位移信息。将计算得到的位移信息和上一时刻位置相加得到机器人当前位置。这一节主要介绍通过 IMU 来进行航位推算，通过不断测量机器人的加速度和角速度获取下一时刻机器人的位姿信息。

2.5.1 IMU 模型建立

IMU 三轴加速度计的真实加速度记为 \mathbf{a} ，三轴陀螺仪的真实角速度记为 \mathbf{w} 。在建立 IMU 模型时需要考虑加速度计和陀螺仪存在误差问题。IMU 的误差主要来自于三部分，分别是：稳定性误差、非线性测量误差以及测量噪声误差。误差会影响 IMU 测量的精度，通过建模对 IMU 误差进行分析，对误差进行补偿。

加速度计的误差主要有静态误差和动态误差，通过加速度计可以进行二次积分进行航位推算，但是需要和其它传感器进行融合，计算过程如式 2-16 所示：

$$\mathbf{a}_{ce} = \mathbf{a} + \mathbf{b}_a + \mathbf{n}_a + \mathbf{g} \quad (2-16)$$

其中 \mathbf{a}_{ce} 是真实的加速度计的测量值， \mathbf{b}_a 是加速度的偏置量， \mathbf{n}_a 是随机噪声， \mathbf{g} 为重力加速度的分量。

陀螺仪同加速度计一样有动态误差和静态误差，陀螺仪的模型用式 2-17 来表示：

$$\mathbf{w}_{ce} = \mathbf{w} + \mathbf{b}_w + \mathbf{n}_w \quad (2-17)$$

其中 w_{ce} 是真实的陀螺仪角速度测量值， b_w 是陀螺仪的静态偏移，静态偏移可以通过计算在 IMU 不动的情况下陀螺仪数据的平均值来获得， n_w 是一个随机游走的噪声。

2.5.2 IMU 姿态解算

通过对 IMU 的三轴陀螺仪获取到的角速度进行积分可以得到在三个轴上的旋转，但是由于三轴陀螺仪存在漂移和噪声，直接通过积分计算角度误差会很大，陀螺仪的动态稳定性相对较好，但在静止状态时数据可靠性较低，而加速度计相反在静止状态时稳定性较好，而在运动时可靠性较低，可以通过滤波的方式用加速度计修正陀螺仪的误差。EKF 和互补滤波是惯性测量单元常用的姿态解算算法。扩展卡尔曼滤波相对于互补滤波动态性能较优，通过扩展卡尔曼滤波进行姿态解算时，输入是三轴陀螺仪获取的数据，状态向量是偏移角以及姿态角，观测模型是加速度计获取的三个方向上的加速度。扩展卡尔曼滤波存在一些问题，由于计算量受状态向量维数的影响，随着状态量的增加计算量越大，所以通过扩展卡尔曼滤波进行姿态解算对实时性和处理器的处理能力有一定影响。通过互补滤波算法进行姿态解算的过程主要分为两步。首先用高通滤波处理陀螺仪积分后的数据，用低通滤波处理加速度计的数据。然后通过求和获取姿态角，相比于 EKF 速度更快。本文使用的互补滤波算法进行姿态解算。主要过程如下所示：

首先对数据进行规范化处理，把角度的单位转化为“弧度每秒”，同时对加速度 $a(ax, ay, az)$ 进行归一化处理得到 a' ，处理过程如式 2-18 所示：

$$\begin{aligned} a_{normal} &= \sqrt{a_x^2 + a_y^2 + a_z^2} \\ a'_x &= a_x / a_{normal} \\ a'_y &= a_y / a_{normal} \\ a'_z &= a_z / a_{normal} \end{aligned} \quad (2-18)$$

构建方向余弦矩阵，矩阵 C 如所式 2-19 所示：

$$C = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_0q_2 + q_1q_3) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (2-19)$$

通过旋转把重力加速度转化到机体坐标系，得到重力加速度在机体三个坐标轴上的分量 a'' ，如式 2-20 所示。

$$a'' = C \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 2(q_1 q_3 - q_0 q_2) \\ 2(q_2 q_3 + q_0 q_1) \\ q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (2-20)$$

计算补偿向量 e ，及理想加速度和实际的加速度之间的差，把标准的加速度 a'' 和归一化后的加速度 a' 进行叉乘。

在这一步中对于上一步求得的误差进行互补滤波，滤波的过程如式 2-21 所示，其中 \hat{q} 是用四元数表示系统的姿态， δ 表示 PI 调节器通过运算产生的新息， e 表示补偿向量。PI 值不能够过大，其中 P 的值要小于 1，I 的值要小于 0.01。

$$\begin{aligned} \dot{\hat{q}} &= \frac{1}{2} \hat{q} \otimes p(\bar{\Omega} + \delta) \\ \delta &= k_p e + k_i \int e \\ e &= \bar{v} \times \hat{v} \end{aligned} \quad (2-21)$$

求解微分方程 $\dot{\hat{q}}$ ，得到新的四元数，然后通过公式 2-14 得到旋转矩阵。除了本文提到的姿态解算算法，还有很多有很好效果的通过融合算法进行姿态解算的算法^[45-46]。

2.5.3 位移计算

通过 IMU 进行位移计算是一种常用的航迹推算方法。在航迹推算过程中要用到 IMU 的加速度计，在使用时需要去掉重力加速度分量。通过对三轴加速度进行二次积分可以得到短时间内比较准确的位移信息。设加速度计的加速度值为 $a(a_x, a_y, a_z)$ ，其中 a_x, a_y, a_z 分别表示加速度计在三个轴上的加速度分量。首先需要把惯导坐标系下的加速度投影到世界坐标系中，设变换矩阵 T_{CL} 为式 2-22：

$$T_{CL} = \begin{bmatrix} R_{CL} & t_{CL} \\ 0^T & 1 \end{bmatrix} \quad (2-22)$$

因为不需要考虑位移的影响所以通过其中的旋转矩阵 R_{CL} 进行坐标系变换的过程如公式 2-23 所示：

$$a_{CL} = R_{CL} a \quad (2-23)$$

获取到 IMU 在世界坐标系下的三轴加速度后，去除掉重力加速度的影响，然后通过一次积分得到位移 $v_c(t)$ ，通过对速度进行积分得到位移信息 $s_c(t)$ ，计算过程如式 2-24 所示：

$$\begin{aligned} v_c(t) &= v_c(0) + \int_0^t (a_{CL} - g) dt \\ s_c(t) &= s_c(0) + \int_0^t v_c(t) dt \end{aligned} \quad (2-24)$$

其中 $v_c(0)$ 是 IMU 初始状态的速度, $s_c(0)$ 是 IMU 初始状态的位置, g 是重力加速度。由于姿态解算存在误差, 同时会有噪声进行干扰, 得到的结果会存在误差, 所以不能直接用来定位, 需要同激光传感器或者视觉传感器滤波融合来提高定位精度。例如 T Qin^[39]等人将惯性数据和视觉数据进行融合, 通过单目相机和加速度计以及陀螺仪获取的数据进行融合进行 SLAM 算法的实现。本文采用双目相机和 IMU 数据进行融合的方式进行同时定位与建图。

2.6 粒子优化算法

2.6.1 粒子群优化算法

由 Kennedy 和 Eberhart^[47]于 1995 年提出的粒子群优化算法(Particle swarm optimization algorithm PSO), 是一种基于群体的优化方法。最早起源是对鸟群寻找食物和迁徙的过程的研究。然后发展为一种搜索复杂空间最优解的算法, 被逐步应用于优化问题, 现在 PSO 算法已经广泛应用于对目标优化和优化约束以及组合优化等问题。

PSO 与遗传算法^[48]一样, 是一种基于群体的优化工具。然而, 与遗传算法不同, PSO 没有交叉和变异等演化算子, 而且 PSO 参数较少。PSO 算法把种群中的每一个单独个体看作一个没有体积和质量的点, 称之为“粒子”(particle), 并且这个粒子是在 n 维搜索空间中不断运动, 在搜索空间中粒子以一定速度运动, 在运动的过程中粒子会不断地向历史上的领域最佳位置和自身最佳位置运动, 逐渐聚集, 从而对候选解进行优化。

PSO 算法整体上的思路就是通过一组随机粒子的多次迭代找到最优解的位置。粒子在迭代探索过程中, 从自身发现以及其他粒子的过去的经验中获益, 并寻找使目标函数值更高的位置。由于 PSO 算法需要的参数少, 同时不需要梯度信息, 又有一定的智能背景, 非常适合科学研究和工程计算。主要过程如下:

首先定义参数, 在 n 维空间中有 m 个粒子, 粒子 $i(0 \leq i \leq m)$ 在 t 时刻的位置为 $x_i^t = (x_{i1}^t, x_{i2}^t, \dots, x_{in}^t)$ 。粒子 i 在 t 时刻的速度为 $v_i^t = (v_{i1}^t, v_{i2}^t, \dots, v_{in}^t)$ 。粒子 i 在 t 时刻的局部最优解为 $p_i^t = (p_{i1}^t, p_{i2}^t, \dots, p_{in}^t)$ 。所有粒子在 t 时刻的全局最优解为 $p_g^t = (p_{g1}^t, p_{g2}^t, \dots, p_{gn}^t)$ 。在每次循环迭代的过程中需要获取粒子 i 的速度和位置, 在 t 时刻粒子 i 在 n 维空间的速度更新公式为 2-25

$$v_{in}^t = \omega v_{in}^{t-1} + c_1 \text{rand}_1(p_{in}^{t-1} - x_{in}^{t-1}) + c_2 \text{rand}_2(p_{gn}^{t-1} - x_{in}^{t-1}) \quad (2-25)$$

其中 c_1 和 c_2 是学习参数，通常 $c_1 = c_2 = 2$ ； v_{in} 的速度最大值为 v_{\max} ，可以得到如式 2-26：

$$v_{in}^t = \begin{cases} v_{\max} & v_{in}^t > v_{\max} \\ v_{in}^t & \text{else} \end{cases} \quad (2-26)$$

rand_1 和 rand_2 是对角阵，并且对角线元素为服从标准正态分布的随机数。 c 在 t 时刻粒子 i 在 n 维空间的位置更新公式为 2-27 所示

$$x_{in}^t = x_{in}^{t-1} + v_{in}^t \quad (2-27)$$

适应性函数 f_{fitness} 是 PSO 寻优过程非常重要的一个函数。通过这个函数判断当前粒子的适应值，可以得知这个粒子所在位置的好坏。

标准 PSO 算法的流程如下所示：

(1) 初始化粒子种群，包括粒子种群的规模 m ；每个粒子的初始位置 x_{in} 和初始速度 v_{in} ；个体粒子的局部最优解 p_{in} 等于粒子自身的开始位置；群体的全局最优解 p_{gn} 等于粒子权重最大粒子所在的位置。

(2) 参数初始化惯性系数 ω 根据经验 $\omega \in (0.8, 1.2)$ 。设置学习参数 $c_1 = c_2 = 2$ 。

(3) 根据公式 2-25，更新粒子速度。

(4) 根据 2-27 公式计算粒子位置。

(5) 根据适应度函数 f_{fitness} 计算粒子群中每个粒子的适应度。

(6) 更新粒子集中的每一个粒子的局部最优解和所有粒子的全局最优解。把每个粒子经过 f_{fitness} 值和 p_{in} 比较，当大于后者时用当前值进行替换。把粒子集中每一个粒子的 f_{fitness} 值和 p_{gn} ，判断大小，使用最大值进行替换。

(7) 如果最优解是否大于最小的界限或者 PSO 算法的迭代次数到达迭代次数的上限就停止迭代，否则返回步骤(3)。

2.6.2 免疫算法

免疫算法(Immune Algorithm, IA)^[49]来源于生物学中的人体免疫的概念，免疫算法的主要理论来源包括免疫学习机制，克隆选择原理以及免疫网络理论。IA 是受生物学启发的一个分支，属于人工智能的领域。IA 通过克隆变异实现全局寻优。由于免疫算法的克隆和亲和性计算，将它和其它算法区别开来。具体过程如下所示：

(1) 识别抗原

当识别到抗原时，表示免疫系统知道了抗原已经进入。将抗原作为优化问题中待求解的函数以及各种约束。通过对待求解问题进行可行性分析，从问题中提取出先验知识，构造亲和度函数。

(2) 产生初始化的抗体群体

在确认抗原入侵以后，抗体开始产生，并且把以前的抗原清除掉，从最优解中选择一些抗体。抗体和优化问题中的可行解相对应，所以在这一步把抗体表示成问题的可行解，同时在解空间的范围内产生一个抗体初始种群。

(3) 亲和力的计算

设免疫算法的亲和力为 f_{affinity} 。 f_{affinity} 主要有两个作用，一个是应用是表示抗体和抗原之间的亲和性， f_{affinity} 越大越匹配。另一个是应用是表示两个抗体之间的亲和性， f_{affinity} 越小，那么表明两个抗体之间的排斥力越大。在免疫算法中 f_{affinity} 是多样性的保证。 f_{affinity} 可以保证免疫算法收敛，且收敛位置是全局最优位置。在这一步需要对所有可行解进行基于 f_{affinity} 的评价，分别计算抗体之间和抗体与抗原之间的 f_{affinity} 。

(4) 记忆细胞分化

通过步骤 3 可以得到和抗原 f_{affinity} 高的抗体，同时将这个抗体添加到记忆细胞。并且用新抗体替换另一个和抗原之间 f_{affinity} 最高的旧抗体。

(5) 抗体促进和抑制

通过计算所有抗体的期望值，对于期望值比设定的阈值低的抗体进行抑制，所以高亲和力抗体得到促进。所以与抗原的亲和力越高的抗体，被克隆的数量越大。相应的变异率也就越低。

(6) 抗体产生

通过抗体之间的交叉和变异产生下一代的抗体。

(7) 终止优化

当结果达到设定的阈值时停止记忆细胞的选取和生成。

2.7 多传感器融合的 SLAM 系统

本文所描述的多传感器融合的 SLAM 系统是通过 IMU 和双目摄相机对周围环境进行观测实现定位与建图。系统框架如图 2-4 所示。

系统中的硬件主要包括双目相机和惯性测量单元。双目相机相对于激光传感器和红外传感器安装方便，成本更加低廉同时获取到的信息更加丰富。相对于单目相机，双目相机能够通过恢复出特征点的尺度信息，更加可靠。IMU 包括加速度

计可以计算三个方向的加速度和陀螺仪计算机器人的旋转信息，短时间内精度很高，但是由于累计误差的存在，如果不进行误差修正几秒内误差就会很大，所以不能直接把 IMU 用于直接计算机器人的位姿。

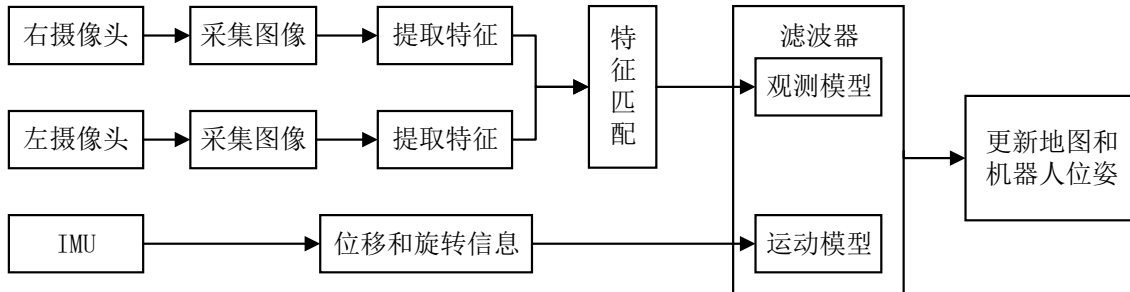


图 2-4 SLAM 系统框架

在系统中左右相机分别采集图像，保证采集的图像质量尽量相同，在采集图像后对于图像进行畸变矫正，然后应用本文改进的特征提取算法提取图像特征，然后进行左右特征的立体匹配，得到匹配的特征点，由于相机的内外参数已知，所以可以得到匹配特征点的世界坐标作为观测方程。另一方面通过 IMU 可以得到短时间内精度较高的机器人位移和旋转信息，得到运动方程，然后通过滤波的方法将两者数据进行融合，使得结果更加精确。在 SLAM 系统的运行过程中地图是由带有空间信息的特征点组成的，当提取出的匹配特征点和 SLAM 地图库中的特征相匹配时用这个特征点更新地图信息。在同时结合 IMU 的信息和机器人的位姿信息进行定位，完成整个系统的运行过程。

2.8 本章小结

本章主要是对论文将会涉及的理论知识和相关技术作简单介绍。首先介绍了 SLAM 算法解决的问题，以及 SLAM 问题解决过程中会遇到的难点。然后介绍了主流的 SLAM 算法，包括基于 EKF 和基于 PF 的 SLAM 算法。然后介绍了本文 SLAM 系统使用的坐标系统和惯性系统模型。最后介绍了 SLAM 系统的框架。

第三章 双目立体匹配关键技术的研究

通过相机计算路标点的空间坐标是本文 SLAM 系统观测模型的重要组成部分。该计算过程使用的相机可以是双目相机和 RGBD 相机，双目相机和 RGBD 相机克服了单目相机不能够获取尺度信息的缺点。RGBD 相机通过红外或者激光来获取深度信息，在室外应用效果较差。双目相机通过实时匹配左右相机的图像恢复特征点的坐标，不受应用场景的影响，是 SLAM 系统中实现立体视觉常用的方法。在本章中主要进行了双目相机模型相关技术的研究。首先从针孔相机模型引出了双目相机模型。然后介绍了双目相机的标定和矫正方法，并且对第五章 SLAM 系统中使用的双目相机进行了矫正。然后通过实验对比选定图像特征提取算法，并根据 SLAM 系统的要求对选取的图像特征提取算法进行改进。

3.1 引言

现在视觉 SLAM 系统中通常使用图像提取的特征进行立体匹配获取路标 (Landmark)。在移动机器人运动的过程中，会不断提取周围环境的视觉特征，这些视觉特征就是作为观测数据的路标。通过观测到的路标点作为观测信息，估计机器人的真实位姿，是实现机器人同时定位与建图的关键。特征之间的匹配主要通过分析匹配图像的纹理，灰度和位置信息以及图像间的映射关系实现的。一定区域范围内的像素匹配法，特征线匹配法以及特征点匹配法是图像匹配常用的三种方法。相对于区域匹配和线匹配，点匹配应用场景更为广阔，匹配方式更为灵活，实现更为容易。如今在点匹配中使用的特征点需要具有良好的尺度缩放不变性，旋转不变性，光照不变性以及平移不变性，这些特征点多为拐点，线段交叉点以及图像的重心。现在常用图像特征算法有 SIFT(Scale-invariant feature transform) 算法^[50-51]，SURF(Speeded Up Robust Features)算法^[52]，ORB^[22]算法。对于视觉 SLAM 相关算法的研究，首先要确定采用何种特征提取以及匹配方法。在本章实验中详细对比了 SIFT，SURF 和 ORB 三种算法的性能，最后选取 ORB 算法作为 SLAM 系统双目立体视觉所使用的特征算法。

在进行图像的特征提取的过程中会对自然环境中的天气情况十分敏感。比如在有雾霾等天气不好或者夜晚和室内等光照不理想的情况下，摄像头获取到的图像存在特征不明显，噪声多等问题，会造成图像特征提取效果不理想，影响后续的图像匹配，SLAM 系统使用在未知环境下运行时需要克服这些问题。所以考虑在采集完成图像以后使用图像增强的方法改善图像成像状况。而这些方法通常是通过

提高图像的对比度来实现的。本文通过将对比度被限制的直方图均衡化算法 (Contrast Limited Adaptive Histogram Equalization , CLAHE)^[53-54]引入, 增强图像的对比度, 改善图像的显示状况, 提高 ORB 特征的提取和匹配效率。

