

分类号: TP242

密 级: 公开

U D C:

单位代码: 11646

宁波大学

硕士学位论文

论文题目: 基于无人机视觉的 SLAM 研究

学 号: 1411082689
姓 名: 郭 润
专 业 名 称: 计算机应用技术
学 院: 信息科学与工程学院
指 导 教 师: 石守东

论文提交日期: 2017 年 4 月 15 日

A Thesis Submitted to Ningbo University for the Master's Degree

Research Of Vision SLAM Based on Unmanned Aerial Vehicle

Candidate: Guo Run

Supervisors: (Associate) Professor Shi Shoudong

Faculty of Electrical Engineering and Computer Science

Ningbo University

Ningbo 315211, Zhejiang P.R.CHINA

Date: April 15, 2017

独创性声明

本人郑重声明：所呈交的论文是我个人在导师指导下进行的研究工作及取得研究成果。尽我所知，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，也不包含为获得宁波大学或其他教育机构的学位或证书所使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中做了明确的说明并表示了谢意。

若有不实之处，本人愿意承担相关法律责任。

签名： 郭润 日期： 2017.6.1

关于论文使用授权的声明

本人完全了解宁波大学有关保留、使用学位论文的规定，即：学校有权保留送交论文的复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存论文。

（保密的论文在解密后应遵循此规定）

签名： 郭润 导师签名： 石宇 日期： 2017.6.1

基于无人机视觉的 SLAM 研究

摘 要

近年来，随着计算机控制技术、计算机视觉技术以及人工智能技术的飞速发展，不仅使得无人机从传统的军事领域逐步融入到民用领域的各个方面，而且在无人机的工作方式上也发生了巨大的变化，从以前传统的手动遥控方式到现在的完全自主化控制方式逐渐过渡。无人机的智能化与自主化发展离不开导航定位技术，传统的导航系统主要采用 GPS 的方式来实现；当无人机进入某些特殊环境作业时，易受到环境的干扰使得 GPS 信号变弱或者完全失效。为了弥补无人机基于 GPS 导航系统的不足，本文采用 SLAM 技术作为无人机的辅助导航方式。通过该技术无人机可以在没有外界信号提供的基础之上，实现周围环境地图的建立并且根据建立的环境地图来进行自身的定位。

本文研究的内容是基于小型无人机单目视觉的 SLAM 技术。由于无人机在实际的飞行过程中是一种非平滑的运动方式，故不能采用输入控制命令的方式来预测下一时刻无人机的位姿，为此本文采用的方法是通过视觉里程计来估计下一时刻无人机的位姿。在视觉里程计的特征点提取过程中计算量非常大，本文采用实时性高的 ORB 算法作为图像特征点提取算法，加快图像特征点的提取速度。对于 SLAM 系统的算法优化部分选择了时间复杂度和空间复杂度均优于扩展卡尔曼滤波的稀疏扩展信息滤波算法；在基于无人机的 SLAM 系统中，系统对算法的实时性、无人机姿态估计精度及一致性方面都有更高的要求；为此在标准的稀疏扩展信息滤波 SLAM 技术的基础上提出了两种改进算法，首先是从稀疏化操作角度出发利用熵性质、综合当前以及下一观测时刻来选择与位姿关联性最弱的环境特征作为稀疏特征点，来提高算法的精度和一致性；然后是根据信息矩阵元素的稀疏性分布特点，利用三对角矩阵来快速求解在状态均值恢复过程中的求逆操作，来提高算法的计算效率。最后建立仿真三维环境模型，在三维环境当中预先设定好无人机的航行轨迹以及环境特征点，将改进的 SEIF SLAM 技术与标准的 SEIF SLAM 技术应用于仿真环境当中进行对比验证。

仿真实验结果表明，基于熵性质的改进算法在位姿估计精度及一致性上相比标准算法有了明显的提高；基于三对角矩阵求逆的改进算法随着信息矩阵阶数逐渐增加，改进算法计算效率远快于标准算法。

关键词：小型无人机， 同时定位与地图构建（SLAM）， 视觉里程计，
稀疏扩展信息滤波

Research of Vision SLAM Based on Unmanned Aerial Vehicle

Abstract

Recently, with the rapid development of computer control technology, computer vision technology and artificial intelligence technology, which not only makes unmanned aerial vehicle(UAV) from the traditional military field gradually into all aspects of the civil field, but also in the aspect of the operation mode of the UAV changes greatly, that is to say, from the traditional way of remote manual control before gradually transition to now fully customized control. As a result of the UAV intelligent and customized development cannot leave the navigation and positioning technology, the traditional navigation system mainly uses the GPS, and when UAV does tasks in some special environment, in which it is vulnerable to the interference of environment, and the GPS signal will become weak or complete failure. In order to make up for the lack of GPS navigation system, the SLAM technology is used in this paper for being an auxiliary navigation way of UAV. The UAV can realize the establishment of environment map and uses it for location without external signal by SLAM.

This topic research is based on the small UAV monocular visual SLAM technology, because the UAV in the actual process of flight is not a kind of smooth movement, so it can't adopt the way of inputting control command to predict the position of the UAV in the next moment. Therefore, this paper uses the visual odometer to estimate the position of the UAV in the next moment. In the process of feature points extraction of visual odometer requires very large amount of calculation, the high real-time ORB is proposed as the image feature extraction algorithm in this thesis, and speeds up the image feature extraction. As to the optimization algorithm of SLAM system, this paper chooses the sparse extended information filter, due to its time complexity and space complexity are better than that of extended kalman filtering algorithm; In the SLAM system based on UAV, the real-time, UAV pose estimation accuracy and consistency of algorithm has higher requirements; so this paper has proposed two kinds of improved algorithm based on the basic algorithm. Firstly, from the perspective of sparse operation, and using the entropy properties, integrated the current and the next observation to choose the sparse feature point which is the weakest correlation to posture, so as to effectively improve the accuracy and consistency of the algorithm. And then according to sparse distribution characteristics of information matrix elements, the tridiagonal matrix is used to quickly solve inverse operation in the state average recovery

process, thus effectively improve the computation efficiency. Finally, establishing the simulation model of 3d environment, which is composed of the pre-set flight path of UAV and environmental feature points, and using the improved SEIF SLAM and the standard SEIF SLAM to the simulation environment for verification and comparing with the simulation results.

the simulation results show , the improved algorithm that is based on the entropy properties in the pose estimation precision and consistency compared with the standard algorithm has obvious improvement; the improved algorithm that is based on the tridiagonal matrix inversion in the computation efficiency is faster than the standard algorithm as the order number of information matrix increased gradually.

Key Words: Small unmanned aerial vehicle, SLAM, Vision odometer, Sparse extended information filter

目 录

引 言.....	1
1 绪论.....	3
1.1 基于无人机视觉 SLAM 的研究背景和意义.....	3
1.2 SLAM 技术发展历史以及研究现状.....	4
1.3 基于单目视觉传感器的 SLAM 技术概述.....	6
1.3.1 图像采集和视觉里程计.....	6
1.3.2 SLAM 技术优化算法.....	6
1.3.3 环境地图构建.....	6
1.3.4 闭环检测.....	6
1.4 本文研究工作和内容安排.....	7
1.4.1 本文研究工作.....	7
1.4.2 论文内容安排.....	8
2 SLAM 技术基本理论与系统模型.....	9
2.1 基于贝叶斯滤波的 SLAM 技术.....	9
2.1.1 贝叶斯滤波基本理论.....	9
2.1.2 将贝叶斯滤波算法应用于 SLAM 技术.....	11
2.2 基于 EKF 的 SLAM 技术.....	12
2.2.1 扩展卡尔曼滤波 (EKF) 基本理论.....	12
2.2.2 将扩展卡尔曼滤波应用于 SLAM 技术.....	18
2.3 基于 SEIF 的 SLAM 技术.....	23
2.3.1 信息滤波基本理论.....	23
2.3.2 将信息滤波应用于 SLAM 技术.....	24
2.4 基于无人机视觉 SLAM 系统模型的建立.....	28
2.4.1 基于无人机视觉 SLAM 系统的前端模型.....	28
2.4.2 基于无人机视觉 SLAM 系统的后端优化方程的建立.....	36
2.4.3 本章小结.....	38

3 视觉 SLAM 技术图像特征点的提取方法.....	39
3.1 图像特征点提取算法的基本原理.....	39
3.1.1 SIFT 算法基本原理.....	39
3.1.2 SURF 算法基本原理.....	45
3.1.3 ORB 算法基本原理.....	49
3.2 通过实验对图像特征点提取算法进行比较.....	54
3.2.1 基于 SIFT 算法的图像特征提取和匹配.....	54
3.2.2 基于 SURF 算法的图像特征提取和匹配.....	56
3.2.3 基于 ORB 算法的图像特征提取与匹配.....	59
3.3 地图特征点的初始化.....	61
3.4 本章小结.....	63
4 同时定位与地图构建技术的研究.....	63
4.1 基于熵规则的 SEIF SLAM 技术的研究.....	63
4.1.1 信息熵特性.....	63
4.1.2 改进稀疏规则.....	65
4.2 基于状态均值恢复的 SEIF SLAM 技术的研究.....	68
4.2.1 三对角矩阵求逆算法.....	68
4.3 本章小结.....	72
5 改进 SEIF SLAM 算法的仿真及分析.....	73
5.1 基于改进的熵规则的 SEIF SLAM 技术的实验结果及分析.....	75
5.2 基于状态均值恢复的 SEIF SLAM 技术的实验结果及分析.....	77
6 总结与展望.....	79
6.1 本文工作总结.....	79
6.2 未来工作展望.....	80
参考文献.....	82
在学研究成果.....	85
致 谢.....	86

引 言

随着计算机控制技术、计算机视觉技术以及人工智能技术的飞速发展，不仅使得无人机从传统的军事领域正逐步融入到民用领域的各个方面，比如：电力巡检、航空拍摄、搜索救援、快递送货等；而且在无人机的工作方式上也发生了巨大的变化，从传统的手动遥控方式到现在的完全自主化控制方式逐渐过渡。要实现无人机的智能化与自主化的关键因素在于导航定位技术，传统的导航定位系统主要采用 GPS 和惯导的方式来实现，对于采用 GPS 方式的导航系统，当无人机在某些特殊环境作业时，易受到环境的干扰使得 GPS 信号变弱或者完全失效；由于惯导本身具有固定的漂移率，导致在定位过程中产生的误差会随着时间而逐渐增大，使其很难单独应用于长时间的导航系统当中；虽然二者的结合能达到较好的效果，但是性价比相对只采用单个摄像机来实现导航定位来说就要高很多。

本文通过引入基于单目视觉的 SLAM 技术来实现无人机的导航与定位技术，视觉 SLAM 是指运动载体通过携带的视觉传感器来获取环境信息，根据获取的环境信息来建立地图，同时根据已建地图实现自我定位的过程。在 SLAM 系统的实现过程当中，要及时且准确地计算出运动载体的实时位置以及环境地图信息，对算法的实时性、精确度以及一致性方面的要求非常高。要达到以上要求的关键在于 SLAM 系统核心算法的优劣，即优化算法。优化算法主要分为两大类：基于概率框架下的 SLAM 技术和基于非概率框架下的 SLAM 技术。在概率框架下主要包括 EKF SLAM 方法，UKF SLAM 方法，EIF SLAM 方法，SEIF SLAM 方法以及 FastSLAM 方法等。在非概率框架下主要包括基于关键帧优化的 SLAM 系统以及基于图优化的 SLAM 方法等。大多数学者还是采用相对比较成熟的基于概率框架的 SLAM 技术。比较典型的是 EKF SLAM 技术，该算法时间复杂度与构建地图中特征点数呈二次方关系，使其难以应用到大环境下。本文选择基于概率框架下实时性最好的 SEIF 作为本文 SLAM 系统的优化算法，在此基础上又提出了两种改进算法；首先是为了进一步提高算法的实时性，在算法稀疏化过程当中，根据信息矩阵元素的稀疏性分布特点，利用三对角矩阵来快速求解在状态均值恢复过程中的求逆操作，从而有效提高了算法的计算效率；其次是从算法稀疏化操作角度出发利用熵性质、综合当前以及下一观测时刻来选择与位姿关联性最弱的环境特征作为稀疏特征点，从而有效提高了算法的精度

和一致性。在视觉传感器信息的提取过程当中，计算量要求非常大，本文通过编程对算法 SIFT、SURF 以及 ORB 在尺度不变性、旋转不变性以及时间性能上进行对比，从中选择实时性能最高且算法稳定性也相对较好的 ORB 作为图像特征点提取算法。

最后通过建立三维仿真环境，在三维环境当中预先设定好无人机的航行轨迹以及环境特征点；通过仿真实验，将改进的 SEIF SLAM 技术与标准 SEIF SLAM 技术应用于仿真三维环境当中进行验证，结果表明了改进算法的有效性。

1 绪论

1.1 基于无人机视觉 SLAM 的研究背景和意义

无人驾驶飞行器（Unmanned Aerial Vehicle, UAV）简称为无人机，它通常是指无人驾驶、自主推进且利用无线电遥控设备或自备的程序操控的不载人飞行器；其具有诸多优点，如使用灵活、生存能力强、无人员伤亡风险、机动性能好等。早期的无人机应用主要投入于军事领域，比如：情报侦察、定位校射、毁伤评估、电子战等。近年来，随着无人机各方面技术的飞速发展，使得无人机作为一种新生产品正逐步融合至各个民用领域；比如：地震灾情勘测、电力巡检、航空摄影、搜索救援、快递送货等。

然而，在无人机的整个飞行过程中导航定位系统起着至关重要的作用^[1-3]。传统的无人机导航方式主要通过卫星导航（GPS）、惯性导航（IMU）以及二者的组合。

惯导系统是一种自主的导航方式，优点在于不依赖于外界信息、且具有短时精度高、隐蔽性好、不易受外界干扰等；但最大的缺点是定位误差随时间而增大，且惯导本身具有固定的漂移率，这样使其很难单独应用于长时间的导航系统当中。

卫星导航系统（GPS）是指运动载体携带的目标接收机接收卫星发射的信号，测算卫星到目标接收机的传播延时，并算出运动载体与卫星之间的实际距离，从而对运动载体进行定位的过程。优点是具有全球、全天候提供的高精度的位置和速度等信息；但缺点是完全依赖于外界因素，易受外界因素干扰等致命弱点。对于任务环境复杂，有建筑物遮挡以及干扰因素强烈等场合，则难以使用 GPS 导航方式，这在很大程度上限制了无人机的应用发展^[5]。

上述导航定位系统自身固有的结构特性，均存在一定的不足；同时定位与地图构建（Simultaneous Localization and Mapping, SLAM）技术能够实现运动载体在某未知环境当中，通过携带的传感器来获取环境相关信息，并建立一个与周围环境一致的地图，同时通过该地图来估算自己在地图中的绝对位置，实现自主导航。SLAM 技术通过自身建立的环境地图来实现定位，防止定位误差的积累，有效提高了运动载体定位的精确度。在 SLAM 系统中可以使用不同的传感器，如：激光测距仪传感器、视觉传感器、超声波传感器以及传感器相融合等；由于视觉传感器的性价比最高，不仅能够获取丰富的环境信息，而且也适合于三

维空间环境的应用。故本文采用视觉传感器的方式来实现无人机环境信息的获取。

当我们把 SLAM 技术应用到无人机上时，便可以有效地解决当无人机进入一个未知且有一定干扰性的环境时，如何进行自适应地导航与定位，同时为进一步的避障以及路径规划等行为打下坚实的基础。这在很大程度上弥补了传统导航系统的不足，为无人机实现真正的自主导航提供了一个很好的解决方案。

通过以上的分析，可以看出对基于无人机视觉的同步定位与构图问题的深入研究。不仅具有深远的理论意义，同时还具有广泛的现实意义。

1.2 SLAM 技术发展历史以及研究现状

SLAM 又称 CML (Concurrent Mapping and Localization)^[6-8]，即同时定位与地图构建，是指运动载体在未知环境中从某一个未知位置出发，在其运动过程中根据携带的传感器观测数据来构建增量式地图，并且利用已建地图实现自定位的过程^[9]，实现运动载体的自主定位和导航。最初的 SLAM 概念起源于 1986 年在美国旧金山举行的 IEEE Robotics and Automation Conference 会议，同时这也预示着机器人和人工智能两大领域研究相结合的开端。

随后，Smith、Self 和 Cheeseman^[10]等人于 1988 年在 Ayache、Faugeras^[11]和 Chatila、Laumond^[12]工作的基础之上，最早提出了基于扩展卡尔曼滤波 (EKF) 的 SLAM 技术。随着 SLAM 技术的不断发展，经过研究者们近三十年的努力，衍生出各种 SLAM 技术的解决方法，主要分为两大类：基于概率框架下的 SLAM 技术和基于非概率框架下的 SLAM 技术。

1、基于概率框架的 SLAM 技术

在概率框架下主要可分为基于卡尔曼滤波 (KF) 的 SLAM 技术和基于粒子滤波 (PF) 的 SLAM 技术；从后验概率的不同出发，前者主要包括 EKF SLAM 方法^[13]，UKF SLAM 方法^[14]，EIF SLAM 方法^[15]以及 SEIF SLAM 方法^[16]等；后者主要包括 RBPF SLAM^[17]方法，FastSLAM 方法^[18]和 DP SLAM 方法^[19]等。除此之外，还有基于 Random Finite Set 的方法^[20]，该方法也是近年比较流行的滤波方法之一^[21]。

2、基于非概率框架下的 SLAM 技术

在非概率框架下主要包括基于关键帧优化算法的 SLAM 系统 PTAM (Parallel Tracking and Mapping)^[22, 23]，以及基于图优化的 SLAM 方法^[24]等。

在早期的 SLAM 系统中，运动载体携带的传感器主要是激光测距仪传感器，由于激光传感器的体积比较大，不适于搭载在较小的无人机上，除此之外，激光测距仪难以在非常混乱或拥挤的场合提取到角或者直线特征，具有一定的感知漂移的问题。随着计算机处理性能的大幅提升以及计算机视觉技术的快速发展，基于视觉的 SLAM 成为了目前一个非常热门的研究方向，并有大量的相关突破性理论出现。基于视觉的 SLAM 技术主要分为三类：

1) 单目摄像头 (Monocular Camera)

单目 SLAM 是利用相邻图像帧之间进行匹配计算出摄像头位姿的变换，而对应点的深度信息是通过两个不同视角上的图像帧进行三角测距而得到；通过不断地迭代上述过程来实现环境地图的构建和运动载体的定位。主要单目 SLAM 技术包括 MonoSLAM^[25]、PTAM^[26]、ORB-SLAM^[27]、SVO^[28]以及 LSD-SLAM^[29]等。

2) 双目摄像头 (Binocular Camera)

双目 SLAM 是直接利用左右摄像头通过三角法计算出对应点的深度信息，除此之外，其它的处理方式与单目类似；然而，当双目摄像头与目标物体距离较远时，会退化成单目视觉。主要双目 SLAM 技术包括 LSD-SLAM^[29]、ORB-SLAM^[27]等。

3) 深度摄像头 (RGB-D Camera)

深度摄像头主要是通过 Time Of Flight 等方法来直接获得图像以及对应图像的深度信息，相对于单目和双目，其优点在于容易获取目标物体的深度信息；其缺点主要是成本高，体积大，室外环境基本不可使用。主要基于 RGB-D 的 SLAM 技术包括 DTAM^[30]、DVO^[31]、RGBD-SLAM-V2^[32]以及 Kintinous^[33]等。

目前的 SLAM 研究大都是处于室内环境，要想在室外大环境下应用 SLAM 技术，必须要解决的是 SLAM 系统优化算法计算复杂度、状态的估计精度以及一致性等问题。为此本文在标准的稀疏扩展信息滤波 SLAM 技术的基础上提出了两种改进算法，首先是从稀疏化操作角度出发利用熵性质、综合当前以及下一观测时刻来选择与位姿关联性最弱的环境特征作为稀疏特征点，来提高算法的精度和一致性；然后是根据信息矩阵元素的稀疏性分布特点，利用三对角矩阵来快速求解在状态均值恢复过程中的求逆操作，来提高算法的计算效率。

1.3 基于单目视觉传感器的 SLAM 技术概述

1.3.1 图像采集和视觉里程计

图像采集是指通过单目摄像头来获取运动载体周围环境的图像信息，图像数据格式以一个图像序列或者是一个视频流的形式存在；然后将获取到的图像序列或者视频流输入到视觉里程计，通过视觉里程计可以增量式地计算出摄像机的方向和位置，即摄像机的运动轨迹；但由于相邻图像帧之间的误差会影响之后的运动轨迹的估计，也就意味着，它会随着时间的推移，不可避免地有漂移存在。实现方法主要有两种：基于特征点的方法和直接法。

1.3.2 SLAM 技术优化算法

从理论上来说，如果视觉里程计（VO）模块估计的摄像机的旋转矩阵和平移向量都是完全正确的话，那么我们就可以得到精确的定位和建图。但是在实际的试验或者实际应用中，我们得到的图像数据往往有很多噪声和干扰，且由于传感器的精度、错误的匹配等原因，都会对结果造成一定误差；并且在里程计算法当中我们只是把新的关键帧与相邻前一个关键帧进行比较，当比较的结果有误差时，就会把误差延续到后面的比较过程当中，而且误差会越来越大。为了解决这一问题，引入了优化算法。优化算法一般采用滤波器和图优化的方式。

1.3.3 环境地图构建

地图构建是指运动载体必须记录环境中特征的位置。构建地图的主要目的是用于导航、规划、通讯、可视化、交互等目的。常见的地图形式有以下几种：拓扑地图、度量地图、路标地图以及他们的组合方式等。

1.3.4 闭环检测

虽然经过优化算法处理之后能够得到一个比较优的解，但是当运动载体回到某一个之前去过的地方时，如果能识别出这个地方，并找到对应的关键帧进行比较，那么就可以得到比只使用优化算法处理更准确的结果。闭环检测的目的就是要解决这个问题。

闭环检测主要有三种方式，第一种方式是根据估计出来的相机位置，检测是否与之前某个位置邻近；第二种方式是根据图像来识别出是否之前到过这个场景，并搜索到之前场景的关键帧。最后一种方式使用的方法是 Bag-of-Words(BOW)，即词袋模型。综上所述，便可以得出基于视觉的 SLAM 技术框架图，如下所示

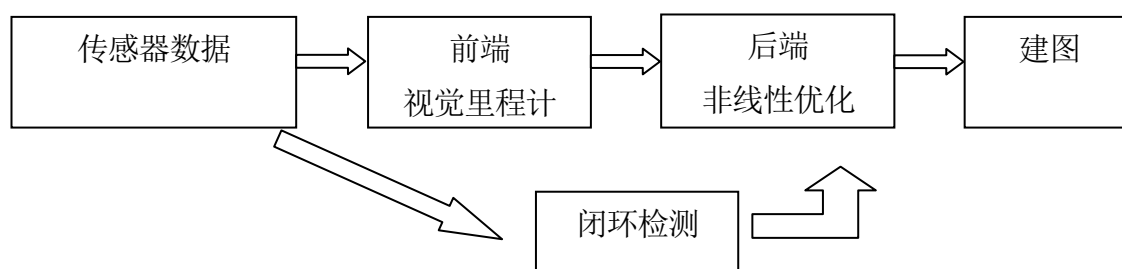


图 1.1 视觉 SLAM 系统流程图

Fig. 1.1 visual SLAM system flow diagram

1.4 本文研究工作和内容安排

1.4.1 本文研究工作

本文首先对基于无人机视觉的同时定位与地图构建技术相关背景及意义进行了阐述，同时对 SLAM 技术发展历史进行了简单的回顾，并综述了 SLAM 技术的发展现状。

其次，对基于无人机的单目视觉 SLAM 技术进行了简要分析；根据目前 SLAM 技术的发展现状，选择了其中的一个关键研究点，即 SLAM 技术的优化算法部分作为本论文的主要研究点。分别讨论了基于贝叶斯滤波的 SLAM 技术、基于扩展卡尔曼滤波的 SLAM 技术以及基于稀疏扩展信息滤波的 SLAM 技术。传统的优化算法一般采用的是基于扩展卡尔曼滤波的方式，该算法时间复杂度与地图中特征点的数量呈二次方关系，使其难以应用到大环境下；本文采用了改进的 SEIF 来替代 EKF 作为 SLAM 技术的优化算法，改进算法主要从 SLAM 技术的计算效率、系统状态估计精度以及一致性角度出发进行相应的改进。

然后，由于无人机在实际的飞行过程中是一种非平滑的运动方式，故不能采用输入控制命令的方式来预测下一时刻无人机的位姿，为此本文采用基于视觉里程计的方式来估计下一时刻无人机的位姿。在视觉里程计的特征点提取过程中计算量非常大，本文采用实时性高的 ORB 算法作为图像特征点提取算法，加快图像特征点的提取速度。

最后，通过建立三维仿真环境，并在仿真环境当中预先设定好无人机的航行轨迹以及环境特征点，将改进稀疏扩展信息滤波 SLAM 算法与标准稀疏扩展信息滤波 SLAM 算法进行仿真实验对比，并对实验结果进行分析。

