



华南理工大学

South China University of Technology

专业学位硕士学位论文

基于点线特征的双目视觉惯性 SLAM 系统研究

作者姓名	韩世豪
学位类别	工程硕士
指导教师	裴海龙 教授
	吴玲玲 高级工程师
所在学院	自动化科学与工程学院
论文提交日期	2020 年 5 月

Research on Visual-Inertial SLAM System Based on Point and Line Features

A Dissertation Submitted for the Degree of Master

Candidate: Han Shihao

Supervisor: Prof. Pei Hailong

South China University of Technology

Guangzhou, China

分类号：TP273

学校代号：10561

学 号：201721017050

华南理工大学硕士学位论文

基于点线特征的双目视觉惯性 SLAM 系统研究

作者姓名： 韩世豪

指导教师姓名、职称：裴海龙 教授

申请学位级别：工程硕士

学科专业名称：控制工程

研究方向：无人机视觉惯性导航

论文提交日期：2020 年 5 月 7 日

论文答辩日期：2020 年 6 月 10 日

学位授予单位： 华南理工大学

学位授予日期： 年 月 日

答辩委员会成员：

主席： 谢巍

委员： 裴海龙、黄战

摘 要

同步定位与地图构建(Simultaneous Localization and Mapping)是计算机视觉和机器人导航领域中的核心技术。SLAM 技术在机器人领域中被广泛应用,其目的是机器人在没有先验知识的情况下,能够根据传感器数据实时估计自身位置,同时构建环境地图。基于点特征的 SLAM 系统最近成为人们研究的重心,但在结构化环境中,线特征和点特征一样的丰富,本文提出基于点线特征的双目视觉惯性 SLAM 系统,目的在于融合点特征、线特征与 IMU 信息提高机器人的定位精度和鲁棒性。本文的主要研究内容如下:

首先,本文搭建了双目 IMU 视觉惯性模组。根据 SLAM 硬件平台的实时性与同步性要求,从元件选型、电路图设计到整体框架对模组做一个整体介绍,同时对双目 IMU 视觉惯性模组的 FPGA 驱动设计和 IMU 数据采集驱动设计进行介绍,然后进行模组性能测试。

其次,本文对线特征的提取、匹配与参数化方法进行研究。其中详细介绍了使用 LSD 方法提取图像中线段的原理和 LBD 描述子的生成方法,然后仔细分析了空间直线的普吕克表示方法以及正交表示法的原理、优缺点以及应用场景。

然后,本文提出了基于点线特征的视觉惯性紧耦合优化算法。基于滑动窗口优化理论,提出了一种同时融合双目视觉约束(点特征和线特征)和 IMU 约束的方法,对优化目标函数即先验残差、IMU 残差、点特征观测残差和线特征观测残差进行了详细推导,同时本文提出了一种关键帧筛选机制,分析了滑动窗口优化算法和边缘化原理。

最后,本文搭建了基于点线特征的双目视觉惯性 SLAM 系统进行实验,包括硬件平台搭建和软件系统。在 EuRoC 数据集上与其他几种主流算法进行实验对比,验证了本文提出的基于点线特征的双目视觉惯性 SLAM 系统具有较高的定位精度和鲁棒性,同时在室外场景下进行实验测试,验证了本文搭建的硬件平台具有较高的稳定性与可靠性。

关键词: 双目 IMU 惯性模组; 点线特征; VIO; 紧耦合优化; SLAM;

Abstract

Simultaneous Localization and Mapping is the core technology in the field of computer vision and robot navigation. SLAM technology is widely used in the field of robots. Its purpose is that robots can estimate their position in real time based on sensor data without the prior knowledge, and at the same time build an environment map. The SLAM system based on point features has recently become the focus of people's research, but in a structured environment, line features are as rich as point features. This paper proposes a binocular visual inertial SLAM system based on point and line features, the purpose is to fuse point features, lines Features and IMU information improve robot positioning accuracy and robustness. The main research contents of this article are as follows:

Firstly, this article builds a binocular IMU visual inertial module. According to the real-time and synchronization requirements of the SLAM hardware platform, the module is introduced from the component selection, circuit design to the overall framework, and the FPGA driver design and IMU data acquisition driver design of the binocular IMU visual inertial module are carried out Introduction, and then perform module performance testing.

Secondly, this paper studies the extraction, matching and parameterization of line features. It introduces in detail the principle of using LSD method to extract line segments in the image and the method of generating LBD descriptors, and then carefully analyzes the principle, advantages and disadvantages and application scenarios of the Pluck representation method and orthogonal representation method of spatial straight lines.

Then, this paper proposes a visual inertial tight coupling optimization algorithm based on point-line features. Based on the theory of sliding window optimization, a method of simultaneously combining binocular visual constraints (point features and line features) and IMU constraints is proposed. The optimization objective functions are priori residual, IMU residual, point feature observation residual and line The feature observation residuals are deduced in detail. At the same time, this paper proposes a key frame selection mechanism, and analyzes the sliding window optimization algorithm and the principle of marginalization.

Finally, this paper builds a binocular visual inertial SLAM system based on point-line

features for experiments, including hardware platform construction and software system. Experimental comparison with several other mainstream algorithms on the EuRoC data set verifies that the binocular visual inertial SLAM system based on point and line features proposed in this paper has high positioning accuracy and robustness, and experiments are conducted in outdoor scenarios , Verified that the hardware platform built in this paper has high stability and reliability.

Keywords: Binocular IMU inertial module; line characteristics; VIO; tightly coupled optimization; SLAM;

目 录

摘 要	I
ABSTRACT	II
目 录	IV
第一章 绪论	1
1.1 研究背景与意义	1
1.2 研究现状	3
1.2.1 基于滤波器的 SLAM 方法	4
1.2.2 基于关键帧优化的 SLAM 方法	5
1.2.3 基于直接法的 SLAM 方法	6
1.2.4 基于点线特征的 SLAM 方法	7
1.3 本文研究内容与结构框架	8
1.4 课题来源	9
第二章 双目 IMU 视觉惯性模组设计与实现	10
2.1 双目 IMU 视觉惯性模组硬件选型与设计	10
2.1.1 硬件选型	10
2.1.2 硬件框架	15
2.2 双目 IMU 视觉惯性模组驱动设计	16
2.2.1 FPGA 驱动设计	16
2.2.2 图像 IMU 数据采集驱动设计	18
2.3 相机采集实验	21
2.4 本章小结	23
第三章 SLAM 中的线特征	24
3.1 线特征的提取	24
3.1.1 LSD 线特征	24
3.1.2 LSD 改进	28
3.2 线特征的描述	28
3.3 空间直线的参数化	31
3.4 本章小结	33
第四章 基于点线特征的双目视觉 IMU 紧耦合优化	34
4.1 滑动窗口优化与边缘化	34

4.1.1 滑动窗口优化.....	34
4.1.2 边缘化与关键帧的选取.....	36
4.2 IMU 观测残差	38
4.3 点特征观测残差	43
4.3.1 两帧间同相机的点特征观测残差.....	43
4.3.2 两帧间两相机的点特征观测残差.....	44
4.3.3 一帧间两相机的点特征观测残差.....	46
4.4 线特征观测残差	47
4.4.1 线特征的观测模型和误差.....	47
4.4.2 线特征的误差雅可比推导.....	49
4.5 本章小结	51
第五章 实验与分析	52
5.1 基于点线特征的双目视觉惯性 SLAM 系统搭建	52
5.1.1 硬件平台.....	52
5.1.2 软件系统.....	53
5.2 基于 EuRoC 数据集的实验.....	56
5.3 基于室外环境的实验	61
5.4 本章小结	64
总结与展望	65
参考文献.....	67
攻读硕士学位期间取得的研究成果.....	71
致谢	72

第一章 绪论

1.1 研究背景与意义

近些年来，SLAM 在机器人导航和计算机视觉领域中已经成为一个核心技术，在飞行器导航，室内机器人避障，仓库 AGV 小车定位导航，扫地机器人，自动驾驶导航，AR,VR 等领域中有越来越多的应用。如图 1-1 所示，Geek+ 的 AGV 小车装有激光 SLAM 导航系统，能够不需要人工的干预就能沿预定轨迹行驶，将负载的货物自动从起点送到终点。小米的扫地机器人装有激光 SLAM 导航系统，能够自动扫描重建房间地图，凭借先验地图进行实时定位与建图完成地板清理工作。谷歌的 Waymo 自动驾驶汽车具有多种传感器，包括激光、雷达和视觉传感器，能够自主进行感知，决策规划，控制避障。微软的 HoloLens 眼镜通过 IMU，RGB-D 相机等传感器实现了沉浸式的交互体验，能够将虚拟世界与真实环境相融合，使得用户感受到多元化的体验。



a) AGV 小车



b) 扫地机器人



c) 自动驾驶



d) HoloLens 眼镜

图 1-1 SLAM 的应用

SLAM 是 Simultaneous Localization and Mapping 的缩写，翻译为同步定位与地图构建^[1,2]，即机器人装有视觉、激光或 IMU 传感器，在没有环境的先验信息情况下，既可以重建周围环境模型，同时又能估计机器人自身的运动轨迹。在进入到一个陌生的环境

中,机器人首先需要解决两个问题:“我在什么地方?”、“我周围的环境是怎么样的?”,这两个问题分别是定位与建图的问题^[3]。要解决这两个问题,需要确定传感器的类型,一种是机器人携带的传感器,例如编码器,激光,相机,IMU等,另一种是安装在环境中的,比如二维码, EuRoC 数据集的维肯动作捕捉系统等。当我们无法对环境提出要求时,那些机器人携带的传感器,比如轮式编码器,激光,相机,IMU 具有更明显的用处,使得这类传感器更加适合用于未知环境。其中激光传感器由于其测量精度高,测量距离远而被自动驾驶广泛使用,同样由于其价格偏高,功率较高无法在嵌入式平台使用。相机传感器由于其采集到的图像特征丰富,且价格低,功率低,体积小,得到了广泛使用,随着 CMOS 技术的日益提升,关注视觉 SLAM 的研究者逐渐变多。

根据相机的工作方式,可以把相机分为三类:单目,双目和 RGB-D 相机,如图 1-2 所示,此外还有 Event 相机,全景相机等新兴相机,本文不做介绍。其中单目相机结构简单,价格低廉,所以单目相机很容易受到研究者的关注,但是单目 SLAM 系统具有尺度不确定性,单目 SLAM 系统估计的轨迹与真实地图之间相差一个真实尺度因子。双目相机由于基线是已知的,所以可以根据双目的基线估计空间中物体的深度,也就消除了尺度不确定性。RGB-D 相机也叫深度相机,通过主动向外界发射光同时接收反射回来的光进而测量环境的深度,比较常用的 RGB-D 相机有 Kinect, Realsense 等。RGB-D 相机由于其测量的深度不远,噪声大,容易受到太阳光的干扰等特性,一般用于室内。

由于摄像头在快速运动场景下、强阳光下、黑暗环境中无法获得稳定可靠的图像,而 IMU 在一段时间的工作后数据会发生偏移,即系统使用单一的传感器往往会存在缺陷。因此 VIO 系统充分利用了相机和 IMU 两种传感器的信息,当摄像头在快速运动场景下,摄像头无法获得准确清晰的图像,但是 IMU 可以在短时间内提供准确的加速度和角速度信息。而当长时间工作后,IMU 的偏置发生改变,数据会累计漂移,此时通过图像计算的姿态不会发生漂移,从而实现了更高的精度,更好的效果。

经过最近几年研究者们对 VIO 系统的持续关注,使得 VIO 技术已经日渐成熟,但是仍然有一些棘手问题待解决,在低纹理、无纹理环境中如何进行精确定位,在弱光照场景下怎么进行精确定位,结合深度学习构建语义地图等^[4]。本文提出基于点线特征的双目视觉惯性 SLAM 系统,对以上情况进行研究。



图 1-2 单目、双目和 RGB-D 相机

1.2 研究现状

SLAM 技术最早由 Cheeseman 和 SmithSelf 于 1986 年提出^[5],他们使用扩展卡尔曼滤波器对机器人的位姿和特征点的深度进行估计,将概率统计学引入到机器人定位与建图中,被认为是 SLAM 技术中的关键。此前 SLAM 早期的传感器主要是里程计,超声波和激光传感器,在基于概率统计方法的定位导航技术开始快速崛起后,视觉 SLAM 技术开始了快速发展,这一时期被称为 SLAM 的“经典时期^[6]”(1986-2004),经典时期见证了基于贝叶斯最大后验概率的 SLAM 解释,涌现出了多种基于概率统计学的方法,包括基于扩展卡尔曼滤波器的方法,基于粒子滤波器的方法等。

随后就是“算法分析时期^[6]”,算法分析时期研究了 SLAM 的本质性质,对 SLAM 系统的可观性,一致性进行了解释,同时分析了矩阵的稀疏性对 SLAM 的求解起到了至关重要的作用,在这一时期也涌现出许多开源库,开源算法。2007 年, Davison 教授提出了第一个可以实时运行在 CPU 上的 SLAM 系统 MonoSLAM^[7,8],其沿用基于扩展卡尔曼滤波器的思路,前端提取了稀疏的角点以保障实时性。同时 Klein 提出了基于非线性优化的 SLAM 系统 PATM^[9],提出了追踪线程和建图线程分开运行的思路,同时在引入了经典的关键帧机制以实现系统的实时运行,同时效率不会有很大的损失。

在此之后, Raul Mur-Artal 等于 2015 年发表基于特征点法的 ORB-SLAM 系统^[10],

ORB-SLAM 是基于 PTAM 的非线性优化架构，ORB-SLAM 系统在运行速度、跟踪效果和地图重建精度上都超过了 PTAM，具有优异的性能表现。J. Engle 等人于 2014 年提出 LSD-SLAM^[11]，在 SLAM 系统中，LSD-SLAM 算法代表着直接法可以成功应用在 CPU 上实现单目半稠密地图重建。

1.2.1 基于滤波器的 SLAM 方法

Davison 教授是视觉 SLAM 领域中的先驱者，Davison 在 2007 年提出了第一个可以实时运行在 CPU 上的 SLAM 系统 MonoSLAM。MonoSLAM 的前端是提取和追踪非常稀疏的特征点，后端是扩展卡尔曼滤波器。在 PTAM 未提出前，基于扩展卡尔曼滤波器的 SLAM 系统一直是研究者们的主要方向，MonoSLAM 在扩展卡尔曼滤波器的基础上，将相机当前状态下的姿态和前端提取的所有特征点设为状态变量，进行状态变量的更新和协方差的更新。在 MonoSLAM 之前，SLAM 系统都不能在 CPU 上实时运行，通过机器人离线采集数据，再在服务器上进行离线定位与建图。随着计算机性能的快速进步，加上前端提取稀疏特征点，后端关键帧的选取策略才使得 SLAM 系统在线运行变为可能。MonoSLAM 系统为基于贝叶斯滤波器方法的 SLAM 系统提供了一个通用的框架。

Montemerlo 基于粒子滤波器提出了 FastSLAM^[12,13]，粒子滤波方法是对系统模型进行近似，对状态向量和观测值是否满足高斯分布不做要求。而 SLAM 系统在实际应用中的矩阵维度很高，粒子滤波方法在高维 SLAM 系统中产生的计算量会很大，所以 FastSLAM 对机器人的位姿估计部分使用粒子滤波的方法，而地图中特征点的深度估计使用扩展卡尔曼滤波器的方法。

Mourikis 在 2007 年提出了一种基于卡尔曼滤波的 VIO 算法 MSCKF^[14] (Multi-State Constraint Kalman Filter)，即多状态约束卡尔曼滤波，相比于以前的 EKF 滤波算法将特征点都加入到状态向量中，当机器人进入到一个场景很大的环境中，提取到的特征点会较多，状态向量的维度会变得非常大，进行迭代更新时会非常耗时，MSCKF 会保留最新帧附近几帧的相机位姿和当前帧的 IMU 位姿，不将特征点加入到状态向量中，特征点被相邻几帧图像观测到形成视觉约束，每当特征点不被最新帧观测到的时候就对这个特征点形成的约束构建观测模型进行 EKF 更新。这样 MSCKF 的状态向量维度远远小于传统的 EKF 系统。

MSCKF 的状态向量如图 1-3 所示。MSCKF 在 EKF 的框架下融合了视觉和 IMU 的信息，相对于纯视觉 SLAM 系统，MSCKF 能够在快速运动场景中稳定运行，具有很好的鲁棒性，相较于目前主流的基于优化的 VIO 算法，MSCKF 的运行速度更快，精度相当，可以在嵌入式平台中实时运行。

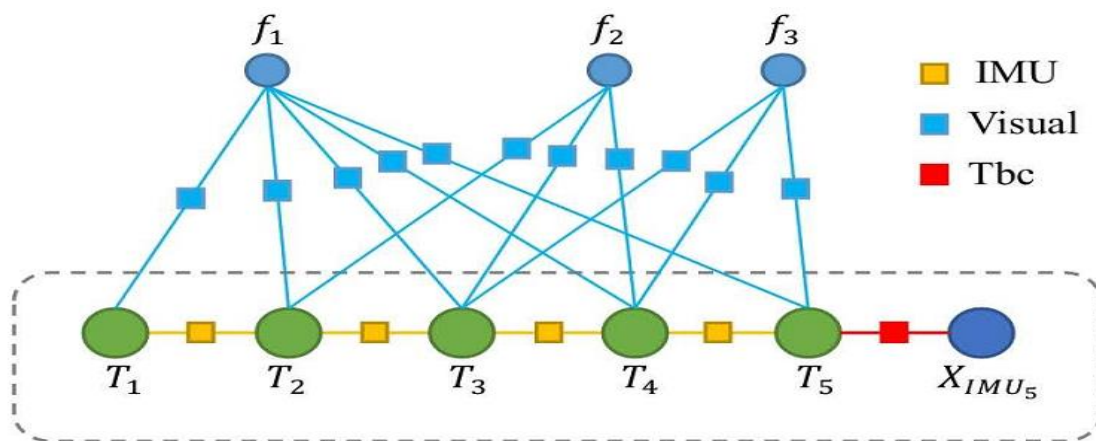


图 1-3 MSCKF 状态向量

1.2.2 基于关键帧优化的 SLAM 方法

在 Davison 教授发布 MonoSLAM 算法的同时，Georg Klein 和 David Murray 提出了基于非线性优化的 PTAM 系统。PTAM 提出了 SLAM 系统可以分成跟踪线程和建图线程两个线程运行，跟踪线程需要实时的追踪图像特征点，但是建图过程没必要进行实时响应，可以在追踪线程完成后将数据送到建图线程再运行^[15]。PTAM 提出 SLAM 系统可以分为跟踪和建图两个线程方法，同时引入了关键帧机制，为以后的基于优化的 SLAM 系统奠定了基础。在 PTAM 系统出现之前，研究员们着重研究基于滤波的 SLAM 系统，早期的 SLAM 系统基本上都是基于扩展卡尔曼滤波器或者粒子滤波器等等，在 PTAM 出现之后，研究员们意识到在后端非线性优化时，矩阵具有稀疏性，于是纷纷开始研究基于非线性优化的视觉 SLAM 系统。同时 PTAM 系统中结合了 AR 的效果，如图 1-4 所示。

其后，基于非线性优化的 SLAM 系统开始了快速发展，在 2015 年，Raul Mur-Artal 等提出了 ORB-SLAM 系统，代表了目前基于特征点法的 SLAM 系统的较高水平。ORB-SLAM 具有单目、双目、RGB-D 三种模式，具有广泛的通用性，系统使用了 Tracking 线程，局部地图 BA 优化线程和回环检测与优化线程三个线程来设计 SLAM 系统。其中

Tracking 线程在每一帧图像上提取 ORB 角点，将这些新提取的角点与相邻关键帧之间进行对比，然后预估新来的一帧图像的初始位姿。局部地图 BA 优化线程保存了最新帧附近几帧的特征点和相机位姿，在本线程内，会对局部地图内的特征点和相机位姿进一步优化，在回环检测线程内保存了全局地图中的关键帧，会对每一个新来的帧进行回环检测，进行全局位姿图优化，避免产生累计误差^[15]。

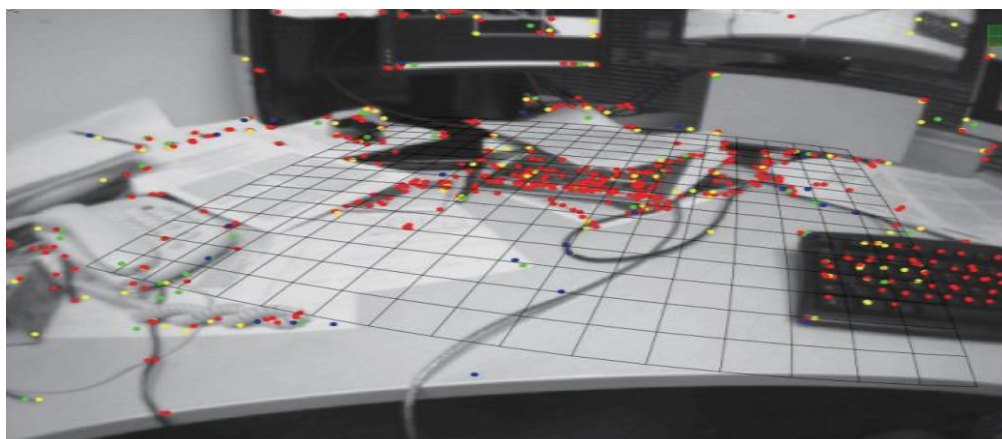


图 1-4 PTAM 的 AR 效果

1.2.3 基于直接法的 SLAM 方法

在低纹理或者黑暗环境等特征缺失的环境下，例如楼道、空旷的房间，特征点的提取较为困难，此时使用基于点特征的 SLAM 系统估计相机位姿，会因为特征点数量不足而估计失败，此时可以假设空间中一个特征点在各个视角下，特征点的灰度值是不变的，此时通过最小化相邻帧中像素点的光度误差来估计相机的位姿，同时直接法可以构建出稠密地图。直接法分为稀疏直接法，半稠密直接法和稠密直接法，稀疏直接法中的像素点来自稀疏关键点，并且也假设这些关键点周围像素是不变的，但是这种方法计算出的地图是稀疏的。半稠密直接法中的像素点是带有梯度的像素点，舍弃掉像素梯度不明显的点，可以构建出一个半稠密地图。稠密直接法中的像素点是图像中的所有像素点，此时需要 GPU 加速来计算，重构出的地图是稠密地图。