本文使用图像提取的 ORB 特征作为路标。路标的位置是通过将 ORB 匹配对加入到双目立体视觉模型中计算得出。为了获得精度较高的路标, 需要获取正确匹配的 ORB 匹配对。然而 ORB 匹配的结果中有很多误匹配, 需要通过误匹配剔除算法进行误匹配剔除, 提高匹配对的精度。本文改进了原有的误匹配剔除算法, 提高了匹配对精度。

3.2 双目立体视觉

3.2.1 双目立体视觉模型

针孔相机模型是 SLAM 领域常使用的一种模型。针孔相机模型是通过一个小孔把真实的三维世界投影到一个二维的成像平面上, 该模型如图 3-1 所示。

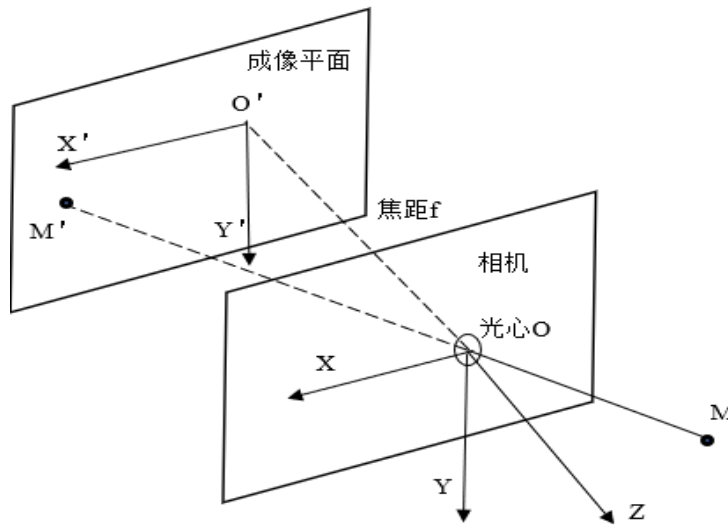


图 3-1 针孔相机成像原理

其中 OXY 是相机的坐标系, $O'X'Y'$ 是成像平面, O 是相机的光心, 光心到成像平面的焦距是 f , 真实空间中点 $M(x, y, z)$ 通过光心投影到 $M'(x', y', z')$ 。 M 和 M' 的关系可以通过三角形相似得到。 M 和 M' 坐标之间的关系如式 3-1 所示:

$$\frac{y}{y'} = \frac{x}{x'} = -\frac{z}{f} \quad (3-1)$$

M' 在成像平面还可以通过像素坐标来表示, 设为 (u, v) , 该像素坐标系横坐标与 X' 轴平行, 纵坐标与 Y' 轴平行。光轴与成像平面交点的坐标是 (u_0, v_0) , dx 和 dy 表示横纵坐标上像素点的缩放比例。转换关系如式 3-2 所示。

$$\begin{cases} u = -d_x f \frac{x}{z} + u_0 \\ v = -d_y f \frac{y}{z} + v_0 \end{cases} \quad (3-2)$$

把像素坐标转换为齐次坐标, 同时把空间点从世界坐标系转换到像素坐标系, 转换后的公式如式 3-3 所示。

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \frac{1}{z} \begin{pmatrix} -d_x f & 0 & u_0 \\ 0 & -d_y f & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix} = \frac{1}{z} K T M_w \quad (3-3)$$

其中 K 是相机的内参矩阵, 有些相机在出厂时就已经固定好了, 也可以通过标定的方法获取到。 R 和 t 组成了变换矩阵 T , 表示相机的平移和旋转, 需要在运动过程中实时计算, 可以用来恢复相机的运动。 M_w 是空间点在世界坐标系下的坐标。

仅仅是通过单目相机是不能够恢复出空间的真实位置的。需要通过双目相机恢复出空间点真实的三维空间坐标。双目相机由左右两个针孔相机组成, 它们水平放置, 光圈位于同一水平线上, 双目相机成像原理如图 3-2 所示。

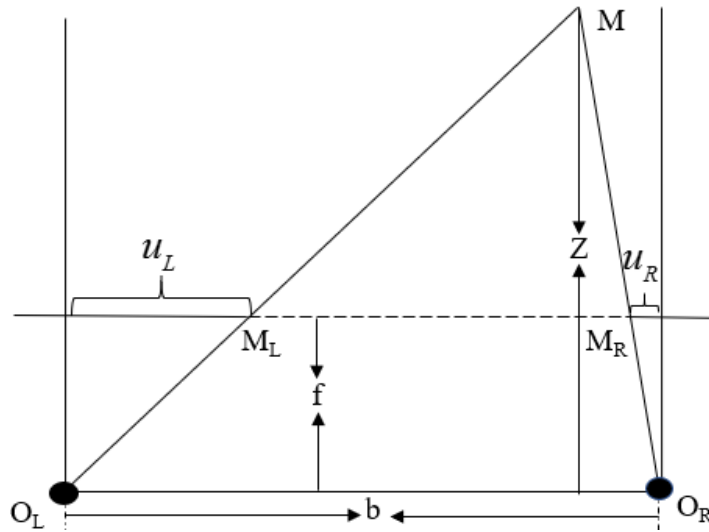


图 3-2 双目相机成像原理

在双目相机模型中 O_L 和 O_R 分别是两个针孔相机的光圈中心，两者之间的连线称之为基线。 f 是双目相机的焦距。空间坐标点 M 在两个相机的成像平面的投影分别是 M_L 和 M_R 。 x_L 和 x_R 是 M_L 和 M_R 在成像平面的横坐标。 d 为视差，可以通过视 d 来计算空间点 $M(x,y,z)$ 的深度信息，该计算通过利用 $\Delta MM_L M_R$ 和 $\Delta MO_L O_R$ 的相似来实现的。该计算过程如式 3-4 所示。

$$z = \frac{b}{d} f, \quad d = u_R + u_L = b - x_R + x_L \quad (3-4)$$

通过式 3-4 可以看出空间中一点的深度信息和视差成反比，又由于视差最小值为 1 所以通过计算的到的深度信息有理论上的最大值，所以可以通过增加基线 b 的距离来提高双目相机的计算距离，降低计算误差。在下一章中的观测模型需要用到空间坐标点 M 的空间坐标，通过本节获取的视差进行计算。

3.2.2 双目相机标定

在上一节的双目相机模型中，把空间路标点转换到像素坐标系时需要使用相机的内参矩阵 K 。 K 可以通过双目相机标定的方法得到。同时会计算相机的畸变系数，用来矫正图像的畸变。

相机的畸变多是由于光学透镜存在着透视失真的问题引起的，主要有径向和切向两类^[55]。径向畸变是由于图像放大率随着光轴之间距离的变大而发生变化，其中放大率变小的是桶形失真，放大率变大的是枕形失真，如图 3-3 所示。

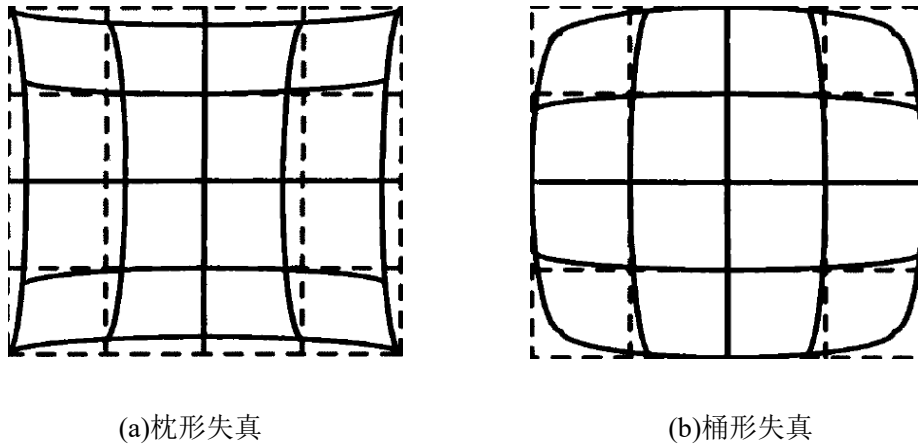


图 3-3 图片径向畸变^[55]

通过式 3-5 对发生畸变的图像进行矫正，其中 (x,y) 是图像平面上点的坐标， $(x_{corrected}, y_{corrected})$ 是该点矫正后的坐标。

$$\begin{cases} x_{corrected} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \\ y_{corrected} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \end{cases} \quad (3-5)$$

切向畸变是由于相机镜头和成像平面不能够严格平行造成的,通过式 3-6 对于切向畸变进行矫正。

$$\begin{cases} x_{corrected} = x + 2p_1 xy + p_2(r^2 + 2x^2) \\ y_{corrected} = y + 2p_2 xy + p_1(r^2 + 2y^2) \end{cases} \quad (3-6)$$

将公式 3-5 和公式 3-6 综合到一起可以得到图像畸变校正的公式 3-7。

$$\begin{cases} x_{corrected} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_1 xy + p_2(r^2 + 2x^2) \\ y_{corrected} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_2 xy + p_1(r^2 + 2y^2) \end{cases} \quad (3-7)$$

校正需要 5 个畸变参数如式 3-8 所示。

$$Distortion = (k_1 \quad k_2 \quad k_3 \quad p_1 \quad p_2) \quad (3-8)$$

在论文使用张正友标定法标定相机参数,标定过程中分别标定左右相机,然后再对双目相机进行立体标定^[56]。标定的主要参数有左右相机的内参矩阵,畸变系数 *distortion* 和矫正矩阵 *rectification*, 投影矩阵 *projection*。使用单元格尺寸是 3cm 的黑白相间的棋盘格图片进行标定。通过标定获取到的左右相机内参矩阵 K_L 和 K_R :

$$K_L = \begin{bmatrix} 477.213066 & 0 & 353.237112 \\ 0 & 465.827522 & 281.683492 \\ 0 & 0 & 1 \end{bmatrix}$$

$$K_R = \begin{bmatrix} 459.193942 & 0 & 382.060013 \\ 0 & 465.767174 & 186.929118 \\ 0 & 0 & 1 \end{bmatrix}$$

左右相机的畸变系数:

$$D_L = (-0.412947 \ 0.160367 \ -0.002278 \ 0.001350 \ 0.000000)$$

$$D_R = (-0.397265 \ 0.127657 \ -0.000240 \ -0.000138 \ 0.000000)$$

左右相机的矫正矩阵:

$$\text{Rectification}_L = \begin{bmatrix} 0.999864 & 0.005615 & 0.006511 \\ -0.005312 & 0.999965 & -0.000397 \\ -0.006713 & 0.000361 & 0.999979 \end{bmatrix}$$

$$\text{Rectification}_R = \begin{bmatrix} 0.999950 & 0.003947 & -0.007036 \\ -0.003864 & 0.999995 & 0.000395 \\ 0.008137 & -0.000363 & 0.999978 \end{bmatrix}$$

左右相机的投影矩阵：

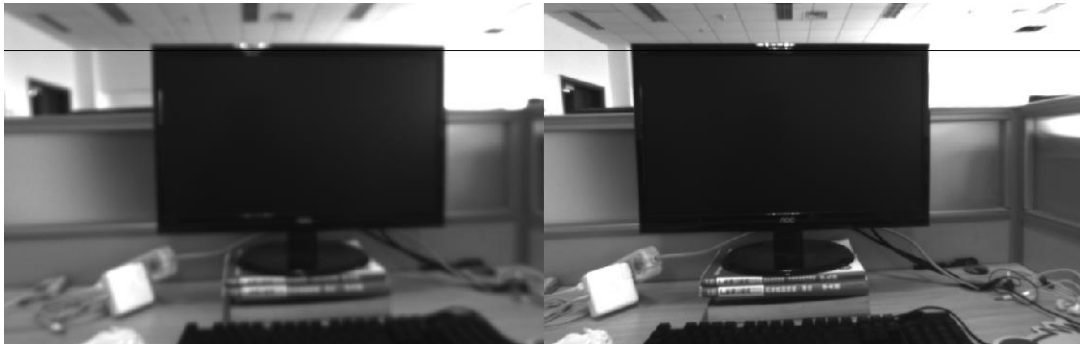
$$\text{projection}_L = \begin{bmatrix} 543.488371 & 0.000000 & 370.335258 & 0.000000 \\ 0.000000 & 543.488371 & 234.186062 & 0.000000 \\ 0.000000 & 0.000000 & 1.000000 & 0.000000 \end{bmatrix}$$

$$\text{projection}_R = \begin{bmatrix} 543.488371 & 0.000000 & 370.335258 & -55.036784 \\ 0.000000 & 543.488371 & 234.186062 & 0.000000 \\ 0.000000 & 0.000000 & 1.000000 & 0.000000 \end{bmatrix}$$

通过投影矩阵可以得到两个相机基线距离 $b=55.0368$ 。双目相机通过畸变系数矫正前后相机成像如图 3-4 所示，图片是在实验室环境下拍摄的。



(a) 矫正前成像



(b) 矫正后成像

图 3-4 畸变矫正前后双目相机成像

3.3 图像特征提取算法和图像增强算法

常用的图像特征提取算法主要有 SIFT 算法, SURF 算法以及 ORB 算法。SIFT 特征通过构建图像差分金字塔模型解决图像尺度变化的影响, 并且具有较好的旋转和亮度不变性。SURF 是在 SIFT 的基础上发展而来的, 是 SIFT 的改进版本, SURF 为了提高计算速度使用了海森矩阵和积分图。ORB 是在 FAST 角点和 BRIEF 描述子的基础上改进得到的。通过灰度质心法得到角点的方向, 通过汉明距离进行特征点间的匹配。在本章实验中对于三种算法的性能做了对比实验, 最后选取了 ORB 算法作为 SLAM 的特征提取算法, 在这一小节当中将详细介绍 ORB 算法的步骤。考虑到 SLAM 系统使用的环境有可能是光线较暗的场景, 需要通过图像增强的算法对图像进行处理, 在这一节介绍了 SLAM 系统使用的图像增强算法。

3.3.1 ORB 算法

2012 年 Rublee 等人提出了 ORB 算法。该算法对于旋转、平移以及光照都有很强的适应性以及不变性。ORB 算法特征提高了提取速度, 远快于其它两种算法。ORB 特征提取过程为:

(1) FAST 角点检测

FAST 是一种角点检测算法。确定一个点是否是 FAST 角点, 计算像素点和周围像素点是否有连续 n 个像素的像素值差大于阈值。FAST 角点提取过程如图:

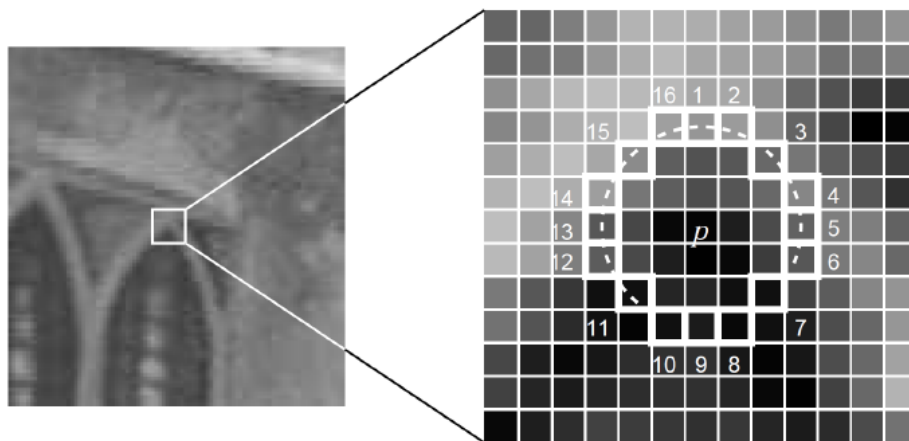


图 3-5 FAST 角点提取^[57]

(2) 计算 FAST 角点的方向

通过计算 FAST 角点的方向保证 ORB 的能够识别出旋转后的特征。这个方向就是计算特征点相邻领域的质量中心。把特征点作为 ORB 特征方向向量的开始的点, 质量中心作为方向向量的最终的点, 从而确定特征点的方向。其中 m_{pq} 是质量

中心， $I(x,y)$ 是图像的灰度值， C 是质心的位置， θ 是特征点的方向，计算过程如式 3-9 所示：

$$\begin{aligned} m_{pq} &= \sum_{x,y} x^p y^q I(x,y) \\ C &= \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \\ \theta &= a \tan 2(m_{01}, m_{10}) \end{aligned} \quad (3-9)$$

(3) 计算 rBRIEF 描述子

rBRIEF 描述子是由 Calonder 提出的。它的主要获取过程是，随机选取邻域内的 128 或者是 256 个点对，然后通过式 3-10 进行计算：

$$\tau(p; x, y) := \begin{cases} 1 & \text{if } p(x) < p(y) \\ 0 & \text{otherwise} \end{cases} \quad (3-10)$$

其中 p 是点在 (u,v) 点的像素值。从而得到 0 和 1 组成的 128 位的二进制码。这个二进制码就是描述子，并且都不相同，计算过程如式 3-11 所示：

$$f_n(p) = \sum 2^{i-1} \tau(p; x_i, y_i) \quad (3-11)$$

在上一个步骤中计算了 FAST 角点的方向，在这里同样需要计算 BRIEF 描述子的方向，使它具备旋转不变性。提取特征点领域内 128 或者 256 个点对，定义一个矩阵， (x_i, y_i) 表示随机点点对的坐标，然后通过上一个步骤获取到的特征点的方向 θ 得到的旋转矩阵 R_θ 来矫正矩阵 Q ，使的描述子具有旋转不变性，如式 3-12 所示。

$$\begin{aligned} Q &= \begin{pmatrix} x_1, x_2, \dots, x_n \\ y_1, y_2, \dots, y_n \end{pmatrix} \\ R_\theta &= \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \\ S_\theta &= R_\theta S \end{aligned} \quad (3-12)$$

(4) rBRIEF 描述子匹配

ORB 特征点的配对是通过描述子的汉明距离作为判断依据。计算两个特征点通过 BRIEF 描述子生成的二进制码之间的汉明距离来判断这两个点是否匹配。当选取的是 256 位的 BRIEF 描述子时，当匹配的位数小于一半的，认为这两个特征点不匹配，然后从所有匹配数量大于一半的特征点中选择匹配数目最多的作为最

后的匹配特征点对。

3.3.2 图像增强算法

通常使用直方图均衡化算法(Histogram Equalization , HE)进行图像增强, 原始的 HE 方法只是考虑了图像的整体, 而没有考虑图某一部分的对比度。而自适应直方图均衡化(Adaptive Histogram Equalization , AHE)在此基础上做出了改进。AHE 通过计算图像局部的直方图来重新分布图像的亮度值。相对于普通的直方图均衡化它考虑到图像的局部对比度, 可以得到更多的细节信息。对比度被限制的自适应直方图均衡化方法是在 AHE 的基础上发展而来的, CLAHE 相比于 AHE 限制了对比度来去除噪声的影响, 同时利用双线性差值的方法加快了计算速度。CLAHE 算法有助于大量减少图像噪声的影响, 同时解决了其它直方图均衡化带来的亮度饱和的问题。CLAHE 算法的主要分为三步:

(1) 划分图像得到 $M=m*n$ 个子图像, 并且子图像间不能有重叠, 根据实际需求调整子图像的个数 M , M 越大增强效果越大, 对于每一个子图像进行直方图均衡化求解灰度变换函数。