1.4.2 论文内容安排

全文共 6 个章节，每个章节内容如下：

第一章：在绪论部分，首先阐述了基于无人机视觉的同时定位与构图技术的背景以及意义，然后简单回顾了同时定位与地图构建技术的发展历史和研究现状；与此同时，对本文即将要讨论的基于单目视觉的同时定位与地图构建技术进行了描述，最后指出本文将要研究的工作内容。

第二章：首先介绍并推导了 SLAM 技术的基本理论，包括贝叶斯滤波理论、扩展卡尔曼滤波理论以及稀疏扩展信息滤波理论；然后介绍了基于无人机单目视觉 SLAM 系统的视觉前端模型；最后是构建基于无人机单目视觉 SLAM 系统的后端优化算法方程模型，包括无人机单目视觉 SLAM 系统的运动方程和观测方程。

第三章：首先对图像特征点的提取方法 *SIFT*、*SURF* 以及 *ORB* 的基本原理进行介绍，并分别对各算法进行推导；然后通过编写代码来实现三种算法，将无人机拍摄的环境图像提取出来，分别从尺度不变性、旋转不变性以及计算效率方面对上述三种算法进行比较；最后对基于单目视觉 SLAM 技术中所特有的特征点的初始化问题进行分析。

第四章：本章主要是对 *SEIF SLAM* 技术进行研究，通过在第二章中对稀疏扩展信息滤波的理论分析以及推导得出改进算法，改进算法主要从算法的计算效率、状态估计精度以及一致性角度出发，从而提出了包括基于熵规则的信息滤波 SLAM 技术和基于状态向量均值恢复的信息滤波 SLAM 技术。

第五章：本章主要是设计并建立本系统的三维仿真环境，包括在三维仿真环境当中预先设定好无人机的航行轨迹以及周围环境特征点信息，然后将改进的 *SEIF SLAM* 算法与标准的 *SEIF SLAM* 算法应用于仿真环境当中进行仿真对比分析。

第六章：本章首先对全文的工作进行总结，然后为进一步完善后续研究内容提出了一些建议。

2 SLAM 技术基本理论与系统模型

2.1 基于贝叶斯滤波的 SLAM 技术

2.1.1 贝叶斯滤波基本理论

1、基本概念

目前对于同时定位与地图构建技术已经存在很多有效算法^[34-35]，在经典概率框架下的各种滤波算法，如 EKF、PF、SEIF 以及 UKF 等均属于贝叶斯滤波理论范畴。基本贝叶斯滤波模型是由先验模型和量测模型构成的。先验模型由参数先验信息构成，而量测模型决定了由先验参数到量测量之间的关系。结合上述融合规则，即贝叶斯法则，就可以由量测量推断出参数的估计值。假设要估计某系统状态量 x_k ，且已知该系统的运动模型和观测模型，那么可以通过贝叶斯滤波来解决，其一般步骤为：

1) 初始化：由先验分布 $bel(x_0)$ 开始进行递归解算。

2) 预测：已知系统的运动模型，在 k 时刻，状态 x_k 的预测分布情况可通过 Chapman-Kolmogorov 方程计算得出，表达式如下所示：

$$\begin{aligned} \overline{bel}(x_k) &= p(x_k | z_{1:k-1}, u_{1:k}) \\ &= \int p(x_k | x_{k-1}, z_{1:k-1}, u_{1:k}) p(x_{k-1} | z_{1:k-1}, u_{1:k}) dx_{k-1} \end{aligned} \quad (2.1)$$

3) 更新：在已知 k 时刻的量测信息 z_k 之后， x_k 的后验分布就可以通过下述贝叶斯公式计算得出：

$$bel(x_k) = \eta p(z_k | x_k) \overline{bel}(x_k) \quad (2.2)$$

2、贝叶斯定理推导过程

第一步，把贝叶斯定理应用到目标系统状态的后验分布中，如下

$$\begin{aligned}
 p(x_k \mid z_{1:k}, u_{1:k}) &= \frac{p(z_k \mid x_k, z_{1:k-1}, u_{1:k})p(x_k \mid z_{1:k-1}, u_{1:k})}{p(z_k \mid z_{1:k-1}, u_{1:k})} \\
 &= \eta p(z_k \mid x_k, z_{1:k-1}, u_{1:k})p(x_k \mid z_{1:k-1}, u_{1:k})
 \end{aligned} \tag{2.3}$$

由于 k 时刻的观测量 z_k ，通过系统的观测方程可知，只与 k 时刻的系统状态相关，与控制量和过去时刻的观测量无关，则有

$$p(z_k \mid x_k, z_{1:k-1}, u_{1:k}) = p(z_k \mid x_k) \tag{2.4}$$

故式 (2.3) 可写为

$$p(x_k \mid z_{1:k}, u_{1:k}) = \eta p(z_k \mid x_k)p(x_k \mid z_{1:k-1}, u_{1:k}) \tag{2.5}$$

第二步，推导式 (2.5) 的后半部分，如下

令 $\overline{bel}(x_k) = p(x_k \mid z_{1:k-1}, u_{1:k})$ ，则由全概率公式可知

$$\begin{aligned}
 \overline{bel}(x_k) &= p(x_k \mid z_{1:k-1}, u_{1:k}) \\
 &= \int p(x_k, x_{k-1} \mid z_{1:k-1}, u_{1:k}) dx_{k-1} \\
 &= \int p(x_k \mid x_{k-1}, z_{1:k-1}, u_{1:k})p(x_{k-1} \mid z_{1:k-1}, u_{1:k}) dx_{k-1} \\
 &= \int p(x_k \mid x_{k-1}, u_k)p(x_{k-1} \mid z_{1:k-1}, u_{1:k}) dx_{k-1} \\
 &= \int p(x_k \mid x_{k-1}, u_k)\overline{bel}(x_{k-1}) dx_{k-1}
 \end{aligned} \tag{2.6}$$

式 (2.6) 中等式一到等式三是根据全概率公式得出的；等式三到等式四是根据系统的状态方程而来，综上所述，贝叶斯滤波可以写成如下递归形式：

$$bel(x_k) = \eta p(z_k \mid x_k) \int p(x_k \mid x_{k-1}, u_k) bel(x_{k-1}) dx_{k-1} \tag{2.7}$$

其中 η 被称为归一化因子。

2.1.2 将贝叶斯滤波算法应用于 SLAM 技术

SLAM 问题解决的是系统中运动载体的位置确定和环境地图的构建，故通过将二者的联合状态作为要估计的系统状态，那么基于贝叶斯滤波算法的 SLAM 技术主要分为两个实现步骤，即预测和更新。

假设要估计的系统状态量由运动载体的位姿 x_t 和环境地图 M 构成，传感器从 1 到 t 时刻获得的观测由 $z_{1:t}$ 表示，从 1 到 t 时刻的系统控制量由 $u_{1:t}$ ，那么 SLAM 问题可以表示为： $p(x_t, M | z_{1:t}, u_{1:t})$ 。

1、预测阶段

在运动载体运动过程中，根据其 $t-1$ 时刻的状态，通过 t 时刻的控制量输入到运动模型中，对 t 时刻的位姿进行预测，其公式如下所示，

$$p(x_{t-1}, M | z_{1:t-1}, u_{1:t-1}) \xrightarrow{u_t} \hat{p}(x_t, M | z_{1:t-1}, u_{1:t}) \quad (2.8)$$

可以把预测过程分为两个子过程进行处理，分别是状态的放大和边缘化上一时刻的位姿。状态放大是指运动载体从 $t-1 \rightarrow t$ 时刻，在输入 t 时刻的控制量之后，将预测的新位姿 x_t 加入到系统状态向量中，如下所示，

$$p(x_{t-1}, M | z_{1:t-1}, u_{1:t-1}) \xrightarrow{u_t} p(x_{t-1}, x_t, M | z_{1:t-1}, u_{1:t}) \quad (2.9)$$

在上面式子中， u_t 为运动载体从 $t-1 \rightarrow t$ 时刻的运动控制输入量， x_t 是运动载体运动后预测的新位姿。边缘化上一时刻位姿的过程是指在状态放大后的基础上对上一时刻运动载体的位姿求积分，其过程如下所示，

$$\begin{aligned} & p(x_t, M | z_{1:t}, u_{1:t}) \\ &= \int p(x_{t-1}, x_t, M | z_{1:t-1}, u_{1:t}) dx_{t-1} \\ &= \int p(x_{t-1} | x_t, u_t) \cdot p(x_{t-1}, M | z_{1:t-1}, u_{1:t-1}) dx_{t-1} \end{aligned} \quad (2.10)$$

2、更新阶段

运动载体根据传感器在 t 时刻观测到的数据来对上面预测结果进行更新，如下所示，

$$p(x_t, M | z_{1:t-1}, u_{1:t}) \xrightarrow{z_t} p(x_t, M | z_{1:t}, u_{1:t}) \quad (2.11)$$

在贝叶斯滤波的过程当中，预测和更新过程是不断在交替迭代进行的，其过程可总结如下，

基于贝叶斯滤波的 SLAM 技术

$$p(x_t, M | z_{1:t}, u_{1:t}) \propto p(z_t | x_t, M) \cdot \int p(x_t | x_{t-1}, u_t) \cdot p(x_{t-1}, M | z_{1:t-1}, u_{1:t-1}) dx_{t-1}$$

1) 预测

$$p(x_{t-1}, M | z_{1:t-1}, u_{1:t-1}) \xrightarrow{u_t} \hat{p}(x_t, M | z_{1:t-1}, u_{1:t})$$

(a) 系统状态的放大

$$p(x_{t-1}, M | z_{1:t-1}, u_{1:t-1}) \xrightarrow{u_t} p(x_{t-1}, x_t, M | z_{1:t-1}, u_{1:t})$$

(b) 对上一时刻的位姿进行边缘化

$$\begin{aligned} & p(x_t, M | z_{1:t-1}, u_{1:t}) \\ &= \int p(x_{t-1}, x_t, M | z_{1:t-1}, u_{1:t}) dx_{t-1} \\ &= \int p(x_{t-1} | x_t, u_t) \cdot p(x_{t-1}, M | z_{1:t-1}, u_{1:t-1}) dx_{t-1} \end{aligned}$$

2) 更新

$$p(x_t, M | z_{1:t-1}, u_{1:t}) \xrightarrow{z_t} p(x_t, M | z_{1:t}, u_{1:t})$$

图 2.1 基于贝叶斯滤波的 SLAM

Fig 2.1 SLAM based on bayesian filter

2.2 基于 EKF 的 SLAM 技术

2.2.1 扩展卡尔曼滤波 (EKF) 基本理论

1、基本概念

卡尔曼滤波理论来源于匈牙利数学家 *Rudolph Emil Kalman*，它是在高斯分布假设的前提下，用于处理线性系统问题^[36]。在实际情况中，很少有线性系统存在，为了解决非线性系统这一问题，文献^[37,38]通过扩展卡尔曼滤波算法，将非线性系统的非线性函数模型在某一估计点处进行泰勒级数展开，实现系统模型的线性化。下图为扩展卡尔曼滤波算法的实现原理，

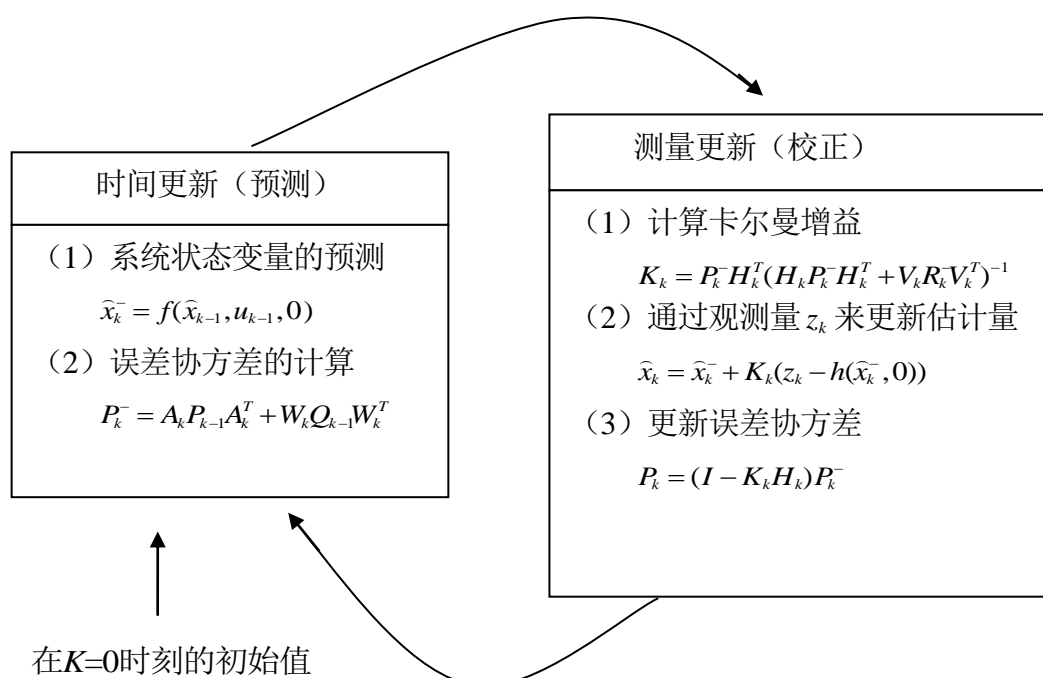


图 2.2 EKF 实现过程

Fig.2.2 EKF realization process

2、扩展卡尔曼滤波的推导过程

1) 系统函数的线性化处理

假设系统的状态方程和观测方程分别由非线性函数 g 和 h 表示，如下

$$x_t = g(u_t, x_{t-1}) + \varepsilon_t \quad (2.12)$$

$$z_t = h(x_t) + \delta_t \quad (2.13)$$

其中随机向量 ε_t 和 δ_t 分别是状态方程和观测方程的噪声，他们均服从均值为 0 的高斯分布。

在扩展卡尔曼滤波当中很重要的一点就是对非线性函数的线性化处理，通过在函数均值 μ_t 处进行一阶泰勒级数展开，然后截断后面的高阶部分进行近似处理，结果如下所示，

$$\begin{aligned} g(u, x_{t-1}) &\approx g(u, \mu_{t-1}) + \underbrace{g'(u, \mu_{t-1})}_{=: G_t} (x_{t-1} - \mu_{t-1}) \\ &= g(u, \mu_{t-1}) + G_t (x_{t-1} - \mu_{t-1}) \end{aligned} \quad (2.14)$$

用高斯概率的形式表示为

$$\begin{aligned} p(x_t | u, x_{t-1}) &\approx \det(2\pi R_t)^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} [x_t - g(u, \mu_{t-1}) - G_t (x_{t-1} - \mu_{t-1})]^T \right. \\ &\quad \left. R_t^{-1} [x_t - g(u, \mu_{t-1}) - G_t (x_{t-1} - \mu_{t-1})] \right\} \end{aligned} \quad (2.15)$$

其中 G_t 是一个 $n \times n$ 的矩阵， n 是系统状态的维数， G_t 也称为雅克比矩阵。

同理，对于系统的观测方程 h 可以在均值 $\bar{\mu}_t$ 处进行泰勒展开，如下所示

$$\begin{aligned} h(x_t) &\approx h(\bar{\mu}_t) + \underbrace{h'(\bar{\mu}_t)}_{=: H_t} (x_t - \bar{\mu}_t) \\ &= h(\bar{\mu}_t) + H_t (x_t - \bar{\mu}_t) \end{aligned} \quad (2.16)$$

其中 $h'(x_t) = \frac{\partial h(x_t)}{\partial x_t}$ 。

将上面写成高斯概率的形式，如下

$$\begin{aligned}
 & p(z_t | x_t) \\
 & \approx \det(2\pi Q_t)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}[z_t - h(\bar{\mu}_t) - H_t(x_t - \bar{\mu}_t)]^T \right. \\
 & \quad \left. Q_t^{-1}[z_t - h(\bar{\mu}_t) - H_t(x_t - \bar{\mu}_t)]\right\}
 \end{aligned} \tag{2.17}$$

2) 算法的数学推导

第一步：初始化。

$$bel(x_0) = p(x_0) = \det(2\pi\Sigma_0)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(x_0 - \mu_0)^T \Sigma_0^{-1}(x_0 - \mu_0)\right\} \tag{2.18}$$

第二步：预测。

$$\overline{bel}(x_t) = \int p(x_t | x_{t-1}, u_t) bel(x_{t-1}) dx_{t-1} \tag{2.19}$$

其中 $p(x_t | x_{t-1}, u_t) \sim N(x_t; g(u_t, \mu_{t-1}) + G(u_t, \mu_{t-1}), R_t)$, $bel(x_{t-1}) \sim N(x_{t-1}; u_{t-1}, \Sigma_{t-1})$ 。

将式 (2.19) 写成高斯的形式，如下

$$\begin{aligned}
 \overline{bel}(x_t) = & \eta \int \exp\left\{-\frac{1}{2}(x_t - g(u_t, \mu_{t-1}) - G(u_t, \mu_{t-1}))^T R_t^{-1}(x_t - g(u_t, \mu_{t-1}) - G(u_t, \mu_{t-1}))\right\} \\
 & \exp\left\{-\frac{1}{2}(x_{t-1} - u_{t-1})^T \Sigma_{t-1}^{-1}(x_{t-1} - u_{t-1})\right\} dx_{t-1}
 \end{aligned} \tag{2.20}$$

将上式简写为如下，

$$\overline{bel}(x_t) = \eta \int \exp(-L_t) dx_{t-1} \tag{2.21}$$

其中

$$\begin{aligned}
 L_t = & \frac{1}{2} (x_t - g(u_t, \mu_{t-1}) - G_t(x_{t-1} - \mu_{t-1}))^T \\
 & R_t^{-1} (x_t - g(u_t, \mu_{t-1}) - G_t(x_{t-1} - \mu_{t-1})) \\
 & + \frac{1}{2} (x_{t-1} - u_{t-1})^T \Sigma_{t-1}^{-1} (x_{t-1} - u_{t-1})
 \end{aligned} \tag{2.22}$$

因为式 (2.21) 是一个积分的形式, 要求其闭环解, 把 L_t 分解成两个函数, $L_t(x_{t-1}, x_t)$ 和 $L_t(x_t)$, 则有

$$\begin{aligned}
 L_t(x_{t-1}, x_t) = & \frac{1}{2} (x_{t-1} - \Phi_t [G_t^T R_t^{-1} (x_t - g(u_t, \mu_{t-1}) + G_t \mu_{t-1}) + \Sigma_{t-1}^{-1} \mu_{t-1}])^T \Phi_t^{-1} \\
 & (x_{t-1} - \Phi_t [G_t^T R_t^{-1} (x_t - g(u_t, \mu_{t-1}) + G_t \mu_{t-1}) + \Sigma_{t-1}^{-1} \mu_{t-1}])
 \end{aligned} \tag{2.23}$$

其中 $\Phi_t = (G_t^T R_t^{-1} G_t + \Sigma_{t-1}^{-1})^{-1}$ 。

$$\begin{aligned}
 L_t(x_t) = & \frac{1}{2} (x_t - g(u_t, \mu_{t-1}) + G_t \mu_{t-1})^T R_t^{-1} (x_t - g(u_t, \mu_{t-1}) + G_t \mu_{t-1}) \\
 & + \frac{1}{2} (x_{t-1} - \mu_{t-1})^T \Sigma_{t-1}^{-1} (x_{t-1} - \mu_{t-1}) \\
 & - \frac{1}{2} [G_t^T R_t^{-1} (x_t - g(u_t, \mu_{t-1}) + G_t \mu_{t-1}) + \Sigma_{t-1}^{-1} \mu_{t-1}]^T \\
 & \Phi_t [G_t^T R_t^{-1} (x_t - g(u_t, \mu_{t-1}) + G_t \mu_{t-1}) + \Sigma_{t-1}^{-1} \mu_{t-1}]
 \end{aligned} \tag{2.24}$$

通过这样分解就可以把 $L_t(x_t)$ 移到积分号外面, 因为它与 x_{t-1} 无关, 则式子 (2.21) 可写为

$$\begin{aligned}
 \overline{bel}(x_t) = & \eta \int \exp(-L_t) dx_{t-1} \\
 = & \eta \int \exp\{-L_t(x_{t-1}, x_t) - L_t(x_t)\} dx_{t-1} \\
 = & \eta \exp\{-L_t(x_t)\} \int \exp\{-L_t(x_{t-1}, x_t)\} dx_{t-1}
 \end{aligned} \tag{2.25}$$

由于通过推算积分 $\int \exp\{-L_t(x_{t-1}, x_t)\} dx_{t-1}$ 最终取值与 x_t 无关, 故上式可简写为

$$\overline{bel}(x_t) = \eta \exp\{-L_t(x_t)\} \quad (2.26)$$

下面就是要决定函数 $L_t(x_t)$ 的均值和协方差，对其一阶求导，令导数为 0，则有

$$\overline{\mu}_t = g(u, \mu_{t-1}) \quad (2.27)$$

然后再求取其二阶导，最终二阶导的逆便是协方差矩阵，如下所示

$$\overline{\Sigma}_t = R_t + G_t \Sigma_{t-1} G_t^T \quad (2.28)$$

第三步：测量更新

$$bel(x_t) = \eta p(z_t | x_t) \overline{bel}(x_t) \quad (2.29)$$

其中 $p(z_t | x_t) \sim N(z_t | h(\overline{\mu}_t) + H_t(x_t - \overline{\mu}_t), Q_t)$ ， $\overline{bel}(x_t) \sim N(x_t; \overline{\mu}_t, \overline{\Sigma}_t)$

利用线性化后的式子 (2.16)，则有如下式子

$$bel(x_t) = \eta \exp\{-J_t\} \quad (2.30)$$

其中

$$\begin{aligned} J_t = & \frac{1}{2} (z_t - h(\overline{\mu}_t) - H_t(x_t - \overline{\mu}_t))^T Q_t^{-1} (z_t - h(\overline{\mu}_t) - H_t(x_t - \overline{\mu}_t)) \\ & + \frac{1}{2} (x_t - \overline{\mu}_t)^T \overline{\Sigma}_t^{-1} (x_t - \overline{\mu}_t) \end{aligned} \quad (2.31)$$

最后再对 J_t 进行一阶和二阶求导，从而得出均值和协方差，如下所示

$$\begin{aligned}\mu_t &= \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t)) \\ \Sigma_t &= (I - K_t H_t) \bar{\Sigma}_t\end{aligned}\quad (2.32)$$

其中卡尔曼增益为 $K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1}$ 。

2.2.2 将扩展卡尔曼滤波应用于 SLAM 技术

1、背景知识

最初的扩展卡尔曼滤波（EKF）和同时定位与地图构建技术（SLAM）的结合是在 Smith, Self 以及 Cheeseman^[39] 的研究成果基础之上，他们将机器人位姿和环境特征点的分布采用离散的状态空间框架，并对系统运动方程和观测方程进行线性化以及高斯分布的假设，利用 EKF 来对状态空间中的运动载体位姿和环境特征点位置进行同时估计来完成 SLAM 技术。

为了方便，我们用 y_t 来表示包含运动载体位姿 x_t 和地图特征 m 组成的联合系统状态向量，如下所示

$$\begin{aligned}y_t &= \begin{pmatrix} x_t \\ m \end{pmatrix} \\ &= (x \quad y \quad \theta \quad m_{1,x} \quad m_{1,y} \quad s_1 \quad m_{2,x} \quad m_{2,y} \quad s_2 \quad \cdots \quad m_{N,x} \quad m_{N,y} \quad s_N)^T\end{aligned}\quad (2.33)$$

上面的 x , y , θ 表示运动载体在 t 时刻的坐标； $m_{i,x}$, $m_{i,y}$ 是第 i 个路标的坐标系， s_i 对应于第 i 个路标的特征码；那么系统状态向量的维数为 $3N+3$ ， N 表示地图中路标数量。则 EKF SLAM 可以通过计算后验概率 $p(y_t | z_{1:t}, u_{1:t})$ 来表示系统的状态，由于系统是基于高斯分布假设，故目的是要求出均值和协方差矩阵，分别表示为

$$\mu = (x \quad y \quad \theta \quad m_{1,x} \quad m_{1,y} \quad m_{2,x} \quad m_{2,y} \quad \cdots \quad m_{N,x} \quad m_{N,y})^T \quad (2.34)$$

$$\Sigma = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{x\theta} & \sigma_{xm_{1,x}} & \sigma_{xm_{1,y}} & \cdots & \sigma_{xm_{N,x}} & \sigma_{xm_{N,y}} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{y\theta} & \sigma_{ym_{1,x}} & \sigma_{ym_{1,y}} & \cdots & \sigma_{ym_{N,x}} & \sigma_{ym_{N,y}} \\ \sigma_{\theta x} & \sigma_{\theta y} & \sigma_{\theta\theta} & \sigma_{\theta m_{1,x}} & \sigma_{\theta m_{1,y}} & \cdots & \sigma_{\theta m_{N,x}} & \sigma_{\theta m_{N,y}} \\ \sigma_{m_{1,x}x} & \sigma_{m_{1,x}y} & \sigma_{\theta} & \sigma_{m_{1,x}m_{1,x}} & \sigma_{m_{1,x}m_{1,y}} & \cdots & \sigma_{m_{1,x}m_{N,x}} & \sigma_{m_{1,x}m_{N,y}} \\ \sigma_{m_{1,y}x} & \sigma_{m_{1,y}y} & \sigma_{\theta} & \sigma_{m_{1,y}m_{1,x}} & \sigma_{m_{1,y}m_{1,y}} & \cdots & \sigma_{m_{1,y}m_{N,x}} & \sigma_{m_{1,y}m_{N,y}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \sigma_{m_{N,x}x} & \sigma_{m_{N,x}y} & \sigma_{\theta} & \sigma_{m_{N,x}m_{1,x}} & \sigma_{m_{N,x}m_{1,y}} & \cdots & \sigma_{m_{N,x}m_{N,x}} & \sigma_{m_{N,x}m_{N,y}} \\ \sigma_{m_{N,y}x} & \sigma_{m_{N,y}y} & \sigma_{\theta} & \sigma_{m_{N,y}m_{1,x}} & \sigma_{m_{N,y}m_{1,y}} & \cdots & \sigma_{m_{N,y}m_{N,x}} & \sigma_{m_{N,y}m_{N,y}} \end{pmatrix}^T \quad (2.35)$$