直接法的代表就是 J. Engle 等人于 2014 年提出 LSD-SLAM，LSD-SLAM 能够在 CPU 上实时运行，构建出半稠密地图，LSD-SLAM 系统没有提取特征点，计算描述子，仅仅通过优化带有梯度的像素点的光度误差就能估计相机的位姿，并且能构建半稠密地图，LSD-SLAM 依赖于直接法的优点，能够在纹理特征不明显的场景下正常运行。但是

LSD-SLAM 对相机的内参和曝光方式很敏感,当相机开启自动曝光时或者相机快速运动时,系统很容易跟丢。

SVO 是 Forster 在 14 年提出的一种基于稀疏直接法 SLAM 系统^[16], SVO 提取一些关键点,然后根据角点周围的像素信息,估计相机的位姿。SVO 由于不用计算描述子,提取的关键点又稀疏,所以具有很高的实时性,在 PC 上可以达到 400 帧/秒,所以 SVO 非常适合在嵌入式设备上运行,例如无人机,AR,VR 设备。SVO 提出了深度滤波器的概念,将深度滤波器用于估计特征点的位置,并且使用了逆深度,使得特征点的估计更加准确。

1.2.4 基于点线特征的 SLAM 方法

近些年来,在 VIO 领域涌现出许多性能优越的开源算法,例如 MSCKF, VINS-MONO^[17], OKVIS^[18]等,但是这些 VIO 系统只是用了点特征作为视觉约束。在低纹理、光线较弱场景下,比如走廊,空旷房间,图像中提取到的特征点少之又少,系统在这些环境下估计相机的位姿具有挑战性。基于直接法的 SLAM 系统对周围环境的纹理信息不敏感,可以在上述场景中进行位姿估计,但是在基于直接法的 SLAM 系统对光照变化很敏感,当使用自动曝光相机时,基于直接法的 SLAM 系统会很容易追踪丢失。此时,线特征相比于点特征在结构化场景中更容易提取,并且由于线特征具有光照不变性,和视角不变性,使得线特征更加直观。

最近研究者们逐渐关注了基于点线特征的视觉 SLAM 系统,J.Neira 最早在 1997 年提出了基于竖直线段的单目视觉 SLAM 系统^[19],由于早期的研究者们主要研究基于扩展卡尔曼滤波器的 SLAM 系统,J.Neira 基于扩展卡尔曼的框架搭建了这个 SLAM 系统。直线一般由两个空间点表示,同样也可以用线段无限延长表示,在相机运动幅度较大后,可能并不能完整的看到上一帧观测到的线段的两个端点,所以 Joan Sola 总结了直线的参数化方法^[20],同时 Bartoli 根据直线的 4 自由度,提出了直线的正交表示法^[21]。Strasdat 分别基于滤波和优化分析了线特征对系统的精度影响^[22],在滤波算法中,增加线特征的个数可以提高系统的精度,而在基于优化的方法中,系统的计算代价和一致性都比基于滤波的方法要高。在基于优化的 PTAM 系统出现之前,研究者们的主要方向都是基于滤波器的方法,其后,Georg Klein 融合了点线特征构建了基于优化的 SLAM 系统,大大

提高了系统的鲁棒性^[23]。Yan Lu 融合了点线特征在 RGB-D 相机基础上构建了 SLAM 系统，给出了点线特征融合的方法能降低位姿估计的不确定度^[24]。

1.3 本文研究内容与结构框架

本文搭建了基于点线特征的双目视觉惯性 SLAM 系统，在 VINS-Fusion 和 PL-VIO^[24] 的工作上进行扩展优化，通过在后端中联合优化点特征，线特征和 IMU 三者的约束进行相机位姿估计，用以研究结构化环境中点特征、线特征和 IMU 信息融合对 SLAM 系统定位精度的影响。

根据本文的研究内容，本文的具体结构安排如下：

第一章主要介绍了 SLAM 系统的研究背景与研究意义，概述了视觉 SLAM 的发展历程，详细介绍了视觉 SLAM 的研究现状，并介绍本文研究内容与结构框架。

第二章主要介绍了双目视觉惯性模组的设计与具体实现。其中具体描述了双目惯性模组的性能要求，对双目视觉惯性模组的硬件选型与设计及整体框架到双目视觉惯性模组的硬件驱动设计做了整体详细介绍，最后对双目 IMU 视觉惯性模组的性能进行测试，验证了双目 IMU 视觉惯性模组具有较高的时间同步性与成像稳定性。

第三章主要介绍了 SLAM 中的线特征，对图像中线段的提取方法，描述子的生成方法和空间直线的参数化方法进行了详细介绍，其中分析了使用 LSD 方法提取直线的步骤和 LBD 描述子的生成原理，其后仔细分析了空间直线的普吕克表示方法以及正交表示法的原理和优缺点。

第四章本章主要介绍了基于点线特征的双目视觉 IMU 紧耦合优化。分析了滑动窗口优化算法、边缘化与关键帧筛选策略，IMU 观测残差，两帧间同相机的点特征观测残差，两帧间两相机的点特征观测残差，一帧间两相机的点特征观测残差和线特征观测残差原理。其中详细介绍了优化目标的构成，优化变量的更新方式，边缘化原理，关键帧的筛选策略，IMU 预积分和 IMU 观测残差的推导，两帧间同相机的点特征重投影误差定义，两帧间两相机的点特征重投影误差定义，一帧间两相机的点特征重投影误差定义，线特征投影图像平面推导、线特征的重投影误差定义与线特征的雅可比推导。

第五章主要介绍了基于点线特征的双目视觉惯性 SLAM 系统的硬件平台搭建和软件系统的组成，为了验证本文提出的基于点线特征的双目视觉惯性 SLAM 算法效果和

硬件平台的性能，分别进行了基于数据集的实验和基于室外环境的实验。在基于数据集的实验中，本文对比了三种主流 VIO 算法，在室外环境的实验中，将飞行器的 RTK 信息作为基准信息，同样对比了三种主流 VIO 算法，表明了本文搭建的基于点线特征的双目 VIO 系统在硬件方面具有很高的可靠性、稳定性，在软件方面具有很高的精度和鲁棒性。

1.4 课题来源

本课题研究来源于华南理工大学自动化科学与工程学院教育部重点实验室裴海龙教授主持的国家自然科学基金重大科研仪器研制项目“海岛海岸带多要素综合飞行测绘仪”和广东省省级科技计划项目“无人系统装备质量安全风险评估与测试技术研究及平台建设”。

项目支持基金：国家自然科学基金重大科研仪器研制项目[615278010]、广东省重大科技专项 [2017B010116005]。

第二章 双目 IMU 视觉惯性模组设计与实现

本章主要讲述双目 IMU 视觉惯性模组的设计要求以及具体实现。本文所设计的双目 IMU 视觉惯性模组用于“海岛海岸带多要素综合飞行测绘仪”项目，项目中需要双目 IMU 视觉惯性模组满足以下性能要求：

（1）同步性

由于项目中飞行器的飞行速度较快，当双目之间的时间同步度较低时，会导致左右摄像头拍摄的图像不同步，可能景物在左摄像头中出现，而在右摄像头中却没有出现，或者出现了一半。此时的双目图像是没有办法用来做深度恢复、SLAM 定位导航或者 SFM 的。市面上常见的双目相机一般是软件时间同步，同步精度较低，无法满足在飞行器上使用的要求，因而本文使用 FPGA 控制器硬件同步触发采集双目图像，实现了双目摄像头之间微秒级的硬件同步，同时根据 CMOS 芯片的 FLASH 曝光信号设计算法实现了双目和 IMU 之间的硬件级同步，满足了项目的实际需求。

（2）高集成度

由于飞行器的体积和载重有限，会对双目 IMU 视觉惯性模组的重量和体积有较大的限制，因此本文设计的双目 IMU 视觉惯性模组选用封装小、精度高、集成度高的传感器芯片、电源芯片、FPGA 芯片和网络芯片，在设计 PCB 时使用 4 层板，在不影响功能的情况下模块之间尽量紧凑。

（3）低功耗

由于是在小型飞行器平台上使用，功耗是一个重要参考因素，因此本文选用 FPGA 作为主控制器，一方面考虑到 FPGA 的时序稳定性高，另一方面考虑到 FPGA 低功耗。本文选用 EP4CE6E22C8N 作为双目 IMU 视觉惯性模组的主控制器，EP4CE6E22C8N 是低成本，低功耗 FPGA 架构，6K 至 150K 逻辑单元，足够满足系统要求，每通道功耗 150mW 左右。

2.1 双目 IMU 视觉惯性模组硬件选型与设计

2.1.1 硬件选型

本文综合上述性能要求，对双目 IMU 视觉惯性模组的硬件选型如下：

(1) 摄像头

AR0144 图像传感器是安森美半导体推出的全局快门 CMOS 芯片, AR0144 的像素尺寸为 $3\mu\text{m}$, 靶面面积为 $1/4$ 英寸。它采用了一种创新的全局快门像素设计, 该设计经过优化, 可以准确, 快速地捕获运动场景。AR0144 能够输出清晰的图像, 并且它具有多种触发模式, 可以捕获连续帧或者单帧图像, 使得 AR0144 成为工业检测, 机器人领域的最佳选择。AR0144 有三种触发模式: Master, Automatic Trigger, Pulsed Trigger, 两种输出方式: 并行 DVP, 串行 HISPI。本文使用 Automatic Trigger 的触发方式, 并行 DVP 的输出方式, 并且将 AR0144 的分辨率配置为 1280×720 , 输入时钟为 24MHz , 像素时钟为 74.25MHz 。

AR0144 的 Automatic Trigger 模式时序如图 2-1 所示, 图像一行输出 1280 个像素, 将 LINE_LENGTH_PCK 的寄存器数值要大于 1280 , 那么每行图像的输出时间为 $18.69\mu\text{s}$, 那么整幅图像的输出时间为 17.9ms , 图像帧率配置为 50Hz , 那么平均一帧图像需要时间 20ms , 这时每一帧图像会多出 2.054ms 的时间, 这部分帧间的时间即是 Vertical Blanking 时间由 FRAME_LENGTH_LINES 和 Y_ADDR_START, Y_ADDR_END 寄存器控制, 同时曝光时间由 COARSE_INTEGRATION_TIME 和 FINE_INTEGRATION_TIME 两个寄存器控制, 曝光时间不宜过大, 曝光时间越长, 高速运动物体拖影越长, 由图 2-1 可以看出在 AR0144 的 Trigger 信号开始上拉触发时, 图像的帧信号, 行信号和像素时钟开始输出, 此时将两个摄像头的 Trigger 信号同步触发即可实现摄像头间时间同步。

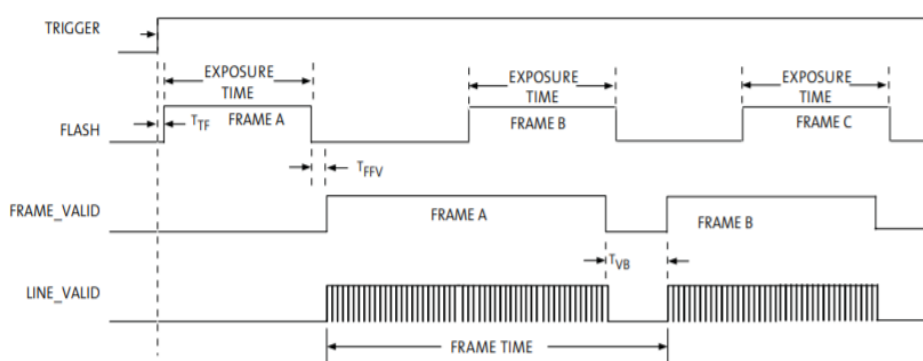


图 2-1 摄像头时序图

AR0144 摄像头实物图与电路图如图 2-2 所示。使用 3.3V 电源输入, 转换为 2.8V 电压供给 AR0144 的 VDDIO 口, 转换为 1.8V 电压供给 AR0144 的 VDD 口, 单独转换一路 2.8V 电压供给 AR0144 的 VAA 口。本文将 AR0144 的曝光方式配置为自动全

局曝光模式，相机触发模式配置为自动触发模式，相机分辨率配置为 1280*720，相机镜头焦距为 2.6mm，视场角宽为 95 度，高为 50 度，双目基线距离为 12cm。

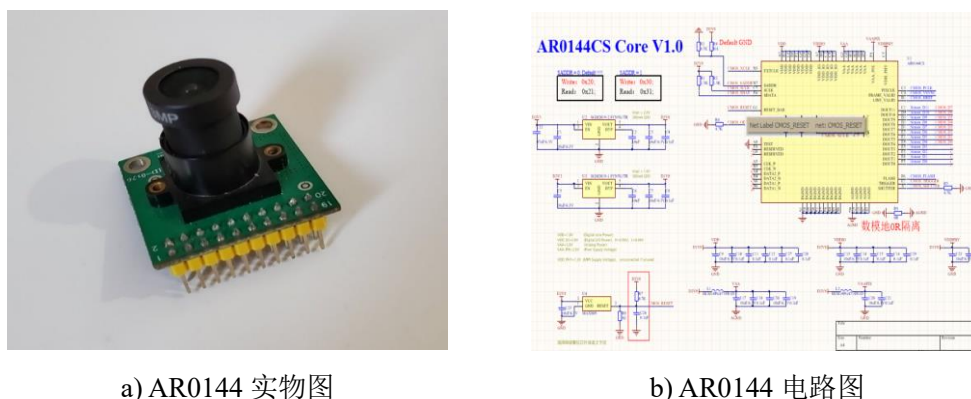


图 2-2 AR0144 摄像头实物图与电路图

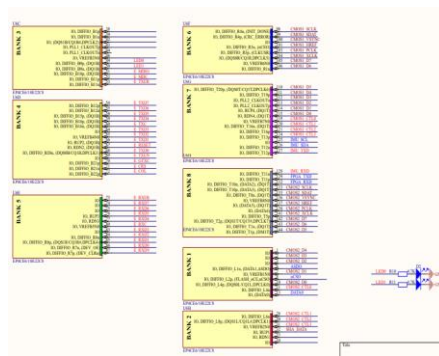
(2) 相机主控

本文选用 FPGA 作为双目 IMU 相机的主控。FPGA 是 Field Programmable Gate Array 的简称，中文为现场可编程门阵列。FPGA 是在通用阵列逻辑、可编程逻辑阵列、复杂可编程逻辑器件等传统逻辑门电路基础上发展起来的一种现场可编程阵列。FPGA 的应用场景主要在 ASIC 领域，能够克服传统可编程器件门电路资源有限的缺点，FPGA 由可配置逻辑模块、输入输出模块和内部连线三个模块组成，具有和复杂可编程逻辑器件、可编程逻辑阵列、通用阵列逻辑不同的电路结构。

FPGA 可以为 ASIC 电路设计提供样片，FPGA 拥有大量的触发器和 I/O 管脚，用户不需要去工厂流片就能得到设计的芯片，并且具有产品质量稳定、投入费用低、设计周期短、生产风险小等优点。本文使用的 FPGA 型号是 Altera EP4CE6E22C8N，EP4CE6E22C8N 是一款 Cyclone IV E FPGA，其速度等级为 8 级，144 引脚 QFP 封装，该芯片具有 392 逻辑阵列块 (LAB)，6272 个逻辑单元 (LE)，RAM 总量 276480bit，电源电压范围: 1.15V 至 1.25V，运行时温度范围 0° C 至 85° C，嵌入式存储器为 270K 字节，PLL 时钟有两路，用户 I/O 多达 179 个，每通道的功耗仅为 150mW。EP4CE6E22C8N 相机主控实物图与电路图如图 2-3 所示。



a) FPGA 实物图



b) FPGA 电路图

图 2-3 Altera EP4CE6E22C8N 实物图与电路图

(3) 惯性测量单元 IMU

由于相机在飞行器上不可避免会有抖动、晃动的情况，即是相机在安装时加上减震片，也不可避免在某些时候照片会模糊，这时我们引入惯性测量单元 IMU 以获得更多的姿态信息从而提高这种情况下 SLAM 算法的精度。惯性测量单元 IMU 包含了三轴陀螺仪和三轴加速度计，三轴陀螺仪测量相机运动的角速度信号，三轴加速度计测量相机运动的加速度信号，但是没有办法准确测出航向角，所以有的 IMU 会加装三轴磁力计，通过地球磁场方向去计算相机的航向角。

本文使用的 IMU 是博世公司的 BMI160，BMI160 将 3 轴低重力加速度计和超低功耗 3 轴陀螺仪集封装成一体，具有功耗低、延时低和精准度高的优点。BMI160 采用 14 管脚 LGA 封装，尺寸仅为 $2.5 \times 3.0 \times 0.8 \text{ mm}^3$ ，是目前体积最小的 6 轴 MEMS IMU，与 BMI055 相比，BMI160 的体积减小了 60%，这是由于博世公司的 MEMS 陀螺仪采用了新的设计工艺，使用一颗 ASIC 芯片控制加速度计和陀螺仪。BMI160 在测量加速度和陀螺仪时，耗电低至 $950\mu\text{A}$ ，耗电量是 BMI055 的 50%，满足了系统低功耗的要求。BMI160 可以通过寄存器直接读取加速度和陀螺仪数值，内部装有 LDO 电源管理芯片，电压范围为 $3.2\sim 6\text{V}$ ，并且 BMI160 装有 I2C 电平转换电路，输入电压兼容 3.3V 和 5V 。惯性测量单元 IMU 实物图与电路图如图 2-4 所示。

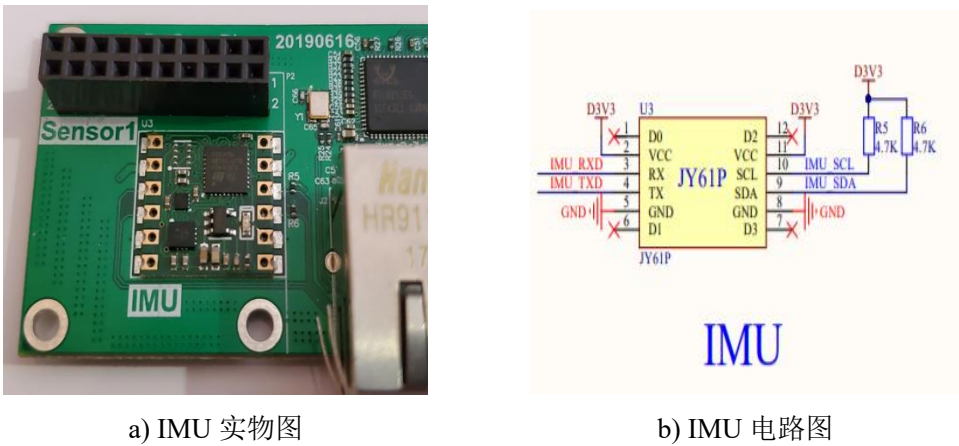


图 2-4 惯性测量单元 IMU 实物图与电路图

(4) 网口通信模块

以太网的应用十分的广泛，包括家庭宽带、工业通信、监控系统、早期的电话线路、服务器数据交互等。本文使用千兆网口的方案实现 FPGA 与 PC 端的数据传输，该千兆网口模块由千兆以太网 PHY 芯片和网络变压器接口组成。当发送数据时，FPGA 通过 8 根数据线把数据发送给 PHY 芯片，然后 PHY 芯片将 FPGA 发送过来的数据编码后送至网络变压器，网络变压器将数据变成电平信号加载到网线上，数据经过网线传递到接收端。发送端传输的数据经过网线传递至网络变压器，网络变压器收到数据后进行电平转换，将数据送到 PHY 芯片，PHY 芯片对数据进行解码得到真实的数据，然后 PHY 芯片将数据传输给 FPGA 芯片。网口通信模块的功能框图如图 2-5 所示。



图 2-5 RTL8211 系统框图

RTL8211 芯片是以太网物理层收发器，支持 GMII, RGMII, MII 三种接口，支持百兆网口通信或者千兆网口通信，本文将其配置为 GMII 模式。以太网最常用的协议是 TCP/IP 协议，在 PC 端或者嵌入式系统中，TCP/IP 协议的应用十分广泛，但是使用 FPGA

实现 TCP/IP 协议只能基于 SOPC 的方式实现，由于其协议十分复杂，使用 Verilog 编写协议层实现，工程量十分的浩大，本文也基于图像传输的实时性去考虑，使用 UDP 协议传输双目 IMU 图像，并且 UDP 在点对点的情况下传输，基本不会产生丢包的现象。RTL821 千兆网口模块实物图与电路图如图 2-6 所示。