(2) 对于每个子图分开处理。计算子图的每个灰阶能分配的像素个数。然后计算平均值 N_{avr} , 如式 3-13 所示:

$$N_{avr} = \frac{N_{CR-X_p} N_{CR-Y_p}}{N_{gray}} \quad (3-13)$$

其中 $N_{CR-X_p} N_{CR-Y_p}$ 表示 x 轴和 y 轴的像素个数。 N_{gray} 表示灰度级的数量。 $N_{cliplimit}$ 表示实际剪切的极限值, 如式 3-14 所示:

$$N_{cliplimit} = N_{clip} N_{aver} \quad (3-14)$$

其中 N_{clip} 是裁剪系数。然后对于子图像的灰度直方图进行裁剪, 由于需要保证灰度直方图面积不变, 把裁剪值均匀的分布在整個灰度区间上, 直到裁剪剩余像素个数为零。其中裁剪的像素数目为 N_{sum} , 平均到每个灰度阶的像素个数为 N , 计算公式如式 3-15 所示:

$$N_{acp} = \frac{N_{\sum clip}}{N_{gray}} \quad (3-15)$$

(3) 裁剪完成后, 对于每个子图的灰度直方图进行直方图均衡化。

(4) 通过变换函数得到子区域的中心像素的灰度值。然后对其它像素通过相邻

的四个子图像的中心像素进行双线性差值的方法得到。CLAHE 算法在对每个子区域进行直方图均衡化后，将中心的像素点作为参考点，参考点的像素值就是它本身的像素值。其它待估计点的像素值通过和它相邻的四个样本点进行双线性插值得到。待估计点 $M=(x,y)$ 。像素值为 I_M 。四个样本点是 $P_1=(x_1,y_1)$, $P_2=(x_2,y_1)$, $P_3=(x_1,y_2)$, $P_4=(x_2,y_2)$ ，它们的像素值为 I_1, I_2, I_3, I_4 。点 M 的像素计算如式 3-16:

$$\begin{aligned} I_M &= C[AI_1 + BI_2] + D[AI_3 + BI_4] \\ A &= \frac{y_2 - y}{y_2 - y_1} \\ B &= \frac{y - y_1}{y_2 - y_1} \\ C &= \frac{x_2 - x}{x_2 - x_1} \\ D &= \frac{x - x_1}{x_2 - x_1} \end{aligned} \quad (3-16)$$

对于分布在样本点形成的正方形区域以外的点，如果分布在图像的角落，就直接使用离这个像素点最近的样本点的像素值作为该点的像素值，如果分布图像的边缘，就使用离这个带估计点最近的两个点进行线性差值得到该点的像素值。

3.4 改进的 ORB 算法

3.4.1 优化 ORB 算法的提取过程

首先对于双目相机获取的图像序列进行预先处理。使用 CLAHE 算法对图像序列进行图像增强。减少因为天气和光照条件影响图像的后续处理，提高图像匹配的准确性。对输入的不同图像类型分别处理。如果图像是彩色图片，直接进行 CLAHE 图像增强效果较差。首先将图像的 RGB 三通道分离，然后对于三个通道分别处理，最后将处理完成的三个通道进行合并。如果是灰度图直接使用 CLAHE 进行处理。在第四小节可以得出 ORB 特征不具备尺度不变性。

本文借鉴了文章[58]的观点，在提取特征之前首先构建图像金字塔。构建的图像金字塔中间每一层仅对上一层图像进行降采样，并不进行高斯模糊处理，然后在相邻的三层图像上通过非极大值抑制找到特征点。

3.4.2 优化 ORB 匹配的筛选过程

ORB 特征的匹配是通过 BRIEF 描述子匹配，计算两个特征点描述子的汉明距离，表现为二进制编码的匹配。当距离小于阈值时认为匹配成功，但是匹配完成后

仍然有很多误匹配,所以仍然需要进一步剔除误匹配点提高匹配的正确率。张毅^[59]等人在论文中使用特征点自身的信息对使用最近邻匹配后的 SURF 进行误匹配剔除。本文考虑结合 Hamming 距离设定阈值的误匹配剔除算法,随机抽样一致性算法(Random Sample Consensus RANSAC)^[24]算法和特征点的自身信息进行误匹配剔除。因为使用多种方法结合操作,所以提高了匹配筛选的精度,为后续的立体匹配提供正确率更高的匹配对。算法的主要过程是:

(1)汉明距离设定阈值的误匹配剔除算法

首先计算所有匹配特征点间的汉明距离。汉明距离越小表示特征点的匹配度越高。得到特征点间汉明距离的最大值和最小值,把最小汉明距离的 10 倍作为阈值,再次遍历匹配的特征点,把汉明距离大于设定阈值的特征点剔除掉,剩下的特征点参与下一步的筛选。该过程的伪代码如算法 1 所示:

算法 1 Hamming 距离设定阈值的误匹配剔除算法

```

输入: 匹配的特征点数目 n, 匹配数据的集合 matches
输出: 筛选后的匹配点集
FUNCTION HammingDistanceLimit(n,matches)
    double max_dist = 0;
    double min_dist = 100;
    for int i=1 to n do    //循环便利获取距离的最大和最小值
        distance = matches[i].distance;
        if(distance < min_dist)
            min_dist = distance;
        else if(distance > max_dist)
            max_dist = distance;
    vector<DMatch> good_matches; //定义匹配点的输出
    for int j=0 to n do    //按照阈值筛选满足条件的匹配
        if(matches[i].distance < 10*min_dist)
            good_matches.push_back(matches[i]);
    return good_matches

```

(2)RANSAC 误匹配剔除

RANSAC 算法中最主要的部分是寻找最佳变换矩阵,所以需要迭代多次,随机采样大量的匹配点。由于需要迭代多次进行重采样和求解变换矩阵,加大了计算量,所以本文在通过最近邻算法对于变换矩阵进行一个筛选判定,然后在进行内点的筛选,提高运算效率。设定算法的迭代次数为 k。主要过程如下:

步骤一: 通过给定的数据集 P 中随机取出四对不共线的点计算变换矩阵 H 得到初始化模型 M。

步骤二: 对于图像中的特征点找到待匹配图片中欧氏距离最近的点和距离次

近的点，距离分别是 $l_{最近}$ 和 $l_{次近}$ 的点，然后计算两者的比值，按照比值大小分成 n 组，然后从比值最小的组中随机抽取四组匹配点，然后根据模型 M 判断内点的个数，如果内点数量大于 2 那么就认为这个模型具有成为最优模型的可能，否则就认为这个模型作为最优模型的概率很小舍弃这个模型，回到步骤 1) 从新计算模型。

步骤三：在得到变换矩阵 H 后对数据集中的数据和通过变换矩阵计算出的结果比较，设定阈值如果这个差值小于阈值就把这个数据加入内点集 S 。

步骤四：更新算法的迭代次数， $n=n+1$ 。比较当前内点集 S 中元素数量和最佳内点集 S^* 数量，然后判断是否更新最佳内点集，当 S 的元素数目大于 S^* 时， $S=S^*$ 。

步骤五：判断迭代次数 n 和设定的迭代次数，如果 $n=k$ 退出循环过程，否则继续从步骤一开始执行。

(3) 通过匹配点自身信息进行筛选

(A) 通过特征点间距离比筛选

通过欧式距离比，利用图片自身的比例信息，可以忽略两幅图片的尺度信息更好的剔除因为尺度变换引起的误匹配，但如果待处理的匹配点满足上述计算但本身有可能是误匹配点。所以在上述计算的基础上在进行特征点间连线的角度差来判定是否是误匹配。主要过程是，设原图像为 S ，待匹配图像为 S' ，设 S 的特征点集合为 A ， S' 的特征点集合是 A' ，它们各自的元素为 $a_1, a_2, \dots, a_n \in A$ ， $a'_1, a'_2, \dots, a'_n \in A'$ 。选取匹配特征点 i 和 i' ，特征点 i 到图像 S 其它特征点的欧式距离设为 D ， $d_{i1}, d_{i2}, \dots, d_{in} \in D$ ，特征点 i' 到图像 S 的欧式距离为 D' ， $d'_{i1}, d'_{i2}, \dots, d'_{in} \in D'$ ，选取 l_{i1} 和 l'_{i1} 作为基准分别计算特征点 i 和 i' 到其它特征点的距离比值的差值，然后和设定的阈值 τ 相比较。伪代码如算法 2 所示：

算法 2 通过欧氏距离之比筛选特征点

输入：原图像点集 $A1$ 和待匹配图像点集 $A2$ ，阈值 τ

输出：筛选后满足筛选条件的特征点对应的链表 $result$

FUNCTION EuclideanDistMatch($A1, A2, \tau$)

$len = \text{length}[A1]$ // 获取点集 $A1$ 的特征点数量

 for $i = 1$ to len // 循环计算任意两个特征点间的距离

 for $j = 1$ to len

$D1[j] = \sqrt{(A1[i].x - A1[j].x)^2 + (A1[i].y - A1[j].y)^2}$

$D2[j] = \sqrt{(A2[i].x - A2[j].x)^2 + (A2[i].y - A2[j].y)^2}$

 for $t = 1$ to len // 计算任意两对匹配的特征点的比值

$sum = 0$

 if $D1[t] \neq 0$ and $D2[t] \neq 0$ then

$temp = |D1[0]/D1[t] - D2[0]/D2[t]|$

```

    if temp <  $\tau$  then
        sum = sum+1
    if sum > length(A)*0.8
        result->next = A1[i] .把筛选通过的特征点添加到链表尾部
return result

```

(B) 通过特征点连线间角度差

主要过程是，选取匹配特征点 i 和 i' 。特征点 i 到图像 S 其它特征点的连线为 L ，并且 $l_{i1}, l_{i2}, \dots, l_{in} \in L$ 。特征点 i' 到图像 S' 的其它特征点的连线为 L' ， $l'_{i1}, l'_{i2}, \dots, l'_{in} \in L'$ 。选取 l_{i1} 和 l'_{i1} 作为基准分别计算它们和图像中其它连线的角度，然后求解角度差值和设定的阈值 ρ 相比较，如果结果小于 ρ 那么认为这个匹配是一个正确的匹配。该过程伪代码如算法 3 所示：

算法 3 通过角度差筛选特征点

```

输入：原图像点集 A1 和待匹配图像点集 A2，阈值  $\rho$ 
输出：筛选后 A1 对应的点集对应的链表 result
FUNCTION AngleMatch(A1,A2,  $\rho$ )
len = length[A1]
for i = 1 to len
    sum = 0
    for j = 1 to len
        Theta1[j] = CalAngle(A1[i],A1[0],A1[j])
        Theta2[j] = CalAngle(A2[i],A2[0],A2[j]) //计算连线间的夹角
    for t = 1 to len
        temp = |Theta1(t) - Theta2[t]|
        if temp <  $\tau$  then
            sum = sum+1
        if sum = length(A)*0.8 then
            result->next = A1[i]
return result

```

3.4.3 改进后的 ORB 算法流程

改进后的 ORB 算法的流程图如图 3-6 所示。

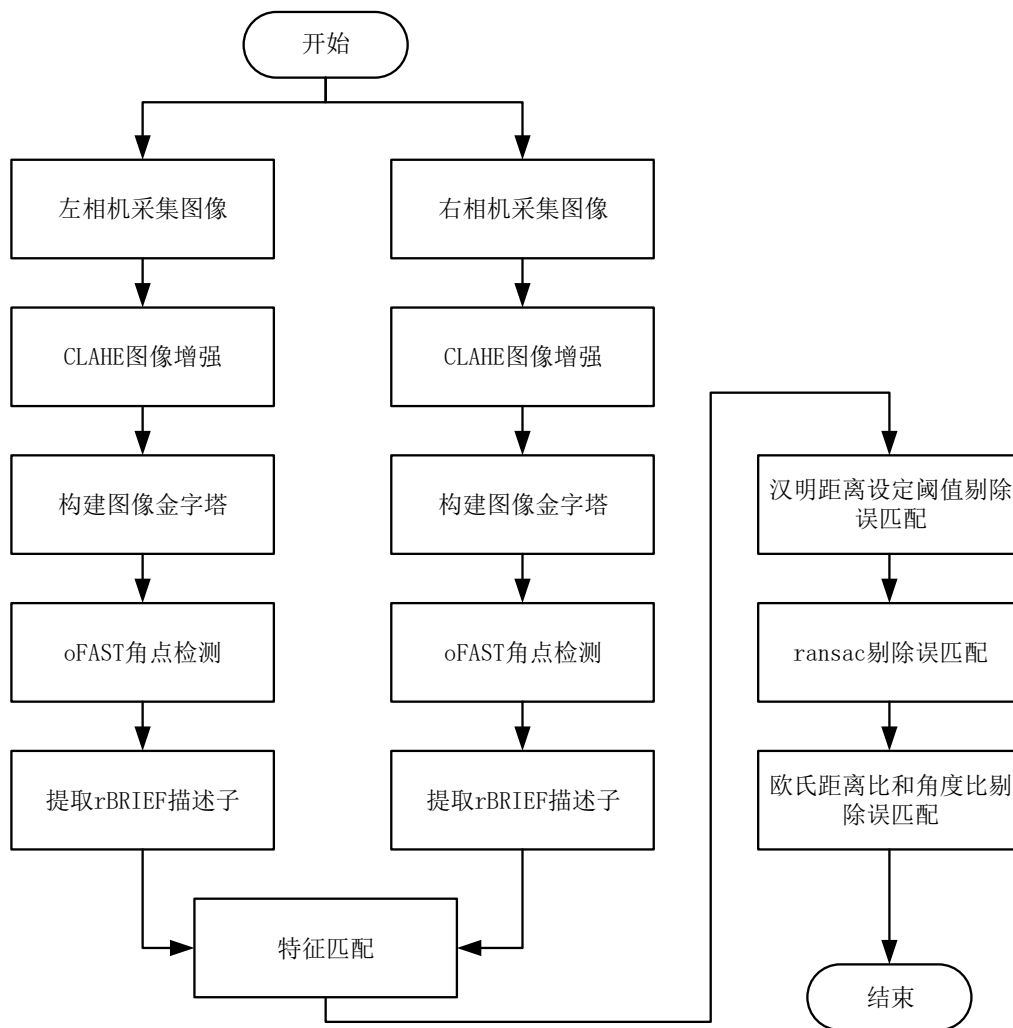


图 3-6 改进的 ORB 算法流程图

首先通过 CLAHE 对于双目相机获取的图像序列进行预处理，得到噪声较少的图像序列。然后构建图像金字塔，提高 ORB 算法的尺度不变性。在前两步的基础上提取 oFAST 角点和 rBRIEF 描述子。对于左右相机图像提取的描述子进行匹配得到初始匹配对。为了提高匹配对的精度，再通过改进的误匹配剔除算法对于误匹配进行剔除，从而得到匹配精度较高的匹配对。

3.5 实验结果及分析

实验仿真环境是 ubuntu16.04 操作系统。实验仪器使用的是 Intel i5 处理器，8G 内存的电脑。在本章实验中，主要将上文介绍过的 SIFT，SURF，ORB 三种特征以及改进后的 ORB 特征进行了对比实验，测试了四中算法的性能，并且进行分析。然后对于提出的误匹配剔除算法和其它剔除算法进行对比实验，并且进行分析。

3.5.1 多种图像特征提取分析

在本节试验中，主要分析了 SIFT，SURF，ORB 以及改进的 ORB 算法提取特征的效果和速度。实验使用的图片是鲁汶大学提供的标准图集集中的 leuven 图集，该图集有 6 张图片，图片是 PPM 格式，拍摄的是同一场景不同光照下的效果。对于 leuven 图集中图像 1 的特征点提取效果如图 3-7 所示，图片中红色点是提取的特征点。



(a)原图



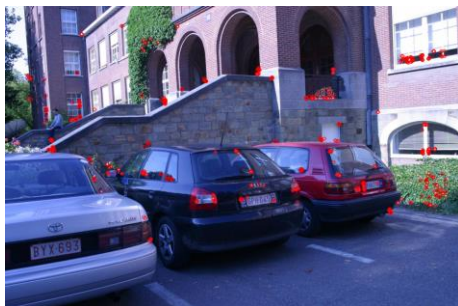
(b)SIFT 特征



(c)SURF 特征



(d)ORB 特征



(e)改进的 ORB 特征

图 3-7 SIFT，SURF，ORB 以及改进的 ORB 特征提取

SLAM 系统对于特征提取的实时性要求很高，所以比较了各个特征的提取时间，由于每个特征提取的特征点数目不一样所以计算了平均提取时间，更加具有代表性。对于图 3-7 所对应的图片进行十次提取实验得到的平均结果如表 3-1 所示：

表 3-1 SIFT, SURF, ORB 以及改进的 ORB 特征提取比较

特征分类 结果参数	SIFT	SURF	ORB	改进的 ORB 特征
提取数量	531	596	450	450
提取时间(ms)	353.02	214.46	18.37	78.90
平均提取时间(ms)	0.665	0.359	0.041	0.175

针对数据集中六幅图片，分别进行四种特征的提取实验。每幅图片提取十次取平均值，得到四种特征的平均提取时间所对应的折线图如图 3-8 所示。

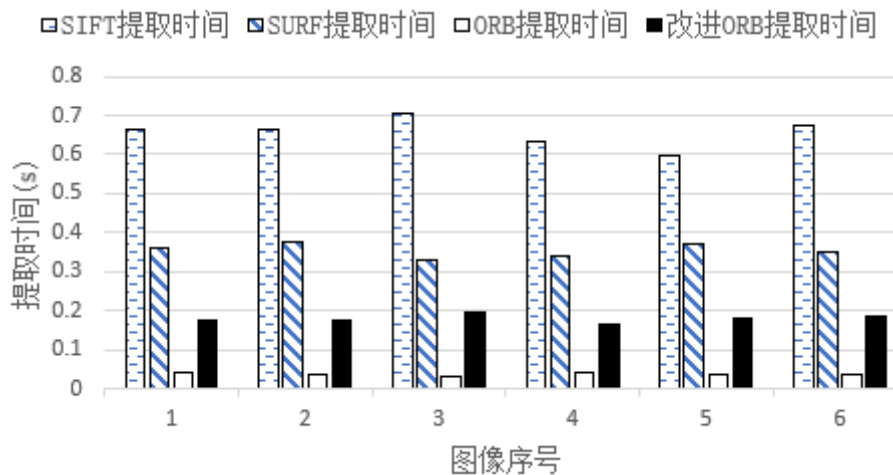


图 3-8 四种算法平均提取时间

通过表 3-1 和图 3-8 可以得到，在平均提取时间上 $SIFT > SURF > \text{改进的 ORB 特征} > ORB \text{ 特征}$ 。从表中可以看出 SIFT 平均提取时间大约是 ORB 的 10 倍，SURF 的平均提取时间大约是 ORB 的 8 倍，所以 ORB 特征的提取速度远快于其它两种特征，更符合 SLAM 系统对于实时性的要求。改进的 ORB 特征由于在特征提取之前使用 CLAHE 算法对图像进行了图像增强，减少噪声增强对比度，并且构建了图像金字塔来解决 ORB 不具备尺度不变性的缺点，所以在平均提取时间上要比原本的 ORB 特征略慢，但是仍然远低于 SIFT 和 SURF 特征的平均提取时间，可以满足 SLAM 系统实时性的要求。

3.5.2 多种图像特征匹配分析

在这一节中，主要分析了 SIFT，SURF，ORB，以及改进的 ORB 特征在不同光照条件下的匹配效果和匹配速率，实验使用的图集和上一实验图集相同，同时 leuven 图集提供了原匹配图片和目标匹配图片之间变换的单应矩阵(homography)。

(1) 不同光照条件下匹配效果如图 3-9 所示：



(a)SIFT 特征匹配



(b)SURF 特征匹配



(c)ORB 特征匹配



(d)改进的 ORB 特征匹配

图 3-9 不同光照条件下四种特征的匹配

通过特征点的匹配数量。正确匹配数量(总匹配数量-误匹配数量)。匹配精度 (匹配精度=(正确匹配数量/总匹配数量))。以及平均匹配时间来分析算法匹配的效果。对于匹配的正确性验证通过式 3-26 计算：