将上式简写为

$$\mu = (x \quad m)^T \quad (2.36)$$

$$\Sigma = \begin{pmatrix} \Sigma_{xx} & \Sigma_{xm} \\ \Sigma_{mx} & \Sigma_{mm} \end{pmatrix} \quad (2.37)$$

2、从数学观点来推导基于 EKF 的 SLAM 技术

第一步：初始时刻只知道运动载体的位姿，且以该位置作为最初的坐标系；同时在此时没有任何的环境特征是已知的，那么可通过初始化均值 μ_0 和协方差 Σ_0 的形式来表达如下：

$$\mu_0 = (0 \quad 0 \quad 0 \quad \cdots \quad 0)^T \quad (2.38)$$

$$\Sigma_0 = \begin{pmatrix} 0 & 0 & 0 & \infty & \cdots & \infty \\ 0 & 0 & 0 & \infty & \cdots & \infty \\ 0 & 0 & 0 & \infty & \cdots & \infty \\ \infty & \infty & \infty & \infty & \cdots & \infty \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \infty & \infty & \infty & \infty & \infty & \infty \end{pmatrix} \quad (2.39)$$

协方差矩阵是一个 $(3N+3) \times (3N+3)$ 大小的矩阵，其左上角的 3×3 维矩阵是由运动载体位姿决定的矩阵，其他的初始值都是无限大。

第二步：运动更新

随着运动载体的移动，系统状态向量可以表示为（以速度运动模型为例）

$$y_t = y_{t-1} + \begin{pmatrix} -\frac{v_t}{w_t} \sin \theta + \frac{v_t}{w_t} \sin(\theta + w_t \Delta_t) \\ \frac{v_t}{w_t} \cos \theta - \frac{v_t}{w_t} \cos(\theta + w_t \Delta_t) \\ w_t \Delta_t + \gamma_t \Delta_t \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (2.40)$$

由于运动载体的移动只改变位置，与环境特征无关，故可将上面简写为

$$y_t = y_{t-1} + F_x \begin{pmatrix} -\frac{v_t}{w_t} \sin \theta + \frac{v_t}{w_t} \sin(\theta + w_t \Delta_t) \\ \frac{v_t}{w_t} \cos \theta - \frac{v_t}{w_t} \cos(\theta + w_t \Delta_t) \\ w_t \Delta_t + \gamma_t \Delta_t \end{pmatrix} \quad (2.41)$$

$$\text{其中 } F_x = \begin{pmatrix} 1 & 0 & 0 & 0 \cdots 0 \\ 0 & 1 & 0 & 0 \cdots 0 \\ 0 & 0 & 1 & 0 \cdots 0 \end{pmatrix}。$$

共有3N列

加入系统运动模型的噪声之后，可变为如下形式

$$y_t = y_{t-1} + \underbrace{F_x \begin{pmatrix} -\frac{v_t}{w_t} \sin \theta + \frac{v_t}{w_t} \sin(\theta + w_t \Delta_t) \\ \frac{v_t}{w_t} \cos \theta - \frac{v_t}{w_t} \cos(\theta + w_t \Delta_t) \\ w_t \Delta_t + \gamma_t \Delta_t \end{pmatrix}}_{g(u_t, y_{t-1})} + N(0, F_x^T R_t F_x) \quad (2.42)$$

接下来对 运动方程 g 进行一阶泰勒展开,

$$g(u_t, y_{t-1}) \approx g(u_t, \mu_{t-1}) + G_t(y_{t-1} - \mu_{t-1}) \quad (2.43)$$

其中 G_t 称为雅克比矩阵, 它是 运动方程 g 在 u_t 处的导数, 其表达形式为 $G_t = g'(u_t, \mu_{t-1})$; 且 G_t 可以简化为如下

$$G_t = I + F_x^T g_t F_x \quad (2.44)$$

其中

$$g_t = \begin{pmatrix} 0 & 0 & \frac{v_t}{w_t} \cos \mu_{t-1, \theta} - \frac{v_t}{w_t} \cos(\mu_{t-1, \theta} + w_t \Delta_t) \\ 0 & 0 & \frac{v_t}{w_t} \sin \mu_{t-1, \theta} - \frac{v_t}{w_t} \sin(\mu_{t-1, \theta} + w_t \Delta_t) \\ 0 & 0 & 0 \end{pmatrix} \quad (2.45)$$

将上面式子带入标准的 EKF 之后, 便得到 运动更新的预测结果, 如下

$$\bar{\mu}_t = \mu_{t-1} + F_x^T \begin{pmatrix} -\frac{v_t}{w_t} \sin \mu_{t-1, \theta} + \frac{v_t}{w_t} \sin(\mu_{t-1, \theta} + w_t \Delta_t) \\ \frac{v_t}{w_t} \cos \mu_{t-1, \theta} - \frac{v_t}{w_t} \cos(\mu_{t-1, \theta} + w_t \Delta_t) \\ w_t \Delta_t \end{pmatrix} \quad (2.46)$$

$$\bar{\Sigma}_t = G_t - \Sigma_{t-1} G_t^T + F_x^T R_t F_x \quad (2.47)$$

第三步: 观测更新

假设系统的观测模型为

$$z_t^i = \underbrace{\begin{pmatrix} \sqrt{(m_{j,x} - x)^2 + (m_{j,y} - y)^2} \\ a \tan 2(m_{j,y} - y, m_{j,x} - x) - \theta \\ m_{j,s} \end{pmatrix}}_{h(y_t, j)} + N \left(0, \underbrace{\begin{pmatrix} \sigma_r & 0 & 0 \\ 0 & \sigma_\phi & 0 \\ 0 & 0 & \sigma_s \end{pmatrix}}_{Q_t} \right) \quad (2.48)$$

其中 x, y, θ 表示运动载体的位姿, i 表示在 t 时刻观测到的第 i 个环境特征点, $j = c_t^i$ 是指在 t 时刻观测到的路标索引; 对系统观测方程通过泰勒展开进行一阶线性化, 得到如下

$$h(y_t, j) \approx h(\bar{\mu}_t, j) + H_t^i (y_t - \bar{\mu}_t) \quad (2.49)$$

其中 H_t^i 是函数 h 在 $\bar{\mu}_t$ 处的雅克比矩阵, 因为 h 只跟状态向量的位姿 x_t 和第 j 个路标 m_j 的位置相关。那么可以 H_t^i 把简写为低维的雅克比矩阵 h_t^i 和 $F_{x,j}$ 的乘积, 如下

$$H_t^i = h_t^i F_{x,j} \quad (2.50)$$

那么 h_t^i 相对于位姿变量 x_t 和路标 m_j 的计算结果如下

$$h_t^i = \begin{pmatrix} \frac{m_{j,x} - \bar{\mu}_{t,x}}{\sqrt{q_t}} & \frac{y_t - \bar{\mu}_{t,y}}{\sqrt{q_t}} & 0 & \frac{\bar{\mu}_{t,x} - m_{j,x}}{\sqrt{q_t}} & \frac{\bar{\mu}_{t,y} - y_t}{\sqrt{q_t}} & 0 \\ \frac{\bar{\mu}_{t,y} - y_t}{q_t} & \frac{m_{j,x} - \bar{\mu}_{t,x}}{q_t} & -1 & \frac{y_t - \bar{\mu}_{t,y}}{q_t} & \frac{\bar{\mu}_{t,x} - m_{j,x}}{\sqrt{q_t}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.51)$$

其中标量

$$q_t = (m_{j,x} - \bar{\mu}_{t,x})^2 + (m_{j,y} - \bar{\mu}_{t,y})^2 \quad (2.52)$$

$$F_{x,j} = \begin{pmatrix} 1 & 0 & 0 & 0 \cdots 0 & 0 & 0 & 0 & 0 \cdots 0 \\ 0 & 1 & 0 & 0 \cdots 0 & 0 & 0 & 0 & 0 \cdots 0 \\ 0 & 0 & 1 & 0 \cdots 0 & 0 & 0 & 0 & 0 \cdots 0 \\ 0 & 0 & 0 & 0 \cdots 0 & 1 & 0 & 0 & 0 \cdots 0 \\ 0 & 0 & 0 & 0 \cdots 0 & 0 & 1 & 0 & 0 \cdots 0 \\ 0 & 0 & 0 & 0 \cdots 0 & 0 & 0 & 1 & 0 \cdots 0 \end{pmatrix} \quad (2.53)$$

$\begin{matrix} & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \end{matrix}$
 $\begin{matrix} & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \end{matrix}$

卡尔曼增益为

$$K_t^i = \bar{\Sigma}_t H_t^{iT} (H_t^i \bar{\Sigma}_t H_t^{iT} + Q_t)^{-1} \quad (2.54)$$

由此，可以得出最后的均值 μ_t 和协方差 Σ_t ，如下所示

$$\mu_t = \bar{\mu}_t + \Sigma_t K_t^i (z_t^i - \hat{z}_t^i) \quad (2.55)$$

$$\Sigma_t = (I - \Sigma_t K_t^i H_t^i) \bar{\Sigma}_t \quad (2.56)$$

2.3 基于 SEIF 的 SLAM 技术

2.3.1 信息滤波基本理论

SLAM 问题可以用一个后验条件概率来表示，该后验条件概率服从一个多维的高斯分布，可表示为：

$$\begin{aligned} p(\xi_t | z^t, u^t) \\ &\propto \exp\{-\frac{1}{2}(\xi_t - \mu_t)^T \Sigma_t^{-1}(\xi_t - \mu_t)\} \\ &\propto \exp\{-\frac{1}{2}\xi_t^T \Sigma_t^{-1} \xi_t + \mu_t^T \Sigma_t^{-1} \xi_t\} \end{aligned} \quad (2.57)$$

其中 $\xi_t = [x_t^T \ M^T]^T$ ，称其为系统的增广状态向量； x_t 为机器人位姿， M 为环境路标集合， μ_t 为分布期望值， Σ_t 为协方差矩阵， z^t 为传感器从 1 到 t 时刻的观测值， u^t 为从 1 到 t 时刻的运动控制量。若定义信息矩阵 $\Lambda_t = \Sigma_t^{-1}$ ，信息向量 $\eta_t = \mu_t^T \Lambda_t$ ，那么把这一表示形式称为高斯分布的标准模型。这种利用高斯分布的标准模型来估计机器人状态向量和环境地图的方法叫信息滤波^[40-42]。

2.3.2 将信息滤波应用于 SLAM 技术

1、 观测更新

假设系统观测模型为

$$z_t = h(\xi_t) + \varepsilon_t \quad (2.58)$$

其中 ε_t 是一个具有高斯分布的噪声随机向量，其均值为 0，协方差矩阵为 R_t ；图 2.3 说明了观测对信息矩阵的影响

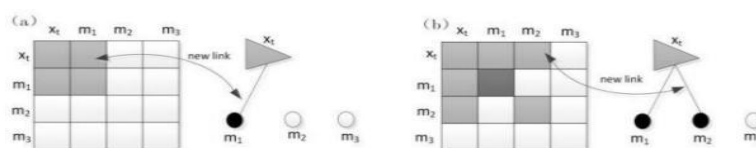


图 2.3 观测更新过程

Fig.2.3 The process of observation update

在图 2.3(a) 中，通过观测把运动载体位姿 x_t 与环境特征 m_1 的位置联系起来；这一观测的更新是在信息矩阵中非主对角线上对应于运动载体位姿 x_t 与环境特征 m_1 之间相联系的元素更新，以及信息矩阵当中主对角线上运动载体位姿 x_t 和环境特征 m_1 的更新。在图 2.3(b) 中观测到新的环境特征 m_2 ，其更新方式与图 2.3(a) 相同。

根据贝叶斯公式，观测更新的后验分布可表示为

$$p(\xi_t | z^t, u^t) \propto p(z_t | \xi_t) p(\xi_t | z^{t-1}, u^t) \quad (2.59)$$

其中 $p(\xi_t | z^{t-1}, u^t)$ 将在运动更新当中进一步分析，暂时用 $\bar{\Lambda}_t$ 和 $\bar{\eta}_t$ 来表示。

那么观测更新可以由以下数学表达形式给出

$$\Lambda_t = \bar{\Lambda}_t + C_t R_t^{-1} C_t^T \quad (2.60)$$

$$\eta_t = \bar{\eta}_t + (z_t - \hat{z}_t + C_t^T \mu_t)^T R_t^{-1} C_t^T \quad (2.61)$$

其中 $C_t = \begin{pmatrix} \frac{\partial h}{\partial x_t} & 0 \cdots 0 & \frac{\partial h}{\partial m_t} & 0 \cdots 0 \end{pmatrix}$, $\hat{z}_t = h(\bar{\eta}_t)$ 。

2、运动更新

假设系统的运动模型为

$$x_t = f(x_{t-1}, u_t) + \delta_t \quad (2.62)$$

其中 x_t 为 t 时刻运动载体的位姿, δ_t 也是一个具有高斯分布的噪声随机向量, 其均值也为 0, 协方差矩阵为 Q_t 。如图 2.4 所示,

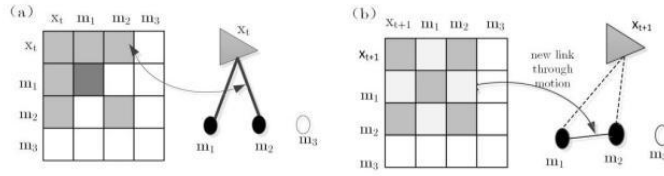


图 2.4 运动更新过程

Fig.2.4 The process of motion update

在图 2.4(a) 中运动载体位姿 x_t 只与环境特征 m_1 和 m_2 相连, 当在运动载体发生运动之后的图 2.4(b) 中, 由于运动中存在干扰, 导致运动后的位姿 x_{t+1} 与环境特征 m_1 和 m_2 的关联性减弱, 但是地图中环境特征点 m_1 和 m_2 之间的相对位置不随运动载体位姿的改变而发生变化, 所以在信息矩阵中这一信息得以保留下来, 如图 2.4(b) 所示。

根据位姿的增广, 可设系统临时增广状态为 $\xi_t' = [x_{t+1}, x_t, M]$, 有

$$\begin{aligned}
 p(\xi_t' | z^t, u^{t+1}) &= p(x_{t+1}, x_t, M | z^t, u^{t+1}) \\
 &\stackrel{Bayes}{=} p(x_{t+1} | x_t, M, z^{t+1}, u^{t+1}) p(x_t, M | z^t, u^{t+1}) \\
 &\stackrel{Markov}{=} p(x_{t+1} | x_t, u_{t+1}) p(x_t, M | z^t, u^t)
 \end{aligned} \tag{2.63}$$

其中 $p(x_t, M | z^t, u^t)$ 是系统在 t 时刻的后验分布，可用 Λ_t 和 η_t 来表示。则运动更新 $p(x_{t+1} | x_t, u_{t+1})$ 的信息矩阵和信息向量可表示为

$$\bar{\Lambda}_t = \bar{\Sigma}_t^{-1} = [(I + A_t)\Sigma_{t-1}(I + A_t)^T + S_x Q_t S_x^T]^{-1} \tag{2.64}$$

$$\begin{aligned}
 \bar{\eta}_t &= \bar{\mu}_t^T \bar{\Sigma}_t = [\mu_{t-1} + \hat{\Delta}_t]^T \bar{\Sigma}_t = [\Lambda_{t-1}^{-1} \eta_{t-1}^T + \hat{\Delta}_t]^T \bar{\Sigma}_t \\
 &= [\eta_{t-1} \Lambda_{t-1}^{-1} + \hat{\Delta}_t^T] \bar{\Sigma}_t
 \end{aligned} \tag{2.65}$$

其中 $A_t = \nabla_x f(\mu_{t-1}, u_t)$, $S_x = (I \ 0 \cdots 0)^T$ 。

3、稀疏化

为了稀疏化信息矩阵，下面分别记运动载体位姿与环境特征之间关联数上限为 θ_x ，环境主动特征之间的关联数上限为 θ_y 。当二者的关联数超过了各自的上限得时候，则执行稀疏化操作。其中，环境主动特征指在当前与运动载体相关的环境特征；环境被动特征指当前与运动载体未关联的环境特征。

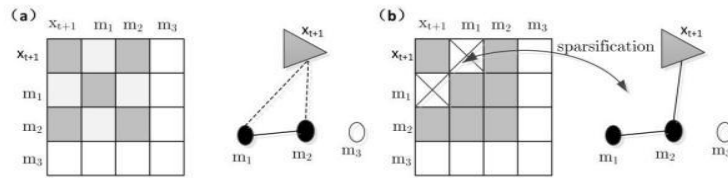


图 2.5 稀疏化过程

Fig.2.5 The process of sparsification

由图 2.5 所示，根据“D-分离”^[43]原理将一个环境主动特征 m_1 与运动载体位姿 x_{t+1} 之间的连接移除，把 m_1 由环境主动特征变为环境被动特征，使运动载体

位姿与环境特征之间的关联数维持在设置的阈值 θ_x 以下，这样使得信息矩阵不至于随着环境特征点的增加而变得很庞大，从而达到真正的稀疏化。

为了更明确的表示，把所有特征分为三个互不连接的子集，

$$m = m^+ + m^0 + m^- \quad (2.66)$$

其中 m^+ 表示保持与运动载体位姿相连的环境主动特征的集合； m^0 表示将要由环境主动特征变为环境被动特征的特征集合； m^- 表示当前所有环境被动特征的集合。所以 $m^+ \cup m^0$ 表示当前所有环境主动特征的集合。那么系统的联合后验分布可用如下数学表达式来表示

$$\begin{aligned} p(\xi_t | z^t, u^t) &= p(x_t, m^0, m^+, m^- | z^t, u^t) \\ &= p(x_t | m^0, m^+, m^-, z^t, u^t) p(m^0, m^+, m^- | z^t, u^t) \\ &= p(x_t | m^0, m^+, m^- = 0, z^t, u^t) p(m^0, m^+, m^- | z^t, u^t) \end{aligned} \quad (2.67)$$

在上面公式最后一步第一项中，如果环境主动特征 m^+ 和 m^0 是已知的，那么运动载体位姿 x_t 与环境被动特征 m^- 无关，因此可以设置 m^- 为任意值，如上面将其设为零。

由图 2.5 所示的稀疏化操作方式，可以用式子 $p(x_t | m^+, m^- = 0)$ 来替代

$p(x_t | \bar{m}^+, \bar{m}^0, \bar{m}^- = 0)$ ；由此得出

$$\begin{aligned} \tilde{p}(x_t, m | z^t, u^t) &= \frac{p(x_t, m^+ | m^- = 0, z^t, u^t)}{p(m^+ | m^- = 0, z^t, u^t)} p(m^+, m^0, m^- | z^t, u^t) \end{aligned} \quad (2.68)$$

4、状态恢复

在基于 SEIF SLAM 技术当中的最后步骤是关于均值 μ_t 的恢复。它在观测更新当中的表达式 (2.61)、运动更新中的表达式 (2.65) 以及稀疏化步骤均有涉及。

那么对于矩阵 μ_t 的精确恢复方法是通过 $\mu_t = \Lambda_t^{-1} \eta_t$ 的方式进行直接计算，由于这里存在一个超高维信息矩阵的求逆问题；然而我们并不要求出所有的均值向量中的元素，只要求出里面的运动载体位姿以及环境当中所有的主动特征点；因此在原文中采用了一种迭代方法。

2.4 基于无人机视觉 SLAM 系统模型的建立

2.4.1 基于无人机视觉 SLAM 系统的前端模型

1、视觉传感器

视觉传感器作为本系统的唯一传感器，是获取整个环境信息的直接来源，也是唯一来源；其主要由一个或者两个图形传感器组成，有时候还需要配上光投射器或者其他辅助设备。下面部分首先介绍小孔成像原理以及基于小孔成像原理的摄像机相应坐标之间的转换，然后再逐步深入到视觉里程计部分进行讲解。

2、小孔成像原理及相应的坐标转换

在计算机视觉模型当中，摄像机的成像原理是将三维场景转换成为二维的图像。然而最常用的成像模型是基于小孔成像原理；其成像原理如下图所示

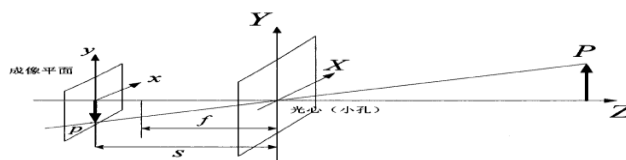


图 2.6 小孔成像原理图

Fig.2.6 The imaging principle of holes

从上图的小孔成像原理模型图当中，把小孔当成光心，且假设光线是满足沿直线传播的。在三维场景中的点 p ，经过光线投射并沿着光心在成像平面上投影成对应的倒像 p 点。从图中可以看出，小孔成像原理主要由光心，光轴，成像平面三个部分构成。

由于基于小孔成像原理的摄像机模型是属于线性成像模型，且包含四种坐标系之间的转换，他们是世界坐标系、摄像机坐标系、物理图像坐标系以及图像像素坐标系，如下图所示

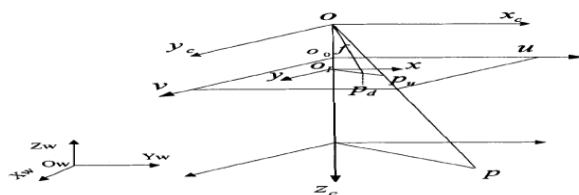


图 2.7 基于小孔成像原理的摄像机坐标之间的转换关系

Fig.2.7 The transformation between the coordinate systems

3、下面分别将介绍各坐标系之间的坐标转换

1) 从世界坐标系中的坐标到摄像机坐标系中的坐标进行转换

首先，世界坐标系的建立，其目的在于把处于自然环境中的任意位置的被拍摄物体以及摄像机的坐标通过世界坐标系表示出来，如上面图 2.7 中的左半部分 $O_w-X_wY_wZ_w$ 。

然后，摄像机坐标系的建立，是通过把摄像机的光心作为坐标原点 O_c ，把光轴作为 Z_c 轴， X_c 坐标轴和 Y_c 坐标轴如上图所示 $O_c-X_cY_cZ_c$ 。

最后，是要实现在世界坐标系中的坐标转换到摄像机坐标系中的坐标，那么可以通过一个旋转矩阵 R 和一个平移向量 t 来完成，其计算方式如下所示

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} = R \begin{pmatrix} X_w \\ Y_w \\ Z_w \end{pmatrix} + t = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \end{pmatrix} + \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} \quad (2.69)$$

以上是直角坐标系中的坐标，其对应的齐次坐标可以用如下来表示

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} = \begin{pmatrix} R & t \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \quad (2.70)$$

在上面的参数当中 t 作为三维的平移向量，且有 $t = (t_1 \ t_2 \ t_3)^T$ ，而旋转矩阵 R 是 3×3 的单位正交矩阵，矩阵中的元素满足如下关系：

$$\begin{cases} r_{11}^2 + r_{12}^2 + r_{13}^2 = 1 \\ r_{21}^2 + r_{22}^2 + r_{23}^2 = 1 \\ r_{31}^2 + r_{32}^2 + r_{33}^2 = 1 \end{cases} \quad (2.71)$$

2) 从摄像机坐标系中的坐标向物理图像坐标系中的坐标进行转换

在基于小孔成像原理的摄像机的线性成像模型当中，其物理图像坐标系是一个二维坐标系，其坐标原点 O_I 是由摄像机的光轴 Z_C 与成像平面之间的交点，也就是光点；对于 x 轴坐标与 y 轴坐标分别 X_C 轴和 Y_C 轴相平行，如图 2.7 所示；对于在三维场景中的任何一个点 p 经过透视摄影原理在摄像机成像平面上投影为 p_u ，该成像点 p_u 是自然场景中的 p 点与摄像机的光点 O_C 的连线与摄像机的成像平面的交点。从上图 2.7 中可以看出摄像机的焦距为 OO_I ，记为 f ，根据几何的三角形相似原理，可以很容易得到坐标之间的转换关系，如下所示

$$\begin{cases} x_u = fX_C / Z_C \\ y_u = fY_C / Z_C \end{cases} \quad (2.72)$$

把上面的转换关系用矩阵和齐次坐标的形式来表示，如下所示

$$Z_C \begin{pmatrix} x_u \\ y_u \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{pmatrix} \quad (2.73)$$