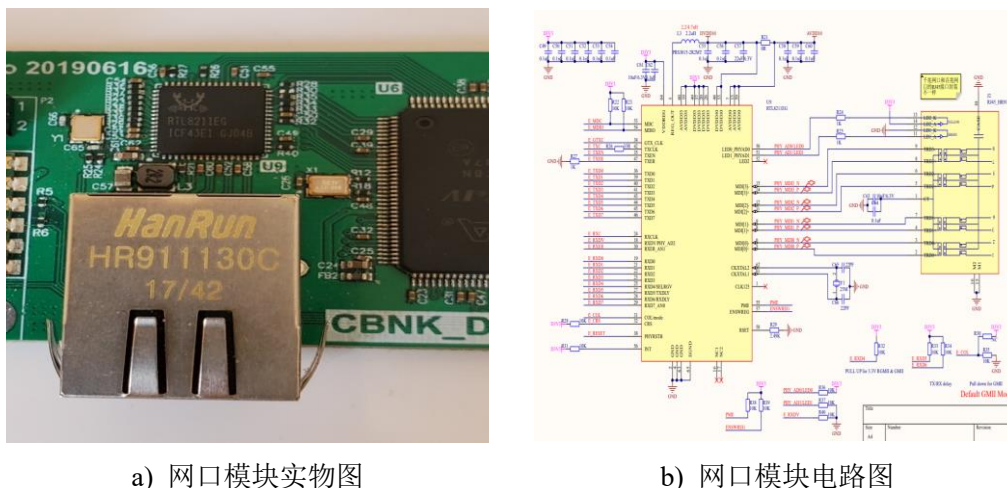


图 2-6 RTL8211 千兆网口模块实物图与电路图

2.1.2 硬件框架

FPGA 通过 I2C 同步初始化两个 AR0144 摄像头的输出模式寄存器，FPGA 然后同步触发采集两个 AR0144 摄像头的图像，同时采集 IMU 数据。FPGA 首先对双目图像数据进行解析，根据行信号、帧信号、像素时钟解析出整幅图像，再通过时间同步模块将相机时间戳和 IMU 时间戳同步，然后将数据缓存在 FPGA 内部的 FIFO 中，最后基于 UDP 协议将双目图像和 IMU 数据传送至 Avnet Ultra96-V2 板中。图像传感器是安森美 AR0144，IMU 是 BMI160，图像处理平台是 Altera 的 EP4CE6E22C8N，网口传输芯片是 RTL8211。双目 IMU 视觉惯性模组如图 2-7 所示。



图 2-7 双目 IMU 视觉惯性模组实物

其中时间同步模块通过相机的 FLASH 的信号对相机和 IMU 数据进行时间同步,同步精度达到 1ms 内,满足了高速运动场景下对时间同步的要求。图像和 IMU 的时间同步算法设计步骤如下,在 AR0144 的 Flash 信号变成高电平的时候,此时代表图像开始曝光,这个时候 FPGA 采集 IMU 的数据。如果采集到偶数个 IMU 数据,取曝光时间采集到的 IMU 数据的中间数据的时间戳作为相机的时间,如果采集到奇数个 IMU 数据,取曝光时间采集到的 IMU 数据的中间两个数据的平均时间戳作为相机的时间。

其中 FIFO 是小容量 8K 字节,FPGA 处理过的图像和 IMU 数据不经过长时间缓存,传输延时很低,FIFO 包括 2 个相机 FIFO 和一个 IMU FIFO。相机 FIFO 的容量不足够缓存整幅图像数据,但是相机 FIFO 的读取时钟为 120MHZ,相机 FIFO 的写入时钟为 74.25MHZ,相机 FIFO 的读取速度几乎是写入速度的两倍,保证了 FIFO 的容量不会被写满,图像能够稳定传输不丢帧。IMU FIFO 缓存 IMU 的数据,三个 FIFO 都连接到双目 IMU 传输 IP 核,然后通过网口模块将图像数据和 IMU 数据传输到 PC 端。双目 IMU 视觉惯性模组的框架如图 2-8 所示。

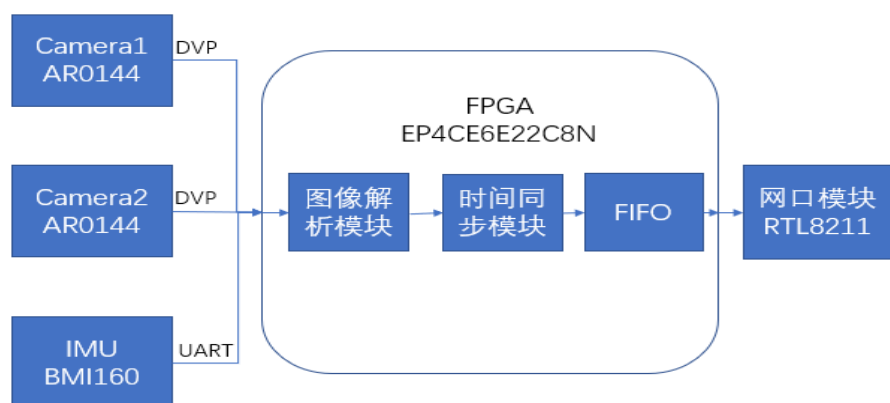


图 2-8 双目 IMU 视觉惯性模组硬件框架

2.2 双目 IMU 视觉惯性模组驱动设计

2.2.1 FPGA 驱动设计

FPGA 通过 I2C 配置 AR0144 的寄存器为 Auto Trigger 模式,同时配置 IMU 寄存器。然后 FPGA 同步触发两个摄像头的 Trigger 信号,使得两个摄像头的帧信号和行信号同步输出。FPGA 首先对数据进行解析,根据行信号、帧信号、像素时钟解析出整幅图像,再通过时间同步模块将相机时间戳和 IMU 时间戳同步,其后 FPGA 同时缓存双目

图像和 IMU 数据到 FIFO 中。最终通过基于 UDP 的双目图像 IMU 传输 IP 核分别传输三个 FIFO 中的图像和 IMU 数据，PC 根据图像协议解析出图像和 IMU 数据。双目 IMU 视觉惯性模组的 FPGA 驱动框架如图 2-9 所示。

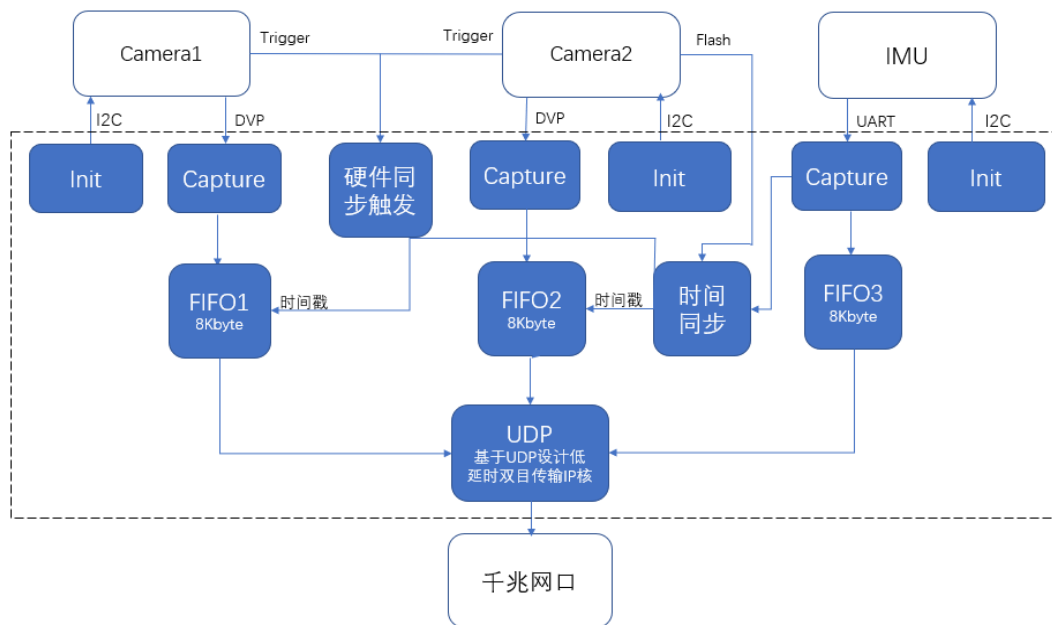


图 2-9 双目 IMU 视觉惯性模组的 FPGA 驱动框架

其中基于 UDP 设计低延时双目 IMU 传输 IP 核，实现图像低延时无损传输，使用 Verilog 语言实现 MAC 层，IP 层，UDP 层，数据层，CRC 校验层，其中数据层的数据根据 3 个 FIFO 中缓存的数据情况灵活切换要发送的数据，设计了快速安全图像传输协议，将双目数据同步低延时传输到 PC 端。其中图像传输协议规定一个包的长度为 1283，第一个数据为相机的型号，第二个数据为该行图像的行数低 8 位，第三个数据为该行图像的行数高 8 位，后 1280 个数据为该行图像数据，通过行数和相机型号解析出双目数据。本文设计的图像传输协议数据稳定传输速度达到 800Mb/s，带宽满足了双目分辨率为 1280*720 的双目图像以 50HZ 帧率高速稳定传输，双目 IMU 传输 IP 核的框架如图 2-10 所示。

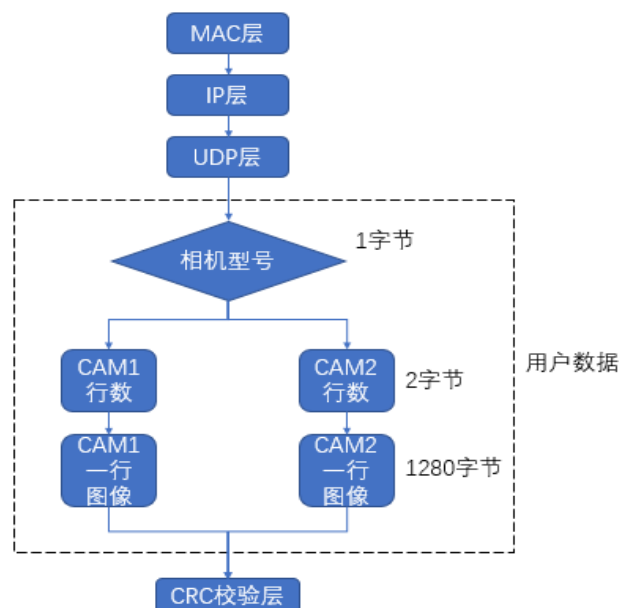


图 2-10 双目 IMU 传输 IP 核框架

2.2.2 图像 IMU 数据采集驱动设计

图像 IMU 数据采集驱动的主要任务是对 FPGA 通过以太网发送过来的数据进行解包，拆分出双目图像数和 IMU 数据。图像 IMU 数据采集驱动设计的关键在于保证数据不丢包，具有实时性、同步性。本文将图像 IMU 数据采集驱动分为三个线程，分别为 UDP 数据采集线程、图像 IMU 数据解析线程和主线程。

（1）UDP 数据采集线程

TCP 协议为了实现网络通信的稳定性，建立了繁琐的握手过程，一旦发现数据发生了丢包，TCP 协议会将后续的数据包缓存起来，等待前面的数据包重传并且接收成功后再会继续发送，所以基于这种机制，TCP 协议的传输延时会越来越大。在对实时性要求较为严格的情况下，基于 UDP 协议能够实现实时传输，同时在点对点的情况下把丢包率降到最低。所以图像 IMU 数据采集驱动为了实时性采用了 UDP 协议，UDP 协议传输数据非常快，在千兆网下传输速度能达到 800M/S，但是没有建立连接，即没有握手的过程，数据包容易丢失，而且没有 TCP 协议那种重传的机制，本文在点对点的情况下，对客户端和服务端进行改进，实现了 800M/S 的情况下不丢包。首先在客户端，本文增加 UDP 每个包之间的发送时间间隔，每个数据包要在上一个包发出去后等待 10ns 再发送下一个。对于服务端，本文单独使用一个线程采集 UDP 数据包，在新线程中使

用 while 死循环，并采用 Socket 默认的阻塞模式接收数据，并且在 Linux 下使用 Linux 的 Socket，摒弃了 Qt 的 QSocket。在收到数据后，本文使用环形缓冲区存储图像和 IMU 数据，在本线程内不对数据处理。UDP 数据采集线程的程序流程图如图 2-11 所示。

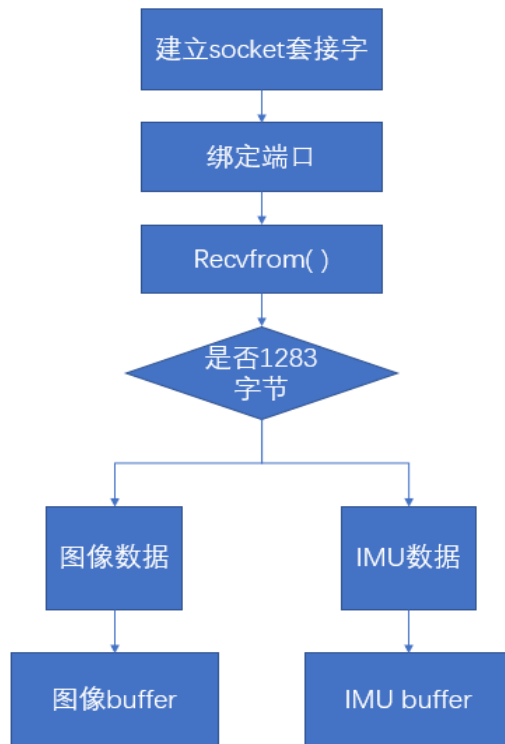


图 2-11 UDP 数据采集线程流程

(2) 图像 IMU 数据解析线程

图像 IMU 数据解析线程从环形缓冲区内读取数据，对于图像数据，图像传输协议规定一个包的长度为 1283，第一个数据为相机的型号，第二个数据为该行图像的行数低 8 位，第三个数据为该行图像的行数高 8 位，后 1280 个数据为该行图像数据，通过行数和相机型号解析出双目数据，对于偶尔的丢包情况可以通过行数来判定，再进一步做丢包处理。加速度计和陀螺仪数据分为高低字节顺序传送，将高低字节结合成一个有符号 short 型数据。例如加速度 X 轴的数据 X 分为两个部分，低字节是 XL，高字节是 XH。将数据 X 转换为加速度数据的方法如下：将高字节 XH 左移 8 位加上低字节 XL，即 $X = (\text{short})(XH \ll 8 | XL)$ ，X 的数据类型为有符号的 short 类型，如此才可表示负数。图像 IMU 数据解析线程程序流程图如图 2-12 所示。

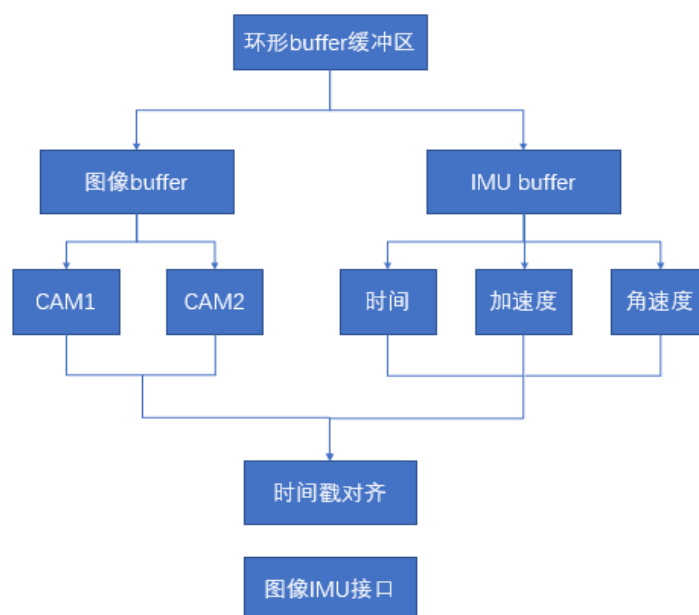


图 2-12 图像 IMU 数据解析线程流程

(3) 主线程

主线程的任务是对各个数据接口进行初始化、开启 UDP 数据采集线程与图像 IMU 数据解析线程、然后对图像 IMU 数据解析线程解析出的图像和 IMU 数据打包成接口。线程之间使用了生产者与消费者模型，UDP 数据采集线程为生产者，图像 IMU 数据解析线程为消费者。在一个系统中，生产者和消费者扮演着两种角色，生产者生产数据，消费者使用数据，生产者与消费者通过缓冲区进行数据通信。具体的说，缓冲区一直不断的存储生产者生产的数据，消费者又不断的从缓冲区中拿取数据，如果生产者生产数据很快，缓冲区容量满了，则生产者线程会阻塞。如果消费者拿取数据很快，缓冲区中没有数据了，则消费者线程会阻塞。生产者与消费者模型的实现有等待唤醒方式和阻塞队列方式，本文选用第一种办法实现生产者与消费者模型。在生产者生产数据的时候，即缓冲区存储数据时唤醒消费者线程，在消费者线程使用完所有的数据后，消费者线程挂起休眠，等待生产者生产数据进行下一次唤醒，生产者与消费者之间使用了互斥锁、挂起与唤醒，保证了线程之间数据交互的安全与效率。本文使用的三个线程中图像 IMU 数据解析线程消耗的资源最多，在不使用生产者消费者模型情况下，CPU 占用率达到单核 100%，使用生产者消费者模型后，CPU 占用率为单核 40%。主线程的程序流程图如图 2-13 所示。

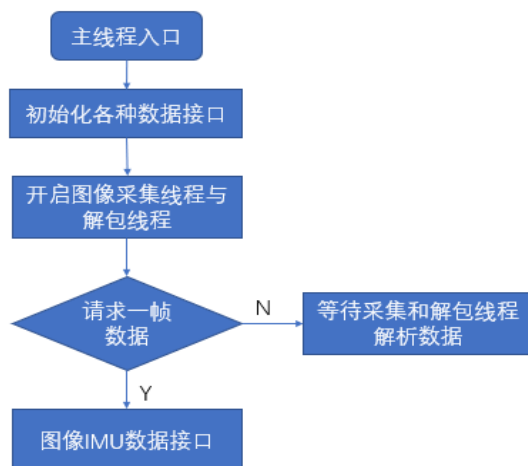


图 2-13 主线程程序流程图

2.3 相机采集实验

(1) 同步性

为了检验设计的双目 IMU 视觉惯性模组性能，本文利用 Altera 的 IDE Quartus II 抓取双目相机的像素时钟、帧信号和行信号，行数，帧信号与行信号波形图（上升沿与下降沿）如图 2-14 所示。其中帧信号和行信号的上升沿与下降沿在 50MHZ 的采样时钟下几乎是同步的，且行数与帧数一致的，没有发生错行的情况，由此可以表明双目 IMU 视觉惯性模组的双目时间同步精度在微秒级。同时使用双目 IMU 视觉惯性模组对秒表进行拍照，相机同步性实验如图 2-15 所示，由相机同步性实验可见双目相机同步精度在 1ms 内，该相机完全满足同步性的要求。

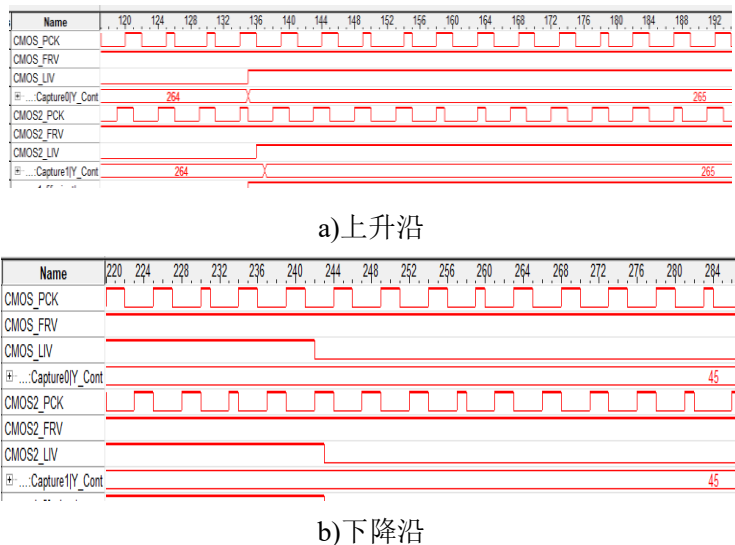


图 2-14 帧信号与行信号波形图（上升沿与下降沿）

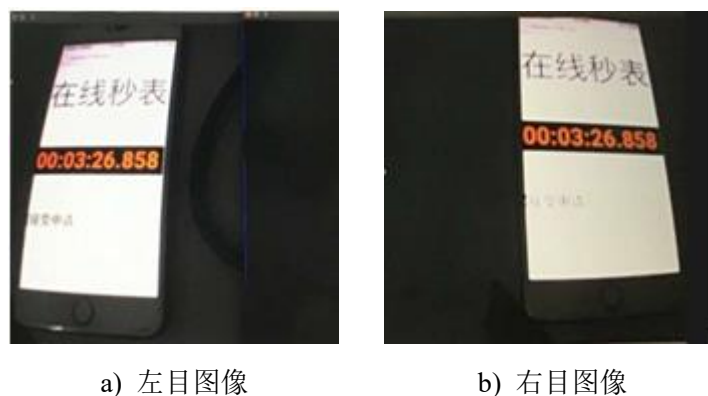


图 2-15 相机同步性实验

(2) 动态场景成像稳定性

双目 IMU 视觉惯性模组中使用的是全局曝光 CMOS 芯片 AR0144, AR0144 成像传感器采用全局快门技术, 可以在快速运动场景中拍摄图像, 获得的图像不会扭曲变形。卷帘快门图像传感器是逐行进行图像曝光, 在快速运动场景中或者场景中有快速运动的物体, 快速运动的物体在逐行曝光时会产生扭曲变形。全局快门图像传感器同时对整幅图像进行曝光, 即是在快速运动场景中物体也不会发生扭曲变形。本实验在飞行器高速飞行状态下抓拍的图像, 如图 2-16 所示, 其中路灯没有扭曲, 篮球架没有模糊变形, 图像中景物完整清晰。本实验也抓拍了高速旋转的风扇, 其中风扇扇叶清晰可见, 没有变形模糊。