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \quad (3-26)$$

$$X_2 = HX_1$$

其中 H 是两幅图片间的单应矩阵，已经由数据库给出。 X_1 是原匹配图片的特征点 I 的齐次坐标。通过该公式可以计算得到特征点 I 在待匹配图片中的坐标 X_2 ，这个结果是理想结果，真实结果会有差距，通过公式 3-27 计算：

$$Z_{distance} = \sqrt{(x_2 - x_3)^2 + (y_2 - y_3)^2} \quad (3-27)$$

计算通过图像匹配得到的坐标和真实匹配点坐标 X_3 之间的距离，当这个距离小于阈值时认为匹配正确。不同光照条件下使用穷举法进行匹配，得到不同特征的匹配。一共进行十次实验取平均值得到的数据如表 3-2 所示：

表 3-2：不同光照条件下四种特征匹配数据

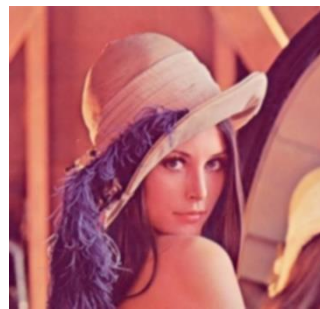
特征分类 结果参数	SIFT	SURF	ORB	改进的 ORB 特征
总匹配数量	531	596	450	450
正确匹配数量	274	337	266	281
匹配精度	51.6%	56.54%	59.11%	62.44%
总匹配时间(ms)	9.23	5.62	5.20	5.57
平均匹配时间(ms)	0.017	0.009	0.016	0.012

通过表可以看出在未经过筛选的匹配精度方面，改进的 ORB 特征 > ORB > SURF > SIFT，ORB 特征在光线发生变化时相对于 SIFT 和 SURF 匹配精度更好，而平均匹配时间三者相差不多。改进的 ORB 特征和原本的 ORB 特征在平均匹配时间上都是通过 BREIF 描述子来进行匹配，所以基本相同。而改进的 ORB 特征通过 CLAHE 先对图像进行增强在进行特征提取匹配，在光线较暗，噪声较多的场景下具有更好的匹配精度，所以改进的 ORB 特征更加符合 SLAM 系统对于光线较暗环境适应性的要求。

(2) 不同尺度条件下的特征匹配，实验结果如图 3-10 所示：



(a)Lenna 原图



(b) $\sigma = 3$ 的高斯模糊图片



(c)SIFT 特征匹配



(d)SURF 特征匹配



(c)ORB 特征匹配



(d)改进的 ORB 特征匹配

图 3-10 不同尺度条件下四种特征的匹配

统计匹配结果的数据如表 3-3 所示:

表 3-3: 不同尺度条件下四种特征的匹配数据

特征分类 结果参数	SIFT	SURF	ORB	改进的 ORB 特征
总匹配数量	426	338	450	450
正确匹配数量	216	187	143	262
匹配精度	50.70%	55.32%	31.77%	58.22%
总匹配时间(ms)	5.74	2.79	5.43	5.95
平均匹配时间(ms)	0.014	0.009	0.012	0.013

通过表 3-3, 在匹配精度方面改进的 ORB 特征>SURF>SIFT>ORB, 当尺度发生变化时, SIFT 和 SURF 相比于 ORB 来说具有更好的尺度不变性。SIFT 和 SURF 具有较好精度的原因是构建了多尺度空间。改进的 ORB 特征在尺度发生变化时匹配精度要好于原本的 ORB 特征, 原因是改进的 ORB 特征构建了图像金字塔, 更能适应图像尺度变化。

3.5.3 改进的误匹配去除算法

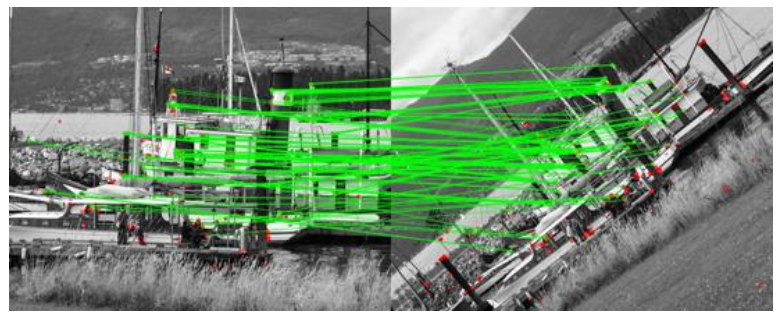
图像在匹配过程中因为旋转, 尺度变化以及噪声等原因会造成一部分的错误匹配, 所以需要对误匹配点进行剔除。在本节实验中会对汉明距离设定阈值的方法(方法一), ransac 误匹配剔除算法(方法二)以及改进的误匹配剔除算法(方法三)分别对改进的 ORB 算法进行验证。实验结果如图 3-11 所示:



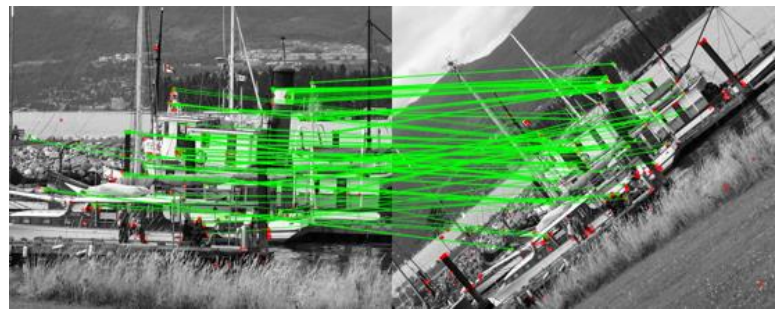
(a)未经筛选匹配



(b)经方法一筛选后的匹配



(c)经方法二筛选后的匹配



(d)经方法三筛选后的匹配

图 3-11 误匹配剔除算法对比

在图 3-11 中第一张是没有进行筛选的图片，第二三四张是分别经过三种算法筛选后的图片。通过十次实验取平均值，得到的误匹配剔除实验数据如表 3-4 所示：

表 3-4: 不同算法筛选匹配的数据

特征分类 结果参数	方法一	方法二	方法三
总匹配总数	450	450	450
筛选后的匹配数量	315	233	205
筛选后的正确匹配	265	213	192
筛选后的匹配精度	84.12%	91.41%	93.65%
匹配筛选时间(ms)	0.0328	2.168	1.732

通过表 3-4 可以看出,在匹配精度上改进的误匹配剔除算法要好于汉明距离设定阈值的方法以及 RANSAC 算法。三种方法中汉明距离设定阈值的方法仅仅对于描述子之间的汉明距离进行了计算,然后通过设定阈值剔除掉阈值大于改设定值得匹配点对,然而由于噪声以及尺度旋转等因素影响描述子之间的汉明距离计算会有一定的误差,所以在最后的匹配效果上最差。RANSAC 算法通过多次迭代寻找一个最优单应矩阵 H ,使得最优矩阵适应更多的特征点,特征点筛选后的匹配精度较高但是相对筛选时间比较长。改进的误匹配剔除算法通过三步来筛选正确的匹配。首先是通过匹配距离设定阈值的方法进行初步筛选,可以剔除掉明显的误匹配点,减少下一步进行筛选的匹配数量,明显加快筛选速度。然后通过改进的 RANSAC 算法,通过最近邻的思想选择匹配距离比最近和次近的比值较小的点验证单应矩阵的正确性,从而加快迭代速度。最后通过距离比和角度差再次剔除误匹配可以提高匹配的精度。从实验结果可以看出方法三在匹配筛选时间上少于 RANSAC 算法,能够满足 SLAM 系统对于实时性的要求。

3.6 本章小结

本章主要研究了双目相机模型和立体匹配的相关技术。首先从针孔模型引出了双目相机模型。然后推导了空间坐标点在世界坐标系下的坐标到相机成像平面的二维坐标的转换公式。推导了通过视差来计算空间坐标点深度计算公式。视觉里程计在进行特征提取之前,需要对双目相机进行标定,本文采用张正友标定法,并且对相机进行畸变矫正。

在介绍完双目相机模型之后,针对双目视觉里程计进行立体匹配需要使用的图像特征做了具体分析。SIFT 特征通过构建 DOG 图像金字塔模型解决图像尺度变化对特征匹配的影响,并且具有较好的旋转和亮度不变性。SIFT 匹配精度较好,特征数目较多,但是提取速度较慢实时性较差。SURF 是在 SIFT 的基础上发展而

来的，是 SIFT 的改进版本，SURF 为了提高计算速度使用了海森矩阵和积分图，并且使用积分图得到 haar 特征得到关键点方向。再然后介绍的是 ORB 算法。ORB 是在 FAST 角点和 BRIEF 描述子的基础上改进得到的。通过灰度质心法得到角点的方向，通过汉明距离进行特征点间的匹配。在尺度变换方面 SURF 性能要优于 SIFT，而由于 ORB 没有构建金字塔模型，在尺度变换时匹配效果最差。在提取速度方面 ORB 提取的是 oFAST 角点所以速度最快，SURF 要优于 SIFT，三者之间各差了一个数量级。而由于 SLAM 对于实时性要求很高，所以选取提取速度最快的 ORB 算法作为立体匹配的特征点。

考虑到应用环境有可能光线较暗，相机图片噪声比较多，所以首先通过使用 CLAHE 提高图像对比度，提高匹配的效果，同时对于 ORB 算法构建空间金字塔模型，提高 ORB 算法对于尺度变化的适应性。

在图像匹配过程中会有一定比例的误匹配。在本章介绍了误匹配剔除的方法。首先获取所有匹配特征之间汉明距离的最大值，把距离大于最大值 80% 作为误匹配点剔除掉。然后通过改进的 RANSAC 算法对匹配特征进行再次筛选，最后通过特征间的距离比和角度差再次筛选出一部分误匹配，通过实验证明这种误匹配剔除算法能够明显提升匹配精度，减少误匹配，在精度上为立体视觉的计算提供保证。

第四章 融合 IMU 和双目相机的 SLAM 算法的研究

4.1 引言

Montemerlo^[60-61]于 2003 年提出了 FastSLAM 算法。FastSLAM 算法是在 Rao-Blackwellised 粒子滤波(RBPF)算法的基础上发展而来的。大部分 SLAM 算法把定位与建图的过程一起考虑, FastSLAM 算法基于 RBPF 的分解思想将状态估计分解成路径估计以及以路径为基础的地图估计两个部分, 降低了计算的复杂性。

FastSLAM 算法是使用递推的方法计算权重, 也就是序贯重要性 ‘SIS’ 方法进行采样。这种重要性采样方法会存在一定的问题, 例如粒子权值退化现象, 也称为粒子贫化, 在论文[62]中, A Doucet 给出了该重要性采样方法必然会导致粒子退化的理论证明。该现象表现为粒子集中的粒子经过多次迭代以后, 除了个别粒子权值较大, 其它粒子权值很小的现象。并且这样通过权值估计机器人的位姿会主要由个别粒子主, 此时需要对于粒子进行重采样。重采样是通过复制权值较大的粒子, 抛弃权值较小的粒子增加有效粒子的数目, 经过多次迭代后可能所有的粒子都来源于相同的一个粒子, 这样又会带来粒子耗尽的问题。

针对粒子贫化和粒子耗尽的问题 Pei F^[63]等人将人工鱼群算法引入 FastSLAM 算法, 可以解决 FastSALM 算法在重采样造成的粒子贫化问题。Wen Shiguang^[64]等人提出了一种基于蚁群算法的重采样方法, 这个方法很好的解决了 FastSLAM 算法中粒子耗尽的问题。周武^[65]等人提出了一种基于遗传算法的 FastSLAM 改进算法, 使用无损卡尔曼滤波进行对于粒子后验分布的估计, 通过遗传算法提高粒子的多样性, 解决粒子耗尽问题。王开宇^[66]于 2016 发表的论文中, 提出将 PSO 算法的思想引入到 SLAM 问题中, 通过 PSO 算法优化粒子的位姿信息, 使得更新后的粒子更加靠近机器人的真实位姿。但是该算法还存在一些问题, 由于所有样本同时向高似然区域运动, 这个运动方向是基于虽然有效的解决了粒子退化问题, 但是会降低粒子的多样性, 可能使粒子陷入局部最优。

本文在王开宇的粒子群优化的 FastSLAM 算法的基础上引入了免疫算法。提出了基于 PSO 和 IA 融合的快速 SLAM 优化算法。通过在 FastSLAM 算法中引入 PSO 算法使粒子向高似然区域运动, 减少粒子退化; 引入免疫算法提高粒子的多样性同时, 防止粒子陷入局部最优, 同时通过复制过程在一定程度上减少了粒子的退化。将 PSO 和 IA 两种算法融合, 提高了粒子集中粒子的多样性, 改善粒子的重采样过程造成的粒子耗尽问题, 提高了位姿的估计精度。

4.2 FastSLAM 算法概述

4.2.1 FastSLAM 算法分析

根据 Murphy 的对于动态贝叶斯网络的分析可以得到, 在机器人路径估计已知的情况下, 路标点(m_i, m_j)是相互独立的, 可以把后验概率分布 $p(s^t, m | z^t, u^t, n^t)$ 分解。分解过程如式 4-1 所示, :

$$p(s^t, m | z^t, u^t, n^t) = p(s^t | z^t, u^t, n^t) p(m | s^t, z^t, u^t, n^t) \quad (4-1)$$

其中 $s^t = \{s_1, s_2, \dots, s_t\}$ 表示机器人在 t 时刻的位姿信息。 $z^t = \{z_1, z_2, \dots, z_t\}$ 表示在 t 时刻的观测信息。 m 表示观测到的特征点, 并且特征点的数量是 M 。 $u^t = \{u_1, u_2, \dots, u_t\}$ 表示外部输入的信息。 n^t 表示机器人所有数据关联的集合。 概率 $p(s^t, m | z^t, u^t)$ 表示机器人的后验概率分布。 概率 $p(s^t | z^t, u^t)$ 表示对机器人运动路径的估计, 用粒子滤波器进行估计。 概率 $p(m | s^t, z^t, u^t)$ 表示特征估计, 使用 EKF 进行估计, 一个特征对应一个扩展卡尔曼滤波器。 如果使用 n 个粒子估计机器人位姿, 每一个粒子代表机器人的一个可能路径, 那么总共需要 $n \cdot M$ 个卡尔曼滤波器。 把地图的估计分解为 M 个独立的特征估计问题, 分解过程如式 4-2 所示:

$$p(m | s^t, z^t, u^t) = \prod_{i=1}^M p(m_i | s^t, z^t, u^t) \quad (4-2)$$

4.2.2 FastSLAM 算法的步骤

FastSLAM 算法的具体过程如下所示:

(1) 粒子的初始化

根据机器人的初始坐标, 随机生成 n 个粒子, 得到粒子的集合 $\Omega = \{s_t^i, w_t^i\}$, 其中 s_t^i 表示第 i 个粒子, w_t^i 表示第 i 个粒子的权值, 协方差矩阵为 P_t , 过程噪声的协方差矩阵是 Q_t 。

(2) 根据运动模型预测采样粒子的位姿

根据 $t-1$ 时刻粒子集中每个粒子的状态和运动模型, 计算 t 时刻粒子所在的位置和姿态, 过程如式 4-1 所示, 同时通过状态转移函数 h 关于状态向量的 Jacobian 矩阵 Δh 和过程噪声方差求解粒子 t 时刻的运动的协方差矩阵, 过程如式 4-3 所示:

$$P_t = \Delta h P_{t-1} \Delta h^T + Q_t \quad (4-3)$$

(3) 将新观测到的路标和历史路标进行数据关联

通过观测方程 4-8 求得 t 时刻所有的观测路标，同时把获取到的结果和 $t-1$ 时刻的粒子集中的粒子所观测到的路标进行数据关联。

(4) 获取重要性概率密度函数并且采样

通过 KF 计算粒子集中每个粒子位姿估计的均值和方差，同时由获取的均值和方差构建新的高斯分布函数作为粒子的重要性概率密度函数，也就是提议分布 $\pi(X_t^R | X_{0:k-1}^{R(i)}, Z_{0:k}^{(i)}, u_k)$ ，根据提议分布进行采样更新。

(5) 计算粒子权重

在上一步中计算出了提议分布，通过目标分布除以提议分布得到粒子权重的公式如式 4-4 所示：

$$\omega_k^i = \frac{p(X_t^R | Z_{0:k}^{(i)}, u_{0:k}, n_{0:k})}{\pi(X_t^R | X_{0:k-1}^{R(i)}, Z_{0:k}^{(i)}, u_k)} \quad (4-4)$$

(6) 权值归一化

对粒子进行归一化处理，处理过程如式 4-5 示：

$$w_t^i = w_t^i / \sum_{i=1}^n w_t^i \quad (4-5)$$

(7) 对粒子进行重采样

首先需要判断是否需要重采样，通过计算有效粒子的数目 N_{eff} ，如式 4-6 示：

$$N_{eff} = 1 / \sum_{i=1}^n (w_k^i)^2 \quad (4-6)$$

设定阈值 Φ 进行判断，如果 $N_{eff} > \Phi$ 就不进行重采样，直接进行地图更新。否则根据粒子的权值 w_t^i 对粒子进行重采样。重采样的过程是复制权值大的粒子，去掉权值小的粒子进行，保持粒子集中粒子数目稳定。

(8) 更新环境地图

机器人在运动过程中，粒子集中每一个粒子对于 M 个观测路标都有自己的估计，所以每个粒子都有自己的地图，这就把地图估计的问题分解为 M 个路标估计的问题。通过 EKF 估计特征的后验分布函数 $p(m_j | x_{1:k}, z_{1:k}, u_{1:k})$ 。

(9) 计算位姿，判断是否需要继续执行。

一次循环完成后，粒子集 Ω 中的粒子完成了状态的更新，通过 n 个粒子获取 t

时刻位姿信息，如式 4-7 所示：

$$X = \sum_{i=1}^n w_t^i s_t^i \quad (4-7)$$

当完成计算后如果还需要计算下一时刻的位姿，算法就会进入下一个循环，跳转回步骤(2)。

4.3 基于免疫算法和粒子群算法融合优化的 FastSLAM 算法

4.3.1 建立 IA-PSO-FASTSLAM 系统模型

1. 运动模型

在 SLAM 系统中机器人的运动模型可以表示为式 4-8：

$$s_t = h(s_{t-1}, u_t) + \delta_t \quad (4-8)$$

其中 s_t 表示机器人在 t 时刻的位姿信息。 s_{t-1} 表示机器人在 $t-1$ 时刻的姿态信息。 u_t 表示， h 表示机器人的运动方程， δ_t 表示机器人运动方程的高斯噪声。在本实验中使用惯性测量单元 IMU 的数据构建运动模型，设状态向量 X 为式 4-9

$$X_t = (S_t, V_t, \delta_t)^T \quad (4-9)$$

其中 $S_t(x, y, z)$ 表示机器人在世界坐标系下的位姿信息。 $V_t(v_x, v_y, v_z)$ 表示机器人在世界坐标系下的速度信息。 $\delta_t = (\delta_x, \delta_y, \delta_z)$ 表示重力加速度的分量和偏置量。在第二章中分析了惯性测量单元 IMU 的加速度计模型，可以得到真实加速度 a 的公式为 4-10：

$$a = a_{ce} - g - b_a - n_a \quad (4-10)$$

真实加速度通过加速度的测量值减去重力的分量和加速度的偏移量以及随机噪声。同时在第二章中分析了 IMU 位移推算的方法，加速度从 IMU 坐标系到世界坐标系的旋转是 R_{CL} ，可以得到在世界坐标系下的加速度如式 4-11 所示：

$$\begin{aligned} \delta &= g + b_a \\ a_{cl} &= R_{CL}(a_{ce} - \delta) - n_{cl} \end{aligned} \quad (4-11)$$