3) 从图像物理坐标系中的坐标向图像像素坐标系中的坐标进行转换

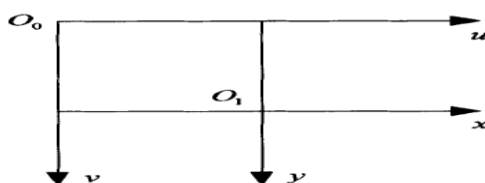


图 2.8 图像像素与物理坐标系之间的转换

Fig.2.8 The transformation between the pixel image and physical image coordinate systems

由于摄像机拍摄到的图像在计算机中是以矩阵的形式来存放的，每一个矩阵元素对应一个叫做像素的单元。然而，在图像上定义的像素坐标系也是一个二维的直角坐标系，如上图 2.8 所示。在图像像素坐标系当中我们一般把原点固定在左上角，用轴 u 来表示图像坐标系的 x 轴，其与图像物理坐标系的 x 轴平行，用轴 v 来表示图像坐标系的 y 轴，其与图像物理坐标系的 y 轴平行；在图像像素坐标系当中用所在的行数和列数来表示某一点的坐标，记为 (u, v) ；由上图 2.8 可知，图像物理坐标系的原点 O_l 位于图像像素坐标系的正中心，称其为主点，假设其在像素坐标系当中的坐标表示为 (u_0, v_0) ，那么要得到二者坐标系之间的转换关系，除了上述情况之外，还需要每个像素 u 和 v 方向分别的物理长度 K 和 L 。有了以上条件之后，其之间的转换关系如下所示

$$\begin{cases} u = \frac{x_u}{K} + u_0 \\ v = \frac{y_u}{L} + v_0 \end{cases} \quad (2.74)$$

把上面的转换关系用矩阵和齐次坐标的形式来表示，如下

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{K} & 0 & u_0 \\ 0 & \frac{1}{L} & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_u \\ y_u \\ 1 \end{pmatrix} \quad (2.75)$$

综合上可知，可以得出图像像素坐标系中的坐标 (u, v) 与世界坐标系中的坐标的转换方式，如下所示

$$\begin{aligned}
 Z_c \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} &= \begin{pmatrix} \frac{1}{K} & 0 & u_0 \\ 0 & \frac{1}{L} & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} R & t \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \\
 &= \begin{pmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} R & t \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \\
 &= M_1 M_2 P = M_{34} P
 \end{aligned} \tag{2.76}$$

上式中的 M_{34} 称为投影矩阵； M_1 称为摄像机的内参矩阵，矩阵元素主要由图像焦距 f_x 和 f_y 以及主点坐标共同来决定； M_1 称为摄像机的外参矩阵，其通过旋转矩阵 R 以及平移向量 t 来共同决定。

4、视觉里程计

我们希望通过某种装置，将该装置安装在运动载体上，从而实现测量该运动载体的轨迹问题；那么在实际生活中可以通过什么方式来实现呢？比如说要估计带轮胎的一个运动载体在环境中的运动轨迹，那么可以在其轮胎上安装一个计数码盘，当轮胎转动便可以测得其距离，从而估计运动载体的轨迹；那么像类似于采用这种装置来估计运动轨迹的方式就叫做里程计。这种装置可以包括硬件和算法。

然而，对于里程计来说，一个非常重要的特性就是它只对局部时间上的运动进行估算，一般是指在两个时刻之间的运动。当我们以某种时间间隔对里程计进行读取时，就可以估计出运动载体在该时间间隔内的运行情况；对于里程计来说不可避免的一个问题就是漂移，即由于里程计在估计的过程当中存在噪声的干扰，那么之前的估计误差情况，会随着时间的积累，延续到后面的估计上来，这就使得估计结果越来越严重。像这种漂移现象是我们不想看到的，因为它会对全局估计造成错乱，故需要借助一些其他的矫正机制来实现更精确的估计。

那么对于视觉里程计来说，它是如何在估计运动载体轨迹中起作用的呢？一个里程计如果它是依靠视觉传感器来获取环境信息，比如采用单目相机、双目相机或者其他视觉传感器的形式；那么对于这样的里程计我们就称它为视觉里程计。视觉里程计不像采用编码器设备的里程计，它可以在任何表面以任何方式的移动来对运动载体进行导航估计。

视觉里程计在机器人领域以及计算机视觉领域都有着非常广泛的应用。其工作原理是通过分析视觉传感器采集到的图像序列，并利用相应的算法来估计运动载体的位置和方向的过程。

视觉里程计采用以下几个步骤来完成运动估计：

步骤一：通过视觉传感器来获取环境图像信息；在本系统中视觉传感器采用的是单目摄像机。

步骤二：把通过摄像机拍摄的图像进行校正，可以通过透视变换和控制点变换等方法进行图像校正。

步骤三：对获取的图像序列进行特征检测，从中提取特征点，并确定其描述符，最后相邻图像帧之间进行特征匹配并构建光流场。

步骤四：检查之前构建的光流场向量是否存在可能的跟踪误差，并将误差点移除。

步骤五：通过步骤四检查过的光流场来估计摄像机的运动情况；可以采用基于卡尔曼滤波的方式进行状态的估计或者通过查询特征之间的几何和 3D 特性，把相邻两帧图像之间的重投影误差的罚函数值最小化，这可以通过随机采样的方式或者最小化的方式来完成。

步骤六：对跟踪点进行周期性的重新定位。

在视觉里程计当中主要采用的方法有基于特征点的方式和直接法。前者也叫作稀疏的方法，而后者称为稠密的方法。

1) 基于特征点匹配的方式

基于图像特征点匹配的方法是目视觉里程计当中最为主流的方式，其研究时间相对来说也比较长。在该方法当中对相邻两帧图像从中选取一些具有特别代表性的点，称之为特征点。然后利用这些特征点来估计摄像机的运动情况，包括位置和方向；与此同时，还对特征点的空间位置进行估计。而图像当中的非特征点信息则被过滤掉。

可以看出特征点方法把对于一个摄像机运动的估计转换成了其拍摄的图像序列中相邻两帧之间的两组特征点之间的估计。那么剩下的主要问题在于：

a、对于采集到的图像，该使用什么方法来获取其特征点信息？当特征点信息提取之后，又如何进行匹配？

b、最后是如何根据求出的特征点来估计摄像机的运动情况？

首先来解决第一个问题，如何提取图像特征点以及如何匹配的问题，由于在过去的特征点研究当中，出现了一些提取方法，比如：图像中的色块、角点等；然而，这些特征提取方法不稳定；随着计算机视觉技术的不断发展，后来出现了许多复杂的提取算法，且能在图像发生一定改变之后，仍然能够提取出相同的特征点，并识别出他们之间的相关性；比如常用的 SIFT、Harris 角点、FAST、SURF 以及 ORB 等等。除了把图像当中的特征点找到之外，为了体现每一个特征点之间的区别，还必须使用“描述子”对每一个特征点进行描述。



图 2.9 ORB 算法特征点提取示意图

Fig.2.9 The schematic diagram of algorithm ORB

图像特征点的描述子通常用一个向量来表示，它包含有特征点以及特征点周围区域的相关信息。那么匹配的方式就是根据两幅图像中对应的特征点以及相应的描述子信息一致，则认为它们是相同的一个点。



图 2.10 特征点匹配的示意图

Fig.2.10 The schematic diagram of feature point match

那么有了匹配好的特征点之后，怎样来估计摄像机的运动呢？接下来，就来解决第二个问题，即根据两组已经匹配好的特征点集来计算出摄像机的运动情况。

由于视觉传感器的不同，比如单目摄像机、双目摄像机以及 RGBD 等，故在这个问题上又出现了多种形式的相机运动估计。分别有

2D 到 2D 形式：通过相邻两个图像的对应特征点来估计摄像机的运动情况。

3D 到 2D 形式：假设有两组特征点，一组特征点的 3D 坐标已知，且另一组特征点的 2D 坐标已知，那么便可以求出摄像机的运动情况。

3D 到 3D 形式：假设有两组特征点的 3D 坐标都是已知的，那么利用这两组特征点的 3D 坐标来估计摄像机的运动情况。

对于上面这三种情况，我们既可以单独为每一种情况设计对应的算法，也可以设计一个统一的框架来解决上面三种情况。就单独的情况来说，2D 到 2D 对应于对极几何的方法，3D 到 2D 对应于使用 PnP 算法，而 3D 到 3D 则可以使用 ICP 的方法。针对上述三种情况的统一框架是指把所有的未知变量都作为优化变量，而其几何关系就作为优化变量之间的约束。但是由于噪声因素的影响，几何约束关系没法很好满足。故采用 *Bundle Adjustment*(BA, 捆集优化或者光束平差法) 通过最小化误差函数来解决各个变量的估计值。

对于在 2D 到 2D 的情形下，我们只知道了摄像机的运动情况，而还不知道特征点在空间中的 3D 位置信息；但是我们可以通过三角化原理，如下图所示，

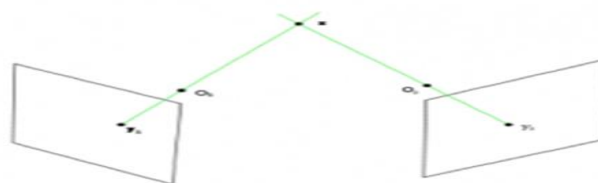


图 2.11 三角化原理示意图

Fig.2.11 The schematic diagram of triangulation theory

即只要我们在三维空间中的不同地点拍摄到同一个特征点，然后利用其在图像中的像素位置，来估计这个特征点在真实三维空间中的实际位置。

2) 基于直接法的方式

直接法为视觉里程计提供了一种全新的思路。其认为基于图像特征点的方法在特征点的提取过程当中过滤了图像当中大量的有用信息。故直接法直接跳过了特征点提取步骤，通过构建一个优化的问题，直接采用图像像素信息来估计摄像

机的运动情况。虽然直接法跳过了特征点的提取过程，但是它利用了所有的图像像素信息使得优化问题规模变得更加庞大。故基于直接法的视觉里程计，大多数情况下需要 GPU 来加速计算，才能使算法做到实时性的效果。除此之外，直接法假设摄像机的运动是一个缓慢的、平滑的过程。只有在图像序列之间足够相似时，该算法才有效。

综上对视觉里程计的分析可知，目前占主流方式的特征点方法，因其能够在噪声比较大且摄像机运动较快速时均能正常工作，故本系统基于特征点的视觉里程计方法来估计无人机的位姿情况。

2.4.2 基于无人机视觉 SLAM 系统的后端优化方程的建立

1、世界坐标系以及摄像机坐标系的建立

在本论文中采用将摄像机固定安装在无人机上，那么摄像机的运动状态便能够用以表示无人机的运动状态，其装置示意图如下所示

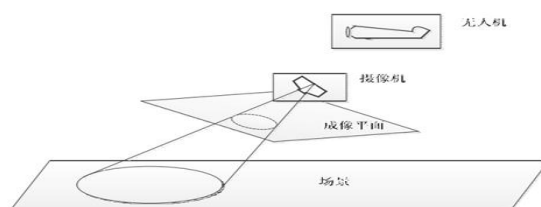


图 2.12 无人机单目视觉 SLAM 装置示意图

Fig.2.12 The schematic diagram of UAV monocular vision SLAM

在本系统中主要涉及两个坐标系，即世界坐标系和摄像机坐标系（无人机机体坐标系），将上图简化为如下图坐标系的形式，在图 2.13 中的 (O_w, X_w, Y_w, Z_w) 称为世界坐标系， (O_c, X_c, Y_c, Z_c) 称为摄像机坐标系。

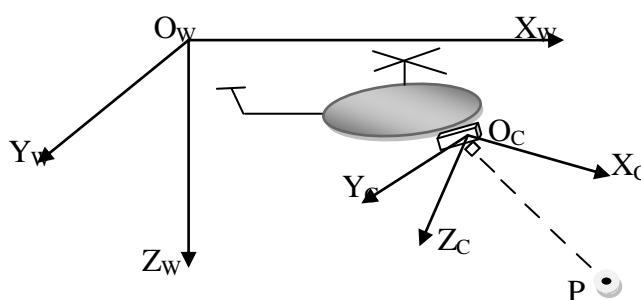


图 2.13 世界以及摄像机坐标系

Fig.2.13 The world and the camera coordinate system

2、运动模型

由文献^[44]可知，在平滑的运动模型当中，单目摄像机运动状态向量可表示为

$$x_v = \begin{bmatrix} r^W \\ q^{WC} \\ v^W \\ \omega^C \end{bmatrix} \quad (2.77)$$

其中 r^W 和 v^W 分别表示相机在世界坐标系下的位置和线速度， q^{WC} 表示在摄像机坐标系和世界坐标系下用四元素的方式来表示相机方向， ω^C 表示摄像机在其坐标系下的角速度。其运动变化过程可以用如下方程来表示

$$r_{K+1}^W = r_K^W + (v_K^W + V^W) \cdot \Delta t \quad (2.78)$$

$$q_{K+1}^{WC} = q_K^{WC} \times (\omega_K^C + \Omega^C) \cdot \Delta t \quad (2.79)$$

$$v_{K+1}^W = v_K^W + V^W \quad (2.80)$$

$$\omega_{K+1}^C = \omega_K^C + \Omega^C \quad (2.81)$$

其中， V^W 和 Ω^C 分别表示摄像机的线速度噪声和角速度噪声。

由于在无人机的飞行过程当中绝大部分情况下都不是处于平滑的运动状态，故需要对上面的运动方程进行修正。本文通过采用视觉里程计的方式来估计摄像机位姿，而非采用传统的输入控制命令的方式来预测下一时刻摄像机的位姿，通过视觉里程计我们可以获取摄像机运动过程的旋转矩阵 R 和平移向量 t 。将二者带入上式 (2.78) 和式 (2.79)，那么就可以得到摄像机相邻时刻的运动状态变化过程，如下

$$r_{K+1}^W = r_K^W - R^{WC} \cdot t_K^{K+1} \quad (2.82)$$

$$q_{K+1}^C = q_K^C \times q_K^{K+1} \quad (2.83)$$

其中， R^{WC} 表示旋转矩阵， t_K^{K+1} 和 q_K^{K+1} 分别表示由几何约束估计而来的相机平移和旋转，通过上面的变换，就可以满足无人机非平滑运动的要求。

假设系统 k 时刻的状态向量表示为

$$X_K = \begin{bmatrix} x_{v,K} \\ y_{i,K} \end{bmatrix} \quad (2.84)$$

其中 $x_{v,K}$ 表示为 k 时刻相机的位姿， $y_{i,K}$ 表示为环境中第 i 个特征点位置。假设环境的特征点是静止不动的，那么系统的运动方程可以表示为

$$\begin{cases} r_{K+1}^W = r_K^W - R^{WC} \cdot t_K^{K+1} \\ q_{K+1}^C = q_K^C \times q_K^{K+1} \\ y_{i,K+1} = y_{i,K} \end{cases} \quad (2.85)$$

3、观测模型

观测模型的目的是利用无人机携带的摄像机对环境特征点进行观测，从而获得环境特征点信息。由于相机位姿与特征点之间的关系可以用如下方程来表示

$$h^c = R^{CW}(y_i^W - r^W) \quad (2.86)$$

根据摄像机的小孔成像原理，假设观测到的特征点在生成的图像中的坐标为 (u, v) ，则系统的观测模型可以表示为如下

$$h_i = \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} u_0 - f_x \frac{h_x^C}{h_z^C} \\ v_0 - f_y \frac{h_y^C}{h_z^C} \end{pmatrix} \quad (2.87)$$

其中 u_0, v_0, f_x, f_y 表示文献^[45]中所定义的摄像机内参矩阵信息。

2.4.3 本章小结

本章首先介绍了 SLAM 技术的几个经典算法的基本理论，并分别对其进行了简要的分析和推导。紧接着是对系统模型的建立进行介绍，分别包括前端模型和后端算法优化模型；在前端模型当中主要介绍了视觉传感器的概念，小孔成像原

理及其对应的各个坐标系之间的转换关系，以及视觉里程计相关的概念和算法。在后端模型当中主要对系统的运动方程和观测方程进行建模，为后文的后端优化算法做好准备。

3 视觉 SLAM 技术图像特征点的提取方法

3.1 图像特征点提取算法的基本原理

3.1.1 SIFT 算法基本原理

SIFT 的英文名称是 Scale Invariant Feature Transform^[46]，中文译作尺度不变特征变换，该算法是由 David G.Lowe 于 1999 年提出来的。其特点主要包括如下：

- 1) 特征点具有唯一性；
- 2) 特征点对旋转、尺度的放大和缩小以及亮度变化都能保持不变；
- 3) 特征点对于在图像中引入的噪声而造成了仿射失真或者拍摄的角度发生了变化等都能够保持一定的鲁棒性。

下面把 SIFT 特征点提取算法分为五个步骤：尺度空间的生成、尺度空间的极值检测、特征点定位、特征方向赋值以及特征点描述。

1、尺度空间的生成

由于计算机不同于人类，其在未知环境当中，通过摄像机等视觉传感器并不能提供三维场景中具体物体的尺度，但是我们可以把该场景下物体的各种不同尺度提供给计算机，让计算机能够了解所有提供的尺度，且可以找出在各个不同尺度下都存在的特征点。在早期研究的尺度空间的生成采用的是图像金字塔，由于其难以保持图像的局部特征，从而也就难以使环境特征的尺度保持不变。

另外一种尺度空间的表示形式可以采用高斯尺度空间，其原理是在保持分辨率不改变的情况下，根据高斯正态分布的标准差因子的不同来模糊图像，从而形成高斯尺度空间，其数学表达形式如下

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (3.1)$$

其中 $I(x,y)$ 表示对应的图像， $G(x,y,\sigma)$ 称为高斯核函数，其表达形式如下

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3.2)$$

上式的标准差因子也叫做尺度空间因子，其体现了图像的模糊程度；标准差因子值越大，则图像的模糊程度就越大，对应的尺度空间也相应变大。

经实践发现，在尺度空间构成中，使用高斯拉普拉斯（LoG）算子，即 $\Delta^2 G$ ，来对特征点进行检测还是相对比较好的方法，其中 $\Delta^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ 。

但是在其检测特征点的过程当中，其运算量相对来说还是比较大，那么我们在实际使用的过程当中，通常采用差分高斯（DoG）来近似替代拉普拉斯算子。其定义表达式如下

$$\begin{aligned} D(x, y, \sigma) &= [G(x, y, k\sigma) - G(x, y, \sigma)] * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \quad (3.3)$$

下图展现了高斯尺度空间与差分高斯尺度空间之间的关系

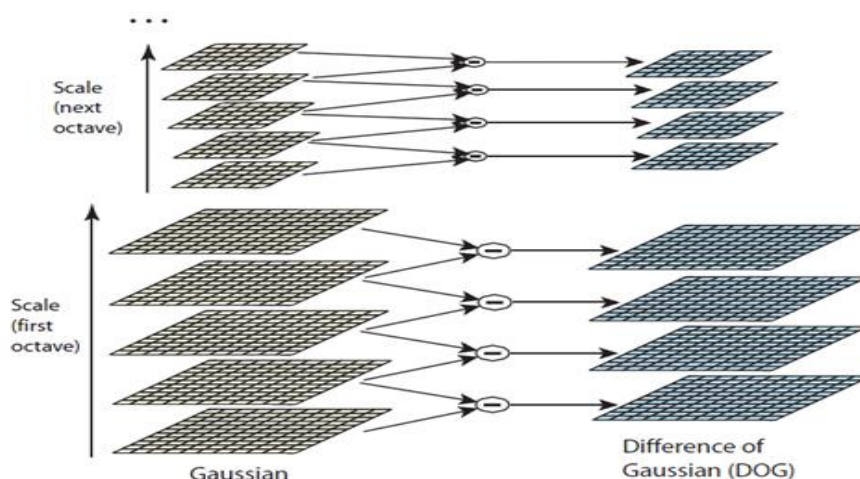


图 3.1 高斯尺度空间与 差分高斯尺度空间

Fig.3.1 Gaussian scale space and differential gaussian scale space

2、基于差分高斯（DoG）算子尺度空间的 极值检测

为了找到基于差分高斯算子尺度空间的极值点，每一个被采样的像素点都要和该像素点相邻的所有像素点（包括同一尺度空间和上下相邻的尺度空间）进行比较，看它是不是所有像素点当中最大或者最小的，如果是，则把该像素点作为极值点。如下图所示

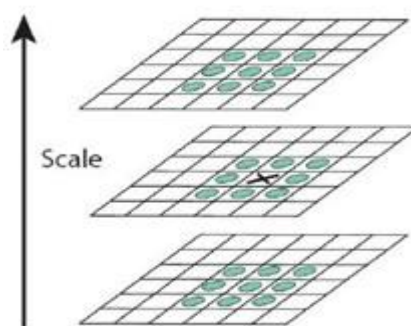


图 3.2 候选特征点的选择

Fig.3.2 The choice of candidate feature points

由于在进行极值点的选取过程中，每组的第一层和最后一层都无法进行比较，为了使选择的特征极值点是连续的，故我们采用在高斯尺度空间中每一组的第一层继续再加 3 幅经过高斯模糊的图像，那么便可以使特征点的选取具有连续性，如下图所示

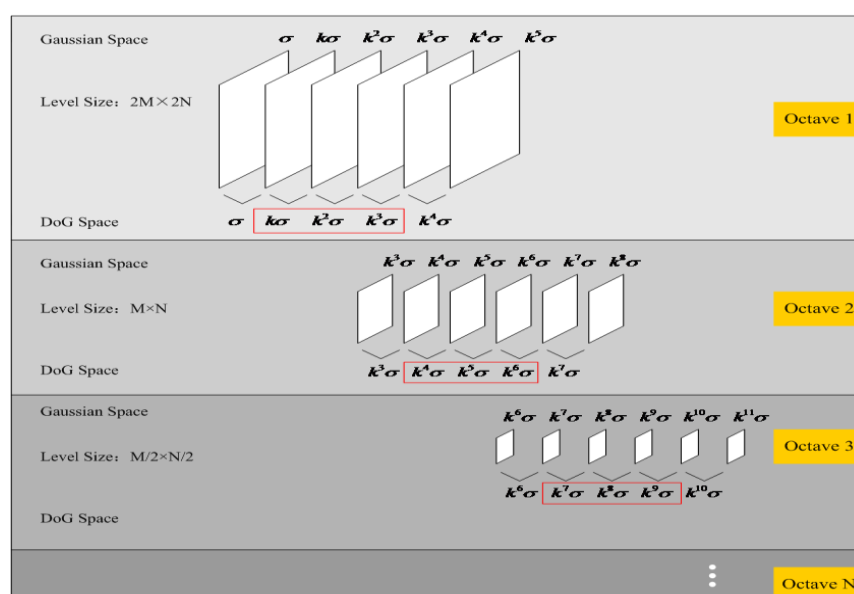


图 3.3 极值点选取的连续性

Fig.3.3 The continuity of extreme value point selection

3、特征点定位

由于在上面的差分高斯空间选取的特征点都是离散地分布的，且之前找到的极值点也不一定就是真正要找的特征点，那么就要通过一定方法来丢弃掉之前选择的不是特征点的极值点。一般都是采用对差分高斯函数进行拟合，找出其中曲

率很不对称的点，主要包括两类：对比度比较低的极值点和不稳定的边缘相应极值点。

对于前者，假设极值点的差分高斯函数为 $D(x)$ ，对它进行二阶泰勒级数展开

$$D(x)=D+\frac{\partial D^T}{\partial_x}\Delta x+\frac{1}{2}\Delta x^T\frac{\partial^2 D}{\partial x^2}\Delta x \quad (3.4)$$

由于 x 是极值点，故对上式求导后，令其为零，有

$$\Delta x=-\frac{\partial^2 D^{-1}}{\partial x^2}\frac{\partial D(x)}{\partial x} \quad (3.5)$$

将 (3.5) 带入式 (3.4)，则有

$$D(\hat{x})=D+\frac{1}{2}\frac{\partial D^T}{\partial x}\hat{x} \quad (3.6)$$

然后将极值点的对比度设置一个阈值 T ，如果有 $|D(\hat{x})|>T$ ，那么就把该极值点保留下来，否则就丢弃掉。

对于后者，由于上述选择的极值点的差分高斯函数 $D(x)$ 的主曲率正比于 $2*2$ 的 *Hessian* 矩阵 H 的特征值。 H 表示如下

$$H=\begin{pmatrix} D_{xx} & D_{yx} \\ D_{xy} & D_{yy} \end{pmatrix} \quad (3.7)$$

其中的 D_{xx} ， D_{xy} 和 D_{yy} 是极值点邻域相应位置的差分所得。

为了利用主曲率来检测，可以使用矩阵 H 的特征值的比值来判断，设 $a=\lambda_{\max}$ ， $\beta=\lambda_{\min}$ 分别为最大和最小特征值，则有

$$Tr(H) = D_{xx} + D_{yy} = \alpha + \beta \quad (3.8)$$

$$Det(H) = D_{xx} + D_{yy} - D_{xy}^2 = \alpha \cdot \beta \quad (3.9)$$

其中 $Tr(H)$ 和 $Det(H)$ 分别称为矩阵 H 的迹和行列式。

令 $\gamma = \frac{\alpha}{\beta}$ ，则有

$$\frac{Tr(H)^2}{Det(H)} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(\gamma\beta + \beta)^2}{\gamma\beta^2} = \frac{(\gamma + 1)^2}{\gamma} \quad (3.10)$$