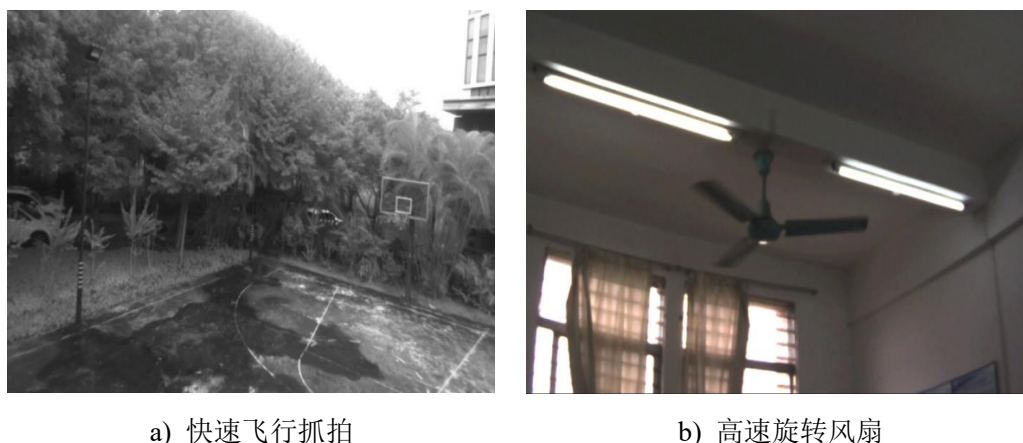


图 2-16 CMOS 相机全局曝光实验

(3) VIO 算法定位效果

本实验使用该双目 IMU 视觉惯性模组进行测试, 该双目 IMU 视觉惯性模组可以流畅运行 ORB-SLAM2, VINS-MONO, VINS-FUSION, MSCKF 等开源算法。该双目 IMU 视觉惯性模组接口为千兆网口, 双目 IMU 视觉惯性模组通过 USB 转接器接入到 Avnet

Ultra96-V2 板，在 Avnet Ultra96-V2 平台上搭建 Linux 环境，在其 SD 卡上烧录 Linaro 文件系统，实现 Avnet Ultra96-V2 板的 4 核 A53 ARM 和 PC 的 X86 Linux 的交叉编译。本实验在飞行器上进行测试，使用本文改进的基于点线特征的双目视觉惯性 SLAM 系统，我们使用 RTK 得到的姿态作为基准信息，效果如下图所示。

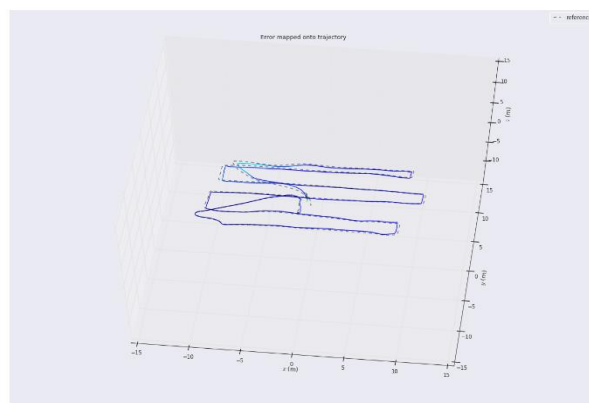


图 2-17 SLAM 算法效果

2.4 本章小结

本章主要介绍了双目 IMU 视觉惯性模组的设计与具体实现。描述了模组的性能要求，从元件选型、电路图设计到硬件整体框架对模组做一个整体介绍；然后就双目 IMU 视觉惯性模组驱动设计部分，本文讲解了 FPGA 驱动和图像 IMU 数据采集驱动的功能与设计实现，仔细分析了图像 IMU 数据采集驱动的 UDP 数据采集线程和图像 IMU 数据解析线程的设计思路，最后对双目 IMU 视觉惯性模组的性能进行测试，验证了双目 IMU 视觉惯性模组具有较高的时间同步性与成像稳定性。

第三章 SLAM 中的线特征

本章对图像中的线特征进行介绍。SLAM 系统中点特征一般是图像的角点，线特征一般是图像中的物体的边缘或者纹理处。在某些环境下，图像能提取的角点比较少，例如在无纹理环境中（走廊）点的检测和环境光线发生变化的环境中点的匹配都是很困难的。而线特征具有光照和视角不变性，且能提供更多的环境结构信息，因此本文融合点特征和线特征的信息来估计相机的运动姿态，有效地提高系统的鲁棒性和精度。

3.1 线特征的提取

3.1.1 LSD 线特征

SLAM 系统的精度一定程度上取决于前端的好坏，所以前端筛选提取出一个好的特征至关重要。本文使用 LSD 算法对图像进行线特征提取，LSD 的核心思想是计算一幅图像每个像素点的梯度方向，将相邻像素点的梯度方向近似相同的像素点合并成一个线段支持域^[25]，如图 3-1 所示。

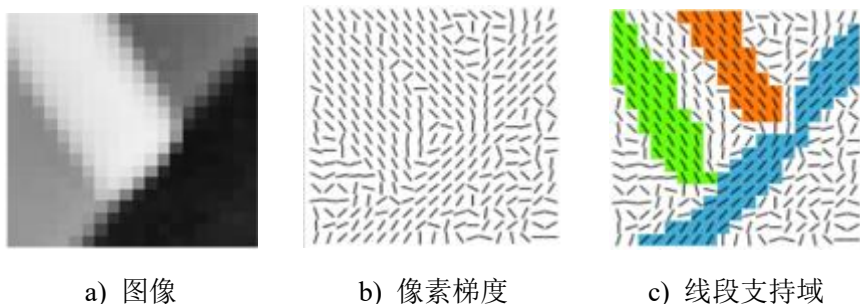


图 3-1 LSD 中的线段支持域

每一个线段支持域都是由一组梯度相似的像素点组成，也都是直线段的候选。首先检测线段支持域的最小外接矩形，然后计算最小外接矩形的最长对称轴的方向作为最小外接矩形的主方向^[26]，最小外接矩形如图 3-2 所示。

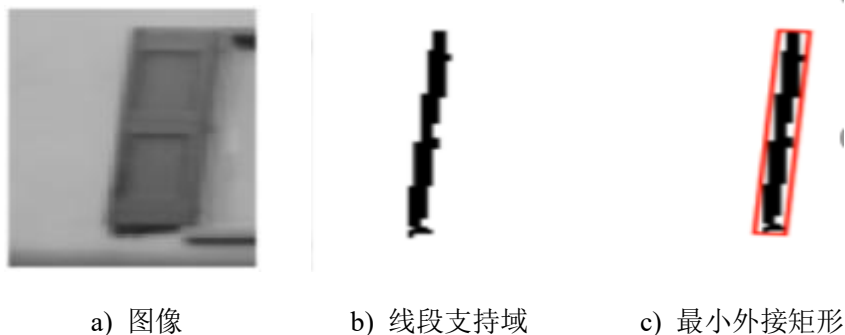


图 3-2 线段支持域的最小外接矩形

同时计算最小外接矩形中的每个像素点的梯度方向与最小外接矩形的主方向的角度差是否在一定阈值内，若是则该点为同性点，通过统计最小外接矩形中同性点与所有像素点的比例来判断该线段支持域是不是一个线端。

LSD 算法的具体解释：

(1) 图像降采样

将图像的 X 和 Y 方向尺寸同时缩小为原尺寸的 80%，由此消除图像中物体边缘的锯齿效应，避免出现一条线段被检测成几条线段，如图 3-3 所示，一条线段因为锯齿效应提取出了四条线段。本文对图像进行高斯降采样，首先对图像进行高斯滤波，避免出现锯齿效应和图像模糊，随后进行 80% 的降采样。

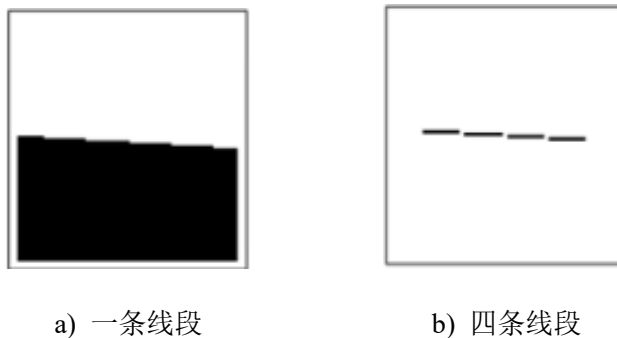


图 3-3 边缘锯齿效应

(2) 梯度计算

对整幅图像计算像素点的梯度值和梯度方向，整理得：

$$\begin{cases} g_x(x, y) = \frac{i(x+1, y) + i(x+1, y+1) - i(x, y) - i(x, y+1)}{2} \\ g_y(x, y) = \frac{i(x, y+1) + i(x+1, y+1) - i(x, y) - i(x+1, y)}{2} \\ G(x, y) = \sqrt{g_x^2(x, y) + g_y^2(x, y)} \\ \theta(x, y) = \arctan\left(-\frac{g_x(x, y)}{g_y(x, y)}\right) \end{cases} \quad (3-1)$$

其中 g_x 为水平方向梯度, g_y 为垂直方向梯度, $G(x, y)$ 是梯度值, $\theta(x, y)$ 是梯度方向。

LSD 提取线段时是确定线段方向的, 即是线段的两个端点是分起点和终点的, 当图像的灰度值相反时, LSD 提取到的线段和原来线段一样, 但是线段的两个端点的起点和终点会相互交换。

(3) 梯度排序

直线边缘的像素点的梯度值一般很高, 其像素点的梯度值越大, 就越适合作为种子点进行搜索。而对所有像素点的梯度值进行排序是一个很耗时的工作, 通常需要 $O(n \log n)$ 的时间复杂度, 因此, 将图像梯度范围设为 $[0, 1023]$, 建立 1024 个链表, 根据每个像素的梯度值的大小分别放入到 1024 个链表中, 然后根据大小将 1024 个链表再合成一个链表。其中将所有的像素点状态设为 **unused**。LSD 算法是一种贪婪的算法, 第一个像素的梯度值高低会影响算法的时间复杂度, 所以需要梯度排序。

(4) 梯度域值

梯度值小的点往往出现在平坦区域, 即图像灰度值变化不明显的地方, 也可能是噪声, 且会严重干扰最小外接矩形的构建, 因此, 将梯度幅值小于某个阈值的像素点的状态设置为 **USED**, 在进行区域增长的时候不对该像素点进行检测, 不让其参与构建最小外接矩形。

(5) 区域增长

线段支持域的生成通过区域生长算法合并梯度方向相似的相邻像素点得到的, 搜索种子点的 8 邻域内 **UNUSED** 的像素点的梯度方向和线段支持域的方向是否相似来决定是否将该像素点加入到线段支持域中。其中线段支持域的初始角度是由种子点的梯度方向决定, 其后线段支持域中每添加一个像素点都需要更新一次线段支持域的角度, 公式如下:

$$\arctan\left(\frac{\sum_j \sin(\text{level-line-angle}_j)}{\sum_j \cos(\text{level-line-angle}_j)}\right) \quad (3-2)$$

区域增长检测每个像素点是不是 UNUSED，对于 UNUSED 的像素点才能加入到线段支持域中，这种机制决定了该像素点只能属于一条线段，当线段被物体遮挡时或者线段的中央图像模糊时，一条线段会断裂成两条线段。

(6) 最小外接矩形

区域增长找到了线段支持域，此时需要寻找线段支持域的最小外接矩形，最小外接矩形的中心为线段支持域的质心，公式如下：

$$\begin{aligned} c_x &= \frac{\sum_{j \in \text{Region}} G(j) \cdot x(j)}{\sum_{j \in \text{Region}} G(j)} \\ c_y &= \frac{\sum_{j \in \text{Region}} G(j) \cdot y(j)}{\sum_{j \in \text{Region}} G(j)} \end{aligned} \quad (3-3)$$

其中 c_x, c_y 是线段支持域的质心， $G(j)$ 是像素梯度值， $x(j), y(j)$ 是像素坐标。

M 矩阵的最小特征值对应的特征向量的角度是最小外接矩形的主方向，整理得：

$$\begin{aligned} M &= \begin{pmatrix} m^{xx} & m^{xy} \\ m^{xy} & m^{yy} \end{pmatrix} \\ m^{xx} &= \frac{\sum_{j \in \text{Region}} G(j) \cdot (x(j) - c_x)^2}{\sum_{j \in \text{Region}} G(j)} \\ m^{xy} &= \frac{\sum_{j \in \text{Region}} G(j) \cdot (y(j) - c_y)^2}{\sum_{j \in \text{Region}} G(j)} \\ m^{yy} &= \frac{\sum_{j \in \text{Region}} G(j) \cdot (x(j) - c_x)(y(j) - c_y)}{\sum_{j \in \text{Region}} G(j)} \end{aligned} \quad (3-4)$$

其中 M 是最小外接矩形的构造矩阵， c_x, c_y 是线段支持域的质心， $G(j)$ 是像素梯度值， $x(j), y(j)$ 是像素坐标。

(7) NFA 计算

即是最小外接矩形中的同性点与最小外接矩形中的所有像素点的比值是否在一定阈值内，进而判断这个最小外接矩形是否是一条直线。

3.1.2 LSD 改进

在某些情况下，比如遮挡、图像模糊，实际环境中的一条直线可能被分割为很多条线段，增加了相邻帧线段匹配的难度，由此改进线特征的 LSD 提取方法。LSD 提取线段是确定线段方向的，即是线段的两个端点是分起点和终点的，根据线段支持域的最小外接矩形的主方向、线段之间的距离、线段的最近两个端点之间的距离和线段的 LBD 描述子之间的距离来判断两个线段是不是在同一条直线上，如果两个线段支持域的最小外接矩形的主方向之差在一定阈值内，且线段之间的距离差和线段的端点最小距离差在一定阈值内，初步判定两个线段在一条直线上，再判断两个线段的 LBD 描述子是否基本吻合，如果吻合则合并两个线段，保留两个线段距离最远的两个端点，重新计算新的线段的 LBD 描述子。

3.2 线特征的描述

由于线特征的端点检测到的位置不准确，没有像点特征一样具有极线几何约束，整条线段极易发生割裂，在相同的纹理处不具有辨识度等等，线特征的匹配是一个很棘手的问题。Zhiheng Wang 提出了 MSLD (mean-standard deviation line descriptor) 描述子，沿用 SIFT 描述子的思路，将线段分成了多个条带，根据每个条带内的像素梯度建立描述子，使得描述子对线段的长度不敏感，同时具有旋转不变性，光照不变性^[27]。随后 Lilian Zhang 提出了 LBD 描述子，LBD 描述子在 MSLD 描述子的基础上进行改进，引入局部高斯权重系数和全局高斯权重系数，使得 LBD 描述子匹配效果更好，计算速度更快^[28]。

本文使用 LBD 描述子对线特征进行特征匹配，LBD 算法的核心是首先在线段处建立一个线段支持域，将线段支持域分割为一组相互平行的条带，假设将线段支持域分割为 m 个条带，用 w 表示每个条带的像素宽度，如图 3-4 所示，线段支持域的 $m=5$, $w=3$ 。

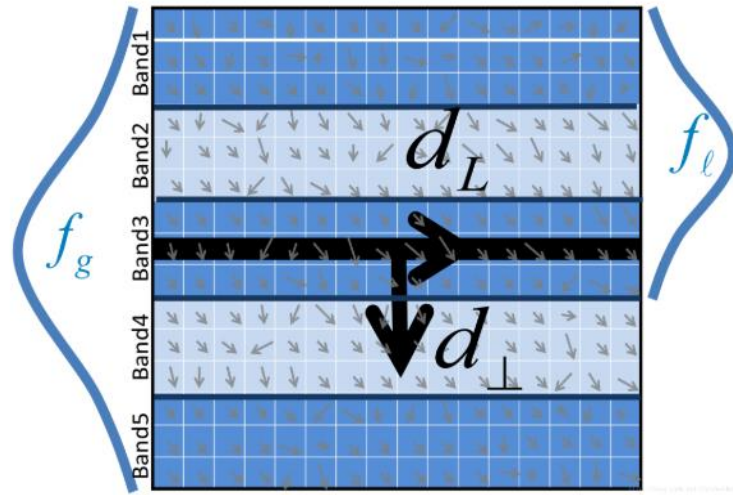


图 3-4 线段支持域的条带表示

其中 d_L 为直线段所朝的方向, d_{\perp} 表示 d_L 方向的顺时针垂直方向, 通过 d_L 与 d_{\perp} 构建了局部坐标系, 局部坐标系的原点是这条线的中点。同时将线段支持域内的每个像素的梯度都投影到局部坐标系中 $g' = (g^T \cdot d_{\perp}, g^T \cdot d_L)^T \triangleq (g'_{d_{\perp}}, g'_{d_L})^T$, 其中 g 是原线段支持域中的像素梯度, g' 是投影到局部坐标系中的像素梯度。为了降低离线段较远的条带的梯度的重要性, 引入了全局高斯权重函数 f_g 应用于线段支持域的每个条带。为了降低条带之间的边界效应, 引入了局部高斯权重函数 f_l 应用于线段支持域的相邻条带。

使用该办法描述线段支持域会使得描述子对 d_L 方向的细微变化具有更好的鲁棒性, 由于线段的端点检测的位置可能不准确, 所以线段在 d_L 方向的位置精度要比 d_{\perp} 方向的精度要低, 这时候会导致线段支持域的边界发生变化, 而线段支持域的大部分内容并不发生改变。

通过 B_j 和它的相邻条带 B_{j-1} 和 B_{j+1} 的像素梯度可以计算出 B_j 的条带描述子 BD_j , BD_j 的计算如下。

因为条带中第 k 行中每个像素有四个方向上的像素梯度, 分别是 d_{\perp} 方向, d_{\perp} 反方向, d_L 方向, d_L 反方向, 所以条带中每一行像素可以计算出 4 个方向的梯度数据, 整理得:

$$\begin{aligned} v1_j^k &= \lambda \sum_{g'_{d_{\perp}} > 0} g'_{d_{\perp}}, \quad v2_j^k = \lambda \sum_{g'_{d_{\perp}} > 0} -g'_{d_{\perp}} \\ v3_j^k &= \lambda \sum_{g'_{d_L} > 0} g'_{d_L}, \quad v4_j^k = \lambda \sum_{g'_{d_L} > 0} -g'_{d_L} \end{aligned} \quad (3-5)$$

其中 $\lambda = f_g(k) f_l(k)$ 是高斯系数, $v1, v2, v3, v4$ 是四个方向的高斯加权梯度数据^[26]。

将 B_j 条带中的所有行的信息累计起来, 可以构造出一个条带的描述子矩阵 BDM_j :

$$\begin{aligned} BDM_j &= \begin{pmatrix} v1_j^1 & v1_j^2 & \cdots & v1_j^n \\ v2_j^1 & v2_j^2 & \cdots & v2_j^n \\ v3_j^1 & v3_j^2 & \cdots & v3_j^n \\ v4_j^1 & v4_j^2 & \cdots & v4_j^n \end{pmatrix} \in \mathbf{R}^{4 \times n} \\ n &= \begin{cases} 2w, & j=1 \parallel m \\ 3w, & else. \end{cases} \end{aligned} \quad (3-6)$$

其中 BDM 是描述子矩阵, $v1, v2, v3, v4$ 是四个方向的高斯加权梯度数据, n 是行数^[26]。

最后 BD_j 由 BDM_j 的均值向量 M_j 和方差向量 S_j 组成, 即 $BD_j = (M_j^T, S_j^T)^T \in \mathbf{R}^8$, 最后将所有条带的特征向量 BD_j 进行合并构成 LBD 描述子:

$$LBD = (BD_1^T, BD_2^T \dots BD_m^T)^T \quad (3-7)$$

Lilian Zhang 实验分析指出当 $m=9, w=7$ 时, LBD 描述子是一个 72 维的浮点型向量, 描述子的效果最好, 在实际应用时, 由于需要大量匹配线特征的描述子, 基于实时性的考虑, 参考 ORB 角点的 Brief 描述子, 将 72 维的浮点型向量转化为 256 维的二进制的描述子, 二进制的 LBD 描述子由 0 和 1 组成, 代表着 72 维的浮点型向量中元素之间的大小关系。此时两个线特征的匹配是计算两个二进制 LBD 描述子之间的汉明距离, 匹配效率很高。

本文对 EUROC 数据集内的一张图像提取线特征, 如图 3-5 所示。



图 3-5 线特征提取

3.3 空间直线的参数化

三维空间中点 X 的参数化方法的非齐次表示为 $\tilde{X} = [x, y, z]^T$ ，具有 3 个参数，3 个自由度。三维空间中的直线具有 4 个自由度，具有两种参数化方法：普吕克参数化方法和正交表示法。

(1) 普吕克参数化方法^[29]

空间中一条直线可以由两个空间点 P_1 和 P_2 确定，其中 $P_1 = (x_1, y_1, z_1, w_1)$ ， $P_2 = (x_2, y_2, z_2, w_2)$ ，直线的普吕克矩阵定义为： $T = P_2 P_1^T - P_1 P_2^T$ ，该表示具有描述一条直线所需的 4 个自由度，该普吕克矩阵是一个反对称矩阵，具有 6 个独立的非零元素，但只有 5 个比率有意义，而且行列式 $\det L = 0$ ，所以元素多了一个约束，故该普吕克矩阵自由度是 4。

普吕克坐标是由普吕克矩阵中的 6 个非零元素构成， \mathbf{n} 为直线的法向量， \mathbf{v} 为直线的方向向量，且 $\mathbf{n}^T \mathbf{v} = 0$ ，整理得：

$$L = \begin{bmatrix} \tilde{P}_1 \times \tilde{P}_2 \\ w_1 \tilde{P}_2 - w_2 \tilde{P}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{n} \\ \mathbf{v} \end{bmatrix} \quad (3-8)$$

其中 \tilde{P}_1 和 \tilde{P}_2 是空间点的坐标， \mathbf{n} 是直线的法向量， \mathbf{v} 是直线的方向向量。

普吕克矩阵 L 和普吕克坐标 l 的相互转换公式如下：

$$T = \begin{bmatrix} \mathbf{n}^\wedge & \mathbf{v} \\ -\mathbf{v}^T & 0 \end{bmatrix} \quad (3-9)$$

对于世界坐标系下的空间直线 L_w 可以变换到相机坐标系下 L_c ：

$$L_c = \begin{bmatrix} n_c \\ v \end{bmatrix} = T_w^{c'} L_w = \begin{bmatrix} R_w^c & \begin{bmatrix} t_w^c \end{bmatrix}_\times R_w^c \\ 0 & R_w^c \end{bmatrix} L_w \quad (3-10)$$