其中 δ 是重力加速度的分量和加速度的偏移量的和， n_{cl} 是随机噪声， a_{cl} 是在世界坐标系下的加速度的值。可以由加速度计得到机器人状态的变化如式 4-12 所

示，其中 W_v ， W_s 是速度和位移的噪声：

$$\begin{aligned} V_t &= V_{t-1} + (R_{CL} a_{ce} - R_{CL} \delta) \cdot \Delta t + W_v \\ S_{t+1} &= S_t + (R_{CL} a_{ce} - R_{CL} \delta) \cdot \Delta t^2 / 2 + W_s \end{aligned} \quad (4-12)$$

通过式 4-13 可以得到系统的运动模型：

$$X_t = \begin{pmatrix} S_t \\ V_t \\ \delta_t \end{pmatrix} = \begin{pmatrix} 1 & \Delta t & -R_{CL} \cdot \Delta t^2 / 2 \\ 0 & 1 & -R_{CL} \cdot \Delta t \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} S_{t-1} \\ V_{t-1} \\ \delta_{t-1} \end{pmatrix} + \begin{pmatrix} R_{CL} \cdot \Delta t^2 / 2 \\ R_{CL} \cdot \Delta t \\ 0 \end{pmatrix} a_{ce} + \begin{pmatrix} W_s \\ W_v \\ 0 \end{pmatrix} \quad (4-13)$$

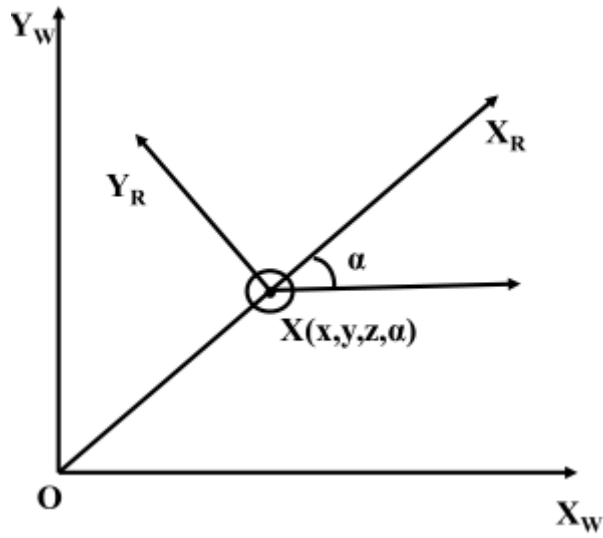


图 4-1 机器人运动模型

通过获取到的机器人在世界坐标系下的位姿 (x,y,z) ，可以得知在 k 时刻当前机器人的俯视图 4-1 所示世界坐标系下的坐标和旋转如式 4-14 所示，其中 α 是机器旋转角度：

$$\begin{aligned} \alpha &= \arctan \frac{y}{x} \\ \hat{X}_k &= \begin{pmatrix} x \\ y \\ z \\ \alpha \end{pmatrix} \end{aligned} \quad (4-14)$$

2. 观测模型

观测模型的作用是在 SLAM 系统中机器人的观测模型可以表示如式 4-15 所示：

$$z_t = f(s_t) + v_t \quad (4-15)$$

其中 z_t 是移动机器人在 t 时刻的观测值。 s_t 是移动机器人在 t 时刻的位姿信息。 f 是观测方程， v_t 是噪声。在本实验中把空间中的路标点作为观测量，而匹配的空间坐标点就是第三章获取的改进的 ORB 特征，通过在第三章求得双目相机匹配的路标点信息，从而求取路标点的空间坐标。求解模型如图 4-2 所示：

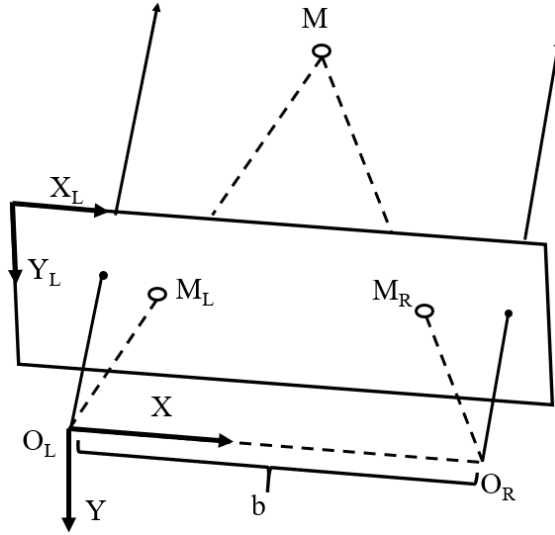


图 4-2 双目立体视觉模型

其中光心 O_L 在成像平面的坐标是 (c_x, c_y) ， O_R 在成像平面的坐标是 (c'_x, c'_y) ，基线距离为 b 。通过光轴平行模型求解 M 点相对于机器人坐标系的坐标 (X_c, Y_c, Z_c) 。 $M_L(x_L, y_L)$ 和 $M_R(x_R, y_R)$ 的视差是 d 。在第三章中计算了空间坐标点的深度信息，在这里将深度信息应用到三维坐标的计算。空间坐标点在相机坐标系下的坐标求解过程如式 4-16 所示：

$$d = x_L - c_x + c'_x - x_R$$

$$\begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 0 & f \\ 0 & 0 & 1/b & 0 \end{bmatrix} \begin{bmatrix} x_L \\ y_L \\ d \\ 1 \end{bmatrix} \quad (4-16)$$

调整三个坐标轴坐标的位置让 Z 轴指向重力加速度方向， X 轴指向机器人前

进的方向，Y 轴是和 X 轴在水平面上垂直向左的方向，如式 4-17 所示：

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} Z/W \\ -X/W \\ Y/W \end{bmatrix} \quad (4-17)$$

然后如图 4-3 所示，需要把空间特征点 M 的坐标从相机坐标系转换到世界坐标系中。

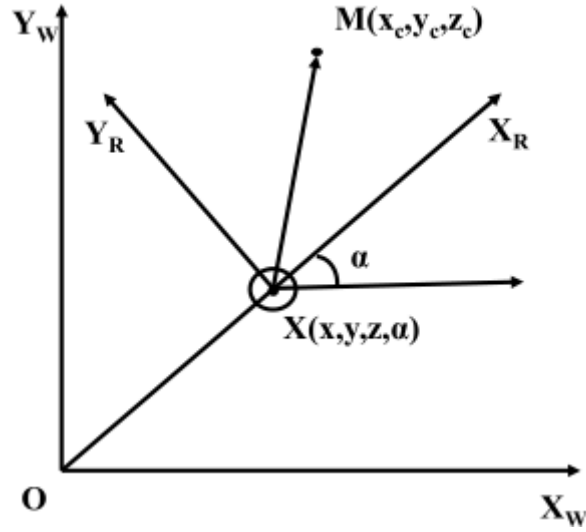


图 4-3 机器人观测模型

其中机器人的旋转角度为 α ，已经在式 4-14 求出。可以得到机器人的观测模型为式 4-18，其中 $v(k)$ 是观测噪声：

$$Z_k = \begin{pmatrix} x_k \\ y_k \\ z_k \end{pmatrix} = \begin{pmatrix} x + x_c \cos \alpha - y_c \sin \alpha \\ y + y_c \cos \alpha + x_c \sin \alpha \\ z_c + z \end{pmatrix} + v(k) \quad (4-18)$$

4.3.2 融合粒子群算法和免疫算法优化粒子集

原生的 FastSLAM 算法由于多次迭代会造成粒子退化现象，造成很少一部分粒子权值较大，而大部分粒子权值较小，造成后续的计算很大一部分用在权值较小的粒子上面，而这些粒子又对最终机器人的位姿估计作用很小，造成资源的浪费。如果通过原本的重采样算法进行重采样，复制权值大的粒子，去除权值小的粒子，增加有效粒子数目，又会造成最后所有的粒子可能是由很少的一部分粒子重采样

得来的，降低了粒子的多样性，造成粒子耗尽的问题。所以本文在王开宇^[66]基于粒子群优化的 FastSLAM 算法的基础上做出改进，通过融合粒子群算法和免疫算法优化粒子集缓解粒子退化和耗尽问题。

基于粒子群优化算法和免疫算法融合的粒子优化算法的具体优化过程如下所示：

(1) 在 t 时刻获取到系统输入的粒子集 $\Omega = \{s_t^i, w_t^i\}$ ，其中 s_t^i 表示粒子位姿， w_t^i 表示粒子的权重，粒子的数目是 M 。

(2) 通过人工免疫算法对于粒子集进行处理。把粒子集 Ω 的粒子作为初始抗体种群存入记忆单元，将机器人的真实位姿作为抗原，粒子集合中的每一个粒子作为抗体。由于每一个粒子的权值可以表明当前粒子位姿估计的准确程度，所以认为每个粒子的亲和力和它自身的权值相关，如果权值大那么亲和力就小，权值小亲和力就大，通过亲和力可以控制粒子的克隆数目以及粒子的变异程度。可以得到每个粒子亲和力的计算过程如式 4-19 所示：

$$f_{\text{affinity}} = 1 - \omega_t^i \quad (4-19)$$

(3) 计算完亲和力以后对于粒子进行克隆，计算粒子需要克隆的数目如式 4-20 所示：

$$N_{cl} = \text{round} \left(M * \cos \left(f_{\text{affinity}} \frac{\pi}{2} \right) \right) \quad (4-20)$$

通过 \cos 函数使得粒子复制的数目随着亲和力的增加而减少，也就是对权值高的粒子进行多复制，权值小的粒子减少复制，通过这种方法可以有效的抑制亲和力高的抗体而促进亲和力低的抗体，加快收敛速度。将新获得的粒子添加到粒子集合当中。然后通过亲和性对于粒子进行变异操作。通过变异操作使权值较大的粒子的位姿变化较小，而权值较小的粒子变化较大，从而使权值较小的粒子去寻找更好的位姿，增加粒子的多样性，变异过程计算如式 4-21 所示：

$$s_t^i = s_{t-1}^i + f_{\text{affinity}} * \text{random} \quad (4-21)$$

其中 random 是一个服从 $N(0,1)$ 分布的随机数。

(4) 然后计算抗体的排斥力，排斥力使用两个粒子位姿差的绝对值来表示，两个粒子离得越近排斥力越小，排斥力的计算公式如式 4-22：

$$f_{\text{reject}} = |x_t^i - x_t^j| \quad (4-22)$$

当粒子数目大于 M 时，把排斥力小于设定阈值的两个粒子中的一个抛弃，抛弃掉最为接近的粒子。通过式 4-23 计算粒子集中剩余粒子的权值。将粒子集中剩余的粒子作为下一阶段的输入。

$$\omega_t = \omega_{t-1} p(z_t | s_t^*) \quad (4-23)$$

(5) 通过上一过程保留了权值较低的粒子的同时增加了粒子集的多样性，缓解了粒子退化问题。然后对粒子进行粒子群优化。粒子群优化的过程使用全局最优解和局部最优解，使粒子向最优方向运动。对粒子集合中的每个粒子的速度和位姿进行更新，过程如式 4-24，4-25 所示

$$v_{in}^t = c_1 rand_1(p_{in}^{t-1} - x_{in}^{t-1}) + c_2 rand_2(p_{gn}^{t-1} - x_{in}^{t-1}) \quad (4-24)$$

$$x_{in}^t = x_{in}^{t-1} + v_{in}^t \quad (4-25)$$

x_{in} 表示当前机器人的位姿信息。其中 p_{in} 是粒子本身的局部最优解，用上一个循环的该粒子位姿估计值来表示。 p_{gn} 是粒子集全局最优解，使用权值最高的粒子位姿来表示。 v_{in} 表示粒子运动的速度，通过上一时刻的速度加上变化求得。 c_1 和 c_2 表示学习因子并且设定 $c_1=c_2=2$ 。通过位姿更新使得粒子向最优方向运动。PSO 寻优过程中使用的是如式 4-26 所示的适应度函数：

$$f_{fitness} = \exp[-\frac{1}{2R_k}(Y_{new} - Y_{pred})^2] \quad (4-26)$$

在本论文中这里 Y_{new} 指的是机器人观测到的信息路标的位姿， Y_{pred} 指的是通过地图信息和粒子位置的预测估计出的路标位姿信息，该适应度函数表明了当前粒子位置估计的准确程度，随着误差预测值和观测值距离的减小，适应度值会逐渐增大。通过设定阈值判断粒子是否到达了机器人真实位姿的附近。如果粒子的适应度值小于阈值，那么重复步骤 5，再次更新速度和位移，直到得到的粒子的适应度值大于阈值。

(6) 经过 PSO 优化完成后，通过式 4-4 再次计算粒子集中粒子的权值，对完成权值计算的粒子集合进行筛选，从中选出 N 个权值最大的粒子。

(7) 然后通过式 4-5 对粒子集中的粒子的权值进行归一化，然后对归一化后的粒子集通过 4-6 计算有效粒子数目，如果粒子数目少于设定的阈值则进行重采样过程。

融合算法的流程图如图 4-4 所示：

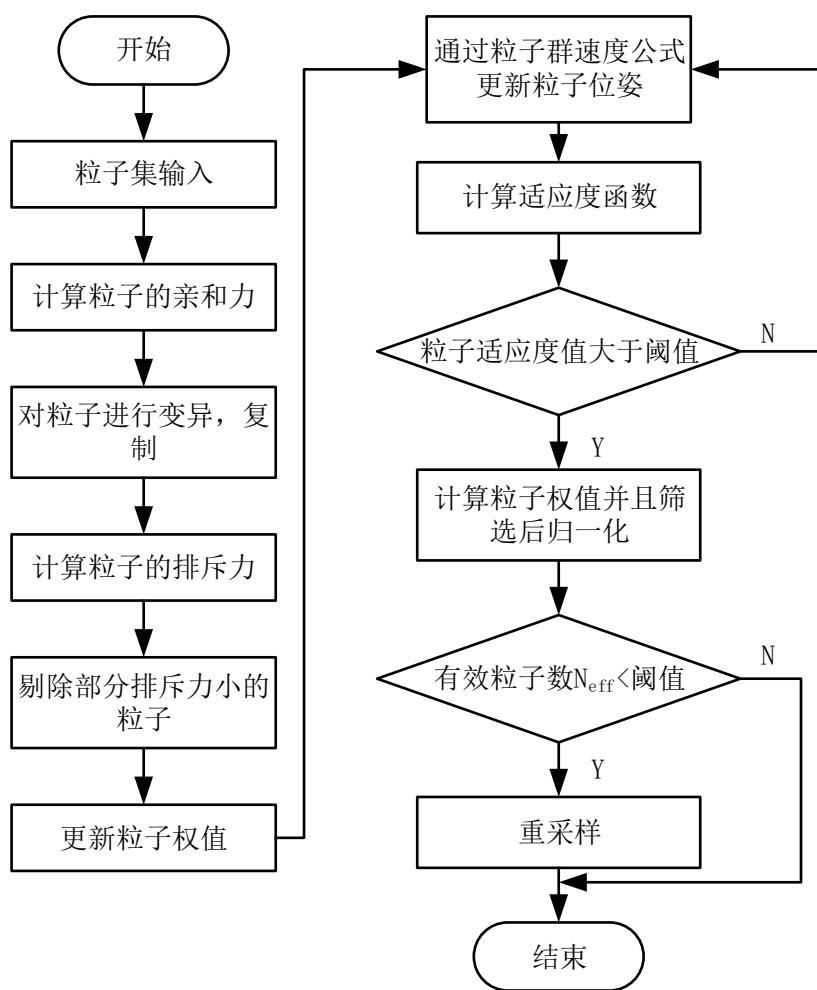


图 4-4 基于 PSO 和 AI 融合的粒子优化算法的流程图

4.3.3 设定优化算法的阈值

在粒子滤波的过程中由于存在粒子退化的问题，所以需要进行重采样来更新粒子增加有效粒子数，缓解粒子退化的现象。但是随着时间增长，有可能造成粒子损耗，使粒子集中的粒子都来源于个别粒子，降低了粒子的多样性。本文提出将人工免疫算法和粒子群优化算法融合来优化粒子集。粒子群算法可以使粒子向高似然区域运动，人工免疫算法可以提高粒子的多样性在一定程度上缓解粒子退化现象。但是在进行该步骤之前，需要通过公式 4-6 计算有效粒子的数目，总的粒子个数为 M ，当有效粒子数目 N_{eff} 在 $(0.5M, 0.7M)$ 之间时不需要进行粒子的更新处理，如果不在这个范围内对粒子集进行优化处理。

4.3.4 IA-PSO-FastSLAM 算法流程

整体算法的流程图如下所示：

(1) 粒子的初始化

根据机器人的初始坐标，随机生成 n 个粒子，得到粒子的集合 $\Omega = \{s_t^i, w_t^i\}$ ，其中 s_t^i 表示第 i 个粒子， w_t^i 表示第 i 个粒子的权值，协方差矩阵为 P_t ，过程噪声的协方差矩阵是 Q_t 。