设置某个阈值 T_γ ，判断主曲率是否在该阈值范围下，通过如下式子来判断

$$\frac{Tr(H)^2}{Det(H)} > \frac{(T_\gamma + 1)^2}{T_\gamma} \quad (3.11)$$

4、特征方向赋值

通过以上的方式便得到了尺度不变的特征点，为了使特征点旋转也具有不变性，那么就必须对特征点相关的方向进行赋值。主要是利用该特征点附近像素点的梯度分布情况从而确定该特征点的方向，同时还利用该图像的梯度直方图来获取该特征点相对稳定的方向。由于得到了特征点，那么对应的空间尺度 σ 便也是已知的，利用式子 $L(x, y) = G(x, y, \sigma) * I(x, y)$ ，其对应的尺度图像也可以求出来。

下面来求解图像中以特征点为圆心，半径为 $3 \times 1.5\sigma$ 的幅角以及幅值；其每一个特征点梯度的模 $m(x, y)$ 和角度 $\theta(x, y)$ ，由下式可得

$$m(x, y) = \sqrt{[L(x+1, y) - L(x-1, y)]^2 + [L(x, y+1) - L(x, y-1)]^2} \quad (3.12)$$

$$\theta(x, y) = \arctan \frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \quad (3.13)$$

那么就可以利用上式 (3.12) 和 (3.13) 计算出特征点附近的像素对应的幅角和幅值，并通过图像直方图表示出来。如下图所示，

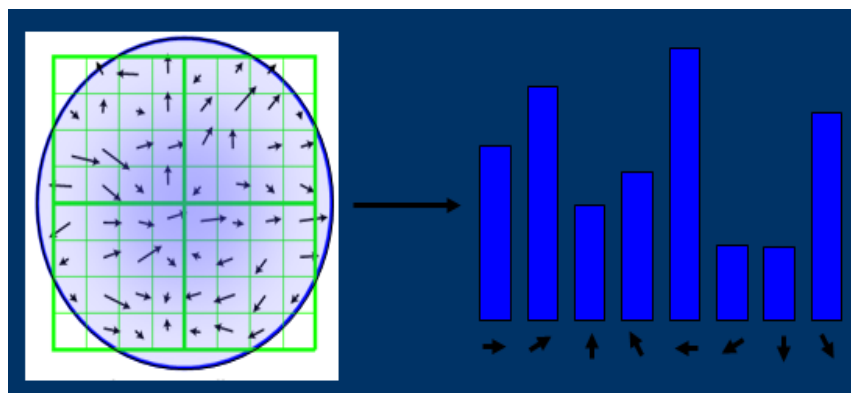


图 3.4 基于特征点像素梯度幅角和幅值的直方图

Fig.3.4 The histogram of feature points pixel gradient Angle and amplitude

图中横坐标轴表示幅角，纵坐标轴表示幅值。幅值最高的就是极值点的主方向，可以把其他大于最高值的 80% 作为辅助方向。通过以上结论便得到了特征点的三个重要信息 (x, y, σ, θ) ，也就是特征点对应的位置，所在的尺度和方向。

5、特征点的描述

在以上步骤的基础之上，我们已经得到了特征点的位置信息，尺度信息以及方向信息，那么接下来的任务就是要通过特征点的描述子来表达这些信息。采用的方法是通过一组向量来表示，该向量当中不仅仅只包括特征点本身，而且还包括其周围的有效像素点。描述子的生成过程分为三个步骤：

- 为了确保特征点旋转的不变性，需要旋转坐标轴至特征的主方向位置；
- 生成一个有 128 维度的特征描述子向量；
- 为了去除噪声的影响，需要对特征向量进行归一化处理。

在步骤一当中，当旋转了坐标轴之后，在特征点领域内的像素对应的新坐标可表示为

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad (3.14)$$

在执行步骤一之后，选取以特征点作为圆心的 8×8 的窗口。如下图所示，在图中的左部分中每一个小格子代表一个像素点，对应有一个箭头，其方向表示对

应梯度的方向，其长度表示对应梯度的长度，然后对该 8×8 的窗口进行加权运算得到右图所示的描述子，在该描述子里面的每一个小格子里面8个方向的像素梯度叠加而成，称其为一个种子点，那么每个特征点由4个种子点构成。

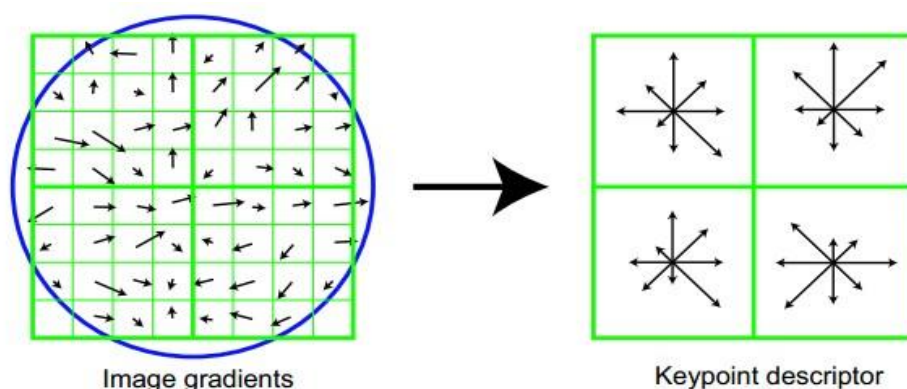


图 3.5 特征点描述子的表示

Fig.3.5 The representation of a feature point descriptor

3.1.2 SURF 算法基本原理

SURF 的英文全称是 Speeded Up Robust Features^[47]，其中文可翻译为加速稳健特征；该算法于 2006 年，由 Herbert Bay 教授首次提出，其可以应用于计算机视觉领域当中的物体识别或者 3D 重构等任务。它可以算得上是尺度不变特征变换算法的一个升级版本，主要体现在算法计算速度以及在检测多张图片时候具有更好的鲁棒性。算法计算速度提高的主要原因在于它利用 harr 特征的方式以及对图像进行积分的方式。

在实现算法之前，先来讲解一下图像积分的概念，该概念是由 Viola 和 Jones 在 2001 年首次提出来的。其大体思路是指在形成的积分图像之后中的任一点 (i, j) 的值来自于原始图像从左上角部分到该点构成的整个矩形区域中灰度值的累加和，如下图所示

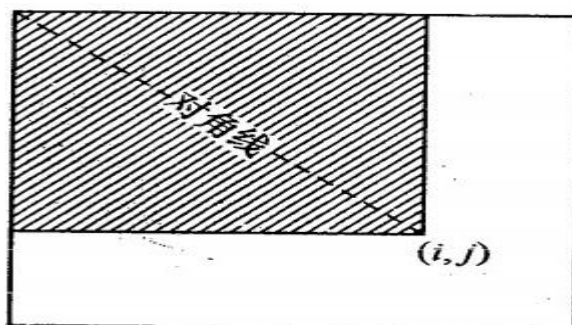


图 3.6 图像积分示意图

Fig.3.6 The schematic diagram of image integral

其数学表达式如下

$$ii(i, j) = \sum_{i' \leq i, j' \leq j} p(i', j') \quad (3.15)$$

通过上面积分图像的概念，如果我们想对一个原始图像的某一个区域进行积分操作，那么在需要对该原始图像的积分图像里面求取该区域四个顶点的值即可，其数学表达如下所示

$$\sum_w = ii(i_4, j_4) - ii(i_2, j_2) - ii(i_3, j_3) + ii(i_1, j_1) \quad (3.16)$$

下面是算法原理的实现，由五个部分组成。

1、Hessian 黑森矩阵构建

为了提高算法的计算性能，算法首先利用 *Hessian* 矩阵来选择候选特征点，设有函数 $f(x, y)$ ，那么 *Hessian* 矩阵 H 由该函数的偏导构成，下面是图像中像素点的 *Hessian* 矩阵定义式，如下：

$$H(f(x, y)) = \begin{pmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{pmatrix} \quad (3.17)$$

下面使用 *Hessian* 矩阵的行列式作为判别式，其表达式如下

$$\det(H) = \frac{\partial^2 f}{\partial x^2} \frac{\partial^2 f}{\partial y^2} - \left(\frac{\partial^2 f}{\partial x \partial y} \right)^2 \quad (3.18)$$

其判别方式是通过 *Hessian* 矩阵的特征值的取值是否正负来选择是不是候选的极值点，通常在算法当中我们使用图像函数 $I(x, y)$ 来取代上面的函数 $f(x, y)$ 。在滤波器方面选择二阶的标准高斯函数，然后与特定核之间的卷积来计算上面的偏导数，从而得出 *Hessian* 矩阵的元素 L_{xx} , L_{xy} , L_{yy} ，即有如下的公式：

$$H(x, \sigma) = \begin{pmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{pmatrix} \quad (3.19)$$

下面再对 *Hessian* 矩阵进行高斯滤波，把尺度因子 σ 去掉，得到如下式子

$$L(x, t) = G(t) \cdot I(x, t) \quad (3.20)$$

其中 $G(t)$ 为高斯核，表达式如下

$$G(t) = \frac{\partial^2 g(t)}{\partial x^2} \quad (3.21)$$

后来 Herbert Bay 又提出了使用近似值来替代 $L(x, t)$ 。得到如下的判别式

$$\det(H_{approx}) = D_{xx} D_{yy} - (0.9 D_{xy})^2 \quad (3.22)$$

2、尺度空间生成

算法中尺度空间的表示是基于图像不一样的解析度来完成的。上文所述的只是得到了一张相近的 *Hessian* 图，与 SIFT 算法相似，他们的尺度空间都是采用金字塔的形式，如下图所示

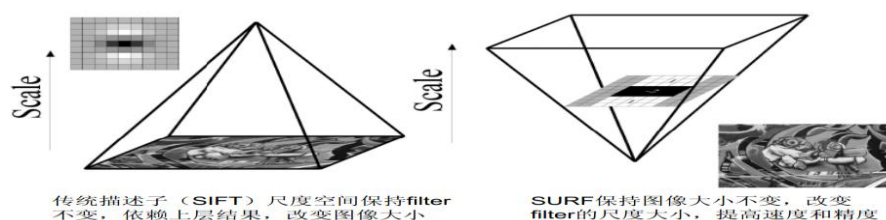


图 3.7 基于 SIFT 与 SURF 尺度空间的对比

Fig.3.7 The contrast of scale space based on SIFT and SURF

然而不同的是 SURF 中图像的尺寸大小是保持一致不变的，只是在金字塔的每一层中使用高斯模糊来改变尺寸大小。并且 SURT 算法并不需要进行二次的抽样，同时还可以实现尺度空间的多个层图像并行处理，这就在计算性能上大大提高。

3、利用非极大值抑制来对特征点进行初步和精确定位

经过上面的 *Hessian* 矩阵计算过的每个图像像素点与它上下层包括自身本层的 3 层进行比较，如下图所示，如果得出该值是最大的值或者最小的值，那么就可以用来作为候选的特征点。

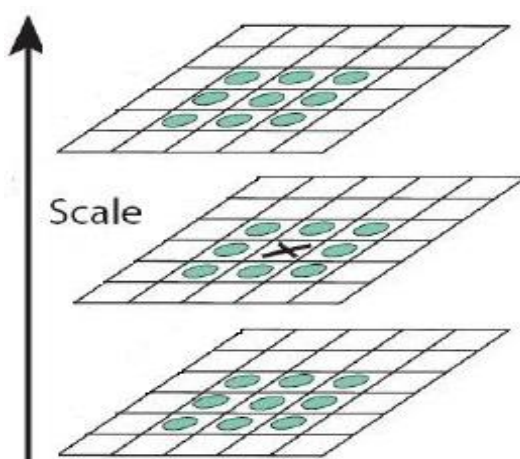


图 3.8 候选特征点的选择

Fig.3.8 The choice of candidate feature points

然后，我们可以设置一个阈值，如果满足阈值要求，则把该候选特征点留下，否则就丢弃；那么留下来的肯定就是相对来说更强的特征点。

4、选取特征点主方向确定

为了确定特征点的主方向，与 *SIFT* 算法不相同的是，在 *SURF* 算法中选择要统计的是由上面步骤计算出的特征点附近的 *Harr* 特征。其计算方法是，以特征点作为圆心，半径为 6 倍的尺度值，以该圆作为搜索区域，且以 60 度作为一次统

计，找出里面在横轴和纵轴方向的 *Harr* 小波响应和，并给每一个响应和附上相应的高斯权值，当遍历完整个区域后，找出其中最长的向量作为主方向，其过程图如下所示：

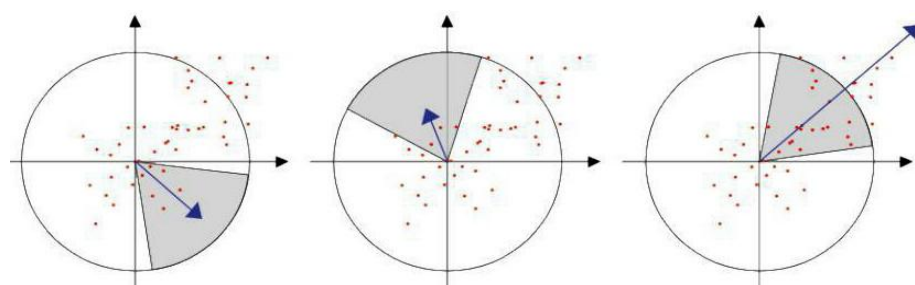


图 3.9 特征点主方向的选取过程

Fig.3.9 The selection process of feature point main direction

5、构造 SURF 特征点描述算子

在 SURF 算法当中，首选以上面步骤选择出来的特征点的方向作为主方向，选取一个正方形的区域，其边长为 20 倍特征点的尺度因子，然后把该区域划分成为 16 个小区域，在每个小区域里面对 25 个像素的横纵方向上的 *Harr* 小波特征进行统计求和，由此而得出 *SURF* 的描述子。如下图所示

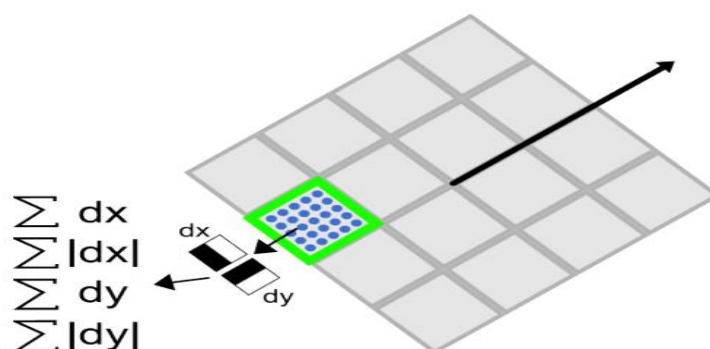


图 3.10 构造 SURF 特征点描述子

Fig.3.10 The construction of SURF feature point descriptor

3.1.3 ORB 算法基本原理

ORB 的英文全称是 Oriented FAST and Rotated BRIEF^[48]，它是由 Ethan Rublee, Vincent Rabaud 等人于 2011 年首次提出，算法主要特点就是速度快；而且完全可以用于替代 SIFT 或者 SURF 算法。它主要由两部分组成，分别是通过算法 FAST

来进行特征点的提取部分以及用算法 BRIEF 对提取的特征点进行描述部分。

ORB 分别都在两个子部分都进行了一定的改进和优化。下面讲解 ORB 的原理实现部分。

1、特征点提取

ORB 的特征点提取方法是通过对算法 FAST 进行改进而来的，在这里把它称作 oFAST。这里的“o”是指 Orientation，即方向；那么就是在由 FAST 算法提取出特征点后为其选择了一个方向，目的是保证提取出来的特征点的旋转性质的不变。ORB 的特征提取算法步骤如下：

第一步：本步骤是对特征点的一个粗略的提取，提取的结果包含大量的特征点，可是其中包含了相当一部分效果不好的特征点。其示意图，如下图所示，从图像当中任意选择一点 p ，以 p 为中心，半径为 3 个像素的画一个圆形区域；判定该点 p 是不是一个特征点的方法是，如果在圆周经过的像素点上有连续的 n 个像素点的灰度值比圆心点 p 都大或者都小，则认为圆心点 p 是一个特征点。一般我们都把 n 取为 12。为了提高特征点被提取的速度，把非特征点快速地排除掉，我们会首先选择检测位置在 1、9、5、13 上的值，如果圆心点 p 是一个特征点的话，那么在上述的四个位置中的像素值至少有一个的值是比圆心点 p 的值都要大或者都要小；否则，就不是一个特征点。

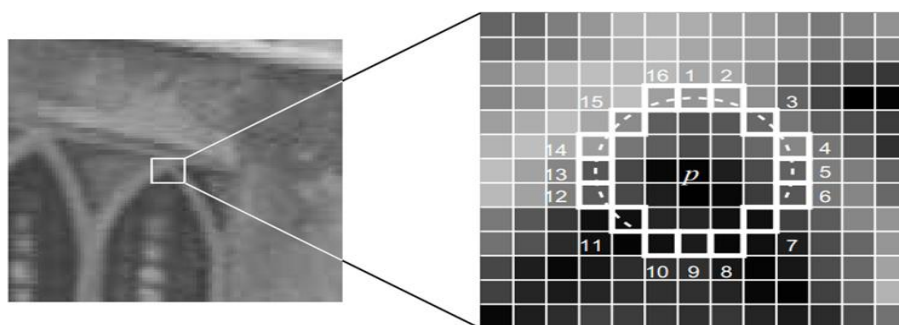


图 3.11 FAST 特征点判断示意图

Fig.3.11 The Judge schematic of FAST feature point

第二步：这一个的目的是从上一步的粗略提取中精细地提取出质量比较好的特征点。使用的方法是利用机器学习当中的决策树，该决策树是通过算法 ID3 训练出来的。那么通过把上述的圆区域的边缘上的 16 个像素点输入到由 ID3 训练出来的决策树里面来选择更优的特征点。

第三步：由于某些局部区域会生成多个特征点，导致该局部的特征点非常密集，那么我们应该采取一定方法来删选当中相对较好的特征点保留下来，使用的是非极大值抑制算法。该算法首先为该局部区域的每一个特征点算出其对应的响应值，然后把响应值相对比较大的点保留下来，丢弃那些响应值小的特征点。

第四步：这一步的目的是使提取到的特征点尺度保持不变性。与大多数特征点提取算法一样，通过建立一个图像金字塔来实现其尺度的不变性。

第五步：在这一步的目的是实现特征点旋转保持不变性。在 *ORB* 算法当中他们采用的是矩的方法来确定 *FAST* 特征点方向。该方法是通过计算以特征点为圆心，半径为 r 的圆区域的质心，那么特征点方向是由该特征点的位置到质心的位置所构成的一个向量来表示。下面是矩的定义

$$m_{pq} = \sum_{x,y \in r} x^p y^q I(x, y) \quad (3.23)$$

在上式当中 $I(x, y)$ 为图像的灰度表达形式。

那么对应矩的质心表示为如下：

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \quad (3.24)$$

那么对应特征点的方向可以表示为如下：

$$\begin{aligned} \theta &= \arctan\left(\frac{m_{01}}{m_{00}} / \frac{m_{10}}{m_{00}}\right) \\ &= \arctan(m_{01}/m_{10}) \end{aligned} \quad (3.25)$$

2、ORB 中特征点的描述

在 *ORB* 算法中，其采用的特征点描述方法是借鉴于 *BRIEF* 算法，并在此基础上把旋转因素考虑进去加以改进的。首先讲解基本的 *BRIEF* 算法，然后再谈它的改进方法

1) *BRIEF* 算法描述

在 *BRIEF* 算法中它把特征的描述符表示称为一个二进制形式的字符串。该方法是通过在特征点的一个区域范围内，从中选择 n 个像素点对 p_i 和 q_i ，其中 i 的取值是 $1, \dots, n$ 。然后对每一个像素点对其进行灰度值的大小比较，即如果有 $I(p_i) > I(q_i)$ ，那么则结果就是二进制中的 1，否则就是二进制中的 0。通过对所有的 n 对像素点进行比较之后，形成一个长度为 n 的二进制字符串。另外，在 *ORB* 算法中，为了提高描述符抗噪能力，利用高斯平滑来对图像进行处理，且在平滑处理之后，又运用了其他的操作方法，使算法的抗噪能力更强。在下图中采用了五种方法来在特征点区域内选择像素点对，如下图所示

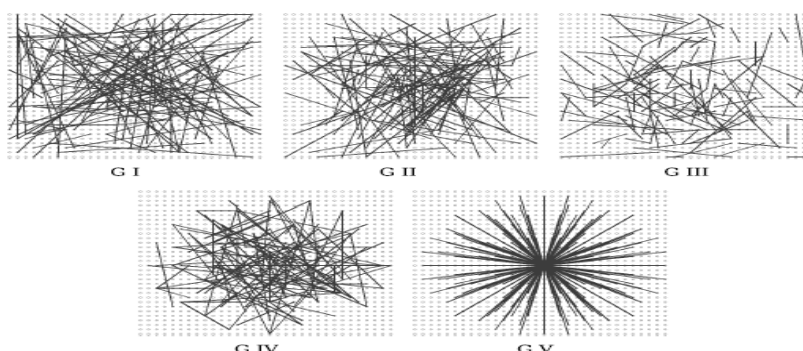


图 3.12 特征点区域内像素点对的选取方式

Fig.3.12 The way of the pixels selection in feature point region

在 GI 图中使用的方法是，在图像区域块里面进行平均采样；在 GII 图中使 p 和 q 都满足高斯分布 $(0, S^2/25)$ 的形式；在 GIII 图中是采用 p 为高斯分布 $(0, S^2/25)$ 的形式，而 q 满足高斯分布 $(0, S^2/100)$ 的形式；在 GIV 图中采用的是在空间中通过量化后的极坐标下的离散位置进行随机采样；在 GV 图中采用的是对像素点 q 的周围进行平均采样，而把 p 固定为点 $(0,0)$ 。经过实验证明，采用 GII 图的方法可以获得更好的匹配效果。但是由于当图像的旋转角度变得大于 30° 以后，它的匹配率几乎为了 0。故提出了如下的改进算法。

2) 对 *BRIEF* 算法的改进，即 *rbRIEF* 算法

a、改进算法 *BRIEF* 的旋转不变的特性

设基本的 *BRIEF* 算法在特征点的某个区域内选取了 n 对像素点集，如下所示

$$D = \begin{pmatrix} x_1, x_2, \dots, x_{2n} \\ y_1, y_2, \dots, y_{2n} \end{pmatrix} \quad (3.26)$$

然后，把上述点集经过一个旋转的角度 θ 之后，从而得到了新的像素点集，如下所示

$$D_\theta = R_\theta D \quad (3.27)$$

最后，就是要在新形成的像素点集上，按照上面描述的方式形成一个新的二进制字符串作为描述符。

b、算法 *rBRIEF* 改善描述子之间的相关性

由于使用 *steeredBRIEF* 算法计算出的描述子虽然特征点旋转能够保持不变的特性，但是它在特征描述子的相关特性上就不如算法 *BRIEF* 了。这个性质对于不同图像之间的相同特征点能否正确匹配至关重要。在作者 Ethan Rublee 等人的论文当中，他们用上述的三种方法来对 100k 个特征点分别计算他们的二进制特征描述符，并对其进行统计如下图所示

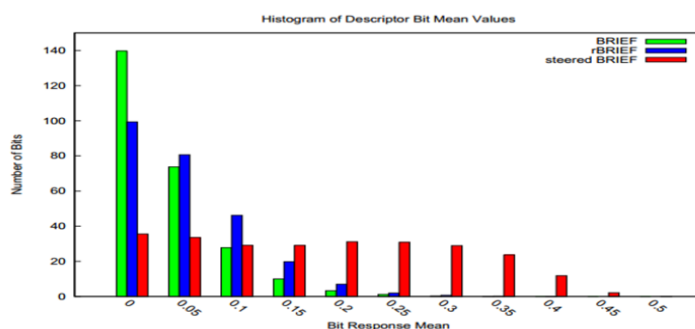


图 3.13 特征描述子的均值分布

Fig.3.13 The average distribution of feature descriptor

在上图中横坐标轴表示离均值 0.5 的距离，纵坐标轴是对在均值为 0.5 下的特征点数进行的统计。首先对上图中算法 *BRIEF* 和 *steeredBRIEF* 进行分析可知，前者在 0 处的特征数量比较多，故其计算的特征描述符的均值在 0.5 左右，其方差比较大，区分性相对就比较强。然后从图中可以看出算法 *steeredBRIEF* 失去了这个特性。

下面主要来分析一下算法 *rBRIEF*，该算法并没有采用原著当中的方法，而是采用的是统计学习来对 n 个像素点集进行选择。算法首先建立 $300k$ 个点集以便测试。对于每一个点，选择其领域为 31×31 。以往不同的是，用某个点的 5×5 区域的平均灰度值来替代该点的值，从而这样更具有抗噪能力。