其中 $T_w^{c'}$ 是世界坐标系到相机坐标系的直线的变换矩阵， R_w^c ， t_w^c 是世界坐标系到相机坐标系的旋转矩阵和平移矩阵， L_w 是世界坐标系下的直线。

对于世界坐标系下的空间直线 L_w 投影到当前相机图像中的投影方程为：

$$L'_{uv} = K' n_c = \begin{bmatrix} f_v & 0 & 0 \\ 0 & f_u & 0 \\ -f_v c_u & f_u c_v & f_u f_v \end{bmatrix} n_c \quad (3-11)$$

其中 K' 是直线从相机坐标系投影到图像平面的内参矩阵。

由上可以看出，普吕克坐标表示法具有直观性，空间直线使用普吕克表示法进行投影变换很简单。但是普吕克坐标在空间中具有 6 个参数，而空间直线是 4 个自由度，因而普吕克坐标表示法是过参的，在优化过程中，因为普吕克坐标需要满足 $n^T v = 0$ ，所以会降低数值稳定性，而且会增加计算代价。

(2) 正交表示法^[21]

由于空间直线只有四个自由度，后端优化时使用 4 个参数的正交表示法 $(U \ W) \in SO(3) \times SO(2)$ 更合适。通过对普吕克坐标进行 QR 分解：

$$\begin{bmatrix} n & v \end{bmatrix} = \begin{bmatrix} \frac{n}{\|n\|} & \frac{v}{\|v\|} & \frac{n \times v}{\|n \times v\|} \end{bmatrix} \begin{bmatrix} \|n\| & 0 \\ 0 & \|v\| \\ 0 & 0 \end{bmatrix} \quad (3-12)$$

其中 $\| \cdot \|$ 表示向量的模值， \wedge 代表向量叉乘。

然后分解出正交矩阵 U ，表示相机坐标系到直线坐标系的旋转矩阵，可得：

$$U = R(\psi) = \begin{bmatrix} \frac{n}{\|n\|} & \frac{v}{\|v\|} & \frac{n \times v}{\|n \times v\|} \end{bmatrix} \quad (3-13)$$

其中 $\psi = [\psi_1, \psi_2, \psi_3]^T$ 代表的是相机坐标系到直线坐标系在 x, y 和 z 轴的旋转角。

使用三角函数矩阵参数化 $(\|n\|, \|v\|)$ 可得：

$$W = \begin{bmatrix} w_1 & -w_2 \\ w_2 & w_1 \end{bmatrix} = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix} = \frac{1}{\sqrt{(\|n\|^2 + v^2)}} \begin{bmatrix} \|n\| & -\|v\| \\ \|v\| & \|n\| \end{bmatrix} \quad (3-14)$$

其中 W 表示直线正交表示法的表示距离信息的 $SO(2)$ 矩阵。

上式中的 W 是包含了坐标原点到空间直线的距离信息的，可以看出 U 包含了旋转的 3 个自由度， W 包含了 1 个距离的自由度。故正交表示法为 $\mathcal{O} = [\psi, \phi]^T$ 。

正交表示可以和普吕克坐标相互转换：

$$L = [w_1 u_1^T, w_2 u_2^T]^T = \frac{1}{\sqrt{(\|n\|^2 + \|v\|^2)}} L \quad (3-15)$$

其中 u_i 是 U 的第 i 列。

正交表示法只有 4 个参数，空间直线是 4 个自由度，因而在后端优化时，正交表示法不会引入额外的约束，而普吕克表示法会引入 $n^T v = 0$ 的额外约束。但是正交表示法在表示直线投影变换时具有困难，所以本文在直线的提取、投影变换时使用普吕克表示法，在优化时使用正交表示法。

3.4 本章小结

本章主要介绍了 SLAM 中的线特征。分析了图像中线段的提取方法，描述子的生成方法和空间直线的参数化方法。其中详细介绍了使用 LSD 方法提取图像中线段的原理和 LBD 描述子的生成方法，然后仔细分析了空间直线的普吕克表示方法以及正交表示法的原理、优缺点以及应用场景。

第四章 基于点线特征的双目视觉 IMU 紧耦合优化

本文基于优化的方式融合双目视觉信息和 IMU 信息，优化目标函数主要包括先验观测残差，IMU 观测残差，两帧间同相机的点特征观测残差，两帧间两相机的点特征观测残差，一帧间两相机的点特征观测残差和线特征重投影误差。其非线性优化过程可以看作因子图^[30]，如图 4-1 所示，其中五角形的节点为路标点，圆形节点为相机姿态，三角形节点为 IMU 的姿态，带正方形的边表示视觉与相机姿态之间的约束，带虚线的边表示两个 IMU 的姿态之间的 IMU 约束。

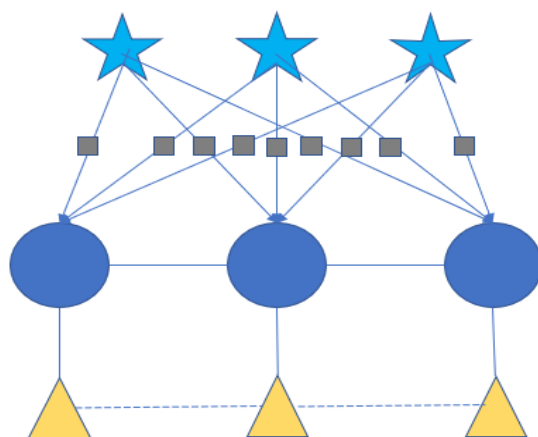


图 4-1 SLAM 系统因子图

4.1 滑动窗口优化与边缘化

4.1.1 滑动窗口优化

本文使用滑动窗口对因子图进行优化，滑动窗口中的优化变量包括所有帧的 IMU 状态（位置，速度，方向，加速度计偏置，陀螺仪偏置），滑动窗口内的相机观测到的特征点的逆深度，滑动窗口内的相机观测到的线特征的正交表示坐标。滑动窗口中的优化变量如下：

$$\begin{aligned} \mathcal{X} &= [x_n, x_{n+1}, \dots, x_{n+N}, \lambda_m, \lambda_{m+1}, \dots, \lambda_{m+M}, \mathcal{O}_o, \mathcal{O}_{o+1}, \dots, \mathcal{O}_{o+O}]^\top \\ x_i &= [p_{b_i}^w, q_{b_i}^w, v_i^w, b_a^{b_i}, b_g^{b_i}]^\top, i \in [n, n+N] \end{aligned} \quad (4-1)$$

其中 x_i 是世界坐标系下第 i 帧的 IMU 的状态（位置 p ，速度 v ，方向 q 和加速度计偏置

$b_a^{b_i}$ 以及陀螺仪偏置 $b_g^{b_i}$)。 l_m 表示滑动窗口内的相机观测到的特征点的逆深度, \mathcal{O} 表示滑动窗口内的相机观测到的线特征的正交表示坐标, N 为滑动窗口中所有帧的数量, M 为滑动窗口中所有帧观测到的特征点的数量, O 为滑动窗口中所有帧观测到的线特征的数量, n 为滑动窗口中起始帧的索引, m 为滑动窗口中特征点的索引, o 为滑动窗口中线特征的索引。

目标函数为观测残差的先验^[31]、IMU 观测残差、点特征观测残差和线特征观测残的马氏距离之和, 通过最小化目标函数来得到系统的最大后验估计^[17], 目标函数如下:

$$\min_{\chi} \rho \left(\|r_p - J_p \chi\|_{\Sigma_p}^2 \right) + \sum_{i \in \mathcal{B}} \rho \left(\|r_b(z_{b_{i+1}}^{b_i}, \chi)\|_{\Sigma_{b_{i+1}}^{b_i}}^2 \right) + \sum_{(i,j) \in K} \rho \left(\|r_k(z_{i_j}^{c_i}, \chi)\|_{\Sigma_{k_i}^{c_i}}^2 \right) + \sum_{(i,l) \in L} \rho \left(\|r_l(z_{l_i}^{c_i}, \chi)\|_{\Sigma_{l_i}^{c_i}}^2 \right) \quad (4-2)$$

其中 ρ 是鲁棒核函数, 通常目标函数使用误差项的二范数平方和来表示, 这种表示具有直观性, 但是由于前端可能出现误匹配的情况, 如果某个误差项存在大量的错误数据, 这会使得优化的时候与该误差项相关的优化变量会使目标函数下降的更多, 即是会抹平其他误差项的影响, 所以使用鲁棒核函数可以限制误差项的变化幅度。

本文使用 Huber 核函数:

$$H(e) = \begin{cases} \frac{1}{2}e^2 & \text{if } |e| \leq d \\ d \left(|e| - \frac{1}{2}d \right) & \text{otherwise} \end{cases} \quad (4-3)$$

其中 d 是阈值, e 是误差。

由图 4-2 可以看出, 当误差 e 由于增长过快大于 d 时, 函数的增长速度开始变慢, 即是误差项由二次形式的增长变成了一次形式的增长, 可以明显看到限制了误差项的增长速度, 同时 Huber 核函数是光滑可导的。

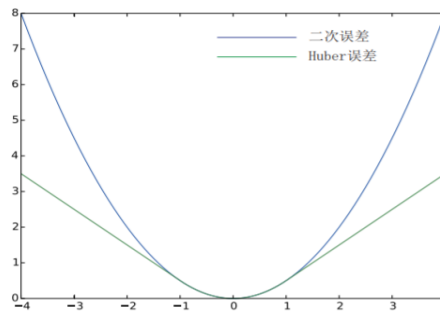


图 4-2 Huber 核函数

本文使用 Levenberg-Marquard 算法进行非线性优化上式目标函数，通过最小化上式目标函数可以得到 dC ，简化为：

$$\begin{aligned} (H_p + H_b + H_k + H_l) \delta \chi &= (b_p + b_b + b_k + b_l) \\ H &= J^T \Sigma^{-1} J \\ b &= -J^T \Sigma^{-1} r \end{aligned} \quad (4-4)$$

其中 H_p 是先验残差的 Hessian 矩阵， H_b 是 IMU 残差的 Hessian 矩阵， H_k 是点残差的 Hessian 矩阵， H_l 是线残差的 Hessian 矩阵。 J 和 S 是各个残差项的雅可比矩阵和协方差矩阵， r 是各个残差项。

进行非线性优化后，得到 dC ，然后对滑动窗口中的优化变量进行更新，滑动窗口中位置 P ，滑动窗口中速度 V ，滑动窗口中方向 Q ，滑动窗口中加速度计偏置 b_a ，滑动窗口中陀螺仪偏置 b_g ，滑动窗口中特征点的逆深度 l 的更新方式如下：

$$\begin{aligned} p' &= p + \delta p, v' = v + \delta v, \lambda' = \lambda + \delta \lambda, b' = b + \delta b, \\ q' &= q \otimes \delta q, \quad \delta q \approx \begin{bmatrix} 1 \\ \frac{1}{2} \delta \theta \end{bmatrix} \end{aligned} \quad (4-5)$$

其中 δp ， δv ， δl ， δb ， δq ， δq 代表优化变量的增量， \otimes 代表四元数乘法。

滑动窗口中线特征的正交表示法参数更新如下：

$$\begin{aligned} \delta O &= [\delta \psi]_{\times}, \delta \phi]^T, U' \approx U (I + [\delta \psi]_{\times}) \\ W' &\approx W \left(I + \begin{bmatrix} 0 & -\delta \phi \\ \delta \phi & 0 \end{bmatrix} \right) \end{aligned} \quad (4-6)$$

其中 W 表示直线正交表示法的表示距离信息的 $SO(2)$ 矩阵。 U 表示直线正交表示法的表示旋转信息的 $SO(3)$ 矩阵， δy ， δf 代表正交表示法优化变量增量。

4.1.2 边缘化与关键帧的选取

由滑动窗口优化中介绍的最小化目标函数解得增量方程：

$$\begin{aligned} H\delta x &= b \\ H &= J^T J, b = J^T e, e = (J^T)^+ b \end{aligned} \quad (4-7)$$

其中 $(J^T)^+$ 表示 J^T 的伪逆, H 为滑动窗口中的 Hessian 矩阵, e 为滑动窗口中的残差, J 为滑动窗口的雅可比。

每当滑动窗口中新到来一帧时, 为了维护固定的滑动窗口帧的数量, 会根据当前帧是不是关键帧来决定丢掉最新帧还是最老帧, 此时直接丢掉滑动窗口中的最老帧会造成约束信息的丢失, 即是最老帧中观测的特征点信息会对第二帧, 第三帧等等产生约束, 而直接丢弃掉这些约束信息, 可能会导致最老几帧共同优化后的优化变量发生偏移, 此时将这些约束信息包装成先验信息, 加入到目标函数中进行非线性优化, 以上即是边缘化的重要性。

假设需要边缘化的优化变量为 x_2 , 需要保留的优化变量为 x_1 , 则边缘化的过程可以整理为:

$$\begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \begin{bmatrix} \delta x_1 \\ \delta x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad (4-8)$$

我们关心如何保留优化变量 x_1 , 而不再需要求解优化变量 x_2 , 但是我们又不能直接将优化变量 x_2 删除掉, 因此, 我们可以猜用舒尔补的方式进行消元:

$$\begin{aligned} \begin{bmatrix} I & -H_{12}H_{22}^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \begin{bmatrix} \delta x_1 \\ \delta x_2 \end{bmatrix} &= \begin{bmatrix} I & -H_{12}H_{22}^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \\ \begin{bmatrix} H_{11} - H_{12}H_{22}^{-1}H_{21} & 0 \\ H_{21} & H_{22} \end{bmatrix} \begin{bmatrix} \delta x_1 \\ \delta x_2 \end{bmatrix} &= \begin{bmatrix} b_1 - H_{12}H_{22}^{-1}b_2 \\ b_2 \end{bmatrix} \end{aligned} \quad (4-9)$$

基于上式, 我们可以得到带有优化变量 x_2 信息的方程, 如下:

$$\begin{aligned} (H_{11} - H_{12}H_{22}^{-1}H_{21})\delta x_1 &= b_1 - H_{12}H_{22}^{-1}b_2 \\ H_0^* &= H_{11} - H_{12}H_{22}^{-1}H_{21} \\ b_0^* &= b_1 - H_{12}H_{22}^{-1}b_2 \\ H_0^*\delta x_1 &= b_0^* \end{aligned} \quad (4-10)$$

其中 H_0^* , b_0^* 是边缘化 x_2 后保留的 H 矩阵和 b 矩阵。

此时既没有损失边缘化的约束信息 x_2 , 又保留了优化变量 x_1 。

同时本文提出了一种关键帧的选取策略来优化滑动窗口中的关键帧的质量，避免不必要的计算的同时，提高了系统的精度，本文提出的关键帧选取策略将 IMU 信息与视觉信息相融合，通过 IMU 积分出的平移将相机当前的运动分为静止状态，平移运动状态，纯旋转运动状态。对于 IMU 积分出的平移 P 较小的帧，我们认为他是静止状态或者纯旋转状态，同时再根据当前最新帧与上一帧的视差量是否大于某个阈值，并且 IMU 积分出的旋转 Q 不能超过某个阈值去确定当前帧是否是关键帧，这样可以有效地排除掉静止状态或者纯旋转状态的帧加入到滑动窗口中。对于 IMU 积分出的平移 P 较大的帧，我们认为他是平移运动状态，同时再根据当前最新帧与上一帧的视差量是否大于某个阈值去确定当前帧是否是关键帧，这样平移较大的帧就可以有效地加入到滑动窗口中，避免了将平移较小的帧加入到滑动窗口中造成不必要的计算。

本文提出的关键帧的筛选策略有效的排除了纯旋转或者静止情况下将图像加入到关键帧中，优先将平移较大的帧加入到滑动窗口中，这样平移较大的帧的特征点三角化精度更高，而在纯旋转情况下，最近两帧图像上的特征点无法三角化，即纯旋转的帧无法提供视觉约束。

本文根据第二最新帧是否是关键帧，将边缘化分为两种策略，当第二最新帧是关键帧时，需要边缘化滑动窗口中的最老帧，将滑动窗口中的最老帧看到的线特征和点特征以及 IMU 信息转化为先验信息加入到目标函数中。当第二最新帧不是关键帧时，需要边缘化滑动窗口中的第二最新帧，将滑动窗口中的第二最新帧看到的线特征和点特征直接丢弃，由于第二最新帧不是关键帧，即代表当前帧观测到的特征点和第二最新帧观测到的特征点产生的约束关系很接近，此时我们直接将第二最新帧的特征点约束和线特征约束丢弃，并不会丢失信息，影响系统的精度，同时我们需要保留第二最新帧的 IMU 信息，以保证 IMU 信息的连贯性。

4.2 IMU 观测残差

在传统的捷联惯性导航算法中，在上一个时刻的 IMU 的位置，方向和速度的基础上对 IMU 新的测量数据进行积分，估计当前 IMU 的状态量（位置，速度和方向）。在基于优化的 SLAM 系统中，滑动窗口中的每一帧都会被优化，即当前 IMU 状态前的几个状态都会被优化，在每次滑动窗口迭代优化后，IMU 的位置，速度和方向都需要从第

一个被优化的 IMU 状态开始重新进行积分计算。因此 IMU 预积分被提了出来，IMU 预积分将 IMU 的观测量和 IMU 在世界坐标系下状态（位置，速度和方向）进行解耦，使得计算 IMU 之间的位姿变化情况不再依赖 IMU 当前的位置，速度和方向，从而避免了在优化时，IMU 的状态被重复积分计算^[32,33,34]。

以下所有的推导公式中，旋转矩阵相乘都是使用了右乘形式，这样角速度是在 IMU 坐标系下的测量值，而在左乘形式下角速度需要变换到世界坐标系下。

IMU 的测量值（加速度计得到的 \hat{a}_t 和陀螺仪得到的 $\hat{\omega}_t$ ）与真值之间的关系如下：

$$\begin{aligned}\hat{a}_t &= a_t + b_{a_t} + R_{\omega}^t g^w + n_a \\ \hat{\omega}_t &= \omega_t + b_{\omega_t} + n_{\omega}\end{aligned}\quad (4-11)$$

其中 \hat{a}_t 是加速度计的测量值， a_t 是加速度计真值， b_{a_t} 是加速度计偏置， g^w 是重力加速度， n_a 是加速度计噪声， R_{ω}^t 是世界坐标系到 IMU 坐标系的变换矩阵， $\hat{\omega}_t$ 是陀螺仪的测量值， ω_t 是陀螺仪的真值， b_{ω_t} 是陀螺仪的偏置， n_{ω} 是陀螺仪的噪声^[17]。

IMU 的测量值受到加速度计和陀螺仪的偏置 b_{a_t} ， b_{ω_t} 的影响，并且收到加速度计和陀螺仪噪声 n_a ， n_{ω} 的影响。其中加速度计和陀螺仪的偏置 b_{a_t} ， b_{ω_t} 是随着时间缓慢变化的，假设加速度计和陀螺仪测量过程中的附加噪声 n_a ， n_{ω} 为高斯噪声。

$$n_a \sim N(0, \sigma_a^2), \quad n_{\omega} \sim N(0, \sigma_{\omega}^2) \quad (4-12)$$

加速度计和陀螺仪的偏置 b_{a_t} ， b_{ω_t} 是随机游走模型，加速度计和陀螺仪偏置的导数是高斯的，整理得：

$$\begin{aligned}\dot{b}_{a_t} &= n_{b_a}, \quad \dot{b}_{\omega_t} = n_{b_{\omega}} \\ n_{b_a} &\sim N(0, \sigma_{b_a}^2), \quad n_{b_{\omega}} \sim N(0, \sigma_{b_{\omega}}^2)\end{aligned}\quad (4-13)$$

其中 S 是噪声方差， n_{b_a} 和 $n_{b_{\omega}}$ 是加速度计和陀螺仪偏置的导数^[17]。

在传统的捷联惯性导航算法中，在上一个时刻的 IMU 的位置，旋转和速度的基础上对 IMU 新的测量数据进行积分，估计当前 IMU 的位置，速度和旋转。通过对第 k 帧的 IMU 的位置 P 、速度 V 和方向 Q 状态和第 $k+1$ 帧内的 IMU 的加速度计和陀螺仪测量值进行积分得到第 $k+1$ 帧的 IMU 的位置 P 、速度 V 和方向 Q ，如下：

$$\begin{aligned}
 p_{b_{k+1}}^{\omega} &= p_{b_k}^{\omega} + v_{b_k}^{\omega} \Delta t_k + \iint_{t \in [t_k, t_{k+1}]} \left(R_t^{\omega} (\hat{a}_t - b_{a_t} - n_a) - g^{\omega} \right) dt^2 \\
 v_{b_{k+1}}^{\omega} &= v_{b_k}^{\omega} + \int_{t \in [t_k, t_{k+1}]} \left(R_t^{\omega} (\hat{a}_t - b_{a_t} - n_a) - g^{\omega} \right) dt \\
 q_{b_{k+1}}^w &= q_{b_k}^w \otimes \int_{t \in [t_k, t_{k+1}]} \frac{1}{2} q_t^{b_k} \otimes \begin{bmatrix} (\hat{w}_t - b_{w_t} - n_w) \\ 0 \end{bmatrix} dt \\
 &= q_{b_k}^w \otimes \int_{t \in [t_k, t_{k+1}]} \frac{1}{2} \Omega(\hat{w}_t - b_{w_t} - n_w) q_t^{b_k} dt \\
 \Omega(\omega) &= \begin{bmatrix} -[\omega]_{\times} & \omega \\ -\omega^T & 0 \end{bmatrix}
 \end{aligned} \tag{4-14}$$

其中 R_t^w 是 t 时刻 IMU 坐标系到世界坐标系的旋转矩阵, $q_{b_k}^w$ 是第 b_k 帧 IMU 坐标系到世界坐标系的旋转矩阵, $q_t^{b_k}$ 是 t 时刻 IMU 坐标系到第 b_k 帧 IMU 坐标系的旋转矩阵^[17]。