(2) 根据运动模型预测采样粒子的位姿

根据 $t-1$ 时刻粒子集中每个粒子的状态和运动模型，计算 t 时刻粒子所在的位置。同时通过状态转移函数 h 关于状态向量的 Jacobian 矩阵 Δh 和过程噪声的方差求解粒子 t 时刻的方差如式 4-27 所示：

$$P_t = \Delta h P_{t-1} \Delta h^T + Q_t \quad (4-27)$$

(3) 将路标进行数据关联

通过观测方程求得 t 时刻所有的观测路标，同时把获取到的结果和 $t-1$ 时刻的粒子所观测到的路标进行数据关联，过程相互独立。

(4) 获取重要性概率密度函数

通过 KF 计算粒子集中每个粒子位姿估计的均值和方差。同时由获取的均值和方差构建新的高斯分布函数作为粒子的重要性概率密度函数。

(5) 通过 4-4 计算粒子权重

(6) 通过 4-5 将粒子权值归一化

(7) 通过 4-6 计算有效粒子的数目 N_{eff}

如果有效粒子数目满足式 4-28:

$$0.5M < N_{eff} < 0.75M \quad (4-28)$$

就进入步骤(9)直接进行地图更新，否则进入步骤(8)。

(8) 根据免疫和粒子群融合算法对加入的粒子集进行优化

得到优化后的粒子集合 $\Omega^* = \{s_t^{*i}, w_t^{*i}\}$ 。

(9) 更新环境地图

通过 EKF 估计特征的后验分布函数 $p(m_j | x_{1:k}, z_{1:k}, u_{1:k})$ 。

(10) 计算位姿，判断是否需要继续执行。

一次循环完成后，粒子集 Ω 中的粒子完成了状态的更新，通过 n 个粒子获取 t 时刻位姿信息，如式 4-29 所示：

$$X = \sum_{i=1}^n w_t^i s_t^i \quad (4-29)$$

当完成计算后如果还需要计算下一时刻的位姿，算法就会进入下一个循环，跳转回步骤(2)。

改进后的 FastSLAM 算法的流程图如图 4-5 所示。

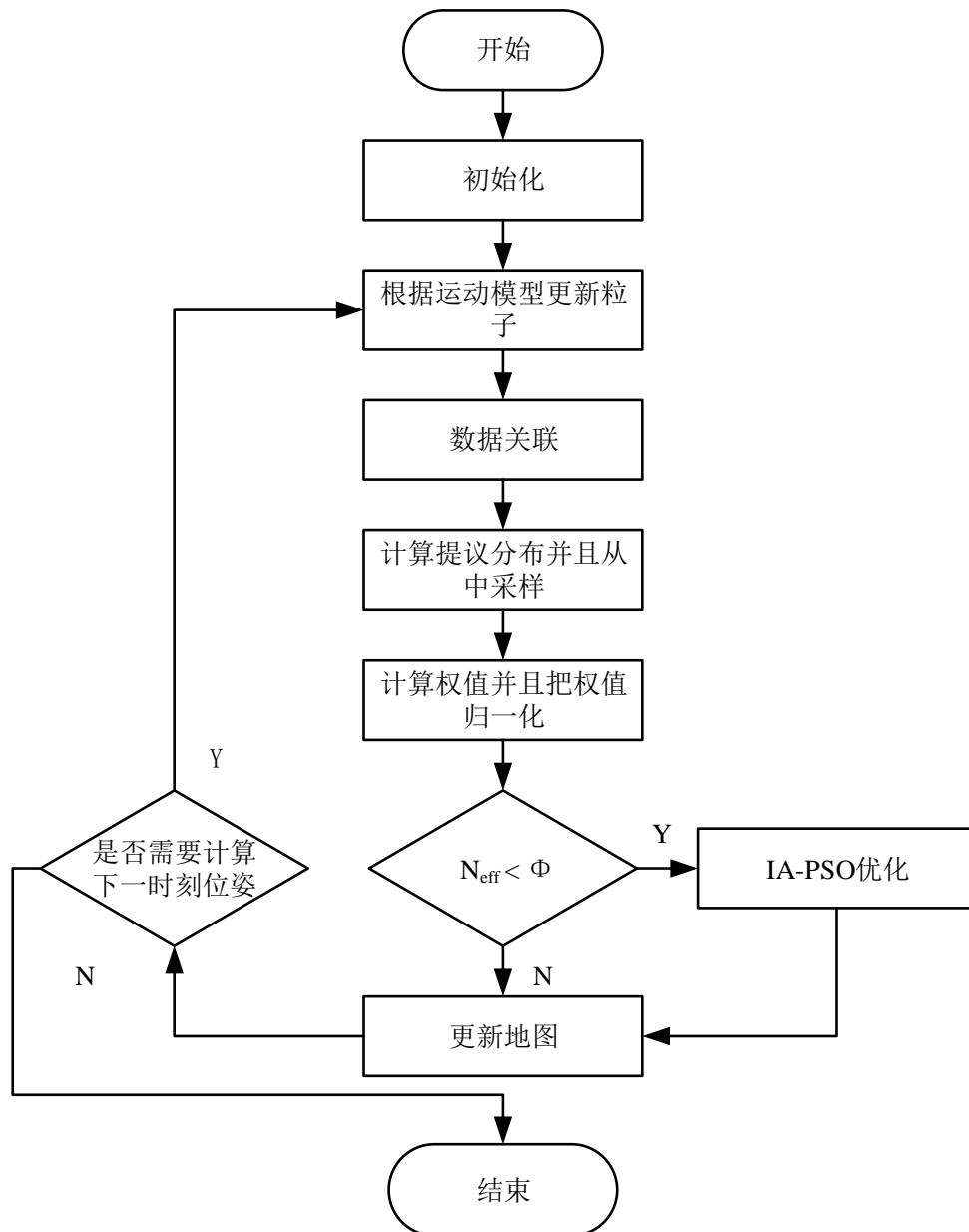


图 4-5 基于 PSO 和 IA 融合优化的快速 SLAM 算法流程图

4.4 仿真结果及分析

实验仿真环境是 Windows10, Matlab2013, 实验仪器使用的是 Intel i5 处理器, 8G 内存的笔记本。本实验采用了 Tim Bailey 教授开源的仿真平台, 对 EKFSLAM, FastSLAM 以及改进的 IA-PSO-FastSLAM 进行对比实验, 然后根据实验结果进行分析。

仿真实验构建了一个 200m*200m 的环形地图。其中路径点有 21 个, 同时在路径点周围标定 65 个路标点用于模仿双目相机观察到的路标信息。在实验开始之前设定机器人从(0,0)开始运动, 运动的速度是 5m/s, 运动的误差是 0.5m/s, 运动最大转向角是 40 度, 最大转向角变化率 25 度/s, 误差是 4 度。观测的距离误差是 0.2m, 角度误差是 1 度。可以得到运动过程的协方差矩阵为:

$$Q = \begin{bmatrix} 0.5^2 & 0 \\ 0 & (3*\pi/180)^2 \end{bmatrix}$$

观测过程的协方差矩阵为:

$$R = \begin{bmatrix} 0.2^2 & 0 \\ 0 & (1*\pi/180)^2 \end{bmatrix}$$

通过仿真记录机器人位姿的平均值, 本次实验的粒子数是 15。仿真使用的地图结果如图 4-6 所示:

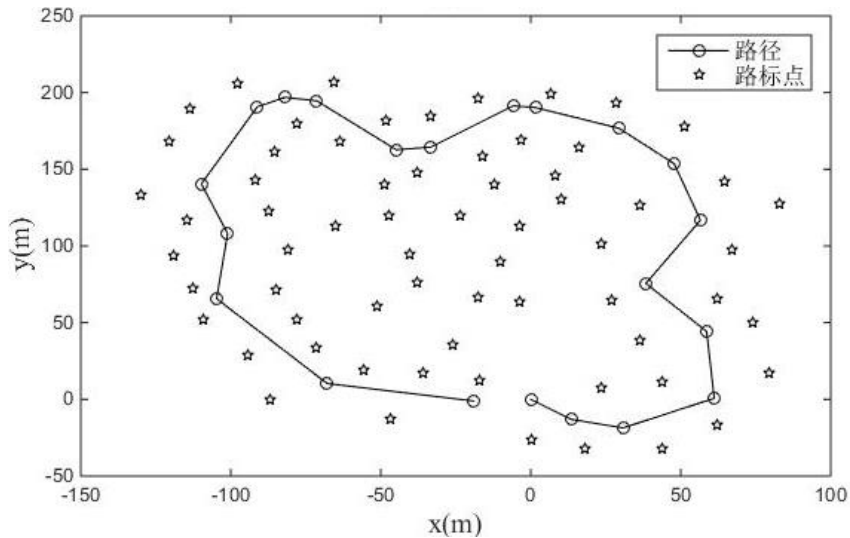
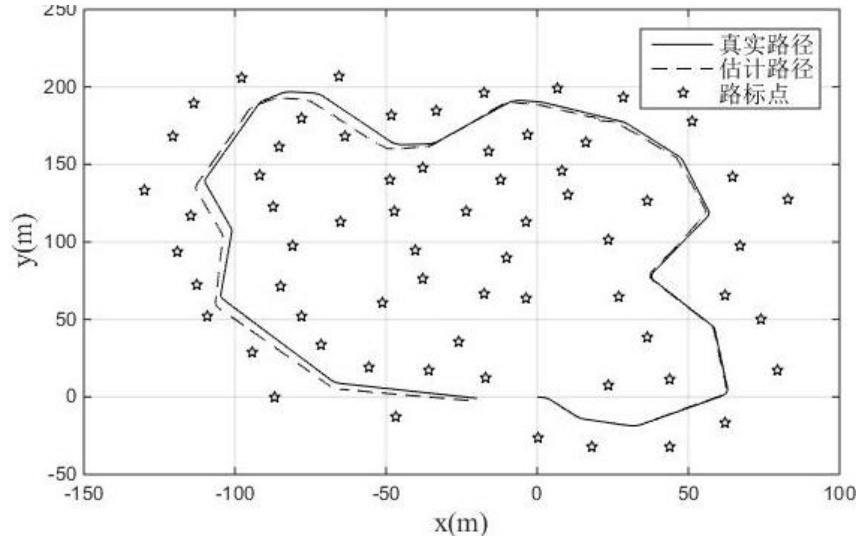


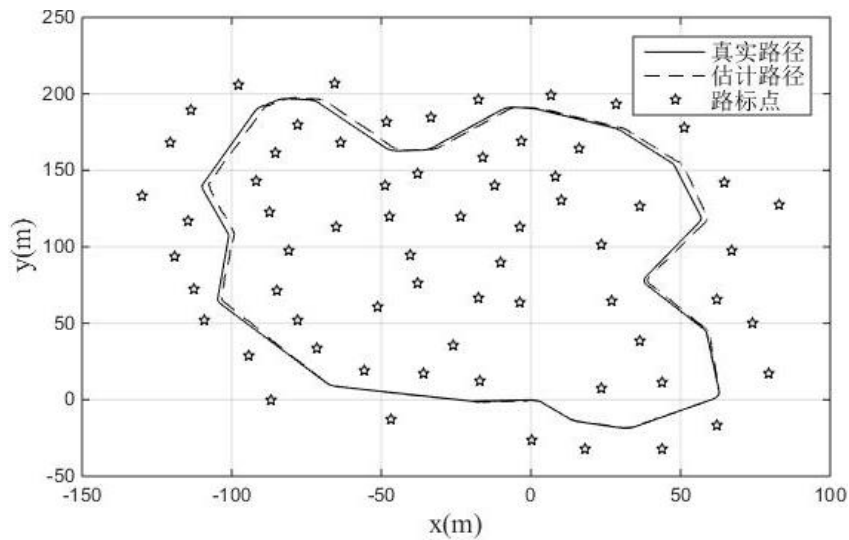
图 4-6 仿真地图和路标点

在图 4-6 中, 实线表示由指定的路径点构建的机器人运动的真实路径, 五角星

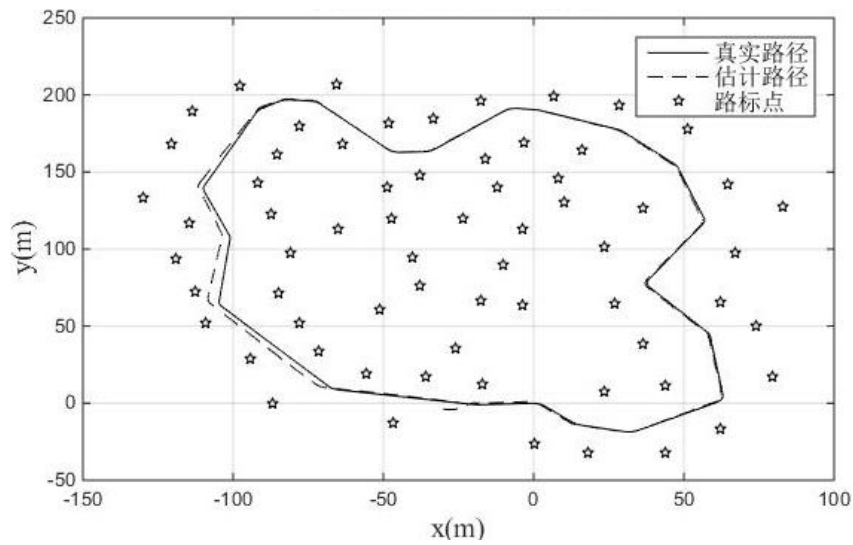
表示运动过程中可能观测到的路标信息。现在使用 EKFSLAM，原始 PFSLAM 已经改进后的 IA-PSO-PFSLAM 分别进行 SLAM 仿真实验，并且将结果和真实路径相比较，实验结果如图 4-7 所示。



(a)EKFSLAM 仿真结果



(b)FastSLAM 仿真结果

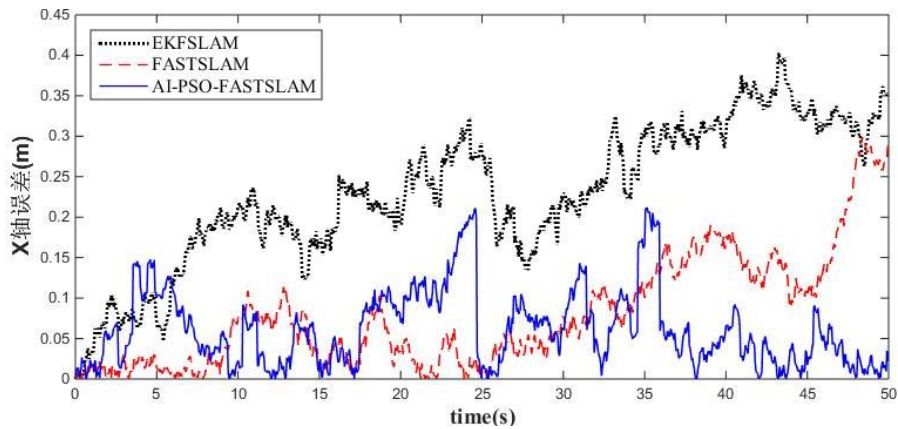


(c)IA-PSO-FastSLAM 算法仿真结果

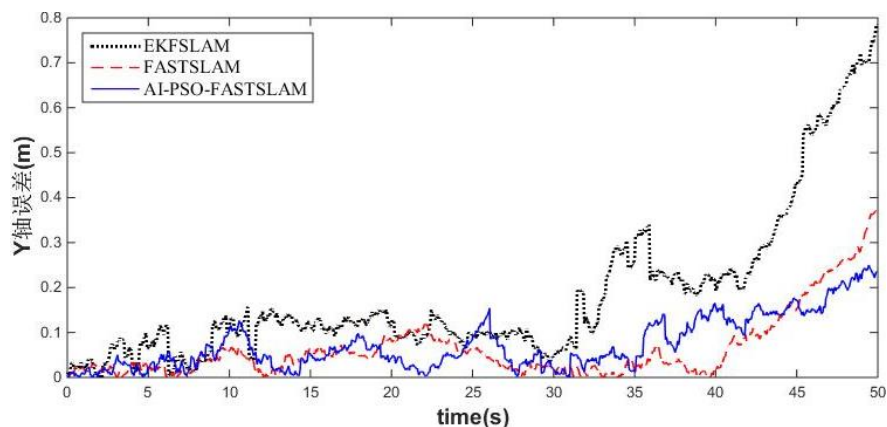
图 4-7 算法仿真结果

从图 4-7 可以看出 EKFSLAM 算法的仿真结果最差，估计的运动轨迹和机器人真实运动轨迹有较大误差，改进的 FastSLAM 算法的结果要好于原始的 FASTSALM 算法，定位更加精准，更加贴合机器人的运动的真实路径。

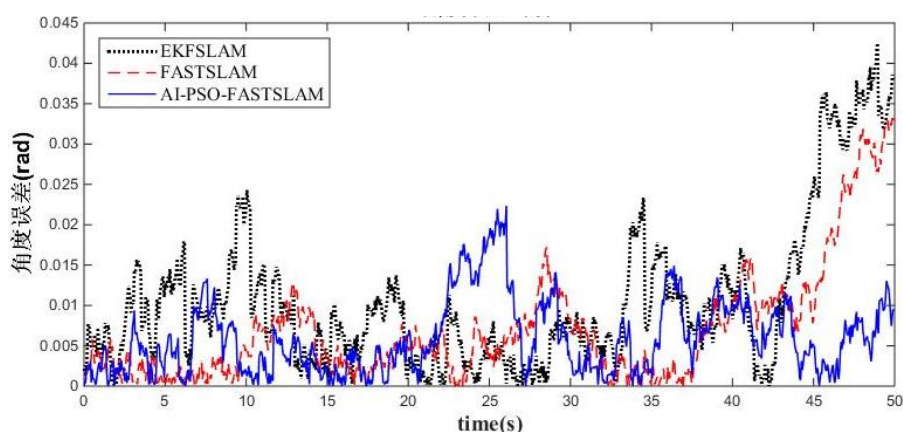
下面比较三种算法在 x 轴和 y 轴以及偏转角度上的最大误差，如图 4-8 所示。



(a)X 轴误差



(b)Y 轴误差



(c)角度误差

图 4-8 X 轴 Y 轴和角度误差比较

从图 4-8 的三个折线图可以看出，FastSLAM 算法在位置估计和角度估计两方面，在大部分时间段都要比 EKFSLAM 误差小。同时可以观察得到改进的 FastSLAM 算法在 X 轴和 Y 轴的误差上又明显要比原始的 FastSLAM 算法估计的更加精准。然后通过表 4-1 对最大误差进行统计：

表 4-1：误差统计

算法	时间/s	X 轴最大误差/m	Y 轴最大误差/m	角度最大误差/rad
EKFSLAM	45.36	2.97	3.71	0.14
FastSLAM	114.11	1.64	2.23	0.068
IA-PSO-PFSLAM	125.31	1.35	1.91	0.079

通过表 4-1 可以看出，在 X 轴和 Y 轴的最大误差，FastSLAM 算法要小于 EKF

算法，而改进的 FASTSALM 算法误差相比于原始 FastSLAM 算法进一步减小。通过统计表可以得出通过融合免疫算法和粒子群算法优化粒子集能够有效的使粒子向真实位置靠近，使粒子的分布更加接近于机器人真实的后验概率。同时通过时间上的比较可以看出，FastSLAM 算法运行时间要长于 EKFSLAM 算法，因为 FastSLAM 更加复杂，所以 EKFSLAM 算法运行时间最短。而改进的 FastSLAM 算法的执行时间又要比 FastSLAM 算法的执行时间更长，因为改进的 FastSLAM 算法为了得到更好的结果，使用免疫算法和粒子群算法代替直接重采样，需要进行迭代收敛，造成执行时间增长，但仍然能够满足 SLAM 算法对于实时性的需求。

最后通过均方根误差函数(RMSE)对三种方法观测结果进行统计评价。RMSE 计算公式为：

$$RMSE = \sqrt{\frac{1}{S} \sum_{j=1}^S (x_k^j - \hat{x}_k^j)^2} \quad (4-30)$$

表 4-2：误差评估

算法	位置估计均方根误差	角度估计均方根误差差
EKFSLAM	2.45	0.2581
FastSLAM	1.4416	0.1910
IA-PSO-PFSLAM	0.9987	0.1613

从表 4-2 可以看出改进的 FastSLAM 算法的均方根误差明显小于 EKFSLAM，略小于 FASTSALM 算法，所以通过实验证明了改进算法对于机器人定位估计精度有所提高。

4.5 本章小结

在这一章中首先介绍了将 IMU 和双目相机融合解决 SLAM 问题所用到的算法，既 FastSLAM 算法，并且指出了 FastSLAM 算法存在的不足。然后分析给出了 SLAM 算法进行滤波预测需要的运动模型和观测模型。运动模型是基于本文第二章分析的 IMU 的模型分析，通过 IMU 的加速度计和陀螺仪获取运动过程方程；观测模型是基于本文第三章分析的双目立体视觉模型，通过双目立体匹配获得较为准确的空间坐标点。针对 FastSLAM 算法存在的不足，本文通过将免疫算法和粒子群算法引入，解决 FastSLAM 算法在粒子滤波过程中粒子退化和粒子耗尽的问题，提高了采样粒子的多样性。最后通过对比分析，比较了 EKFSLAM 算法和 FastSLAM 算法以及改进后的 FastSLAM 算法的性能，得出改进算法的有效性。

第五章 SLAM 系统搭建与实验

5.1 系统概述

本章主要是介绍基于本文的算法开发的 SLAM 系统，包括硬件和软件两部分。系统硬件主要是双目相机和 IMU，采集运动过程数据。系统软件是在机器人操作系统(Robot Operating System，ROS)上开发的，实现的功能是订阅双目相机和 IMU 节点发布的数据，然后使用本文改进的 FastSLAM 算法处理传感器数据。系统软件主要分为六个模块，分别是数据获取模块，图像序列处理模块，IMU 数据处理模块，SLAM 模型模块，SLAM 算法处理模块，以及输出模块。