由以上可知，在区域为 31×31 内一共有 $(31-5+1) \times (31-5+1) = 7 \times 7$ 个这样的 5×5 子窗口，基于这些子窗口，我们选取点对的方式总共有 $M = 265356$ 种，但是只能从中选取 256 种，其准则是这些选取的点对之间的关联性是最小的。选取方法的步骤如下

第一步：通过在候选点集中每个特征点的区域内按照 M 种方式进行选择特征点，同时比较选择的点对之间的大小，从而形成一个 $300K \times M$ 的矩阵 Q 。 Q 的列是由一串二进制数构成。

第二步：通过把 Q 矩阵的列进行平均，然后分析平均后的值与 0.5 的距离来重新对 Q 进行排序，从而形成 T 矩阵。

第三步：把矩阵 T 的首列向量加入到矩阵 R 中。

第四步：计算矩阵 T 接下来的列向量与 R 中的全部列向量的相关性，然后判断该相关性与预先设定的值，如果小于，则在矩阵 R 中加入对应于 T 中的列向量。

第五步：照着第四步的方法一直进行，直到矩阵 R 中有 256 个列向量为止。通过这种方法通过以上步骤就可以完成这 *rBRIEF* 算法。

3.2 通过实验对图像特征点提取算法进行比较

以下图像特征提取算法实验的素材来自于通过无人机通过摄像机获取到的环境图像序列中的一部分。通过以下三个实验来验证三种算法在旋转不变性、尺度不变性以及时间效率方面的性能关系。

3.2.1 基于 SIFT 算法的图像特征提取和匹配

1、利用 SIFT 算法来验证尺度不变的特性

首先，编写 *SIFT* 算法分别对图像 1 和图像 2 进行特征点的提取，对图像 1 提取到的特征点个数是 992 个，而对图像 2 提取到的特征点个数是 984 个，每帧图像平均消耗时间是 2.73s；然后使用近似最近邻（FLANN）特征点匹配算法来进行两幅图像之间的一个匹配操作，如下图所示



图 3.14 SIFT 算法检测尺度不变性优化前匹配效果

Fig.3.14 The matching effect of SIFT's scale invariance before optimization

得到匹配特征点总个数为 962 个，可以看到上面匹配图当中有许多的误匹配，因此要采取一定方式把这些误匹配过滤掉。因为在最近邻匹配算法当中它会返回一个 `DMatch` 这样的结构体，该结构体里面包含有三个元素：第一幅图像的特征描述子索引（`queryIdx`）、第二幅图像的特征描述子索引（`trainIdx`）以及对应特征之间的匹配距离（`distance`）。我们把特征点之间匹配距离大于 4 倍设置的最小距离的匹配特征点给去掉，经过距离优化匹配处理之后的图像如下所示



图 3.15 SIFT 算法检测尺度不变性优化后的匹配效果

Fig.3.15 The matching effect of SIFT's scale invariance after optimization

上面经过优化后的匹配图像得到了 72 个特征匹配点。可以看出算法在不是很平滑的图像下也能使尺度不变的特保持较好的稳定性。

2、利用 SIFT 算法来验证旋转不变的特性

同样是利用编写好的 SIFT 算法对两张图片进行特征点提取以及匹配操作，只是为了验证图像在发生旋转之后，SIFT 算法还能不能保证图像特征之间的匹配具有较好特性。同样也是利用最近邻匹配算法进行的匹配，如下图所示

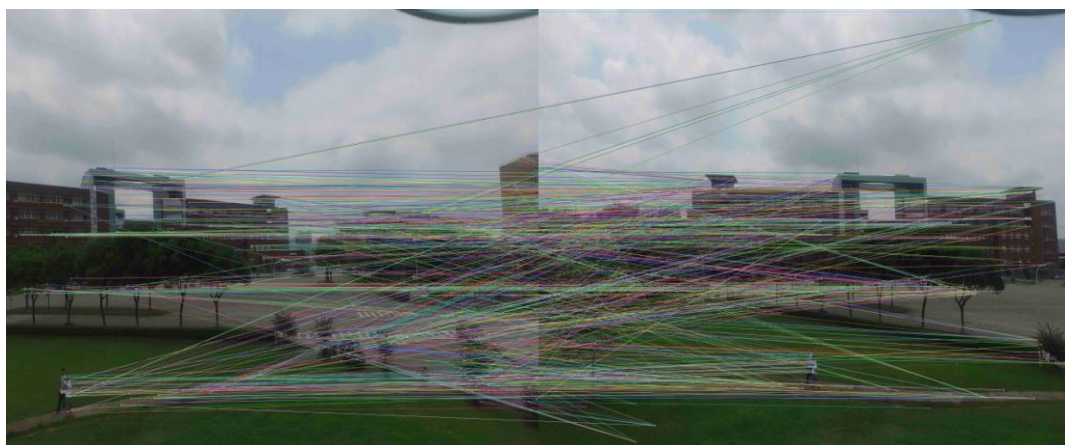


图 3.16 SIFT 算法检测旋转不变性优化前匹配效果

Fig.3.16 The matching effect of SIFT's rotation invariance before optimization

在两幅图当中都提取到了 746 个特征点，总共的特征点匹配数也是 746 个，但是由于其匹配距离设置得相对较大，使得需要特征点出现误匹配，那么跟上面处理方式一样，使特征点之间的匹配距离变小，相应的匹配效果就会更好，如下图所示



图 3.17 SIFT 算法检测旋转不变性优化后匹配效果

Fig.3.17 The matching effect of SIFT's rotation invariance after optimization

在经过距离优化处理之后，可以得到 58 个较优的特征匹配点。可以看出算法在图像发生旋转之后也能保持很好的不变性。

3.2.2 基于 SURF 算法的图像特征提取和匹配

1、利用 SURF 算法来验证尺度不变的特性

首先，编写 SURF 算法来对之前 SIFT 处理过的相同图像进行特征点提取以及匹配，分别对两张图像总共提取到的特征点总数都是 992 个，每帧图像平均消耗时间是 3.25s；其对应的匹配数也是 992 个，图 3.18 是在算法 SURF 中使用 FLANN 算法进行的匹配效果，



图 3.18 SURF 算法检测尺度不变性优化前匹配效果

Fig.3.18 The matching effect of SURF's scale invariance before optimization

同样的由于部分匹配距离设置得过大以及源图像和目标图像之间并不是平滑过渡，那么会产生很多的误匹配，在这里使用缩短特征点匹配距离的方法来优化匹配效果，如下图所示



图 3.19 SURF 算法检测尺度不变性优化后匹配效果

Fig.3.19 The matching effect of SURF's scale invariance after optimization

从优化后的匹配图当中得到 42 个较优的图像匹配点。从匹配图当中可以看出算法还有具有较好的尺度不变特性。

2、利用 SURF 算法来验证旋转不变的特性

同样也是在与 SIFT 算法相同的场景图像下进行旋转不变性的验证，也是利用编写的 SURF 算法，其处理方式与上面一样，如下图是直接经过最近邻匹配算后的图像匹配效果图

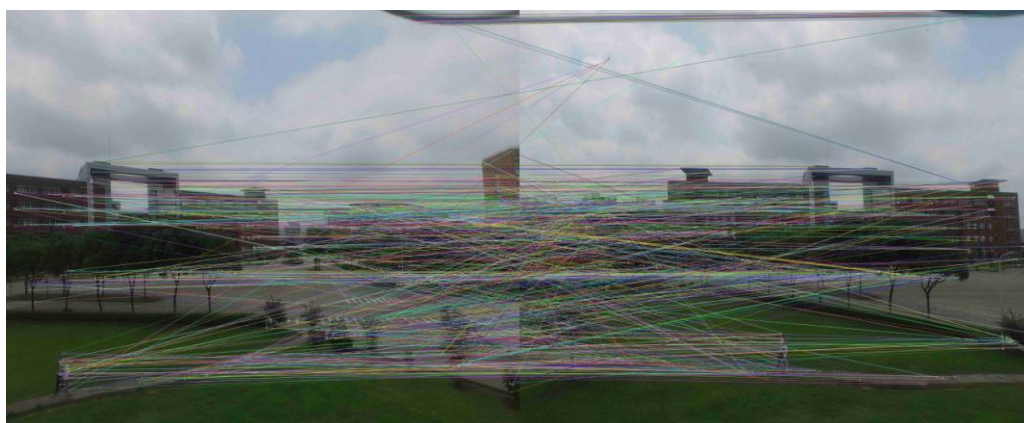


图 3.20 SURF 算法检测旋转不变性优化前匹配效果

Fig.3.20 The matching effect of SURF's rotation invariance before optimization

上图左中提取到的特征点数为 806 个，上图右中提取到的特征点数为 774 个，其对应的特征匹配点总共有 774 个。然后，经过最小距离优化之后得到的匹配效果图如下所示



图 3.21 SURF 算法检测旋转不变性优化后匹配效果

Fig.3.21 The matching effect of SURF's rotation invariance after optimization

上图当中总共获得 78 个较优的匹配点；从图 3.23 中可以看出其具有较好的旋转不变性。

3.2.3 基于 ORB 算法的图像特征提取与匹配

1、利用 ORB 算法来验证尺度不变的特性

首先，也是编写 *ORB* 算法来对图像进行特征点提取与匹配操作在下图场景当中，经过算法处理之后，提取到的特征点都是 500 个，每帧图像平均消耗时间是 0.13s；其匹配的总数也是 500 个。



图 3.22 ORB 算法检测尺度不变性优化前匹配效果

Fig.3.22 The matching effect of ORB's scale invariance before optimization

经过把最近邻匹配算法的匹配距离减小之后，得到了下面的特征匹配图，如下所示



图 3.23 ORB 算法检测尺度不变性优化后匹配效果

Fig.3.23 The matching effect of ORB's scale invariance after optimization

但是经过优化之后的匹配图，算法只得到了 12 对较好的匹配特征点，相对于算法 *SIFT* 以及算法 *SURF* 来说，其效果相对要更差些。

2、利用 ORB 算法来验证旋转不变的特性

同样的也是利用上面的编写的 ORB 算法来对相同的旋转场景进行测试，如下所示的匹配图，在匹配图当中的两幅图像都提取到了 500 个特征点，而且对应的匹配特征点也是 500 个。

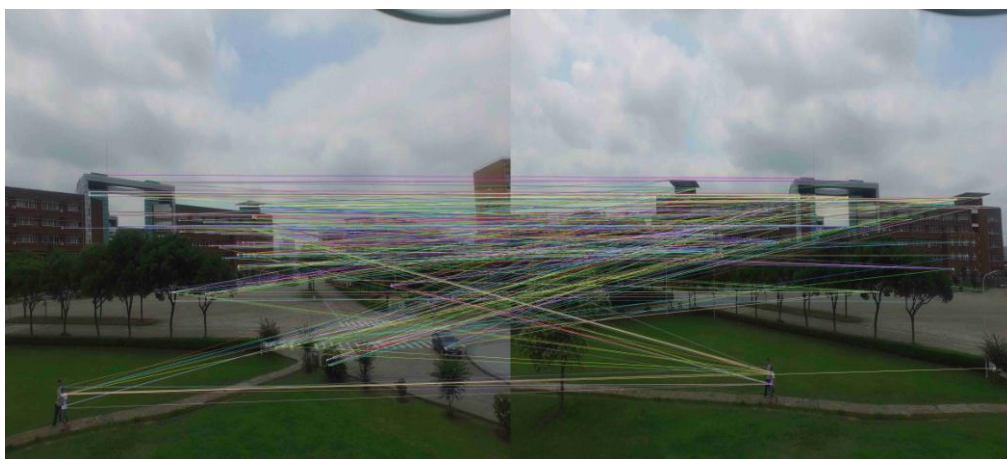


图 3.24 ORB 算法检测旋转不变性优化前匹配效果

Fig.3.24 The matching effect of ORB's rotation invariance before optimization

由于上图当中存在许多误匹配的特征点，故同样利用减小最小匹配距离来优化匹配图，得到的优化后的匹配图，如下所示



图 3.25 ORB 算法检测旋转不变性优化后匹配效果

Fig.3.25 The matching effect of ORB's rotation invariance before optimization

在优化后的匹配图当中总共得到了 47 对较好的匹配特征点。说明 ORB 在旋转不变的特性上还是能保持较好的稳定性。

综上所述，在旋转不变性方面，ORB 算法与 SURF 算法以及 SIFT 算法都能保持较好的性能；而在尺度不变性方面 ORB 算法取得的效果比后两者稍微差了一点，由于 ORB 算法是在 FAST 的基础上发展而来的，故其计算速度相比于 SURF 算法以及 SIFT 算法要快很多，综合考虑到算法的实时性，故本文选择 ORB 算法来作为环境图像的特征点提取以及匹配。

3.3 地图特征点的初始化

1、地图特征点的初始化方法

在各种基于视觉的同时定位与构图技术当中，唯有单目视觉是无法直接提取到三维场景当中物体的距离信息的，也就是说在一次观测当中只能获得目标物体的方位信息，而无法获得其深度信息。所以在建立环境特征地图的时候就存在环境特征的初始化问题。地图特征点的初始化问题可以理解为，在初始时刻，对环境特征点坐标位置的初始化估计，从而可以获得该特征点的位置信息。目前特征点的初始化问题可以分为两种类型：基于特征点延迟的初始化方法和基于特征点非延迟的初始化方法。

1) 基于特征点延迟的初始化方法在首次观测到新的环境特征点之后，它并没有去立即估计该特征点的坐标位置信息，而是把该时刻的运动载体的位姿信息以及测量信息存储下来，当下一次在另一个不同的位姿观测到该特征点时，然后根据相关的约束信息来计算该特征点的坐标位置。例如文献^[49]利用束调整的方式来实现地图特征点的初始化，其主要思想是利用批处理的方法来对观测到的信息进行更新从而实现对运动载体位姿以及特征点位置的估计，这种方式在观测量数据很大的时候，难以运用到实时环境当中。文献^[50]利用粒子滤波的原理来实现地图特征的初始化方法。该算法将运动载体所携带的相机位置与在其视图中的特征点位置连成一条直线，那么就利用粒子滤波方法把该特征点的位置信息作为粒子在该条直线上的分布，然后利用贝叶斯估计的方式对该距离信息进行粒子权值的更新计算，当距离信息满足高斯条件的时候，那么就把该分布的平均值作为地图特征点的初始位置估计。

2) 基于特征点非延迟的初始化方法是指在首次观测到特征点之后，就可以对该特征点进行初始化的相关操作，使之刚开始就可以融合到算法的整个过程当中。文献^[51]采用一种称为多假设的滤波方式，该方法提出在运动载体到目标特征点的连线上有多个目标特征位置，当运动载体继续运动后，便可以通过对之前特征点的具体位置进行估计。相比于基于延迟的初始化方法，非延迟的初始化方

法还是更有优势的；首先是，它并不需要过去时刻运动载体位姿以及观测量信息进行存储，可以减少算法的空间负担；然后就是如果运动载体的运动速度稍微比较快的时候，那么之前观测到的特征点可能就很难跟踪，从而造成无法对其进行初始化。故在本文当中采用的是基于非延迟的方法来对地图特征点进行初始化，所采用的算法是基于逆深度参数化方法。

2、基于逆深度参数化的地图特征点初始化方法

基于逆深度参数化的特征点初始化方法最初是由 Javier Civera 等人^[51]所提出来的，它是一种非延迟的地图特征点初始化方法，它可以比较有效地表示出环境特征点的距离的不确定性。就目前来说，该方法是在单目视觉同时定位与地图构建技术当中应用得最为广泛的地图特征初始化方法。其表示形式如下图所示

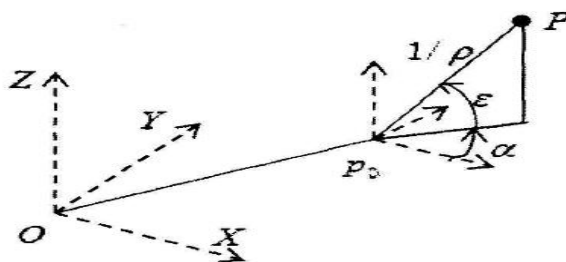


图 3.26 逆深度点参数化

Fig.3.26 The parameterization of inverse depth point

在上图当中一个逆深度特征点可以用 6 维的一个向量来表示，如下所示

$$L = [p_0^T \quad \varepsilon \quad \alpha \quad \rho]^T = [x_0 \quad y_0 \quad z_0 \quad \varepsilon \quad \alpha \quad \rho]^T \quad (3.28)$$

其中参数 p_0 表示初始化特征点时的光学中心， (ε, α) 表示初始化的光射线， ε 表示高度角信息，而 α 表示方向角信息， ρ 表示逆深度信息。那么逆深度特征点所对应的欧几里点可以表示为

$$p = p_0 + v^*(\varepsilon, \alpha) / \rho \quad (3.29)$$

上式当中 $v^*(\varepsilon, \alpha)$ 表示在该方向上的单位向量，其可以表示为如下

$$v^*(\varepsilon, \alpha) = [\cos(\varepsilon)\cos(\alpha) \quad \cos(\varepsilon)\sin(\alpha) \quad \sin(\varepsilon)]^T \quad (3.30)$$

3.4 本章小结

本章首先对常用的几种特征点提取方法的基本原理进行了详细的分析，并分别通过具体例子来验证各个特征提取算法；然后对以上特征点提取算法进行比较。最后是提出了基于单目视觉的同时定位与构图算法当中特有的地图特征初始化问题。

4 同时定位与地图构建技术的研究

由于早期使用的基于扩展卡尔曼滤波的 SLAM 技术基本上都是应用于机器人上，而且随着机器人观测到更多的环境特征的时候，其算法的时间复杂度以及空间复杂度随着特征点的增加呈平方增加，这样就难以使其应用到基于无人机视觉的 SLAM 技术当中。故本文所采用的 SLAM 技术是基于概率框架下的稀疏扩展信息滤波算法，它是信息滤波类算法当中最为著名的算法，该算法采用后验高斯分布的标准形式，即利用信息矩阵和信息向量的形式来估计系统的状态，使其保持了 EKF 具有的数学推导的严密性以及准确性等特点；下面主要是从稀疏扩展信息滤波的稀疏化操作的时机以及对状态均值恢复的计算效率角度出发而得到相应的改进算法。

4.1 基于熵规则的 SEIF SLAM 技术的研究

本改进方法将从信息滤波的核心——稀疏规则角度出发，在对稀疏化过程深入分析的基础之上，提出了基于熵稀疏规则的改进 SEIF SLAM 技术，该稀疏规则利用熵性质选出当前观测时刻以及下一观测时刻与位姿关联性相对最弱的环境主动特征作为稀疏特征点，从而提高 SLAM 技术的估计精度和一致性。

4.1.1 信息熵特性

信息熵^[52]的定义式为

$$H(X) = \sum_x P(X) \log \frac{1}{P(X)} = - \sum_x P(X) \log P(X) \quad (4.1)$$

其中， x 为离散随机变量， $P(X)$ 为其概率分布。信息熵 $H(X)$ 衡量的是随机变量 x 的不确定性，信息熵的值随着随机变量 x 不确定性的增大也随之增加。

下面通过把条件概率的概念引入到信息熵中的条件熵当中，其定义表达式如下

$$\begin{aligned} H(X|Y=y) &= \sum_x P(X|Y=y) \log \frac{1}{P(X|Y=y)} \\ &= -\sum_x P(X|Y=y) \log P(X|Y=y) \end{aligned} \quad (4.2)$$

其中, 表达式 $P(X|Y=y)$ 为随机变量 x 在条件 $Y=y$ 下的条件概率, 条件熵 $H(X|Y=y)$ 衡量的是在已知随机变量 $Y=y$ 的条件下, 随机变量 x 值的不确定性, 即完全知道随机变量 Y 的情况下, 随机变量 x 的熵还有多少。

由于在同时定位与构图问题当中, 本文把运动载体的位姿 x_t 表达为三维的随机向量, 其包括横坐标 x , 纵坐标 y 以及方向角 θ ; 其概率分布表达式可以表示为

$$P(x_t) = \frac{1}{(2\pi)^{3/2} \Sigma_t^{1/2}} \exp \left\{ -\frac{1}{2} (x_t - \mu_t)^T \Sigma_t^{-1} (x_t - \mu_t) \right\} \quad (4.3)$$

如果在 t 时刻有观测到环境特征点 m_i , 那么假设在观测到该环境特征点之后的运动载体位姿 x_t 的条件概率分布表达式如下:

$$P(x_t | m_i) = \frac{P(x_t, m_i)}{P(m_i)} \quad (4.4)$$

那么运动载体位姿 x_t 与环境特征点 m_i 的联合分布 $P(x_t, m_i)$ 以及环境特征点 m_i 的概率分布 $P(m_i)$ 分别是五维和二维的高斯分布, 那么其对应的概率分布表达式分别为

$$P(x_t, m_i) = \frac{1}{(2\pi)^{5/2} \Sigma_t^{1/2}} \exp \left\{ -\frac{1}{2} (\xi_t - \mu_t)^T \Sigma_t^{-1} (\xi_t - \mu_t) \right\} \quad (4.5)$$

$$P(m_i) = \frac{1}{2\pi \Sigma_t^{1/2}} \exp \left\{ -\frac{1}{2} (x_{m_i} - \mu_t)^T \Sigma_{m_i}^{-1} (x_{m_i} - \mu_t) \right\} \quad (4.6)$$

根据以上信息熵的性质, 可以计算出运动载体位姿 x_t 的熵以及在观测到环境特征点条件下的条件熵, 由于条件熵可能会使运动载体位姿的信息熵发生变化。那么就可以从观测到的环境特征点当中选出使运动载体位姿条件熵值最大的环境特征点, 即关联性最弱的环境特征点, 进行稀疏化操作。

4.1.2 改进稀疏规则

通过对标准稀疏扩展信息滤波 SLAM 技术的稀疏化过程分析可知，其在执行稀疏化操作的时候，只考虑到了当前时刻 t 与运动载体位姿 x_t 相关联的主动特征点，很难保证在每一次执行稀疏化操作之后，保留下来的环境特征点与都与运动载体位姿的关联性最强。通过对 SEIF SLAM 技术的稀疏化规则以及上面的信息熵概念的分析可以，如果稀疏化操作发生在运动载体位姿 x_t 运动到 $t+1$ 时刻的位姿 x_{t+1} 且同时执行一次环境的观测操作之后，那么就可以利用在 t 时刻位姿 x_t 所观测到的环境特征的条件熵以及 $t+1$ 时刻位姿 x_{t+1} 所观测到的环境特征的条件熵来综合决定相关性最弱的环境特征点，即

$$\max_i \{H(x_t|m=m_i)*H(x_{t+1}|m=m_i)\} \quad (4.7)$$

其中 m_i 为 t 时刻位姿 x_t 观测到的环境主动特征；在这里，如果在 $t+1$ 未观测到 t 时刻所观测到的环境特征 m_i ，那么在 $t+1$ 时刻运动载体位姿 x_{t+1} 的条件熵 $H(x_{t+1}|m=m_i)$ 的值取最大值。

下面分为四种情形来讨论（设在 t 时刻会超过预先设定的稀疏阈值；不失一般性，设经过改进稀疏规则计算得到，在观测到的所有特征点中，环境特征 m_i 使得运动载体位姿 x_t 的条件熵值达到最大。）

情形 1：运动载体在 $t+1$ 时刻除了观测到 t 时刻所对应的主动特征以外，还观测到了新的环境特征，见图 4.1(b)；此时新观测到的环境特征 m_j 有利于 $t+1$ 时刻位姿的精确定位，可改善系统状态向量的后验分布。因此有必要作为选择 t 时刻与位姿关联最弱的主动特征的一个参考量。此时可类比表达式 (4.7) 把判断条件作为

$$\max_i \{H(x_t|m=m_i)*H(x_{t+1}|m=m_i)*H(x_{t+1}|m=m_j)\} \quad (4.8)$$

其中 m_j 是运动载体在 t 时刻未观测到而在 $t+1$ 时刻观测到的环境特征。

此时可以通过式 (4.8) 来判断关联性最弱的环境特征点，并对其计算出关联性最弱的环境特征点进行稀疏化操作，见图 4.1 所示，

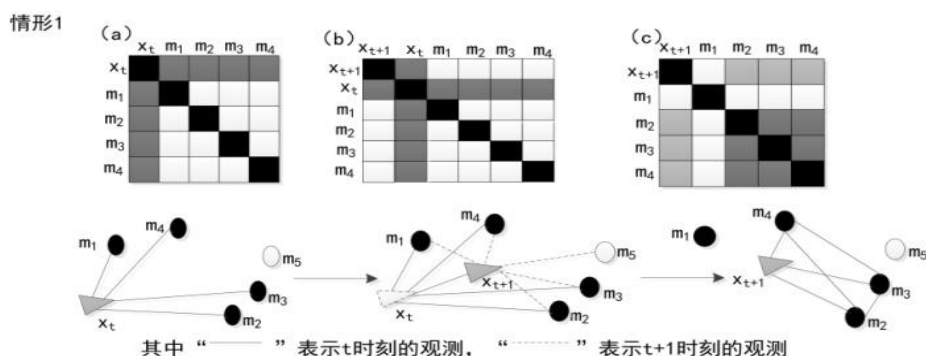

 图 4.1 在 $t+1$ 时刻观测到新特征以及 t 时刻所有特征条件下的稀疏化过程

 Fig.4.1 Sparse process under the condition of new feature and all features of time t are observed at time $t+1$