通过上式可以看出, 每一帧的 IMU 的位置 P 、速度 V 和方向 Q 是在世界坐标系下进行传递, 如果直接将 IMU 在世界坐标系下的 $p_{b_{k+1}}^{\omega}$ 、 $v_{b_{k+1}}^{\omega}$ 、 $q_{b_{k+1}}^{\omega}$ 作为优化变量, 那么在每次非线性优化后, IMU 的这些优化变量会进一步迭代更新, 此时 IMU 的位置 P 、速度 V 和方向 Q 需要这些更新后的优化变量进行积分重新传递, 会消耗很大的计算量。

IMU 预积分的思想是将 IMU 的观测量和 IMU 的在世界坐标系下状态 (位置, 速度和方向) 进行解耦, 即将优化变量的 $p_{b_{k+1}}^{\omega}$ 、 $v_{b_{k+1}}^{\omega}$ 、 $q_{b_{k+1}}^{\omega}$ 参考坐标系从世界坐标系转变为第 k 帧的 IMU 坐标系, 在 IMU 的积分公式上同时左乘 $R_{\omega}^{b_k}$, 整理得:

$$\begin{aligned}
 R_{\omega}^{b_k} p_{b_{k+1}}^{\omega} &= R_{\omega}^{b_k} \left(p_{b_k}^{\omega} + v_{b_k}^{\omega} \Delta t_k - \frac{1}{2} g^{\omega} \Delta t_k^2 \right) + \alpha_{b_{k+1}}^{b_k} \\
 R_{\omega}^{b_k} v_{b_{k+1}}^{\omega} &= R_{\omega}^{b_k} \left(v_{b_k}^{\omega} - g^{\omega} \Delta t_k \right) + \beta_{b_{k+1}}^{b_k} \\
 q_{\omega}^{b_k} \otimes q_{b_{k+1}}^{\omega} &= \gamma_{b_{k+1}}^{b_k} \\
 \alpha_{b_{k+1}}^{b_k} &= \iint_{t \in [t_k, t_{k+1}]} R_t^{b_k} (\hat{a}_t - b_{a_t} - n_a) dt^2 \\
 \beta_{b_{k+1}}^{b_k} &= \int_{t \in [t_k, t_{k+1}]} R_t^{b_k} (\hat{a}_t - b_{a_t} - n_a) dt \\
 \gamma_{b_{k+1}}^{b_k} &= \int_{t \in [t_k, t_{k+1}]} \frac{1}{2} \Omega(\hat{w}_t - b_{w_t} - n_w) \gamma_t^{b_k} dt
 \end{aligned} \tag{4-15}$$

其中 $\alpha_{b_{k+1}}^{b_k}$ 、 $\beta_{b_{k+1}}^{b_k}$ 、 $\gamma_{b_{k+1}}^{b_k}$ 是第 $k+1$ 帧 IMU 相对于第 k 帧 IMU 的位置 P 、速度 V 和方向 Q 的状态变化量^[17]。

此时第 k 帧的 IMU 的位置 P 、速度 V 和方向 Q 状态在优化后也不需要重新计算 $\alpha_{b_{k+1}}^{b_k}$ 、 $\beta_{b_{k+1}}^{b_k}$ 、 $\gamma_{b_{k+1}}^{b_k}$ ，这样避免了重复计算。

由于 IMU 的加速度计和陀螺仪的偏置也在改变，但是实际上这些偏置的变化量很小，将 $a_{b_{k+1}}^{b_k}$ 、 $b_{b_{k+1}}^{b_k}$ 、 $g_{b_{k+1}}^{b_k}$ 对 IMU 的加速度计和陀螺仪偏置进行一阶近似来调整。但是由于初始化时，计算的 IMU 的加速度计和陀螺仪偏置可能和初始值相差较大，此时需要重新进行积分迭代计算，不能进行一阶近似，整理得：

$$\begin{aligned}\alpha_{b_{k+1}}^{b_k} &\approx \hat{\alpha}_{b_{k+1}}^{b_k} + J_{b_a}^{\alpha} \delta b_a + J_{b_{\omega}}^{\alpha} \delta b_{\omega} \\ \beta_{b_{k+1}}^{b_k} &\approx \hat{\beta}_{b_{k+1}}^{b_k} + J_{b_a}^{\beta} \delta b_a + J_{b_{\omega}}^{\beta} \delta b_{\omega} \\ \gamma_{b_{k+1}}^{b_k} &\approx \hat{\gamma}_{b_{k+1}}^{b_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2} J_{b_{\omega}}^{\gamma} \delta b_{\omega} \end{bmatrix}\end{aligned}\quad (4-16)$$

其中 $J_{b_a}^{\alpha}$ 、 $J_{b_{\omega}}^{\alpha}$ 、 $J_{b_a}^{\beta}$ 、 $J_{b_{\omega}}^{\beta}$ 、 $J_{b_{\omega}}^{\gamma}$ 是 $\alpha_{b_{k+1}}^{b_k}$ 、 $\beta_{b_{k+1}}^{b_k}$ 、 $\gamma_{b_{k+1}}^{b_k}$ 对加速度计和陀螺仪偏置的一阶近似雅可比^[17]。

由于 IMU 在每一个时刻积分出来的状态都是有误差的，建立误差状态递推方程，下面给出在 t 时刻 IMU 的位置 P 、速度 V 和方向 Q 状态增量的误差项的导数，整理得：

$$\begin{bmatrix} \delta \dot{\alpha}_t^{b_k} \\ \delta \dot{\beta}_t^{b_k} \\ \delta \dot{\theta}_t^{b_k} \\ \delta \dot{b}_{a_t} \\ \delta \dot{b}_{w_t} \end{bmatrix} = \begin{bmatrix} 0 & I & 0 & 0 & 0 \\ 0 & 0 & -R_t^{b_k} [\hat{a}_t - b_{a_t}]_{\times} & -R_t^{b_k} & 0 \\ 0 & 0 & -[\hat{w}_t - b_{w_t}]_{\times} & 0 & -I \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta \alpha_t^{b_k} \\ \delta \beta_t^{b_k} \\ \delta \theta_t^{b_k} \\ \delta b_{a_t} \\ \delta b_{w_t} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ -R_t^{b_k} & 0 & 0 & 0 \\ 0 & -I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix} \begin{bmatrix} n_a \\ n_w \\ n_{b_a} \\ n_{b_w} \end{bmatrix}\quad (4-17)$$

其中， $[\]_{\times}$ 表示向量的反对称矩阵^[17]。

上式简化为：

$$\delta \dot{z}_t^{b_k} = F_t \delta z_t^{b_k} + G_t n_t \quad (4-18)$$

根据上式 IMU 的位置 P 、速度 V 和方向 Q 状态增量的线性化增量的递推方程，那么可以根据上一个时刻的 IMU 状态增量的雅可比矩阵递推下一个时刻的 IMU 状态增量的雅可比矩阵为：

$$J_{t+\delta t} = (I + F_t \delta t) J_t \quad (4-19)$$

同样根据上式 IMU 的位置 P 、速度 V 和方向 Q 状态增量的线性化增量的递推方程，

那么可以根据上一个时刻的 IMU 状态增量的协方差矩阵递推下一个时刻的 IMU 状态增量的协方差矩阵，整理得：

$$P_{t+\delta t}^{b_k} = (I + F_t \delta t) P_t^{b_k} (I + F_t \delta t)^T + (G_t \delta t) Q (G_t \delta t)^T \quad (4-20)$$

其中 P 是优化变量的协方差矩阵， Q 是噪声的协方差矩阵。

在两帧之间 IMU 预积分的初始协方差设为 0，雅可比设为单位矩阵 1。

由公式，可以得到 IMU 的观测残差为滑动窗口中两帧之间的位置 P 、速度 V 和方向 Q 的变化量与 IMU 预积分的位置 P 、速度 V 和方向 Q 的增量之间的差值为：

$$r_B \left(\hat{z}_{b_{k+1}}^{b_k}, \chi \right) = \begin{bmatrix} \delta \alpha_{b_{k+1}}^{b_k} \\ \delta \beta_{b_{k+1}}^{b_k} \\ \delta \theta_{b_{k+1}}^{b_k} \\ \delta b_a \\ \delta b_g \end{bmatrix} = \begin{bmatrix} R_{\omega}^{b_k} \left(p_{b_{k+1}}^{\omega} - p_{b_k}^{\omega} + \frac{1}{2} g^{\omega} \Delta t_k^2 - v_{b_k}^{\omega} \Delta t_k \right) - \hat{\alpha}_{b_{k+1}}^{b_k} \\ R_{\omega}^{b_k} \left(v_{b_{k+1}}^{\omega} + g^{\omega} \Delta t_k - v_{b_k}^{\omega} \right) - \hat{\beta}_{b_{k+1}}^{b_k} \\ 2 \left[q_{b_k}^{\omega^{-1}} \otimes q_{b_{k+1}}^{\omega} \otimes \left(\hat{\gamma}_{b_{k+1}}^{b_k} \right)^{-1} \right]_{xyz} \\ b_{a_{b_{k+1}}} - b_{a_{b_k}} \\ b_{\omega_{b_{k+1}}} - b_{\omega_{b_k}} \end{bmatrix} \quad (4-21)$$

其中 $[\]_{xyz}$ 为取四元数的虚部， $\delta \alpha_{b_{k+1}}^{b_k}$ 为第 k 帧和第 $k+1$ 帧之间 IMU 的位置 P 增量残差， $\delta \beta_{b_{k+1}}^{b_k}$ 为第 k 帧和第 $k+1$ 帧之间 IMU 的速度 V 增量残差， $\delta \theta_{b_{k+1}}^{b_k}$ 为第 k 帧和第 $k+1$ 帧之间 IMU 的方向 Q 增量残差， δb_a 为第 k 帧和第 $k+1$ 帧之间 IMU 的加速度偏置增量残差， δb_g 为第 k 帧和第 $k+1$ 帧之间 IMU 的陀螺仪偏置增量残差， $\hat{\alpha}_{b_{k+1}}^{b_k}$ 为第 k 帧和第 $k+1$ 帧之间 IMU 的位置 P 的预积分增量， $\hat{\beta}_{b_{k+1}}^{b_k}$ 为第 k 帧和第 $k+1$ 帧之间 IMU 的速度 V 的预积分增量， $\hat{\gamma}_{b_{k+1}}^{b_k}$ 为第 k 帧和第 $k+1$ 帧之间 IMU 的方向 Q 的预积分增量^[17]。

IMU 约束的优化变量为第 i 帧和第 j 帧的 P ， V ， Q ， b_a ， b_g ，即：

$$\left[p_{b_i}^w, q_{b_i}^w \right], \left[v_{b_i}^w, b_{a_i}, b_{\omega_i} \right], \left[p_{b_j}^w, q_{b_j}^w \right], \left[v_{b_j}^w, b_{a_j}, b_{\omega_j} \right]$$

4.3 点特征观测残差

4.3.1 两帧间同相机的点特征观测残差

在归一化相机坐标系下，空间中第 k 个特征点 P_k 被第 i 帧的相机 1 观测到和第 j 帧的相机 1 都观测到，其中相机 1 在第 i 帧观测到特征点为 $\hat{P}_k^{c_i}$ ，相机 1 在第 j 帧观测到特征点为 $\hat{P}_k^{c_j}$ ，如图 4-3 所示。

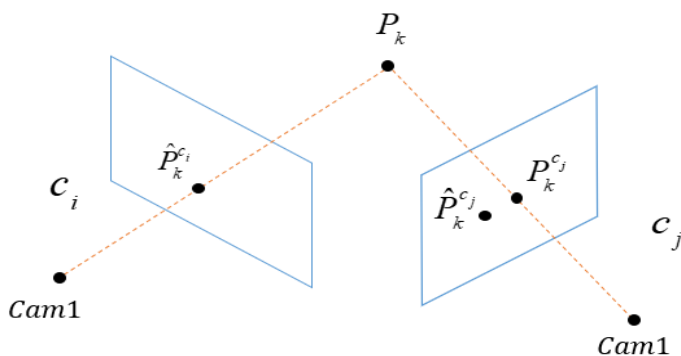


图 4-3 相机 1 的点重投影误差

将相机 1 在第 i 帧观测到的点 $\hat{P}_k^{c_i}$ 转换到在第 j 帧相机 1 的相机坐标系下得到 $P_k^{c_j}$ ，其中点 $\hat{P}_k^{c_i}$ 在第 i 帧的相机 1 的相机坐标系下坐标为 $\begin{bmatrix} u_{c_i} & v_{c_i} & 1 \end{bmatrix}^T$ ，使用逆深度来表示特征点在相机坐标系下的坐标：

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \frac{1}{\lambda} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (4-22)$$

使用逆深度作为优化参数主要是考虑到特征点的深度值在很大时，迭代优化时较为困难，并且逆深度更加服从高斯分布。例如场景深度可能为 0-20m，远处的特征点深度可能为 20m，但是近处的特征点深度不可能小于焦距，即深度不可能是 0，用原始的深度值表示的分布不会是一个对称的形状，不是高斯分布。但是用逆深度表示会以 1m 作为对称轴，深度大于 1m，则会在对称轴的左边，深度小于 1m，则会出现现在对称轴的右边。所以使用逆深度表示数值则是一个对称的形状，更近似高斯分布。

将 $\hat{P}_k^{c_i}$ 转换到第 j 帧相机 1 的相机坐标系下得到 $P_k^{c_j}$ ，整理得：

$$\begin{bmatrix} x_{c_j} \\ y_{c_j} \\ z_{c_j} \\ 1 \end{bmatrix} = T_{c_1}^b {}^{-1} T_{b_j}^{w-1} T_{b_i}^w T_{c_1}^b \begin{bmatrix} \frac{1}{\lambda} u_{c_i} \\ \frac{1}{\lambda} v_{c_i} \\ \frac{1}{\lambda} \\ 1 \end{bmatrix} \quad (4-23)$$

其中 $T_{c_1}^b$ 是相机 1 坐标系到 IMU 坐标系的变换矩阵, $T_{b_j}^w$ 是第 j 帧 IMU 坐标系到世界坐标系的变换矩阵, $T_{b_i}^w$ 是第 i 帧 IMU 坐标系到世界坐标系的变换矩阵。

特征点 P_k 在第 j 帧的相机 1 的相机坐标系下观测到的特征点的 $\hat{P}_k^{c_j}$ 坐标为 $\begin{bmatrix} u_{c_i} & v_{c_i} & 1 \end{bmatrix}^T$, 则此时同相机两帧间的点的重投影误差即是第 j 帧的两个特征点 $\hat{P}_k^{c_j}$ 与 $P_k^{c_j}$ 的坐标之差^[35], 整理得:

$$r_k = \begin{bmatrix} \frac{x_{c_j}}{z_{c_j}} - u_{c_j} \\ \frac{y_{c_j}}{z_{c_j}} - v_{c_j} \end{bmatrix} \quad (4-24)$$

两相机两帧间的点特征观测残差约束中的优化变量为滑动窗口中第 i 帧和第 j 帧的 IMU 在世界坐标系下的位置, 旋转和特征点 P_k 的逆深度, 即 $\begin{bmatrix} p_{b_i}^w & q_{b_i}^w & p_{b_j}^w & q_{b_j}^w & l_k \end{bmatrix}$ 。

4.3.2 两帧间两相机的点特征观测残差

在归一化相机坐标系下, 空间中第 k 个特征点 P_k 被第 i 帧的相机 1 和相机 2 观测到, 第 j 帧的相机 1 和相机 2 都观测到, 其中相机 1 在第 i 帧观测到特征点为 $\hat{P}_k^{c_i}$, 相机 2 在第 j 帧观测到特征点为 $\hat{P}_k^{c_j}$, 如图 4-4 所示。

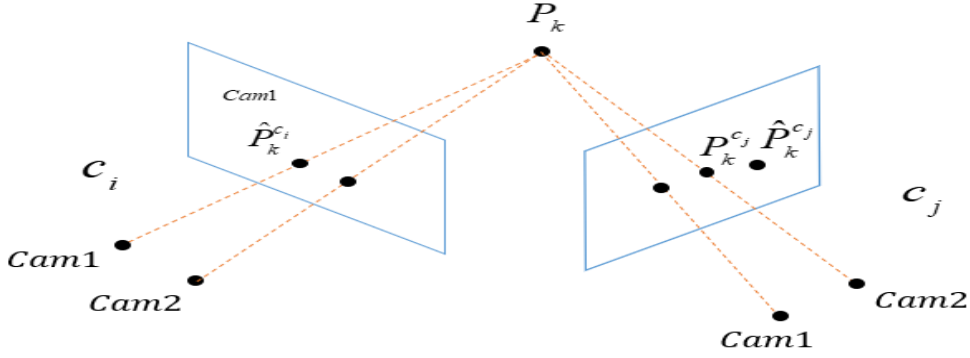


图 4-4 两相机间点重投影误差

将相机 1 在第 i 帧观测到的点 $\hat{P}_k^{c_i}$ 转换到在第 j 帧相机 2 的相机坐标系下得到 $P_k^{c_j}$ ，其中点 $\hat{P}_k^{c_i}$ 在第 i 帧的相机 1 的相机坐标系下坐标为 $\begin{bmatrix} u_{c_i} & v_{c_i} & 1 \end{bmatrix}^T$ ，使用逆深度来表示特征点在相机坐标系下的坐标为：

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \frac{1}{\lambda} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (4-25)$$

将 $\hat{P}_k^{c_i}$ 转换到第 j 帧相机 2 的相机坐标系下得到 $P_k^{c_j}$ ，整理得：

$$\begin{bmatrix} x_{c_j} \\ y_{c_j} \\ z_{c_j} \\ 1 \end{bmatrix} = T_{c_2}^b {}^{-1} T_{b_j}^{w-1} T_{b_i}^w T_{c_1}^b \begin{bmatrix} \frac{1}{\lambda} u_{c_i} \\ \frac{1}{\lambda} v_{c_i} \\ \frac{1}{\lambda} \\ 1 \end{bmatrix} \quad (4-26)$$

其中 $T_{c_2}^b$ 是相机 2 坐标系到 IMU 坐标系的变换矩阵， $T_{b_j}^w$ 是第 j 帧 IMU 坐标系到世界坐标系的变换矩阵， $T_{b_i}^w$ 是第 i 帧 IMU 坐标系到世界坐标系的变换矩阵， $T_{c_1}^b$ 是相机 1 坐标系到 IMU 坐标系的变换矩阵。

特征点 P_k 在第 j 帧的相机 2 的相机坐标系下观测到的特征点的 $\hat{P}_k^{c_j}$ 坐标为 $\begin{bmatrix} u_{c_i} & v_{c_i} & 1 \end{bmatrix}^T$ ，则此时两相机两帧间的点的重投影误差即是第 j 帧的两个特征点 $\hat{P}_k^{c_j}$ 与

$P_k^{c_j}$ 的坐标之差^[35], 整理得:

$$r_k = \begin{bmatrix} \frac{x_{c_j}}{z_{c_j}} - u_{c_j} \\ \frac{y_{c_j}}{z_{c_j}} - v_{c_j} \end{bmatrix} \quad (4-27)$$

两相机两帧间的点特征观测残差约束中的优化变量为滑动窗口中第 i 帧和第 j 帧的 IMU 在世界坐标系下的位置, 旋转和特征点 P_k 的逆深度, 即 $\begin{bmatrix} p_{b_i}^w & q_{b_i}^w & p_{b_j}^w & q_{b_j}^w & l_k \end{bmatrix}$ 。

4.3.3 一帧间两相机的点特征观测残差

在归一化相机坐标系下, 空间中第 k 个特征点 P_k 被第 i 帧的相机 1 和相机 2 观测到, 其中相机 1 在第 i 帧观测到特征点为 $\hat{P}_k^{c_i}$, 相机 2 在第 i 帧观测到特征点为 $\tilde{P}_k^{c_i}$, 如图 4-5 所示。

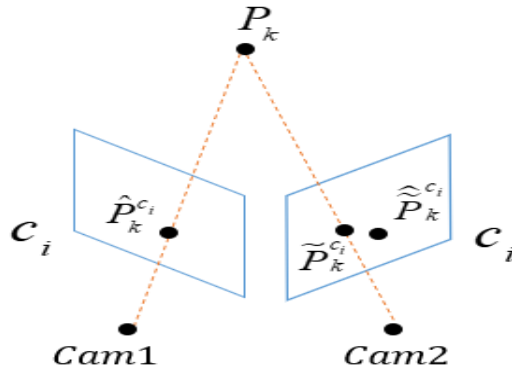


图 4-5 两相机一帧间的点重投影误差

将相机 1 在第 i 帧观测到的点 $\hat{P}_k^{c_i}$ 转换到在第 i 帧相机 2 的相机坐标系下得到 $\tilde{P}_k^{c_i}$, 其中点 $\hat{P}_k^{c_i}$ 在第 i 帧的相机 1 的相机坐标系下坐标为 $\begin{bmatrix} u_{c_i} & v_{c_i} & 1 \end{bmatrix}^T$, 使用逆深度来表示特征点在相机坐标系下的坐标为:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \frac{1}{\lambda} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (4-28)$$

将 $\hat{P}_k^{c_i}$ 转换到第 i 帧相机 2 的相机坐标系下得到 $\tilde{P}_k^{c_i}$ ，整理得：

$$\begin{bmatrix} x_{c_j} \\ y_{c_j} \\ z_{c_j} \\ 1 \end{bmatrix} = T_{c_2}^b {}^{-1} T_{c_1}^b \begin{bmatrix} \frac{1}{\lambda} u_{c_i} \\ \frac{1}{\lambda} v_{c_i} \\ \frac{1}{\lambda} \\ 1 \end{bmatrix} \quad (4-29)$$