5.2 实验环境介绍

5.2.1 ROS 系统简介

本文系统是在机器人操作系统(Robot Operating System，ROS)上开发的。ROS 系统是由斯坦福大学开发的一个机器人系统，现在由 OSRF 进行维护的一个开源项目。ROS 系统是一个用来管理计算机软硬件资源的操作系统，作为一个开源系统，提供了包括进程间消息传递，底层设备控制以及硬件抽象等功能，通过对硬件系统进行封装，使应用软件不需要关注硬件系统，提高了应用软件的开发效率。ROS 系统支持包括 Python、C++以及 Lisp 等多种开发语言，大幅度提高了研究者的开发效率，节省了计算机语言的学习时间。

ROS 操作系统分为三个层次，包括文件系统级(Filesystemlevel)，计算图层级(Computationgraphlevel)，以及交流层级(Communicationlevel)。首先介绍一下文件系统级，文件系统级是可以获取到的系统文件和代码主要有：

(1) 功能包(Packages)，它是 ROS 系统的基本组织，可以包含所有文件格式，包括 ROS 的依赖库，执行任务时处理的文件以及数据集合等有用的数据。

(2) Manifests，提供许可信息以及和 Package 之间的依赖。

(3) Message types(msg)，系统信息的描述，定义了 ROS 系统下的数据类型。

(4) Service types(srv)，系统服务的描述，定义了 ROS 系统下请求的数据类型。

计算图层级是 ROS 系统数据传输的方式，主要包括：

(1) Nodes(节点)，节点就是 ROS 系统中正在运行的程序，ROS 系统中可以同时包含多个节点，同时进行不同操作，比如说监控双目相机，一个提供图形界面。

(2) Master(主导)，通过 Master，可以查找其他节点，进行信息的交换，调用其它服务。

(3) ServerParameterServer(参数服务器), 参数服务器设定了数据的存储方式。

(4) Messages(消息), Messages 是节点之间传递消息的主要方式。Messages 确定了传递消息的数据结构, 可以包括基本数据类型和数组以及各种嵌套关系。

(5) Topics(主题), 主题通过发布和订阅实现消息的传递。一个节点在给定的 Topic 发送消息, 另一个节点可以订阅这个 Topic 获取消息, 实现订阅者和消费者之间的解耦。

(6) Services(服务), 服务主要实现消息的请求和回应, 比 Topic 更加灵活, 可以实现远程访问服务。

(7) Bags(数据包), 数据包是 ROS 系统主要的数据存储和发布方式, 实现对于实验数据的收集。

5.2.2 传感器介绍

本次实验使用的传感器是由岚瞳视觉开发的 LOITOR 视觉惯性相机如图 5-1 所示, 加装在如图 5-2 的六轮电动小车上。



图 5-1 视觉惯性相机



图 5-2 实验搭载平台

双目相机的摄像头采用的是 MT9V034，IMU 采用的是 MPU-6050，对相机和 IMU 的重要参数做了总结如表 5-1 所示。

表 5-1 双目相机和 IMU 参数

型号	MT9V034	MPU-6050
控制器	CY68013	STM-32
数据传输延迟	1000ms/当前帧率	100us
帧率	24-65fps	200fps
帧同步方式	通过 SDK 自动控制，精确到 40 微秒	微米级时间戳

在第二章中已经对于系统使用的双目相机进行标定，可以直接使用标定结果。

5.3 系统详细设计

本文在第三章和第四章理论的指导下实现了一个基于双目相机和 IMU 融合的 IA-PSO-FastSLAM 系统。系统主要包括六个模块，如图 5-3 所示：

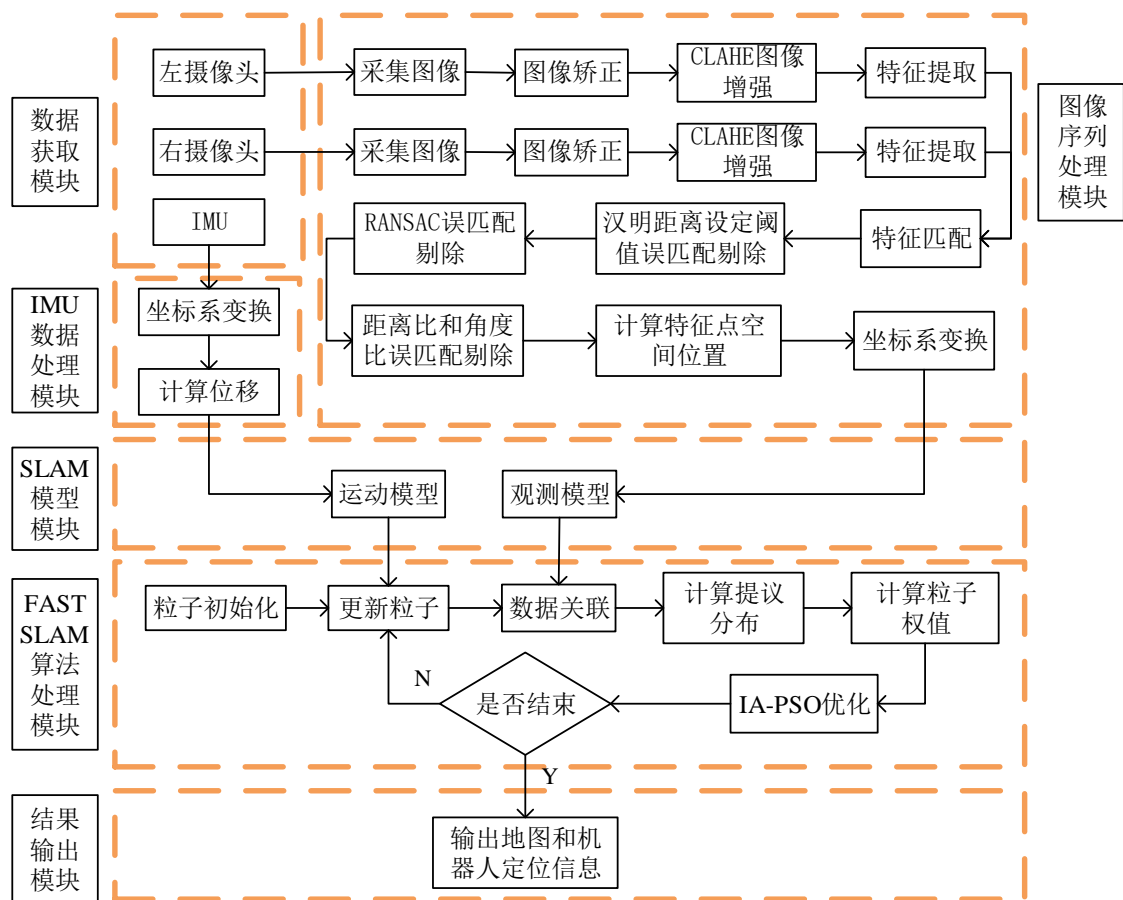


图 5-3 系统模块

如图 5-3 所示整体系统主要分为六个模块分别功能是：

- (1) 数据获取模块，获取到 IMU 和双目相机的信息，通过传感器自带 SDK 对于 IMU 和相机进行时间戳同步，并且同步误差在 2.5ms，满足系统需求。
- (2) 图像序列处理模块，处理双目相机得到的图像序列，通过矫正，增强，匹配和筛选得到精度较高的图像匹配，最后通过立体视觉计算空间匹配点的坐标。
- (3) IMU 数据处理模块，主要用来处理 IMU 数据，计算短时间内的位移信息。
- (4) SLAM 模型计算模块，计算运动模型和观测模型。
- (5) FastSLAM 算法处理模块，通过粒子滤波定位，通过 EKF 进行建图。
- (6) 结果输出模块，输出地图信息和机器人的定位信息。

pzg 在 ROS 系统下的节点关系图如图 5-4 所示。传感器通过发布 IMU 数据以及左右摄像头数据，IA_PSO_FastSLAM 节点订阅这些消息，然后处理完成后输出地图。

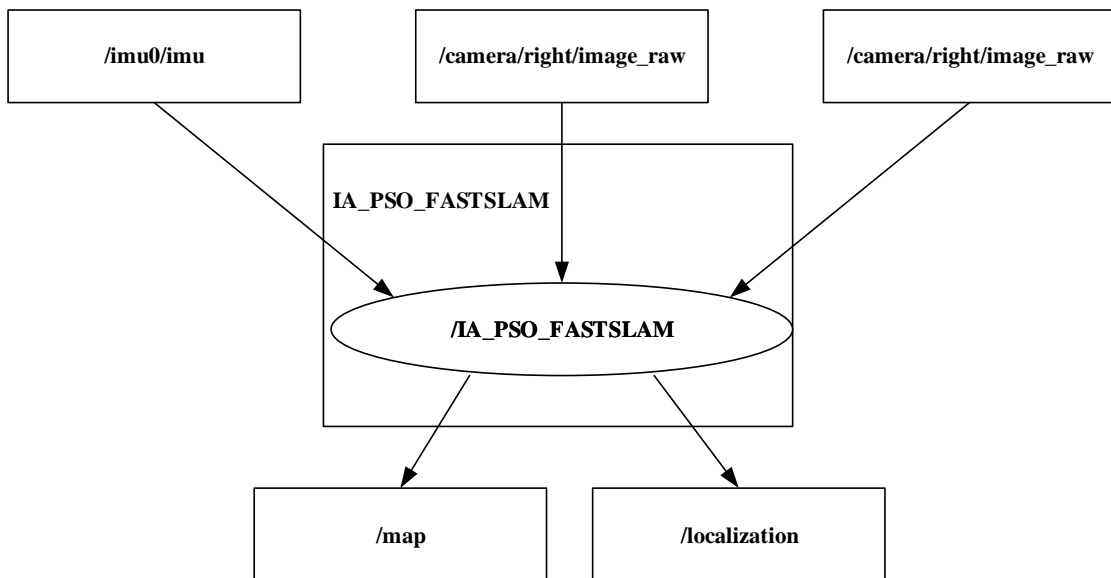


图 5-4 节点关系图

5.4 实验结果

5.4.1 基于数据集的实验

本文采用 Karlsruhe 的数据集进行测试，该数据集使用的是灰度双目相机，双目相机成像效果如图所示。该数据集记录运动过程中了 OXTS RT 3000 IMU 的数据。同时该数据集自带高精度 GPS 数据可以作为真实的运动路径。



图 5-5 Karlsruhe 的数据集成像效果

在本次试验中主要对比了单独使用双目相机的视觉里程计方法和本文介绍的融合双目相机和 IMU 的 FastSLAM 算法的定位效果。双目立体视觉里程计实现的具体过程是，首先使用第三章改进的 ORB 算法进行特征点提取然后使用第四章介绍的方法计算空间坐标点的坐标，然后使用 PnP(Perspective-n-Point)计算 3D 到 2D 的运动，同时通过 BA(Bundle Adjustment)优化 PnP 计算的旋转矩阵和平移矩阵，得到一系列机器人的位姿信息。将这种方法和本文改进的 FastSLAM 算法进行比较。可以得到结果如图所示：

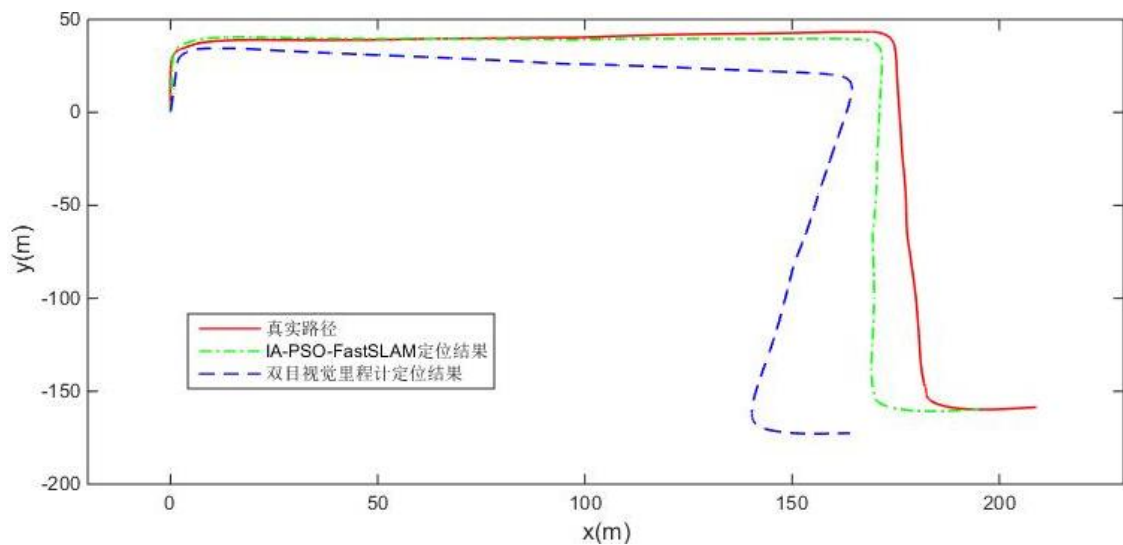


图 5-6 在数据集 2009_09_08_drive_0010 上的实验轨迹

表 5-2 均方根误差统计

实验数据集	IA-PSO-FastSLAM 均方根误差	双目视觉里程计均方根误差
2009_09_08_drive_0010	7.2925	28.2792
2009_09_08_drive_0012	11.3728	19.5283
2010_03_04_drive_0033	6.6871	13.1843

从图 5-6 可以得到, 本文基于双目相机和 IMU 融合的改进 FastSLAM 算法获得了较好的定位结果。通过表 5-2 可以得出, 相比于单独使用双目相机的视觉里程计算法, 本文融合 IMU 和双目相机的 AI-PSO-FastSLAM 算法具有更好的定位效果, 本文改进的 IA-PSO-FastSLAM 算法在均方根误差上明显小于双目视觉里程计算法, 可以满足 SLAM 系统对于定位准确度的需求, 证明了本文算法的有效性。

5.4.2 基于现实场景的实验

本文实验地点选取的是电子科技大学主楼实验室内场景。实验场景实景如图 5-7 所示。模拟场景如图 5-8 所示, 是一个接近 U 形的实验场地, 机器人运动过程中纵向长度大约 9 米, 横向长度 2 米。机器人从 U 形实验场地的 A 点出发运动一周, 到达 U 形场地的 B 点, 结束运行。将机器人运动过程中的双目相机数据和 IMU 数据保存成 ROS 系统下的 Bag 文件, 以供后期重复实验使用。



图 5-7 真实实验场地

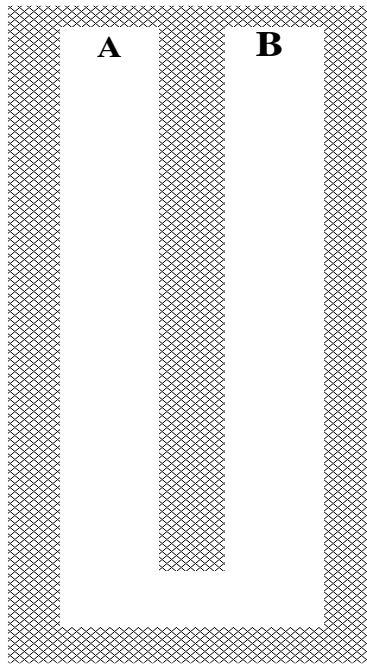


图 5-8 模拟实验场地

记录数据以后，通过将 Bag 文件进行回放，回放过程发布双目相机和 IMU 的运行数据。通过 IA_PSO_FastSLAM 节点获取信息并进行计算。获取定位机器人的定位信息和空间路标的位置信息。程序执行的过程如图 5-9 所示：

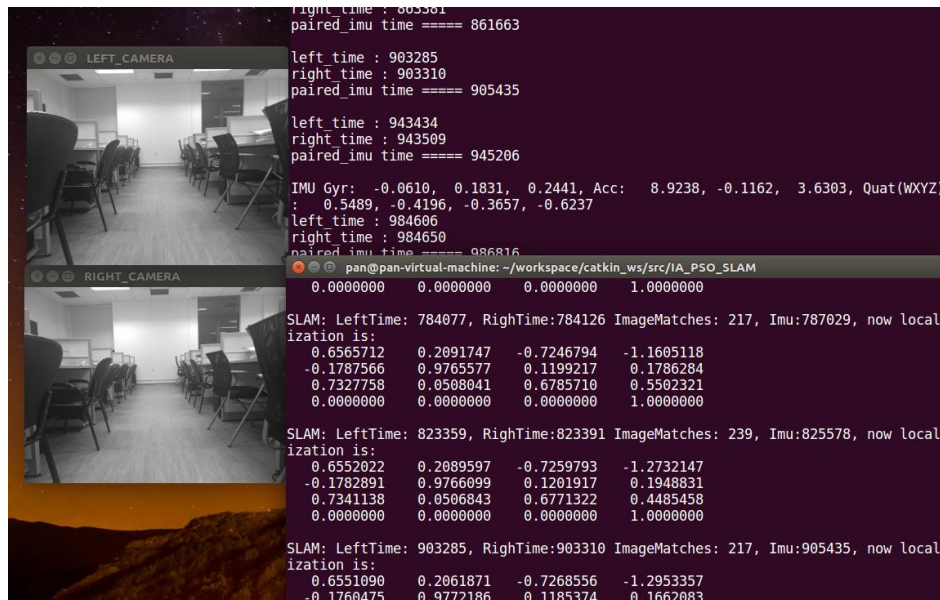


图 5-9 运行时截图

系统在运行过程中，左边两个窗口分别是左右相机的成像，右上角是传感器发布的双目相机时间戳信息，以及 IMU 时间戳和 IMU 当前的姿态信息。右下角是在

接收到双目相机信息和 IMU 信息后，通过改进的算法得到的权值最大的粒子在世界坐标系下的位置信息。将位姿数据转换到坐标系下可以得到机器人运动定位信息，机器人一直在运动过程中，得到的定位信息是不连贯的如图 5-10 所示。