情形 2: 运动载体在位姿从 x_t 移动到 x_{t+1} 之后, 在 $t+1$ 时刻没有观测到新的环境特征, 即只观测到了 m_1, m_2, m_3, m_4 , 见图 4.2 (b) 所示, 利用式 (4.7) 便可判断出关联性最弱的环境特征点, 并将其进行稀疏化操作, 见图 4.2 所示,

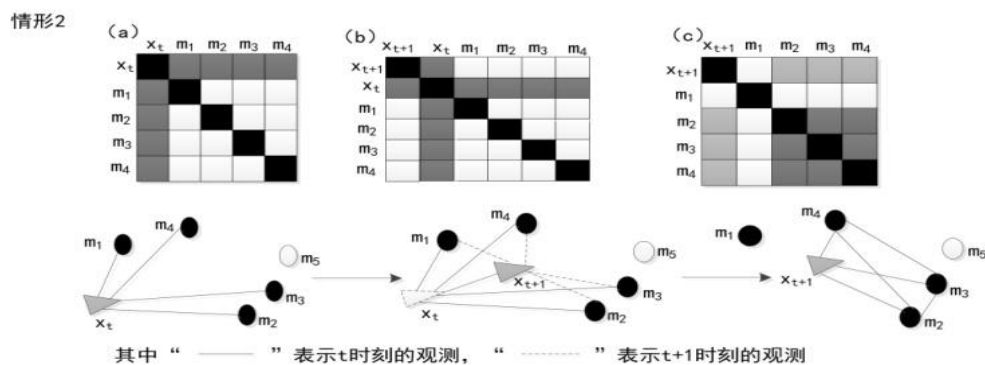

 图 4.2 在 $t+1$ 时刻观测到 t 时刻所有特征条件下的稀疏化过程

 Fig.4.2 Sparse process under the condition of all features of time t are observed at time $t+1$

情形 3: 运动载体在位姿从 x_t 移动到 x_{t+1} 之后, 其在 $t+1$ 时刻只观测到 t 时刻的部分环境特征且没有观测到新环境特征, 见图 4.3 (b) 所示, 不是一般性, 假设只观测到环境特征 m_3, m_4 , 而环境特征 m_1, m_2 没有被观测到, 在这里只需对环境特征 m_1, m_2 进行判断, 利用式 (4.7) 判断出关联性最弱的环境特征, 然后对其执行稀疏化操作。见图 4 所示,

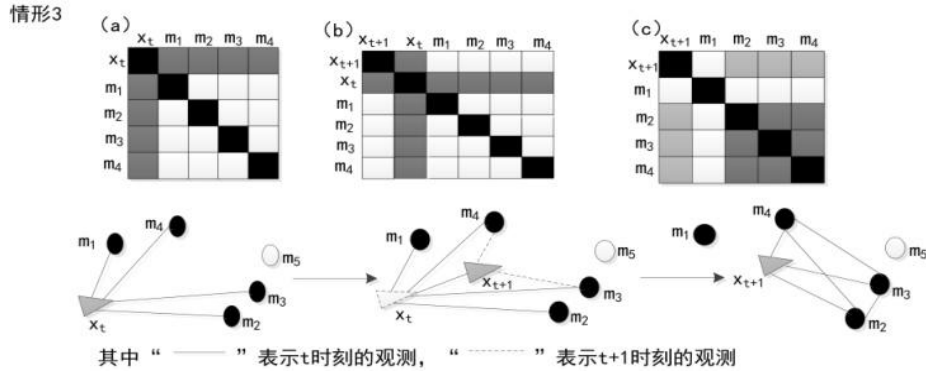

 图 4.3 在 $t+1$ 时刻观测到 t 时刻部分特征条件下的稀疏化过程

 Fig.4.3 Sparse process under the condition of some features of time t are observed at time $t+1$

情形 4: 运动载体在位姿从 x_t 移动到 x_{t+1} 之后, 在 $t+1$ 时刻观测到 t 时刻的部分环境特征 m_3, m_4 的同时也观测到新环境特征 m_5 , 见图 4.4(b); 在这里只需对环境 m_1, m_2 进行判断, 利用式 (4.8) 判断关联性最弱的环境特征, 并对其执行稀疏化操作。见图 4.4 所示,

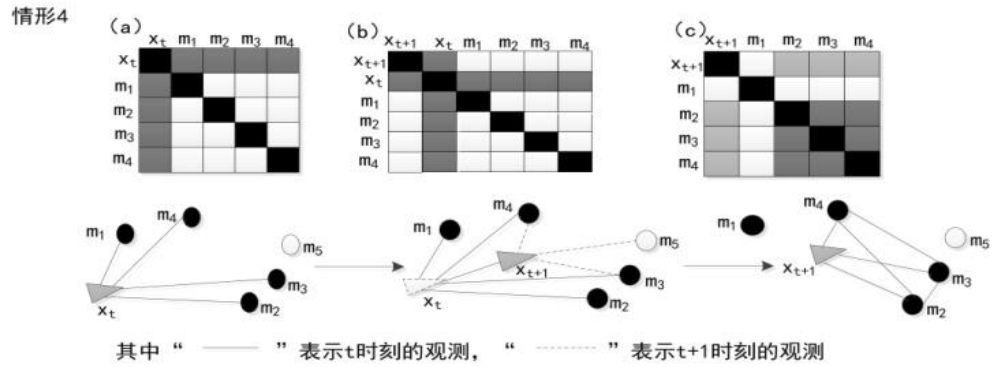

 图 4.4 在 $t+1$ 时刻观测到新特征以及 t 时刻部分特征条件下的稀疏化过程

 Fig.4.4 Sparse process under the condition of new feature and some features of time t are observed at time $t+1$

综合以上分析可知, 改进的环境特征点稀疏化方法, 即通过把稀疏化操作的时机改变为在在 t 时刻运动载体位姿 x_t 运动到 $t+1$ 时刻的位姿 x_{t+1} 且同时执行一次环境特征的观测操作之后, 且综合运用了信息熵的概念, 从而可以把与运动载体位姿关联性最强的特征保留下来, 使得改进的 SEIF SLAM 技术相比于标准的 SEIF SLAM 技术在系统状态的估计精度以及一致性方面均得到很大程度的提高。

4.2 基于状态均值恢复的 SEIF SLAM 技术的研究

针对 SLAM 技术在大环境下计算效率较低问题, 本文提出一种改进的基于稀疏扩展信息滤波的 SLAM 技术, 该改进算法是从状态估计 $\mu_t = \Lambda_t^{-1} \eta_t$ 的恢复角度出发, 即在对稀疏扩展信息滤波 (SEIF) 深入分析之后, 发现其对状态向量均值的恢复存在一定的不足, 在每个时间步上利用多次迭代算法来求解状态向量均值, 导致算法的整体计算性能下降; 根据均值恢复过程中需要对信息矩阵求逆以及信息矩阵稀疏性的特点, 提出快速三对角矩阵求逆方法来直接恢复状态均值, 改进算法的整体计算效率。

4.2.1 三对角矩阵求逆算法

针对大环境下设计的 SLAM 算法, 要求算法的实时性能必须较高, 因为本文所采用的基于稀疏扩展信息滤波的 SLAM 技术中需要恢复状态向量均值

$\mu_t = \Lambda_t^{-1} \eta_t$, 其中存在信息矩阵 Λ_t^{-1} 求逆的问题。随着运动载体的不断运动, 地图特征数量也随之不断增加, SLAM 中的信息矩阵将构成超多维矩阵, 且通常对矩阵求逆的计算复杂度为 $O(n^3)$ 。

为了进一步提高标准稀疏扩展信息滤波 SLAM 技术的计算效率, 本文提出一种快速三对角矩阵求逆的算法来解决状态向量均值恢复问题。根据信息矩阵数据分布的基本形态, 其数据主要集中在主对角线附近, 是对称的且对角占优的 Hermite 矩阵。根据 SLAM 信息矩阵的特点, 用矩阵分解的理论, 提出一种快速三对角求逆方法, 由于不需要直接对信息矩阵进行求逆, 从而算法的运算量得到大大降低。

1、信息矩阵分解

假设信息矩阵为 Λ , 将其分解为 Λ_1 和 Λ_2 , 其中 Λ_1 为三对角矩阵。

$$\Lambda = h(ij) = \Lambda_1 + \Lambda_2 \quad (4.9)$$

$$\Lambda_1 = \begin{pmatrix} h_{11} & h_{12} & \cdots & & \\ h_{21} & h_{22} & h_{23} & & \\ & h_{32} & h_{33} & & \\ \ddots & \ddots & \ddots & \ddots & \ddots \\ & & \cdots & h_{n-1,n-1} & h_{n-1,n} \\ & & & h_{n,n-1} & h_{n,n} \end{pmatrix} \quad (4.10)$$

$$\Lambda_2 = \begin{pmatrix} 0 & 0 & \cdots & h_{1,n-1} & h_{1,n} \\ 0 & 0 & 0 & h_{2,n-1} & h_{2,n} \\ h_{3,1} & 0 & 0 & & h_{3,n} \\ \ddots & \ddots & \ddots & \ddots & \ddots \\ h_{n-1,1} & h_{n-1,2} & \cdots & 0 & 0 \\ h_{n,1} & h_{n,2} & h_{n,3} & 0 & 0 \end{pmatrix} \quad (4.11)$$

由于矩阵 Λ 满足前述条件，根据矩阵公式，则有如下结果

$$\Lambda^{-1} = (I - \Lambda_1^{-1}\Lambda_2 + (\Lambda_1^{-1}\Lambda_2)^2 + \cdots + (-1)^n(\Lambda_1^{-1}\Lambda_2)^n + \cdots)\Lambda_1^{-1} \quad (4.12)$$

那么， Λ^{-1} 的一阶近似为

$$\Lambda^{-1} \approx \Lambda_1^{-1} - \Lambda_1^{-1}\Lambda_2\Lambda_1^{-1} \quad (4.13)$$

信息矩阵求逆问题转化为三对角阵 Λ_1 逆的计算和矩阵乘积问题。

2、三角矩阵求逆

首先讨论三对角阵 Λ_1 逆的计算快速实现，为证明方便，不妨设

$\Lambda_1 = \text{diag}(a_1, b_1, c_1)$ ，即

$$\Lambda_1 = \begin{pmatrix} b_1 & c_1 & & & \\ a_2 & b_2 & c_2 & & \\ & a_3 & b_3 & c_3 & \\ & & \ddots & \ddots & \ddots \\ & & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & & a_n & b_n \end{pmatrix} \quad (4.14)$$

满足如下关系式：

$$p_i \triangleq b_i - a_{i-1}a_i \neq 0, \quad i = 1, \dots, n. \quad (4.15)$$

$$q_i \triangleq b_{i+1}c_i \neq b_i, \quad i = 1, \dots, n. \quad (4.16)$$

$$p_i - q_i \neq 0, \quad i = 2, \dots, n-1. \quad (4.17)$$

其中

$$a_0 \triangleq 0, \quad a_i \triangleq \frac{c_i}{p_i}, \quad i = 1, \dots, n-1. \quad (4.18)$$

$$\beta_{n+1} \triangleq 0, \beta_i \triangleq \frac{a_i}{b_i - q_i}, \quad i = n, \dots, 2. \quad (4.19)$$

令 $\Lambda_1^{-1} \triangleq (x_{i,j})_{n \times n}$ ，可由下述快速算法求得，

步骤一：令

$$p_1 \triangleq b_1, \quad p_i \triangleq b_i - a_{i-1}a_i, \quad i = 2, \dots, n, \quad (4.20)$$

$$q_n \triangleq 0, \quad q_i \triangleq \beta_{i+1}c_i, \quad i = n-1, \dots, 1, \quad (4.21)$$

$$a_i \triangleq \frac{c_i}{p_i}, \quad i = 1, \dots, n-1, \quad (4.22)$$

$$\beta_n \triangleq \frac{a_n}{b_n}, \quad \beta_i \triangleq \frac{a_i}{b_i - q_i}, \quad i = n-1, \dots, 2, \quad (4.23)$$

步骤二：计算

$$x_{1,1} = \frac{1}{b_1 - q_1}, \quad x_{i,1} = -\beta_i x_{i-1,1}, \quad i = 2, \dots, n, \quad (4.24)$$

$$x_{n,n} = \frac{1}{p_n}, \quad x_{i,n} = -a_i x_{i+1,n}, \quad i = n-1, \dots, 1, \quad (4.25)$$

步骤三：对于 $j = 2, \dots, n$ - 分别计算

$$x_{j,j} = \frac{1}{p_j - q_j}, \quad (4.26)$$

$$x_{i,j} = -\beta_i x_{i-1,j}, \quad i = j+1, \dots, n, \quad (4.27)$$

$$x_{i,j} = -a_i x_{i+1,j}, \quad i = j-1, \dots, 1, \quad (4.28)$$

3、算法证明过程

设 $X_j = (x_{1,j}, x_{2,j}, \dots, x_{n,j})^T$, 并用 E_j 表示 n 阶单位方阵的第 j 列, 分三种情况:

1) 先考虑方程组 $\Omega_1 X_1 = E_1$, 由 (4.16)、(4.17) 式可知, 对于 $i = n, n-1, \dots$, 可分别从第 i 个方程解出 $x_{i,1}$, 然后代入第 $i-1$ 个方程当中, 由 (4.19) 式逐个可得出

$$x_{i,1} = -\beta_i x_{i-1,1}, \quad i = n, \dots, 2, \quad (4.29)$$

$$x_{1,1} = \frac{1}{b_1 - q_1} \quad (4.30)$$

再从 (4.30) 式出发, 利用 (4.29) 式依次顺推, 逐个可得

$$x_{i,1} = -\beta_i x_{i-1,1}, \quad i = 2, \dots, n \quad (4.31)$$

2) 再考虑方程组 $\Omega_1 X_n = E_n$ 由 (4.15)、(4.17) 式可知, 对于 $i = 1, \dots, n-1$, 可分别从第 i 个方程当中解出 $x_{i,n}$, 代入第 $i+1$ 个方程当中, 由 (4.18) 式逐个可得

$$x_{i,n} = -a_i x_{i+1,n}, \quad i = 1, \dots, n-1, \quad (4.32)$$

$$x_{n,n} = \frac{1}{p_n} \quad (4.33)$$

再从 (4.33) 式出发, 利用 (4.32) 式依次倒推, 逐个可得

$$x_{i,n} = -a_i x_{i+1,n}, \quad i = n-1, \dots, 1 \quad (4.34)$$

3) 然后考虑方程组 $\Omega_1 X_j = E_j$, $j = 2, \dots, n-1$. 对任意给定的 $j, 2 \leq j \leq n-1$ 。首先, 仿照 1) 的方式, 从第 n 个方程开始一次倒推, 由 (4.17)、(4.19) 式逐个可得

$$x_{i,j} = -\beta_i x_{i-1,j}, \quad i = n, \dots, j+1 \quad (4.35)$$

再仿照 2) 的方式, 从第一个方程开始一次顺推, 由 (4.17)、(4.18) 式逐个可得

$$x_{i,j} = -a_i x_{i+1,j}, \quad i = 1, \dots, j-1 \quad (4.36)$$

然后, 分别从 (4.35)、(4.36) 式得出 $x_{j+1,j}$, $x_{j-1,j}$ 的表达式, 将它们代入第 j 个方程, 由 (4.17) 式得到

$$x_{j,j} = \frac{1}{p_j - q_j} \quad (4.37)$$

最后, 从 (4.37) 式出发, 利用 (4.35) 式依次倒推, 逐个可得

$$x_{i,j} = -a_i x_{i+1,j}, \quad i = j-1, \dots, 1 \quad (4.38)$$

再从 (4.37) 出发, 利用 (4.36) 式依次顺推, 逐个可得

$$x_{i,j} = -\beta_i x_{i-1,j}, \quad i = j+1, \dots, n \quad (4.39)$$

由上面得到的 (4.30)、(4.31)、(4.33)、(4.34)、(4.37) - (4.39) 及 (4.17)、(4.18) 式, 从而给出了求逆的快速算法。

4.3 本章小结

本章主要分析和推导了基于稀疏扩展信息滤波同时定位与地图构建技术 (SEIF SLAM) 的改进算法, 分别提出了基于熵规则的信息滤波改进算法, 其目的在于通过信息熵的原理使 SLAM 技术的估计精度以及一致性得到提升; 以及提出了基于状态向量均值恢复改进的算法, 该算法利用三角矩阵求逆的方式提高了算法的计算速度, 从而有利于大环境下的 SLAM 技术的研究。

5 改进 SEIF SLAM 算法的仿真及分析

为了验证改进算法对基于无人机单目视觉同时定位与地图构建技术的可行性和有效性。本文通过在无人机上搭载视觉传感器并在预先设定的航行轨迹上来获取周围环境的图像序列，通过图像特征提取算法 ORB 对所获取的环境图像进行特征点提取，然后使用近似最近邻（FLANN）特征点匹配算法来选择在匹配过程当中相对稳定的特征点，将这些较稳定的特征点作为环境路标；下图 5.1 及 5.2 是从图像序列中选取的相邻两张帧图像经过 ORB 特征点提取算法处理过的场景图，如下所示



图 5.1 经过 ORB 算法提取的前一帧图像特征点示意图

Fig.5.1 The schematic diagram of the previous frame image features extracted by ORB



图 5.2 经过 ORB 算法提取的后一帧图像特征点示意图

Fig.5.2 The schematic diagram of the next frame image features extracted by ORB

然后，利用 FLANN 特征点匹配算法，对上面两帧图像进行特征点匹配，如下图 5.3 所示，将较好的匹配特征点保留下来，作为构建仿真环境中的环境路标。



图 5.3 相邻帧间较好的匹配特征点示意图

Fig.5.3 The schematic diagram of good matching feature points between adjacent frames

通过把所获取的图像视频按照时间间隔为 1s 分隔成一套图像序列，即可以理解为无人机的运动时间步为 1s；按照上面的 ORB 图像处理方法把所有图像序列的特征点提取出来，保留相对较好的特征点，计算出这些特征点的坐标位置，然后把这些较好的特征点放入模拟的仿真环境当中，如下图所示

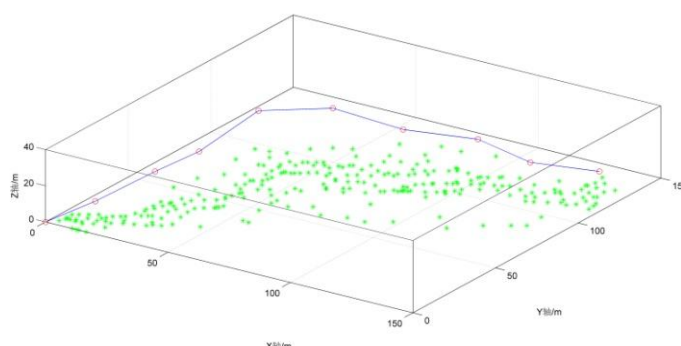


图 5.4 模拟的三维仿真环境

Fig.5.4 The simulation of 3d simulation environment

在上图的仿真环境当中绿色的星状部分表示的是提取的环境特征点，选择的每个特征点之间的最小距离不低于 1m；在上图中位于特征点上方的蓝色点线部分是模拟的无人机的航行轨迹，其航行轨迹的初始坐标设置为坐标原点 (0,0,0)，摄像机的观测视角为 70°，观测视距为 40m。无人机以 0.5m/s 的速度航行，假设无

人机在每个时间步里面都是做在直线运动^[42]；通过视觉里程计可以获得无人机每一时间步的速度和偏转角度；利用单目摄像机在不同时间步上观测到相同特征点的方式来获得无人机与该特征点之间的相对位置，即利用三角化原理来获得二者之间的相对位置。

在此基础上，分别利用基于熵规则的改进 SLAM 技术以及基于三对角矩阵求逆方法的改进 SLAM 技术来建立仿真实验，本实验的仿真环境是在 Matlab R2014a 下运行，其具体实验结果及实验分析分别如下所示。

5.1 基于改进的熵规则的 SEIF SLAM 技术的实验结果及分析

在进行仿真实验当中，把基于稀疏扩展信息滤波的活动特征点的稀疏阈值固定为 8，即在环境主动特征点的个数超过 8 的时候，就必须执行稀疏化操作。

本实验将在相同的无人机初始状态、仿真环境以及系统噪声和观测噪声条件下，对基于熵规则的改进 SEIF SLAM 技术和基于标准的 SEIF SLAM 技术进行比较分析。见图 5.5 所示为基于改进的 SEIF SLAM 技术和基于标准 SEIF SLAM 技术估计无人机位姿状态的误差比较情况，由于改进算法引入了熵规则且把稀疏化时刻更改为发生在无人机在 t 时刻位姿 x_t 运动到 $t+1$ 时刻的位姿 x_{t+1} 且同时执行一次环境的观测操作之后，故从图中可以看出改进算法估计位姿偏差相比于标准 SEIF SLAM 技术有了明显的减小。

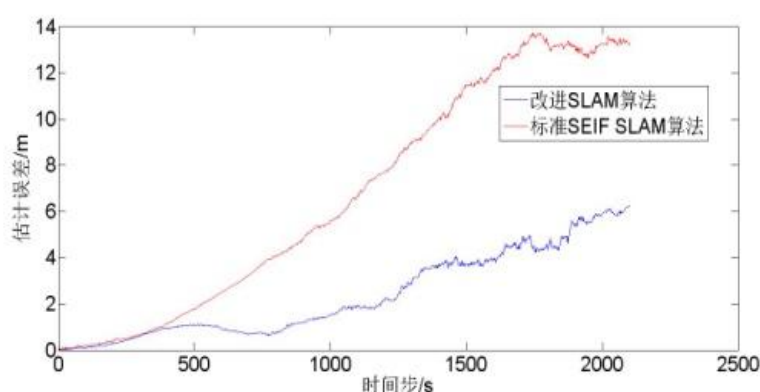


图 5.5 改进算法与标准 SEIF SLAM 误差比较

Fig.5.5 Error comparison between improved algorithm and standard SEIF SLAM

为了进一步统计改进算法与标准算法处理过程的实际效果，本实验采用 Monte-Carlo 仿真来分析改进的稀疏扩展信息滤波算法与标准稀疏扩展信息滤波算

法的处理性能。在经过 30 次 Monte-Carlo 仿真实验之后，从图 5.6 可以看出，改进算法的估计结果相比标准的稀疏扩展信息滤波更好。

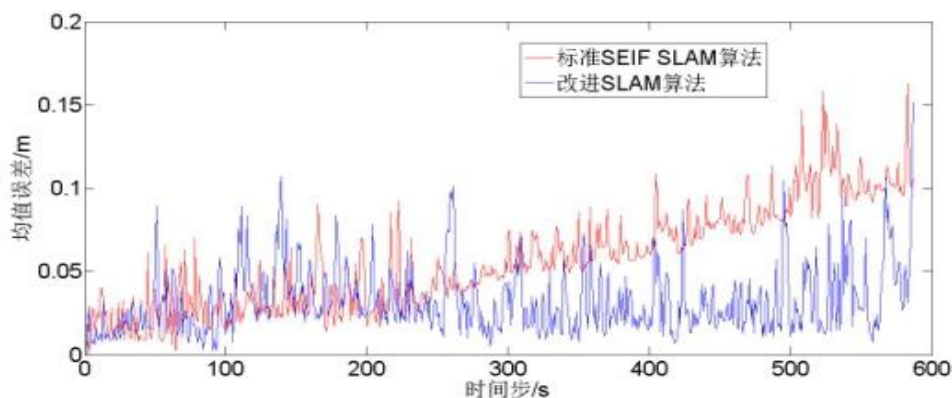


图 5.6 改进算法与标准 SEIF SLAM 均值误差比较

Fig.5.6 Comparison of the mean error between the improved algorithm and standard SEIF SLAM