其中 $T_{c_2}^b$ 是相机 2 坐标系到 IMU 坐标系的变换矩阵， $T_{c_1}^b$ 是相机 1 坐标系到 IMU 坐标系的变换矩阵。

特征点 P_k 在第 i 帧的相机 2 的相机坐标系下观测到的特征点 $\hat{P}_k^{c_i}$ 的坐标为 $\begin{bmatrix} u_{c_i} & v_{c_i} & 1 \end{bmatrix}^T$ ，则此时一帧间两相机的点的重投影误差即是第 i 帧的两个特征点 $\hat{P}_k^{c_i}$ 与 $\hat{P}_k^{c_j}$ 的坐标之差^[35]，整理得：

$$r_k = \begin{bmatrix} \frac{x_{c_j}}{z_{c_j}} - u_{c_j} \\ \frac{y_{c_j}}{z_{c_j}} - v_{c_j} \end{bmatrix} \quad (4-30)$$

一帧两相机间的点特征观测残差约束中的优化变量为滑动窗口中第 i 帧观测到的特征点 P_k 的逆深度 $1/z_k$ 。

4.4 线特征观测残差

4.4.1 线特征的观测模型和误差

空间中一条线段在不同的视角下观测到的长度可能并不相同，所以线的重投影误差不能从两幅图像中观测到的线段比较长短，即是空间中线的观测残差不能通过 2D-2D 的方式进行测量。但是空间中线的观测残差可以通过 3D-3D 的方式进行测量，即是两帧图像中观测到的线段都投影到三维空间，在三维空间中比较两条直线的距离。同时也可以通过 3D-2D 的方式进行比较，将 3D 空间中的直线投影到 2D 空间中，测量 2D 空间

中两条线段的距离^[37]。

本文使用 3D-2D 的方法测量线的观测残差，首先将空间中的直线重投影回当前图像归一化相机坐标系得到 l^c ，如图 4-6 所示。直线 \mathcal{L} 上的两点 C 和 D 分别投影到图像平面上成两点 c 和 d，其中空间中直线 \mathcal{L} 投影到图像平面得到线段 l 的投影公式^[36]推导如下：

$$\begin{aligned} c &= KC, d = KD \\ \mathcal{L} &= (n^\top, v^\top)^\top \\ n &= [C]_\times D \\ l &= [c]_\times d \\ l &= K'n \end{aligned} \quad (4-31)$$

其中 C, D 为空间中两个点， c, d 为图像中两个点， \mathcal{L} 为空间直线， l 为图像平面线段， n 为直线的法向量， v 为直线的方向向量， K' 为直线内参矩阵， K 为点内参矩阵。

由上式推导过程可以看出，投影到图像平面中的线段 l 只与空间中直线 \mathcal{L} 的法向量有关而与直线的方向向量无关。

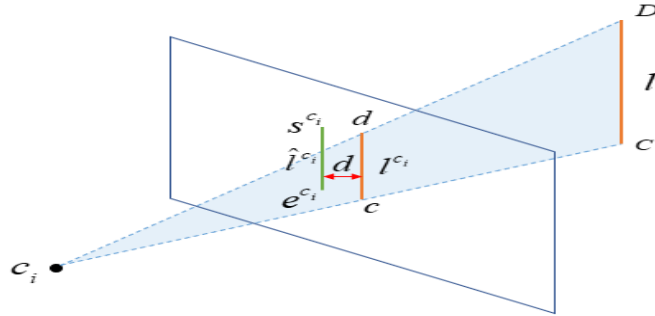


图 4-6 线的重投影误差

第 i 帧图像观测空间中的直线得到线段 \hat{l}^c_i ，观测到的线段可以由线段的两个端点表示， s^c_i 和 e^c_i ，此时，Bartoli 提出了多种直线的误差定义方式，首先是基于直线上的点和该直线点乘为 0，定义了代数距离的直线观测误差， $r = \begin{bmatrix} s^c_i l^c_i & e^c_i l^c_i \end{bmatrix}^T$ ，然后基于两条相同的直线叉乘为 0，定义了无几何意义的直线观测误差 $r = l^c_i \times \hat{l}^c_i$ ，最后基于两条相同的直线，一条直线上的点到另一条直线的距离应为 0，定义了几何意义的直线观测误差

^[38], $r = \left[\frac{s^{c_i} l^{c_i}}{\sqrt{l_1^2 + l_2^2}}, \frac{e^{c_i} l^{c_i}}{\sqrt{l_1^2 + l_2^2}} \right]^T$, 其中 $l^{c_i} = [l_1, l_2, l_3]^T$ 。本文使用基于几何距离定义的直线观测误差进行测量。首先将世界坐标系下的空间直线 \mathcal{L} 变换到相机坐标系 c_i 下, 然后将直线 \mathcal{L}^{c_i} 投影到相机归一化坐标系得到 l^{c_i} , 线的重投影误差定义为两条直线 l^{c_i} 和 \hat{l}^{c_i} 的距离,

根据基于几何意义的直线观测误差计算方式, 计算图像中观测到的线段 \hat{l}^{c_i} 的两个端点到 l^{c_i} 的几何距离, 整理得:

$$r_l(z_{\mathcal{L}^{c_i}}, \mathcal{X}) = \begin{bmatrix} d(s_l^{c_i}, l_l^{c_i}) \\ d(e_l^{c_i}, l_l^{c_i}) \end{bmatrix} \quad (4-32)$$

$$d(e_l^{c_i}, l_l^{c_i}) = \frac{e_l^{c_i \top} l_l^{c_i}}{\sqrt{l_1^2 + l_2^2}}$$

其中 s^{c_i} 和 e^{c_i} 是第 i 帧图像观测到的线段 l^{c_i} 的端点, \hat{l}^{c_i} 是第 i 帧图像观测到的线段, l^{c_i} 是空间中直线 \mathcal{L} 重投影到图像平面得到的线段^[39]。

线特征观测残差的优化变量为第 i 帧图像的位置, 旋转和直线 l 的正交表示法坐标 $\begin{bmatrix} p^{c_i} & q^{c_i} & \mathcal{O}_l \end{bmatrix}$ 。

4.4.2 线特征的误差雅可比推导

由线特征的误差观测模型推导, 可以得到线特征的投影误差的计算步骤, 即:

$$\mathcal{O}_l \rightarrow \mathcal{L}^w \rightarrow \mathcal{L}^{c_i} \rightarrow l^{c_i} \rightarrow r_l$$

由上式根据链式法则, 可以计算出线特征的观测误差相对于优化变量的雅可比矩阵, 整理可得:

$$J_l = \frac{\partial r_l}{\partial l^{c_i}} \frac{\partial l^{c_i}}{\partial \mathcal{L}^{c_i}} \begin{bmatrix} \frac{\partial \mathcal{L}^{c_i}}{\partial \delta x^i} & \frac{\partial \mathcal{L}^{c_i}}{\partial \mathcal{L}^w} \frac{\partial \mathcal{L}^w}{\partial \delta \mathcal{O}} \end{bmatrix} \quad (4-33)$$

其中 $\frac{\partial r_l}{\partial l^{c_i}}$ 为直线残差对图像坐标系内线段的导数, $\frac{\partial l^{c_i}}{\partial \mathcal{L}^{c_i}}$ 为图像坐标系内线段对相机坐标

系内直线的导数, $\frac{\partial \mathcal{L}^i}{\partial \delta x^i}$ 为相机坐标系内直线对优化变量的导数, $\frac{\partial \mathcal{L}^i}{\partial \mathcal{L}^w}$ 为相机坐标系内

直线对世界坐标系内直线的导数, $\frac{\partial \mathcal{L}^w}{\partial \delta O}$ 为世界坐标系内直线对直线正交参数的导数^[39]。

第一项 $\frac{\partial r_l}{\partial l^i}$ 的推导如下, 由于线特征的观测残差为:

$$r_l = \begin{bmatrix} \frac{s^{c_i T} l}{\sqrt{l_1^2 + l_2^2}} \\ \frac{e^{c_i T} l}{\sqrt{l_1^2 + l_2^2}} \end{bmatrix} = \begin{bmatrix} \frac{u_s l_1 + v_s l_2}{\sqrt{l_1^2 + l_2^2}} \\ \frac{u_e l_1 + v_e l_2}{\sqrt{l_1^2 + l_2^2}} \end{bmatrix} \quad (4-34)$$

$$s^{c_i} = [u_s \quad v_s \quad 1]$$

$$e^{c_i} = [u_e \quad v_e \quad 1]$$

$$l = [l_1 \quad l_2 \quad l_3]$$

其中 s^{c_i} , e^{c_i} 为线段的两个端点, l_1 , l_2 , l_3 为直线的三个系数^[39]。

则结合上式可以推得 $\frac{\partial r_l}{\partial l^i}$, 如下:

$$\frac{\partial r_l}{\partial l} = \begin{bmatrix} \frac{\partial r_1}{\partial l_1} & \frac{\partial r_1}{\partial l_2} & \frac{\partial r_1}{\partial l_3} \\ \frac{\partial r_2}{\partial l_1} & \frac{\partial r_2}{\partial l_2} & \frac{\partial r_2}{\partial l_3} \end{bmatrix} \quad (4-35)$$

第二项 $\frac{\partial l^i}{\partial \mathcal{L}^i}$ 的推导如下, 由于直线投影到图像平面只与直线的法向量有关, 则可以

得到:

$$l = K' n$$

$$\mathcal{L} = [n \quad v]$$
(4-36)

则结合上式可以推出 $\frac{\partial l^i}{\partial \mathcal{L}^i}$, 如下:

$$\frac{\partial l^i}{\partial \mathcal{L}^i} = \begin{bmatrix} \frac{\partial l}{\partial n} & \frac{\partial l}{\partial v} \end{bmatrix} = \begin{bmatrix} K' & 0 \end{bmatrix}_{3 \times 6} \quad (4-37)$$

第三项是由相机坐标系下线特征的观测误差对当前帧的平移和旋转的导数和空间中直线对直线的正交表示法的坐标的导数组成, 上式中状态变量的增量

$dx_i = [dp, dq, dv, db_a^b, db_g^b]$, 则线特征对状态变量中的旋转的导数为:

$$\frac{\partial \mathcal{L}_c}{\partial \delta \theta_b^b} = T_c^{b^{-1}} \begin{bmatrix} \left[R_b^{w^T} \left(n^w + [v^w]_{\times} p_b^w \right) \right]_{\times} \\ \left[R_b^{w^T} v^w \right]_{\times} \end{bmatrix}_{6 \times 3} \quad (4-38)$$

其中 $\frac{\partial \mathcal{L}_c}{\partial \delta \theta_b^b}$ 为相机坐标系下空间直线对优化变量 q 的导数, p_b^w 为当前帧 IMU 坐标系到

世界坐标系的平移, $R_b^{w^T}$ 为当前帧 IMU 坐标系到世界坐标系的旋转, T_c^b 为相机坐标系到 IMU 坐标系的变换矩阵^[39]。

线特征对状态变量中的平移的导数为:

$$\frac{\partial \mathcal{L}_c}{\partial \delta p_b^b} = T_c^{b^{-1}} \begin{bmatrix} R_b^{w^T} [v^w]_{\times} \\ 0 \end{bmatrix}_{6 \times 3} \quad (4-39)$$

空间中直线对直线的正交表示法的坐标的导数如下:

$$\begin{aligned} \frac{\partial \mathcal{L}^w}{\partial \delta \mathcal{O}} &= \begin{bmatrix} \frac{\partial \mathcal{L}^w}{\partial \psi_1} & \frac{\partial \mathcal{L}^w}{\partial \psi_2} & \frac{\partial \mathcal{L}^w}{\partial \psi_3} & \frac{\partial \mathcal{L}^w}{\partial \phi} \end{bmatrix} \\ &= \begin{bmatrix} 0 & -w_1 u_3 & w_1 u_2 & -w_2 u_1 \\ w_2 u_3 & 0 & -w_2 u_1 & w_1 u_2 \end{bmatrix}_{6 \times 4} \end{aligned} \quad (4-40)$$

其中, $\frac{\partial \mathcal{L}^w}{\partial \psi_1}$, $\frac{\partial \mathcal{L}^w}{\partial \psi_2}$, $\frac{\partial \mathcal{L}^w}{\partial \psi_3}$, $\frac{\partial \mathcal{L}^w}{\partial \phi}$ 是空间中直线对 4 个正交参数的导数, u_i 和 w_i 分别是正交表示法的第 i 列坐标^[39]。

4.5 本章小结

本章主要介绍了基于点线特征的双目视觉 IMU 紧耦合优化。分析了滑动窗口优化算法、边缘化与关键帧筛选策略, IMU 观测残差, 两帧间同相机的点特征观测残差, 两帧间两相机的点特征观测残, 一帧间两相机的点特征观测残差和线特征观测残差原理。其中详细介绍了优化目标的构成, 优化变量的更新方式, 边缘化原理, 关键帧的筛选策略, IMU 预积分和 IMU 观测残差的推导, 两帧间同相机的点特征重投影误差定义, 两帧间两相机的点特征重投影误差定义, 一帧间两相机的点特征重投影误差定义, 线特征投影图像平面推导、线特征的重投影误差定义与线特征的雅可比推导。

第五章 实验与分析

5.1 基于点线特征的双目视觉惯性 SLAM 系统搭建

本文搭建了基于点线特征的双目视觉惯性 SLAM 系统，包括软件系统和硬件平台。其中硬件平台基于四轴飞行器搭建，包括双目 IMU 视觉惯性模组，Avnet Ultra96-V2。软件系统包括前端和后端，前端用于处理 IMU 和双目图像数据，后端基于点线特征的双目视觉 IMU 紧耦合优化估计相机位姿。

5.1.1 硬件平台

本文搭建的硬件平台由双目相机、IMU、FPGA、网络通信模块、ZYNQ MPSOC 等模块组成。FPGA 同步触发采集两个摄像头图像，同时采集 IMU 数据，再通过时间同步模块将相机时间戳和 IMU 时间戳同步，然后基于 UDP 协议将图像 IMU 数据传至 Avnet Ultra96-V2 板中。Avnet Ultra96-V2 板的 PS 端接收数据，使用 UDP 数据采集线程和图像 IMU 数据解析线程两个线程对数据进行解包，然后将图像 IMU 数据传至基于点线特征的 VIO 系统中。SLAM 硬件平台框架如图 5-1 所示。

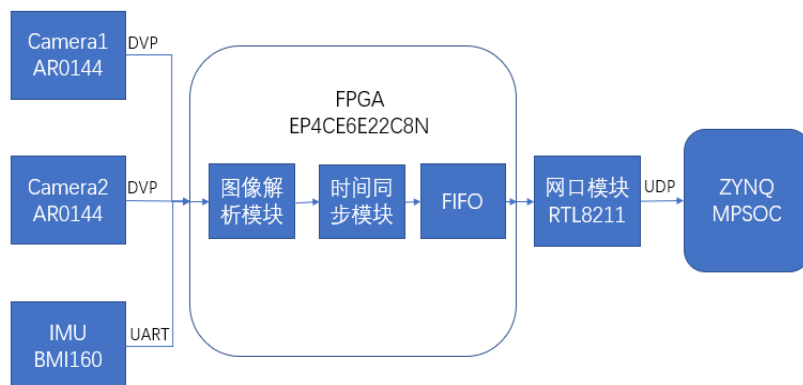


图 5-1 硬件平台框架

本文选用 Avnet Ultra96-V2 作为图像处理平台，如图 5-2 所示，Ultra96-V2 是基于 Arm 的 Xilinx Zynq UltraScale +MPSoc 开发板。Ultra96-V2 提供 40 引脚低速扩展接头用于连接传感器模块，同时具有 2xUSB3.0、1xUSB2.0A 型下游端口和 1xUSB3.0 型 Micro-B 上游端口提供给外设，具有 Microchip Wi-Fi /蓝牙调试模块方便调试程序，可供下载的 SD 卡映像，提供基于 linaro 文件系统的 LINUX OS。Ultra96-V2 是开源的 linux 开

发平台，提供了 Linaro 操作系统、硬件设备和软件驱动。Avnet Ultra96-V2 采用了 Xilinx Zynq UltraScale+ MPSoC ZU3EGA484 芯片，EG 器件采用运行速率高达 1.5GHZ 的四核 ARM® Cortex-A53 平台，与双核 Cortex-R5 实时处理器、Mali-400 MP2 图形处理单元及 16nm FinFET 与可编程逻辑相结合以实现大规模的并行计算能力。



图 5-2 Avnet Ultra96-V2

本文的硬件系统基于四轴飞行器搭建，如图 5-3 所示，四轴飞行器提供 12V 电源供给 Avnet Ultra96-V2 板，双目 IMU 视觉惯性模组通过 USB 转接器接入到 Avnet Ultra96-V2 板，在 Avnet Ultra96-V2 平台上搭建 Linux 环境，在其 SD 卡上烧录 Linaro 文件系统，实现 Avnet Ultra96-V2 板的 4 核 A53 的 ARM 芯片和 PC 的 X86 Linux 的交叉编译，在 Linaro 的系统上装载双目 IMU 视觉惯性模组驱动。



图 5-3 硬件平台实物图

5.1.2 软件系统

(1) 前端

SLAM 系统的精度高低很大程度上取决于前端性能的好坏，所以一个 SLAM 系统前端的设计是重中之重。前端从图像中提取点特征和线特征并追踪，同时对 IMU 进行预积分。

前端分两个线程独立检测点特征和线特征，当一帧新图像来时，使用 KLT 稀疏光流算法^[40]对上一帧已经观测到的特征点进行追踪，剔除跟踪上一帧失败的特征点和超出当前帧图像边界的点，同时使用 RANSAC 的方法通过基本矩阵 F 剔除外点，剔除外点后，重新更新所有当前帧的所有特征点的 ID 编号，同时当特征点数量不够时重新在当前帧提取 FAST 角点^[41]，然后对当前帧检测到的特征点进行畸变矫正，投影到当前帧的归一化相机坐标系。

当新的一帧图像来时，使用 LSD 检测器检测当前帧图像中的线段，并提取线段的 LBD 描述子，然后根据 LBD 描述子与上一帧图像中的线段进行匹配，剔除跟踪失败的线段和线段端点超出图像边界的线端，对线段的两个端点用 RANSAC 的方法通过基本矩阵 F 剔除外点，剔除外点后，重新更新所有特征点的 ID 编号，同时当线段数量不够时重新使用 LSD 检测器提取新的线段，然后对线段的两个端点进行畸变矫正，投影到归一化图像坐标系。线段的检测和匹配都基于 OpenCV3 实现^[42]，前端的程序流程图 5-4 所示。

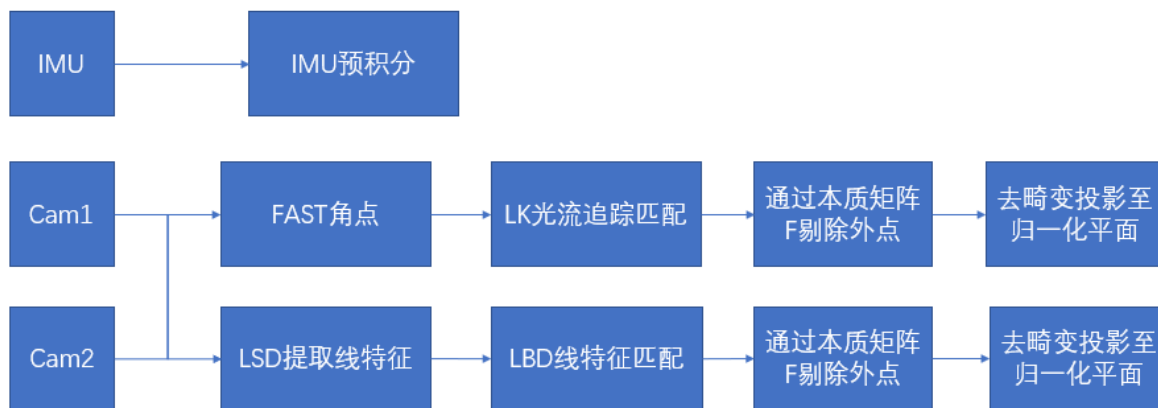


图 5-4 前端流程图

（2） 后端

当后端接收到前端处理后的当前帧的点特征和线特征后，通过本文设计的关键帧筛选机制判断第二最新帧是否为关键帧，本文提出的关键帧选取策略将 IMU 信息与视觉信息相融合，通过 IMU 积分出的平移将相机当前的运动分为静止状态，平移运动状态，纯旋转运动状态。对于 IMU 积分出的平移 P 较小的帧，我们认为他是静止状态或者纯

旋转状态,同时再根据当前最新帧与上一帧的视差量是否大于某个阈值,并且 IMU 积分出的旋转 Q 不能超过某个阈值去确定当前帧是否是关键帧,这样可以有效地排除掉静止状态或者纯旋转状态的帧加入到滑动窗口中。对于 IMU 积分出的平移 P 较大的帧,我们认为他是平移运动状态,同时再根据当前最新帧与上一帧的视差量是否大于某个阈值去确定当前帧是否是关键帧,这样平移较大的帧就可以有效地加入到滑动窗口中,避免了将平移较小的帧加入到滑动窗口中造成不必要的计算。根据本文设计的关键帧的筛选策略有效的排除了纯旋转或者静止情况下将图像加入到关键帧中,优先将平移较大的帧加入到滑动窗口中。

然后判断系统是否已经初始化选择进行系统初始化,首先在静止状态下计算第 0 帧的 IMU 在世界坐标系下的方向 R_b^w ,然后将第 0 帧以内的所有加速度平均值 g^b 转到世界坐标系下 g^w ,即通过 $g^w = q_b^w g^b$,求得 q_b^w 。然后滑动窗口中的每一帧的旋转和平移通过视觉 PnP 计算,而不是 IMU 预积分去计算,最后通过由视觉得到的 $q_{b_{k+1}}^{b_k}$ 和 IMU 预积分得到的 $I_{b_{k+1}}^{b_k}$ 去计算陀螺仪的偏置 b_w ,在初始化过程中,并没有计算加速度计偏置,因为加速度计的偏置与重力耦合,系统的初始加速度数值相对于重力加速度较小,所以加速度计偏置在初始化的时候一般不能求出,因此在初始化的时候并不计算加速度计偏置。