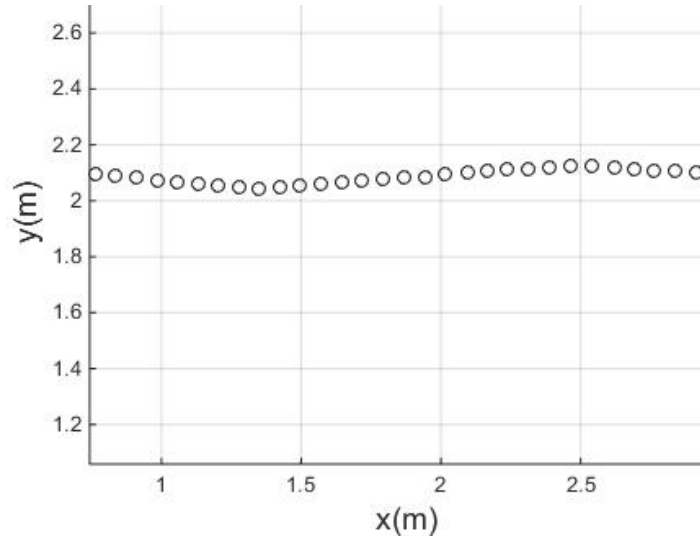


图 5-10 机器人定位信息

将机器人的定位坐标相连接可以得到机器人的轨迹地图，如图 5-9 所示，机器人初始运动的方向是 X 轴。

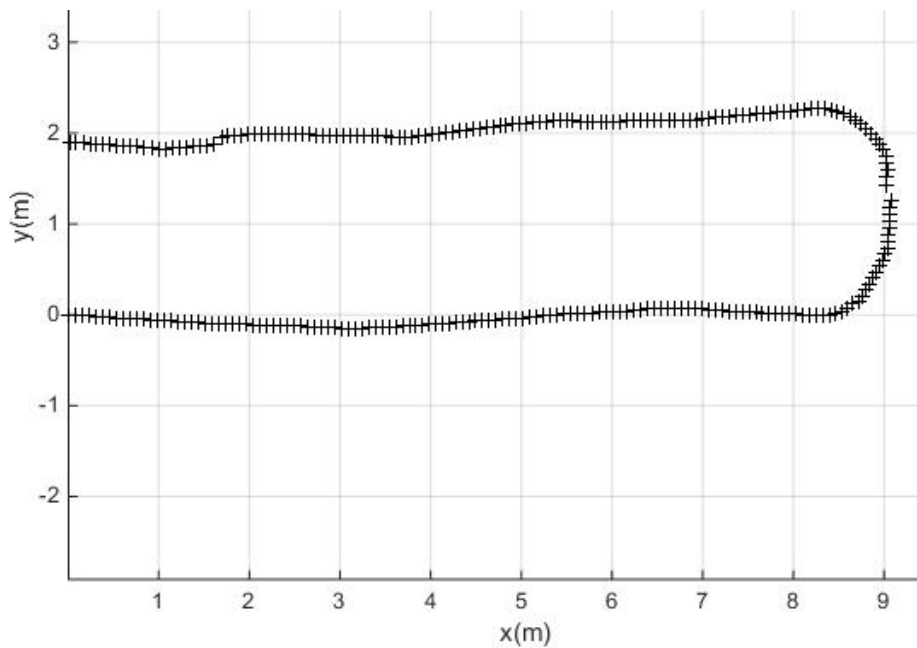


图 5-11 机器人轨迹

因为实验条件有限，没有办法获取到真实路径信息，但是从图 5-9 可以看出估

计的运动轨迹和图 5-8 实验场地所对应的真实运行轨迹基本一致,该实验验证了算法的有效性。

5.5 本章小结

本章主要介绍了试验环境,体统的详细设计以及在真实环境下进行实验。首先介绍了具体的试验环境包括软件和硬件平台,本章系统软件是在机器人系统上开发的。然后介绍了系统的详细设计,包括各个模块的具体功能,以及系统在 ROS 下的节点关系图。最后对基于 FastSLAM 提出的改进算法 IA-PSO-FastSLAM 算法通过真实环境进行了验证,并且证明了该算法的有效性。

第六章 总结与展望

6.1 总结

机器人技术是当今科学技术领域发展最快的领域之一。而 SLAM 是移动机器人实现自主运动的关键。本文通过双目相机和 IMU 融合实现机器人同时定位与建图。本文首先详细的分析了当今国内外 SLAM 领域最新的研究与发展现状，并对当下通过多传感器融合解决 SLAM 问题的国内外最新成果和问题进行了详细分析。当前 SLAM 问题的主要难点集中在未知环境的影响，多传感器融合方法，以及特征信息关联和 SLAM 问题的复杂度。然后研究了双目立体视觉系统。双目立体视觉是本文 SLAM 系统的重要组成部分。在本章中主要研究了双目立体视觉的关键技术。首先研究了双目相机的相机模型和双目相机的标定和矫正方法，并且对本文使用的双目相机进行了矫正。然后分析了多种常用的图像特征，并且比较了这几种特征的优劣。经过对比研究，决定在 ORB 算法的基础进行改进，主要是对双目相机获取的图像进行预处理，进行直方图均衡化对图像进行增强，同时改进了 ORB 算法的误匹配剔除策略，通过结合多种匹配方法明显提高匹配结果的精度，提供给观测方程更加准确的路标坐标信息。本文通过 IMU 构建运动模型，通过双目相机观测路标信息构建观测模型。由于 FastSLAM 算法的定位过程是基于粒子滤波算法，存在着粒子退化问题。本文通过引入免疫算法和粒子群融合的算法，改善了 FastSLAM 粒子退化问题和由于多次重采样造成的粒子耗尽问题。

最后，本文基于前文对于 FastSLAM 算法的改进，实现了多传感器融合的 SLAM 系统，并进行了测试。本文通过理论分析和实验证明改进的 FastSLAM 算法能够有效的改善由于匹配精度问题以及粒子退化和耗尽问题引起的问题，提高了机器人定位的精度。

6.2 展望

本文在原有 ORB 特征提取和 FastSLAM 算法基础上进行改进，取得了一定成果但是本实验还可以从以下几个方面进行改善。

(1) 本文通过融合 IMU 和双目相机进行多传感融合的 SLAM 算法的研究，在后期可以考虑把 GPS 加入到系统当中来，因为 GPS 在开阔地带的定位精度较高，在室外可以矫正系统的定位误差。

(2) 双目相机在运行过程中可能收到光线和移动目标的干扰，在未来的研究中如何解决环境的动态干扰是一个重点的探索方向。

(3)本文使用的 IMU 型号是 MPU6050, 精度较差, 存在着较大的噪声和零漂, 可以考虑使用精度更高的 IMU 作为实验传感器, 可以提高机器人的定位精度。本文现在通过实验前测量零漂从而对 IMU 进行矫正, 可以考虑通过滤波算法对于零漂进行矫正。

(4)本文只是对于一个机器人的同时定位与建图问题进行了研究, 而现今多机器人协作是一个研究的热点, 后期考虑使用多个机器人进行协同工作, 通过分布式的方式进行同时定位与建图, 提高效率和精度。

致 谢

经过三年的研究生生涯，我积累了海量的知识，结识了一群志趣相投的朋友，收获了一段可以回味一辈子的记忆。想起自己在大四备考的那段时光，曾经怀疑过研究生三年的学习是否有必要，现在我可以自豪的对当时的自己说：“你的选择无比正确”。研究生三年的学习生活本应该是孤独的，但是有了老师和同学的陪伴，枯燥的生活也变得欢乐起来。在这里我有很多需要感谢的人。

首先我要感谢的是我的导师罗光春老师，是您给了我三年宝贵的学习机会。当初在宣讲会上听到实验室的简介，实验室众多的研究成果以及学术造诣深厚的老师令我神往不已。最终有幸得到您的认可，加入到实验室这个大家庭当中来。您严谨的作风深深影响着我对于科研和生活的态度。在这里让我再次对您致以深深的谢意。

还要特别感谢段琢华老师，非常有幸能够在研二一年时间里，接受您无微不至的指导。是您让我对于研究方向有了清晰深入的了解，加快了学习和研究的进度。是您的指导，让我懂得了从论文汲取养分的方式，提高了自身的知识储备。更加感谢您对于我的论文无私的指导，在繁重的工作当中抽出时间，检查我的论文并且帮我指出论文的不足，让我的论文更加科学和丰满。以后能够跟您探讨问题的时间越来越短，希望在将来的学习生活中仍然能够得到您的指引。

还要特别感谢段贵多老师，在我研究生三年期间遇到问题时，您总是在无私的帮忙解决问题，同时教我解决问题的方法，从您身上我学到了很多做人做事的道理。同时还要感谢赵太银和田玲老师，感谢两位老师对我论文的指导以及细心地修改。感谢实验室的秦科老师，陈爱国老师，卢国明老师，张栗棕老师，以及实验室师兄师姐师妹师弟，感谢同一届的各位同窗，感谢你们这三年的陪伴。

最后还要感谢我亲爱的父母，大姐二姐，是你们的爱和鼓励让我克服困难走到现在。

最后，再一次感谢所有帮助过我的人，谢谢！

参考文献

- [1] Durrant - Whyte H. Where am I? A tutorial on mobile vehicle localization[J]. Industrial Robot, 1994, 21(2):11-16
- [2] 权美香, 朴松昊, 李国. 视觉 SLAM 综述[J]. 智能系统学报, 2016, 11(6):768-776
- [3] 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
- [4] Mur-Artal R, Tardós J D. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras[J]. IEEE Transactions on Robotics, 2016, 33(5):1255-1262
- [5] Smith R, Cheeseman P. On the Estimation and Representations of Spatial Uncertainty[C]. International Journal of Robotics Research. 1987:1-22
- [6] Smith R, Self M, Cheeseman P. Estimating Uncertain Spatial Relationships in Robotics[J]. Autonomous Robot Vehicles, 1988, 5(5):435-461
- [7] Holmes S, Klein G, Murray D W. A Square Root Unscented Kalman Filter for visual monoSLAM[C] IEEE International Conference on Robotics and Automation. IEEE, 2008:3710-3716.
- [8] Dellaert F, Fox D, Burgard W, et al. Monte Carlo localization for mobile robots[C]. IEEE International Conference on Robotics and Automation, 1999. Proceedings. IEEE, 1999:1322-1328 vol.2
- [9] Doucet A, Freitas N D, Murphy K, et al. Rao-blackwellised particle filtering for dynamic Bayesian networks[C].Sixteenth Conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann Publishers Inc. 2000:176-183
- [10] Zhou H, Ni K, Zhou Q, et al. An SfM Algorithm With Good Convergence That Addresses Outliers for Realizing Mono-SLAM[J]. IEEE Transactions on Industrial Informatics, 2017, 12(2):515-523.
- [11] Bellavia F, Fanfani M, Pazzaglia F, et al. Robust Selective Stereo SLAM without Loop Closure and Bundle Adjustment[J]. 2013, 8156:462-471.
- [12] Zhang G, Jin H L, Lim J, et al. Building a 3-D Line-Based Map Using Stereo SLAM[J]. IEEE Transactions on Robotics, 2015, 31(6):1364-1377.
- [13] Montemerlo M, Thrun S. Simultaneous localization and mapping with unknown data association using FastSLAM[C]. IEEE International Conference on Robotics and Automation, 2003. Proceedings. ICRA. IEEE, 2003:1985-1991 vol.2

- [14] Des Bouvrie S L. Improving RGBD Indoor Mapping with IMU data[D]. Rgbd Camera, 2011.
- [15] Hjelmare F, Rangsjö J. Simultaneous Localization And Mapping Using a Kinect In a Sparse Feature Indoor Environment[J]. Signal Processing, 2012,32(2):121-112
- [16] 宋宇, 李庆玲, 康轶非,等. 平方根容积 Rao-Blackwillised 粒子滤波 SLAM 算法[J]. 自动化学报, 2014, 40(2):357-367
- [17] 杜钊君, 吴怀宇. 基于激光测距与双目视觉信息融合的移动机器人 SLAM 研究[J]. 计算机测量与控制, 2013, 21(1):180-183
- [18] 陈炜楠, 刘冠峰, 李俊良,等. 室内环境的元胞自动机 SLAM 算法[J]. 机器人, 2016, 38(2):169-177
- [19] 吕丹娅, 姚剑敏, 郭太良. 改进 ORB 算法在单目视觉 SLAM 特征匹配中的应用[J]. 电视技术, 2016, 40(11):107-111.
- [20] 熊斯睿. 基于立体全景视觉的移动机器人 3DSLAM 研究[D]. 哈尔滨工业大学, 2015.
- [21] 白云汉. 基于 SLAM 算法和深度神经网络的语义地图构建研究[J]. 计算机应用与软件, 2018(1):183-190.
- [22] Rublee E, Rabaud V, Konolige K, et al. ORB: An efficient alternative to SIFT or SURF[C]. IEEE International Conference on Computer Vision. IEEE, 2012:2564-2571
- [23] Reza A M. Realization of the Contrast Limited Adaptive Histogram Equalization (CLAHE) for Real-Time Image Enhancement[J]. Journal of Vlsi Signal Processing Systems for Signal Image and Video Technology, 2004, 38(1):35-44
- [24] Fischler M A. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography[J]. Readings in Computer Vision, 1987, 24(6):726-740
- [25] Thrun S, Montemerlo M, Koller D, et al. FastSLAM: An Efficient Solution to the Simultaneous Localization And Mapping Problem with Unknown Data[J]. Archives of Environmental Contamination and Toxicology, 2002, 50(2):240-248
- [26] Montemerlo M S. FastSLAM: a factored solution to the simultaneous localization and mapping problem with unknown data association[M]. Carnegie Mellon University, 2003:593-598
- [27] Hahnel D, Burgard W, Fox D, et al. An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements[C].Ieee/rsj International Conference on Intelligent Robots and Systems. IEEE, 2003:206-211 vol.1
- [28] . Monjazebe A, Sasiadek J Z, Neculescu D. Autonomous navigation among large number of nearby landmarks using FastSLAM and EKF-SLAM - A comparative study[C].International Conference on Methods and MODELS in Automation and Robotics. IEEE, 2011:369-374
- [29] Brooks A, Bailey T. HybridSLAM: Combining FastSLAM and EKF-SLAM for Reliable

- Mapping[C]. Algorithmic Foundation of Robotics Viii, Selected Contributions of the Eight International Workshop on the Algorithmic Foundations of Robotics, Wafr 2008, Guanajuato, México, December. DBLP, 2009:647-661
- [30] Nieto J, Bailey T, Nebot E. Scan-SLAM: Combining EKF-SLAM and Scan Correlation[C]. Field and Service Robotics, Results of the, International Conference, Fsr 2005, July 29-31, 2005, Port Douglas, Qld, Australia. DBLP, 2006:167-178
- [31] 石杏喜, 赵春霞, 郭剑辉. 基于 PF/CUKF/EKF 的移动机器人 SLAM 框架算法[J]. 电子学报, 2009, 37(8):1865-1868
- [32] Jia S, Yin X, Li X. Mobile robot parallel PF-SLAM based on OpenMP[C]. IEEE International Conference on Robotics and Biomimetics. IEEE, 2013:508-513
- [33] Zhou W, Shiju E, Cao Z, et al. Review of SLAM Data Association Study[C]. International Conference on Sensor Network and Computer Engineering. 2016: 2991
- [34] Bakkay M C, Arafa M, Zagrouba E. Dense 3D SLAM in Dynamic Scenes Using Kinect[M]. Pattern Recognition and Image Analysis. Springer International Publishing, 2015:121-129
- [35] Fang F, Ma X, Dai X, et al. A new multisensor fusion SLAM approach for mobile robots[J]. Journal of Control Theory and Applications, 2009, 7(4):389-394
- [36] Lee C W, Jeon H S. A Study on Improving the Computational Complexity of SLAM for Intelligent Robot Utilizing Smart Phone[J]. 2014, 19(11):25-31
- [37] Choi J, Choi M, Wan K C, et al. Data association using relative compatibility of multiple observations for EKF-SLAM[J]. Intelligent Service Robotics, 2016, 9(3):177-185
- [38] Alcantarilla P F, Yebes J J, Almazán J, et al. On combining visual SLAM and dense scene flow to increase the robustness of localization and mapping in dynamic environments[C]. IEEE International Conference on Robotics and Automation. IEEE, 2012:1290-1297
- [39] Qin T, Li P, Shen S. VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator[J]. 2017: 1708
- [40] Chen C, Cheng Y H. Computational complexity reduced CDKF based SLAM algorithm[J]. Application Research of Computers, 2012, 29(9):3280-3275
- [41] Xie X, Yu Y, Lin X, et al. An EKF SLAM algorithm for mobile robot with sensor bias estimation[C]. Automation. IEEE, 2017:281-285
- [42] Wang P P, Huang Y Q. EKF-SLAM Algorithm Based on Time Varying Regulation Factor[J]. Journal of Anhui Polytechnic University, 2017
- [43] Jia S, Yin X, Li X. Mobile robot parallel PF-SLAM based on OpenMP[C]. IEEE International Conference on Robotics and Biomimetics. IEEE, 2013:508-513

- [44] 朱福利, 曾碧, 曹军. 基于粒子滤波的 SLAM 算法并行优化与实现[J]. 广东工业大学学报, 2017, 34(2):92-96
- [45] 曹旭东, 胡越黎, 王涛. 基于 ROS 与四元数互补滤波的 IMU 姿态解算研究[J]. 工业控制计算机, 2017, 30(11):63-64
- [46] 张栋, 焦嵩鸣, 刘延泉. 互补滤波和卡尔曼滤波的融合姿态解算方法[J]. 传感器与微系统, 2017, 36(3):62-65
- [47] Kennedy J, Eberhart R. Particle swarm optimization[J]. Proc. of 1995 IEEE Int. Conf. Neural Networks, (Perth, Australia), Nov. 27-Dec. 1995, 4(8):1942-1948 vol.4
- [48] Goldberg D E. Genetic Algorithm in Search Optimization and Machine Learning[J]. Addison Wesley, 1989, xiii(7):2104-2116
- [49] Ge H, Mao Z Y. Immune algorithm[C]. Intelligent Control and Automation, 2002. Proceedings of the, World Congress on. IEEE, 2002:1784-1788 vol.3
- [50] Lowe D G. Object Recognition from Local Scale-Invariant Features[C]. iccv. IEEE Computer Society, 1999:1150
- [51] Lowe D G, Lowe D G. Distinctive Image Features from Scale-Invariant Keypoints[J]. International Journal of Computer Vision, 2004, 60(2):91-110
- [52] Bay H, Ess A, Tuytelaars T, et al. Speeded-Up Robust Features (SURF)[J]. Computer Vision and Image Understanding, 2008, 110(3):346-359
- [53] Ortega J C P, Arreguin J M R. A Feature Extraction Using SIFT with a Preprocessing by Adding CLAHE Algorithm to Enhance Image Histograms[C]. International Conference on Mechatronics, Electronics and Automotive Engineering. IEEE Computer Society, 2014:20-25
- [54] Schatz V. Low-latency histogram equalization for infrared image sequences: a hardware implementation[J]. Journal of Real-Time Image Processing, 2013, 8(2):193-206
- [55] 卢宇峰. 基于 FPGA 的实时图像矫正系统设计[D]. 浙江大学, 2016
- [56] Zhang Z. A flexible new technique for camera calibration[J]. Tpami, 2000, 22(11):1330-1334
- [57] 陈天华, 王福龙, 张彬彬. 基于改进 ORB 和对称匹配的图像特征点匹配[J]. 计算机系统应用, 2016, 25(5):147-152
- [58] 张洋, 吕强, 林辉灿,等. 一种基于改进 ORB 的视觉 SLAM 图像匹配算法[J]. 装甲兵工程学院学报, 2016, 30(6):82-88
- [59] 张毅, 童学容, 罗元. 一种改进 SURF 算法的单目视觉里程计[J]. 重庆邮电大学学报(自然科学版), 2014, 26(3).
- [60] Montemerlo M, Thrun S. Simultaneous localization and mapping with unknown data association using FastSLAM[C]. IEEE International Conference on Robotics and Automation, 2003.

- Proceedings. ICRA. IEEE, 2003:1985-1991 vol.2
- [61] Montemerlo M, Thrun S, Roller D, et al. FastSLAM 2.0: an improved particle filtering algorithm for simultaneous localization and mapping that provably converges[C]. International Joint Conference on Artificial Intelligence. Morgan Kaufmann Publishers Inc. 2003:1151-1156
- [62] Doucet A, Godsill S, Andrieu C. On sequential Monte Carlo sampling methods for Bayesian filtering[J]. Statistics & Computing, 2000, 10(3):197-208.
- [63] Pei F, Yan H, Zhu M. Improved FastSLAM System Using Artificial Fish-Swarm Optimized Distributed Unscented Particle Filter[C]. 中国控制会议. 2017:6951-6956.
- [64] Wen S, Yao M, Wu C, et al. An improved FastSLAM2.0 algorithm based on ant colony optimization[C]. Control and Decision Conference. IEEE, 2017: 1948-9447
- [65] 周武, 赵春霞. 一种基于遗传算法的 FastSLAM 2.0 算法[J]. 机器人, 2009, 31(1):25-32.
- [66] 王开宇. 基于全景视觉机器人的粒子群优化 FastSLAM 算法研究[J]. 科技创新与应用, 2016(24):10-11.

攻硕期间取得的成果

获奖情况:

- [1] 2016 年 10 月, 学业三等奖学金
- [2] 2017 年 10 月, 学业三等奖学金