为了验证基于熵规则的改进算法 SLAM 技术的一致性，本实验从一致性估计的定义出发来对其进行验证。

对于在一个系统当中真实状态向量 $x(k)$ 对应的估计量假设为 $\hat{x}(k/k)$ ，其对应的系统估计误差状态向量设为 $x(k/k) = x(k) - \hat{x}(k/k)$ ，系统状态估计协方差假设为 $P(k/k)$ ，如果下面的式子成立

$$E[x(k/k)] = 0 \quad (5.1)$$

$$E[x(k/k)x(k/k)^T] = P(k/k) \quad (5.2)$$

那么就称该系统状态的估计 $\hat{x}(k/k)$ 是一致的。

那么从上式可以看出，通过对系统真实状态值 x_k 和状态协方差 P_k 以及系统状态估计值 $\hat{x}(k/k)$ 和状态估计协方差 $P(k/k)$ 的仿真数据进行对比验证，就可以检验出对系统的估计结果是否一致。由式 (5.2) 可知，系统状态估计一致性应满足：

$$x_k - \hat{x}(k/k) \leq \sqrt{\text{diag}(P(k/k))} \quad (5.3)$$

在本次仿真实验当中只研究系统各状态的估计误差，故利用式(5.3)对系统各状态的估计误差以及协方差矩阵估计值进行比较，下面通过把仿真实验数

据绘出比较的结果曲线，便可以直观判断出对系统状态估计的一致性性能。

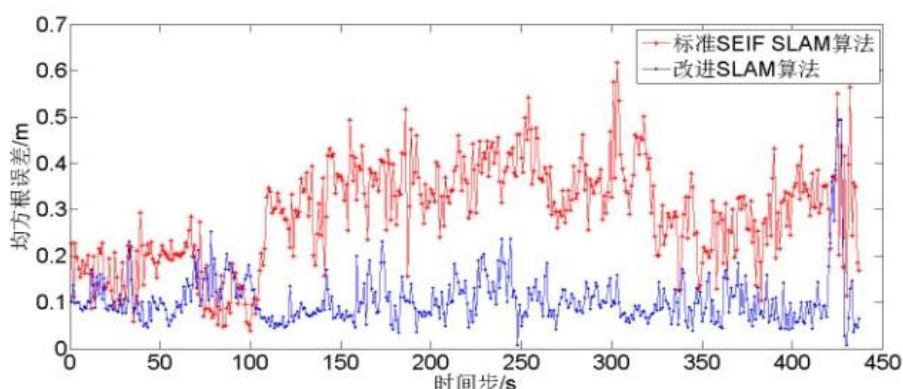


图 5.7 改进算法与标准 SEIF SLAM 一致性比较

Fig.5.7 Consistency comparison between improved algorithm and standard SEIF SLAM

通过 30 次 Monte-Carlo 仿真实验之后，由图 5.7 可以看得出，在基于改进后的稀疏规则的稀疏扩展信息滤波 SLAM 技术的一致性相比于基于标准稀疏扩展信息滤波 SLAM 技术有了比较明显的提高。

5.2 基于状态均值恢复的 SEIF SLAM 技术的实验结果及分析

在本仿真实验当中主要考虑的是算法的实时性，在对 SLAM 技术进行仿真实验之前，先对该进的矩阵求逆方式进行实验验证。

下面是本文设计的测试矩阵分别由快速三对角矩阵求逆和 Matlab 的求逆方法进行测试和比较。设计的测试矩阵是随机产生的，且满足对角占优、实对称性的稀疏矩阵。图 5.8 给出了改进求逆算法与常规 Matlab 求逆算法计算时间的一个比较，实验结果显示在 100 阶以下两种求逆方法的求逆时间差别并不很大，但是随着矩阵的阶数不断增加，快速三对角方法求逆计算时间远远快于使用 Matlab 的求逆方法。

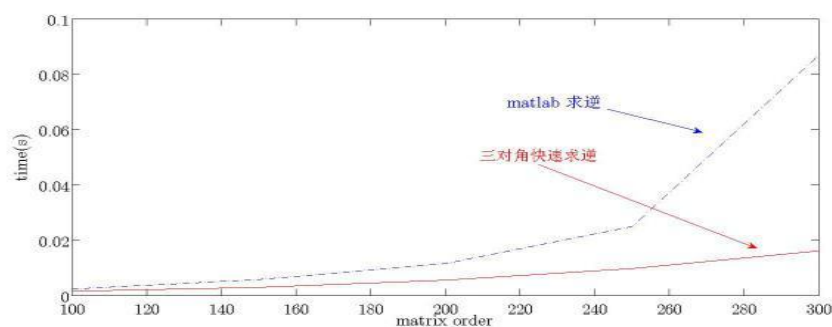


图 5.8 三对角求逆和 Matlab 求逆计算时间比较

Fig.5.8 The comparison of computation time between tridiagonal matrix inverse and Matlab inverse

下面把改进的三对角矩阵求逆方法带入稀疏扩展信息滤波算法当中，然后与标准稀疏扩展信息滤波算法在相同的初始状态和环境条件下进行实验对比分析，

下图是算法的整体性能的对比分析，可以看出基于改进算法的 SLAM 技术和标准 SEIF SLAM 技术的计算效率之间的差别，即基于改进算法的 SLAM 技术的计算速度相比标准 SEIF SLAM 技术的计算效率明显提高。

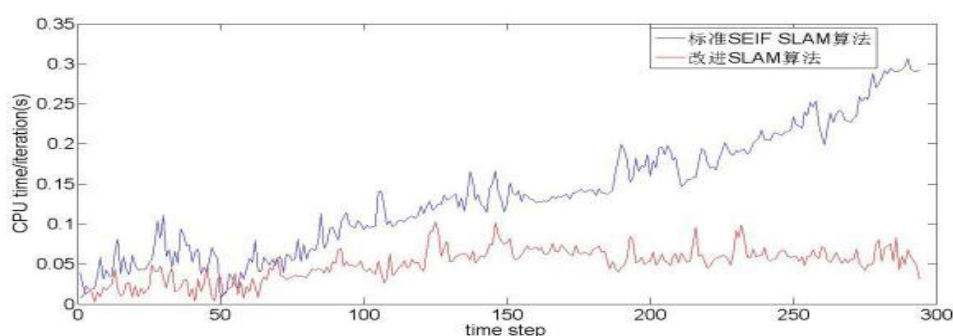


图 5.9 改进算法与标准 SEIF SLAM 算法 CPU 平均消耗时间对比

Fig.5.9 The comparison of CPU average computation time between improved algorithm and standard SEIF SLAM algorithm

为了更进一步探讨稀疏扩展信息滤波的最后一步对整个算法效率的性能作用，即状态均值恢复效率的提高对整个算法的影响，下面分别对基于改进算法的 SLAM 技术和基于标准 SEIF SLAM 技术在运动更新阶段以及观测更新阶段的计算效率进行对比分析；由于在运动更新阶段和观测更新阶段的计算当中均需要对状态向量均值进行计算，见式(2-61)以及(2-65)；故在其他因子不变的前提下，当状态均值计算效率提高，那么运动更新阶段以及观测更新阶段的计算效率也会随之提高，见图 5.10 和图 5.11 所示。

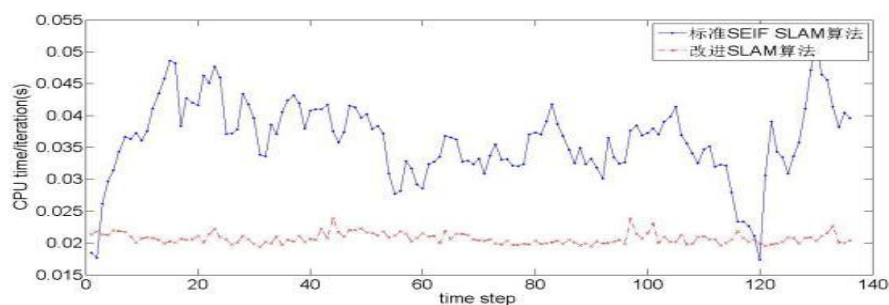


图 5.10 改进算法与标准 SEIF SLAM 算法运动更新阶段 CPU 平均消耗时间对比

Fig.5.10 The comparison of motion update's CPU average computation time between improved algorithm and standard SEIF SLAM algorithm

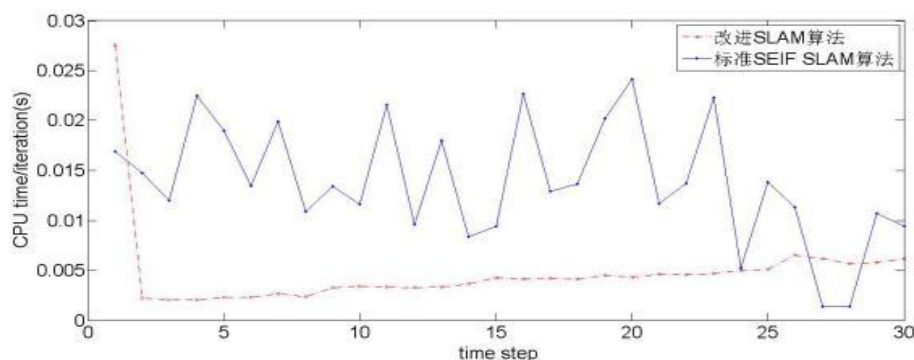


图5.11 改进算法与标准SEIF SLAM算法观测更新阶段CPU平均消耗时间对比

Fig.5.11 The comparison of observation update's CPU average computation time between improved algorithm and standard SEIF SLAM algorithm

综合上述的仿真实验对比可知，基于三对角求逆的稀疏扩展信息滤波 SLAM 技术在整体性能以及每个阶段性能上均比基于标准的稀疏扩展信息滤波 SLAM 技术更优。

6 总结与展望

6.1 本文工作总结

对基于无人机视觉的同时定位与地图构建技术来说，它主要涉及到两个方面的研究领域，一是计算机视觉技术，二是基于无人机的导航与定位技术。随着计算机视觉技术研究的不断深入发展，它已逐渐成为导航定位技术领域当中不可或缺的一个组成部分。随着无人机作业范围的不断扩大，在有些比较特殊的环境 GPS 信号受干扰情况比较严重，或者会直接导致 GPS 信号失效。所以本文采用了基于单目视觉 SLAM 的无人机自主导航定位方式来作为无人机导航定位系统的一种辅助方式。

目前在这方面的技术相对于国外来说，不论是在理论上还是在具体实践上，国内在这方面的研究都相对要少。基于这样的一个背景，本课题采用真实的三维场景结合算法仿真的方式来实现无人机的同时定位与地图构建技术，即通过无人机携带的视觉传感器来获取环境特征信息，并将获得的环境特征信息通过仿真的方式将其模拟为对应的三维场景，然后再结合改进的稀疏扩展信息滤波算法来实现无人机的同时定位与地图构建技术。本论文所研究的工作内容可归纳为如下几点：

1) 首先对无人机基于 GPS 的导航定位技术的一些不足进行了分析，针对这些不足，提出了基于无人机视觉的同时定位与地图构建技术作为无人机的辅助导

航方式，同时介绍了该技术框架下的一些常用算法；然后把该框架分成五个组成部分：图像数据的采集、视觉里程计、算法后端优化、地图构建以及闭环检测，并提出本文的研究点以及创新点是针对算法的优化部分。

2) 针对本文要研究的后端优化算法，分别分析并详细推导了基于贝叶斯滤波的同时定位技术、基于扩展卡尔曼滤波的同时定位技术以及基于稀疏扩展信息滤波的同时定位技术；并在分析基于稀疏扩展信息滤波的同时定位与地图构建技术当中，提出了两个创新点：针对稀疏操作的改进以及加快恢复系统状态均值。在以上理论基础之上，建立了本课题基于无人机视觉的系统运动方程和观测方程。

3) 探讨了包括 SIFT、SURF 以及 ORB 图像特征提取算法的基本理论和原理推导，然后分别编写对应的算法程序，把上述三种图像特征点提取算法应用到由无人机拍摄的环境图像序列上，分别进行特征点提取以及特征点匹配实验，并综合实验结果将 ORB 图像特征提取算法应用到本课题当中；然后介绍了基于单目视觉 SLAM 技术所特有的地图特征点的初始化问题，并将逆深度参数化方法应用到本研究课题当中。

4) 提出了对基于稀疏扩展信息滤波的 SLAM 技术的改进算法，本文主要从系统状态估计的精确度以及一致性和算法计算效率出发，分别提出了基于熵规则的稀疏扩展信息滤波 SLAM 技术以及基于三对角矩阵求逆方法的 SLAM 技术。前者的改进之处在于利用熵的性质、综合当前以及下一观测时刻来选择与位姿关联性最弱的环境特征作为稀疏特征点，从而使保留下来的环境特征点与位姿关联性最强。后者的改进之处在于利用信息矩阵数据分布的基本形态，即数据主要集中在主对角线附近，且是对称的对角占优的 Hermite 矩阵，然后利用矩阵的分解理论，从中分解出一个三对角矩阵，最后将原始矩阵的求逆转换为对三角矩阵的求逆，从而使算法的计算效率得到大大提高。

5) 利用软件 Matlab R2014a 设计本文的仿真实验，首先将获得到的环境特征点以及无人机的航行轨迹模拟成仿真三维环境；然后分别利用两种改进的算法与标准算法进行对比来验证改进算法的有效性，通过实验仿真结果可知，均达到了预期的效果。

6.2 未来工作展望

基于无人机视觉的同时定位与地图构建技术本就属于非常复杂的一个系统，且鉴于自身理论知识以及时间的限制，本论文的研究工作还主要是通过仿真实验

的方式来实现基于改进算法的无人机视觉 SLAM 技术；对于本课题的后续研究内容，主要体现在以下几个方面：

1) 在本论文的传感器部分只使用了基于单目视觉的传感器，视觉传感器在获取环境图像信息的时候容易受到环境外部信息的干扰，比如光照情况、物体遮挡等的影响，为了增加系统的鲁棒性，可以使用多传感器相融合的方式来实现系统的可靠性以及稳定性。

2) 在仿真实验当中所模拟的三维环境是由 ORB 算法经过离线优化处理之后的环境特征点所组成的，如果在实际应用过程当中经过 ORB 算法提取的众多特征点中选择出比较好的特征点，并实现相邻两帧图片之间特征的快速匹配，那么图像特征提取算法在实时性、特征点提取质量和匹配方面还需要进一步提高和改进。

3) 本文采用的是编写仿真程序的方式来实现 SLAM 技术的，如果要实现 SLAM 技术的可实践性，必须对 SLAM 技术中的仿真程序进行跨平台移植，使用其他的编程语言来实现，因此在编程语言方面还需进一步完善，从而实现程序的精简性、高效性。

4) 一整套 SLAM 系统的完成，不仅仅只包括软件部分，同时还包括硬件部分，因为 SLAM 技术是由多个不同算法组成的，所以在硬件实现部分需要考虑使用不同模块来完成不同的算法，仅靠单一的硬件模块来完成 SLAM 系统是难以办到的，所以在这一方面还需做进一步研究。

参考文献

- [1] Fravolini M,Campa G,Ficola A,et al.Modeling and control issues or machine vision-based autonomous aerial refueling for UAVs using a probe-drogue refueling systems[J].Aerospace Science and Technology, 2004,8(7):611-618.
- [2] Valasek J,Gunnam K,Kimmet J,et al.Vision based sensor and navigation system for autonomous air refueling[J].Journal of Guidance,Control,and Dynamics,2005,28(5):979-989.
- [3] Webb T,Prazenica R,Kurdial A,et al.Vision-based state estimation for autonomous micro air vehicles[J].Journal of Guidance,Control,and Dynamics,2007,30(3):816-826.
- [4] 黄帅, 蔡洪, 丁智坚. 惯性导航系统动基座传递对准可观测性分析[J]. 北京航空航天大学学报, 2016, 42(11):2548-2554.
- [5] 王伟, 陈华庆, 韩卫. 无人机自主导航控制的 FastSLAM 算法研究[J]. 计算机仿真, 2015, 32(8):68-72.
- [6] K Ikeuchi.Concurrent Mapping and Localization(CML)[J].Computer Vision,2014:146-146.
- [7] 罗荣华, 洪炳熔. 移动机器人同时定位与地图创建研究进展[J]. 机器人, 2004, 26(2):182-186.
- [8] 张恒, 樊晓平, 刘艳丽. 移动机器人同步定位与地图构建研究进展[J]. 数据采集与处理, 2005, 20(4):458-465.
- [9] 王璐, 蔡自兴. 未知环境中移动机器人并发建图与定位(CML)的研究进展[J]. 机器人, 2004, 26(4):380-384.
- [10] R Smith, M Self, P Cheeseman. Estimating uncertain relationships in robotics[C]. Amsterdam: Proceedings of Conference on Uncertainty in Artificial Intelligence. 1988.
- [11] N Ayache,O Faugeras.Building registrating and fusing noisy visual maps[J].Int. J.Robotics Research. 1988,7(6):45-65.
- [12] R Chatila,J P Lanrnond.Position referencing and consistent world modeling for mobile robots[C]. Leoven: Proc. IEEE Int. Conf. Robotics and Automation.1985.
- [13] 薛永胜, 王姮, 张华等. EKF-SLAM 算法的改进及其在 Turlebot 的实现[J].西南科技大学学报, 2015, 30(1):54-59.
- [14] 陈晨, 程荫杭.基于迭代无迹卡尔曼滤波的 SLAM 算法仿真研究[J].系统仿真学报, 2012,24(8):1643-1650.
- [15] FA Cheein,G Steiner,GP Paina et al.Optimized EIF-SLAM algorithm for precision agriculture mapping based on stems detection[J].Computers & Electronics in Agriculture,2011,78(2):195-207.
- [16] Sebastian Thrun,Yufeng Liu,Daphne Koller,et al. Simultaneous Localization and Mapping With Sparse Extended Information Filters[J].International Journal of Robotics Research, 2004, 23(7-8):693-716.
- [17] 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.

- [18] 朱奇光, 袁梅, 王梓巍等. 机器人球面单径容积 FastSLAM 算法[J]. 机器人, 2015, 37(6): 709-717.
- [19] Austin, Eliazar and Ronald Parr. DP-SLAM 2.0[J]. Proceedings of the 2004 IEEE International Conference on Robotics & Automation, 2004, 4(3): 1314-1320.
- [20] Mullane, J., et al., A Random-Finite-Set Approach to Bayesian SLAM[J]. IEEE Transactions on Robotics, 2011, 27(2): 268-282.
- [21] Adams, M. SLAM Gets a PHD: New Concepts in Map Estimation[J]. IEEE Robotics Automation Magazine, 2014, 21(2): 26-37.
- [22] KLEIN G, MURRAY D. Parallel tracking and mapping for small AR workspaces[C]//6th IEEE and ACM International Symposium on Mixed and Augmented Reality. Nara: IEEE, 2007: 225-234.
- [23] KELIN G, MURRAY D. Improving the agility of keyframe based SLAM[M]//FORSYTH D, TORR P, ZISSERMANA, et al. Computer Vision-ECCV 2008. Berlin: Springer, 2008: 802-815.
- [24] EADE E, DRUMMOND T. Monocular SLAM as a graph of coalesced observations[C]//IEEE 11th International Conference on Computer Vision. Rio de Janeiro: IEEE, 2007: 1-8.
- [25] STRASDAT H, MONTIEL J M M, DAVISON A J. Scale drift-aware large scale monocular SLAM[J]. Robotics: Science and Systems, 2010, 2(3): 5-5.
- [26] Y Kameda. Parallel Tracking and Mapping for Small AR Workspaces (PTAM) Augmented Reality[J]. Journal of the Institute of Image Information & Television Engineers, 2012, 66(1): 45-51.
- [27] Raul Mur-Artal, J.M.M. Montiel, Juan D. Tardos. ORB-SLAM: A Versatile and Accurate Monocular SLAM System[J]. IEEE Transactions on Robotics, 2015, 31(5): 1147-1163.
- [28] C Forster, M Pizzoli, D Scaramuzza. SVO: Fast semi-direct monocular visual odometry[J]. IEEE International Conference on Robotics & Automation, 2014: 15-22.
- [29] J Engel, T Schops, D Cremers. LSD-SLAM: Large-Scale Direct Monocular SLAM[C], 2014: 1-31.
- [30] RA Newcombe, SJ Lovegrove, AJ Davison. DTAM: Dense tracking and mapping in real-time[J]. IEEE International Conference on Computer Vision, 2011, 24(4): 2320-2327.
- [31] D Gutierrez-Gomez, W Mayol-Cuevas, JJ Guerrero. Dense RGB-D visual odometry using inverse depth[J]. Robotics & Autonomous Systems, 2016, 75(PB): 571-583.
- [32] K Melbouci, SN Collette, V Gay-Bellile, et. al. Model based RGBD SLAM[J]. IEEE International Conference on Image Processing, 2016: 2618-2622.
- [33] T. Whelan, M. Kaess, M.F. Fallon, et. al. KinectFusion: Spatially Extended KinectFusion[C], 2013: 1-8.
- [34] 谭民, 王硕, 曹志强. 多机器人系统 [M]. 北京: 清华大学出版社, 2005.
- [35] 陈卫东, 张飞. 移动机器人的同步自定位与地图创建研究进展[J]. 控制理论与应用, 2005, 22(3): 455-460.
- [36] RE Kalman. A New Approach to Linear Filtering and Prediction Problems[J]. Transactions of the ASME-Journal of Basic Engineering, 1960(82): 35-45.
- [37] Y. Sunahara. An Approximate Method of State Estimation for Nonlinear Dynamical Systems[J]. International Journal of Control. 1970, 11(6): 957-972.

- [38] R. S. Bucy, K. D. Renne. Digital Synthesis of Nonlinear Filter[J]. Automatica. 1971, 7(3):287-289.
- [39] Smith R, Cheeseman P. On the representation and estimation of spatial uncertainty[J]. International Journal of Robotics Research, 1987, 5(4):56-68.
- [40] FA Cheein, G Steiner, GP Paina *et al.* Optimized EIF-SLAM algorithm for precision agriculture mapping based on stems detection[J]. Computers & Electronics in Agriculture, 2011, 78(2):195-207.
- [41] Sebastian Thrun, Yufeng Liu, Daphne Koller, *et al.* Simultaneous Localization and Mapping With Sparse Extended Information Filters[J]. International Journal of Robotics Research, 2004, 23(7-8): 693-716.
- [42] Matthew R. Walter, Ryan M. Eustice, John J. Leonard. Eustice RM and Leonard J J. Exactly Sparse Extended Information Filters for Feature-Based SLAM[J]. International Journal of Robotics Research, 2007, 26 (4): 335-359.
- [43] JUDEA PEARL. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference[J]. Artificial Intelligence. 1991, 48(1): 117-124.
- [44] DAVISON A J, REID I D, MOLTON N D., *et al.* Mono SLAM: real-time single camera slam[J]. IEEE Trans. Pattern Analysis and Machine Intelligence, 2007, 29(6):1052-1067.
- [45] HARTLEY R, ZISSERMAN A. Multiple view geometry in computer vision[M]. Cambridge University Press, 2000.
- [46] DG Lowe. Distinctive Image Features from Scale-Invariant Keypoints[J]. International Journal of Computer Vision, 2004, 60(2):91-110.
- [47] H Bay, A Ess, T Tuytelaars, L Van Gool. Speeded-Up Robust Features (SURF)[J]. Computer Vision & Image Understanding, 2008, 110(3):346-359.
- [48] Rublee Ethan, Rabaud Vincent, Konolige Kurt *et al.* ORB: an efficient alternative to SIFT or SURF[J]. IEEE International Conference on Computer Vision, 2011(2):1-8.
- [49] M Deans and M Hebert. Experimental Comparison of Techniques for Localization and Mapping using a Bearings Only Sensor[C]. Honolulu: 7th International Symposium on Experimental Robotics, 2000.
- [50] Davison A J. Real-time simultaneous localization and mapping with a single camera[C]. Washington: ICCV '03 Proceedings of the Ninth IEEE International Conference on Computer Vision, 2003.
- [51] N Kwok, G Dissanayake. An Efficient Multiple Hypothesis Filter for Bearing Only SLAM[J]. IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004(4), 736-741.
- [52] 曹乐, 王朝英, 孔云波等. 基于信息熵的多无源传感器数据关联[J]. 传感器与微系统, 2015, 34(11):33-37.

在 学 研 究 成 果

发表的论文:

[1]郭润, 石守东. 基于熵规则的信息滤波 SLAM 算法研究[J]. 传感器与微系统, 2016(12):132-136。

[2]郭润, 石守东. 一种改进状态向量均值恢复的信息滤波 SLAM 算法[J]. 无线通信技术, 2016, 12(4):31-35。

致 谢

光阴似箭，日月如梭，在不知不觉中三年的硕士研究生生涯即将结束，心中难免有许多的不舍，不舍在母校优秀的学习以及科研氛围中获取知识，掌握能力的那种畅然惬意，不舍与尊敬的师长们以及可爱的同学们的各种欢声笑语；不舍在遇到困难之时，正是在你们的帮助和指导之下，我才能顺利克服困难、解决疑惑的感激之情。

首先，要感谢我的导师石守东副教授，他不管是在生活上还是学业上都对我们关怀备至。在学业上，从最初论文选题、商讨到最后论文的完成，石守东老师都付出了大量的心血。在生活上，他就是一位和蔼可敬的长辈对晚辈的那种无微不至的关心和照顾。不仅如此，石老师对工作以及科研的那种一丝不苟、精益求精的态度是非常值得尊敬和学习的。至此，向石守东老师表达我最诚挚的感谢和敬意。

其次，要感谢陪伴我度过难忘的研究生生涯的实验室小伙伴们，我的实验室同学熊泽，刘大军，孙书单，池萧，经贞，容钰，有你们的陪伴，我的生活中增添了更多的欢声笑语，有你们的陪伴，我的学习之路才不会感觉到孤独乏味；感谢这一路上有你们陪伴左右。

最后，我要特别感谢的是我的父母。感谢他们在我求学路上无微不至的关爱和支持，他们对我无私的付出是我不断前行的动力源，没有他们做我坚强的后盾，我是难以完成自己深造之路。

再次感谢所有帮助和关心过我的人，谢谢你们！