在系统初始化后,首先对当前帧中追踪上最近几帧刚观测到的特征点进行三角化得到点的初始深度,对于已经被很多帧观测到并且追踪上的点,由于在前面几帧已经进行过三角化,并且在滑动窗口中优化过了,所以当前帧并不对这些特征点进行重新三角化,对于刚刚被当前帧观测到的特征点,也不能三角化得到深度。

同样对线特征,首先对当前帧中追踪上最近几帧刚观测到的线特征进行构造普吕克矩阵,得到普吕克坐标,对于已经被很多帧观测到并且追踪上的线特征,由于在前面几帧已经进行过三角化,并且在滑动窗口中优化过了,所以当前帧并不对这些线特征进行重新三角化,对于刚刚被当前帧观测到的线特征,也不能三角化得到深度。

为了限制滑动窗口的大小,根据第二最新帧是否是关键帧判断边缘化第二最新帧还是最老帧,边缘化第二最新帧只会添加上一次边缘化的先验约束,而视觉约束即点约束和线约束会直接丢掉,最新帧的 IMU 信息会保留但不会边缘化。边缘化最老帧会将上一次边缘化的先验约束,第 0 帧到第 1 帧的 IMU 约束,第一帧观测到的视觉约束即点

约束和线约束都进行边缘化。

随后进行基于点线特征的双目视觉 IMU 紧耦合优化，优化目标为先验观测残差，IMU 观测残差，两帧间同相机的点特征观测残差，两帧间两相机的点特征观测残，一帧间两相机的点特征观测残差和线特征重投影误差的马氏距离之和。使用基于滑动窗口的非线性优化算法最小化目标函数，通过最小化目标函数来得到系统的最大后验估计，使用 Ceres 优化库中的 Levenberg-Marquard 算法进行非线性优化^[43]，求得优化变量的增量，然后对优化变量进行更新迭代，找到优化变量最优值使得目标函数最小，即得到系统的最大后验估计。

最后，对优化后的点特征和线特征进一步筛选，计算被多帧追踪观测到的特征点的平均重投影误差，如果大于 3 个像素点则剔除，同样对优化后的线进一步筛选，计算多帧追踪观测到的线的端到直线距离的平均重投影误差，如果大于 3 个像素点则剔除。后端的程序流程图如图 5-5 所示。

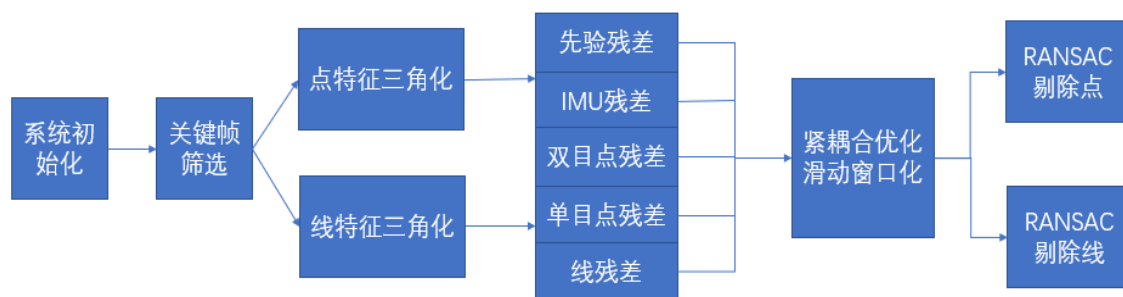


图 5-5 后端流程图

5.2 基于 EuRoC 数据集的实验

EuRoC 数据集是由飞行器在室内采集双目和 IMU 的数据集，包括了两个场景，苏黎世联邦理工学院 ETH 的机器房内和一个贴有标定板的普通房间^[44]，如图 5-6 所示，飞行器的双目相机型号为 MT9V034，相机的采集频率为 20HZ，IMU 的型号为 ADIS16448,IMU 的采集频率为 200HZ，具有相机和 IMU 硬件同步功能，同时硬件设备配有维肯动作捕捉系统 Vicon motion capture system，通过在 MAV 上贴上一组反射标志测量，帧率达到 100HZ，精度在毫米级^[44]。基于 EuRoC 数据集的实验在 2.8GHZ Intel Core i7-7700HQ CPU，8 GB RAM 和 ROS Kinetic^[45]的计算机上进行的。



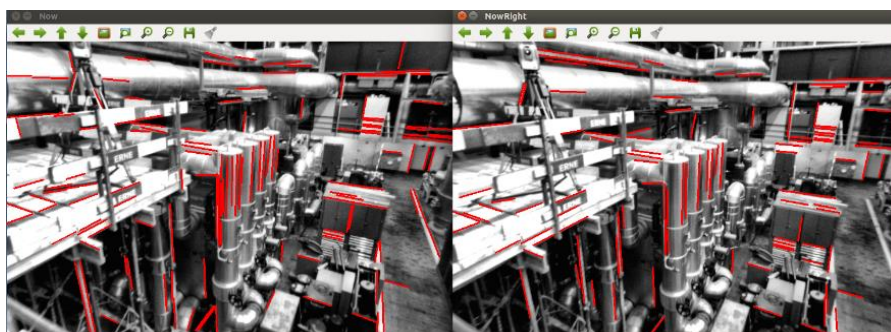
a) 机房



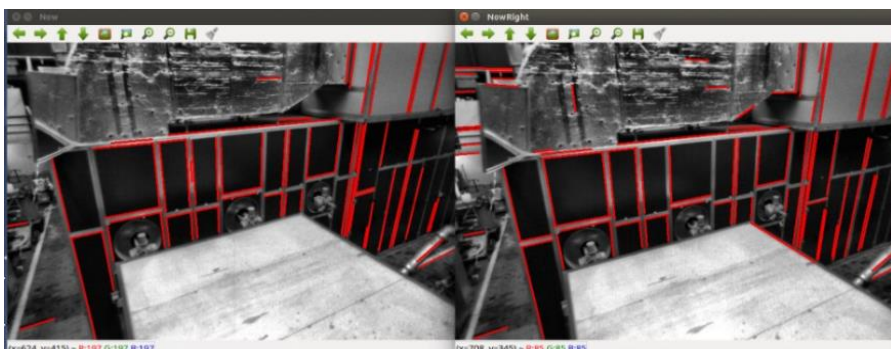
b) 普通房间

图 5-6 EuRoC 数据集

本文分别对左右双目图像提取线特征，如下图 5-7 所示，由图可以看到，线特征会比点特征提供更多的空间环境结构信息，点特征在图像中主要分布在图像的角点处，四周有明显的亮度明暗变化，线特征在图像中主要分布在物体的边缘处，例如图像中管道的边缘，黑色的瓷砖，箱子的边缘，支撑的木板等等。由图可以看出，在结构化场景中能够提取出较多的直线特征，且线特征在图像中会比点特征占据更多的像素，在空间中也会表现出更大的几何特征，因此将点特征与线特征结合，可以提供一个更加直观的结构环境信息。



a) 管道上的双目线特征



b) 墙壁上的双目线特征

图 5-7 双目线特征

本文对灯光较暗环境下，点特征和线特征的跟踪效果进行对比，如图 5-8 所示。四幅图的中间图像描述了点特征的跟踪效果，绿色的点代表当前图像新生成的特征点，红色的点代表当前图像中追踪到上一帧的特征点。由图可以看出，在光线较暗的情况下，点特征的跟踪效果很差，只有图像的右上角有十几个特征点跟踪上了上一帧图像，且角点选取的位置较为集中，不利于系统进行位姿估计。由图可以看出，线特征在光线较暗的环境跟踪效果比较理想，大量的木板边缘被识别成线特征，且提取的线特征均匀分布在整幅图像上。

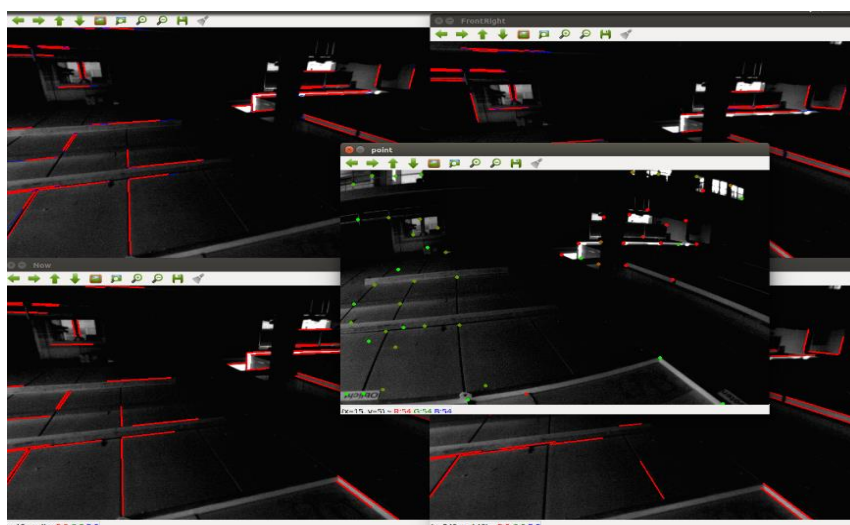


图 5-8 点线特征追踪效果对比

本文对目前几种主流 VIO 开源算法进行对比，分别是 VINS-MONO, VINS-fusion 和 MSCKF_VIO。其中 VINS-MONO 是 HKUST 的沈老师团队开源的一种单目 VIO 算法，具有在线标定相机 IMU 外参，鲁棒的初始化过程，回环检测与重定位等特点，是目前性能卓越的基于紧耦合优化的单目视觉里程计。VINS-fusion 是 VINS-MONO 的双目版本，支持了双目摄像头的接入，具有双目，单目 IMU, 双目 IMU 三种接口，同时引入了 GPS 约束，是一种较为完善的 SLAM 系统。MSCKF 是一种流行的基于 EKF 框架下的进行 IMU 和视觉信息融合的 VIO 方法，MSCKF 在其状态向量中维护了 20 个左右的相机位姿与当前 IMU 的状态，摒弃了传统的将特征点加入到状态向量中的方法，使用特征点不被当前帧观测到就进行观测更新的机制，提高了系统的运算效率，且有较高的精度，相比于 VINS-MONO, VINS-fusion 优化算法，MSCKF 的计算速度很快，精度相当。

本文对 VINS-MONO, VINS-fusion 和 MSCKF_VIO 四种算法进行横向对比，使用绝对位姿误差 APE 作为四种算法的评估指标，评估了四种算法在 EuRoC 数据集上的位置

误差。绝对位姿误差是通过计算预估轨迹和真实轨迹的相对距离,飞机的估计位姿为 T_i' , 真实轨迹为 T_i , 则绝对位姿误差 APE^[46]为:

$$APE = T_i (T_i')^{-1} \quad (5-1)$$

然后对所有时刻的均方根误差,由于位置误差受旋转误差影响,故本文通过评估位置误差即可以分析系统性能,整理得:

$$RMSE(APE) = \left(\frac{1}{n} \sum_{i=1}^n (\|trans(APE_i)\|^2) \right)^{1/2} \quad (5-2)$$

本文使用 EVO 来评估四种算法的结果,同时显示系统的轨迹,如表 5-1 所示,其中本文提出的基于点线特征的双目视觉惯性 SLAM 算法在 EuRoC 的 MH_01_easy, MH_02_easy, MH_05_difficult, V1_01_easy 数据集上表现最佳。可以看到,MY-VIO 在相机运动较慢的数据集表现最佳,位置平均误差在 13cm 左右,优于 VINS-MONO,VINS-Fusion 和 MSCKF 等 3cm 以上。而在相机运动较快的数据集上,基于点线特征的双目视觉惯性 SLAM 算法的平均位置误差为 30cm,优于 VINS-Fusion,和 MSCKF 效果相当,略低于 VINS-MONO。本文的算法是基于 VINS-fusion 上改进优化的,相比于 VINS-fusion 双目版本,本文提出的 SLAM 算法具有更好的性能提升,本文的算法在 EuRoC 数据集上的位置平均误差为 20.15cm,VINS-Fusion 在 EuRoC 数据集上的位置平均误差为 26.25cm,MY-VIO 优于 VINS-Fusion 达到 6cm,提高了 30%的定位精度。由表 5-1 的结果可以说明,线特征和点特征的信息融合可以更好的提升 VIO 系统的定位性能,并且能为机器人提供更加直观的几何结构信息。

表 5-1 四种算法性能对比 (m)

数据集编号	MY-VIO	VINS-Fusion	VINS-MONO	MSCKF
MH_01_easy	0.123443	0.246977	0.155931	0.156265
MH_02_easy	0.144239	0.203462	0.178405	0.169544
MH_03_medium	0.252820	0.285297	0.194973	0.278652
MH_04_difficult	0.351471	0.422478	0.346134	0.338994
MH_05_difficult	0.279661	0.309305	0.303196	0.395397
V1_01_easy	0.065840	0.116945	0.088909	0.122988

由于本文是在 VINS-fusion 上搭建，所以本文在 MH_05_difficult 数据集上对 VINS-fusion 进行了详细对比，同样使用 EVO 进行轨迹的显示，计算 XYZ 三个方向和真实轨迹的位置误差以及 Yaw, Pitch, Roll 三个角度和真实轨迹的角度误差。如图 5-9 所示。其中 MY-VIO 是蓝色实线，VINS-Fusion 是绿色实线，真实轨迹是灰色虚线，从图中可以看出，蓝色实线比绿色实线更贴近灰色虚线，即蓝色实线具有更好的追踪效果。同时，蓝色实线的 X、Y、Z 三轴的跟踪效果和 Yaw、Pitch、Roll 的效果均优于绿色实线。

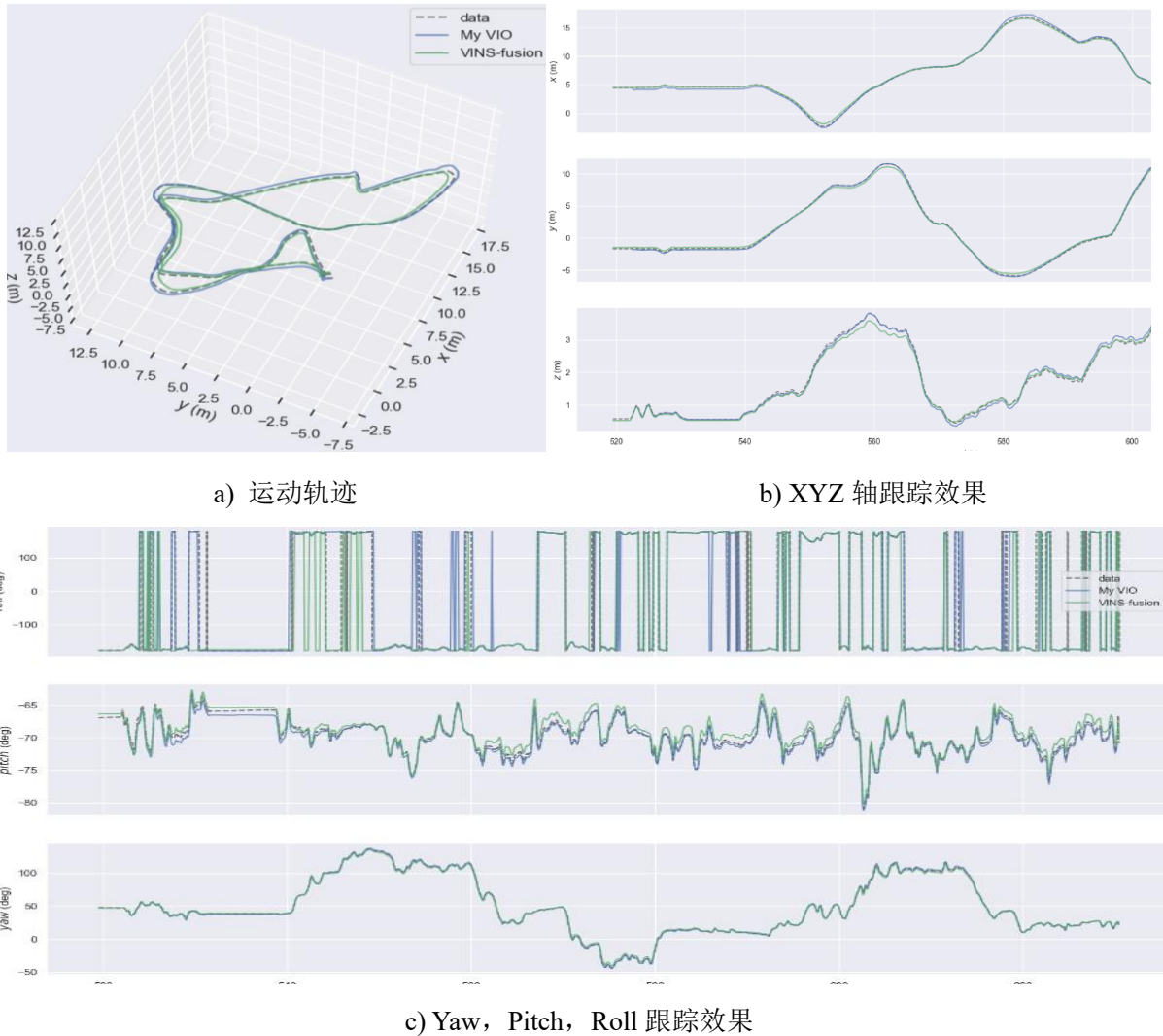


图 5-9 MY-VIO 与 VINS-fusion 性能对比

线特征在图像中比点特征占据了更多的像素点，在空间中，线特征相比点特征提供更多的空间信息，具有更多的几何表达能力，更有利于飞行器进行定位导航，路径规划。如图 5-10 所示，本文对提供的具有线特征的 3D 地图具有更多的几何信息，对语义地图构建，和机器人路径规划具有很重要的参考价值。

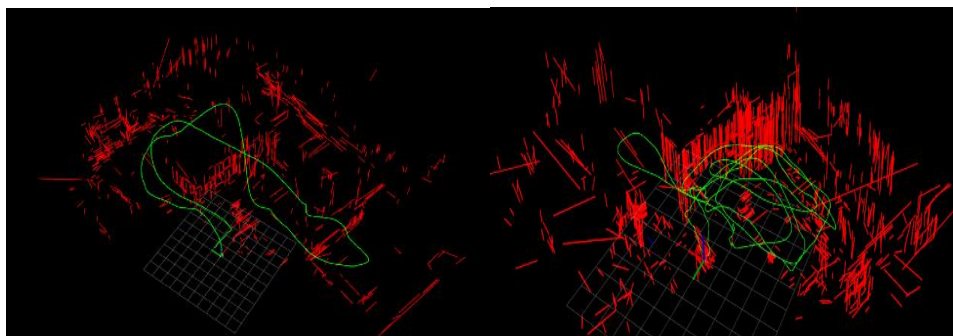


图 5-10 线特征地图

5.3 基于室外环境的实验

为了验证本文提出的 VIO 算法效果和硬件平台的性能，本文进行室外实验。将 AR0144 的曝光方式配置为自动全局曝光模式，相机触发模式配置为自动触发模式，相机分辨率配置为 1280*720，相机镜头焦距为 2.6mm，视场角宽为 95 度，高为 50 度，双目基线距离为 12cm。本文通过 Kalibr 标定工具箱对双目惯性模组进行标定，标定板选用 Aprilgrid，Aprilgrid 标定板能够提供每个标定块的序号，有效的防止了前后帧匹配标定块出现错行或错列的情况，Aprilgrid 标定板如图 5-11 所示。为了使陀螺仪的三个轴得到充分激励，在采集图像和 IMU 数据时，将相机镜头对着 Aprilgrid 标定板依次绕三个轴旋转，同时为了让加速度三个轴得到充分激励，将相机镜头对着 Aprilgrid 标定板在 XYZ 三个方向上匀速运动，最后将相机镜头对着 Aprilgrid 标定板在空中画个 8 字完成数据采集。

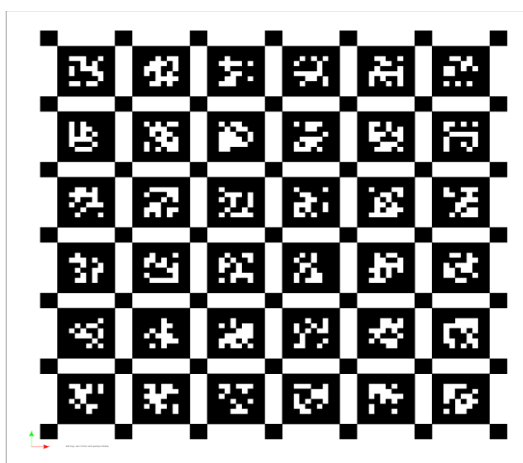


图 5-11 Aprilgrid 标定板

本文通过 Kalibr 工具箱标定出相机 0 内参、相机 1 内参、相机 0 坐标系到 IMU 坐标系的变换矩阵，相机 1 坐标系到 IMU 坐标系的变换矩阵和 IMU 的参数。其中相机焦

距和缩放系数的乘积 f_x 与 f_y 为 891, 相机原点的平移距离 c_x 与 c_y 分别为 691 与 283, 加速度计测量噪声标准差 n_a 为 0.3, 陀螺仪测量噪声标准差 n_w 为 0.08, 加速度计偏置随机游走噪声标准差 n_{b_a} 为 0.015, 陀螺仪偏置随机游走噪声标准差 n_{b_w} 为 $5.2e-5$, 当地重力加速度 g 为 9.788。相机内参和相机 0 坐标系到 IMU 坐标系的变换矩阵如下:

$$K = \begin{bmatrix} 891 & & 691 \\ & 891 & 283 \\ & & 1 \end{bmatrix}$$

$$T = \begin{bmatrix} 0.012 & -0.016 & 0.999 & 0.213 \\ 0.998 & 0.056 & -0.011 & 0.069 \\ -0.056 & 0.998 & 0.017 & 0.060 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5-3)$$

本文室外实验场地为篮球场, 实验时间为中午, 太阳光线较强, 所以整幅图片偏白。操场具有篮球架, 白色线条和路灯, 四周具有车辆、树木和楼房, 为 SLAM 系统提供了丰富的线特征。飞行器从中间圆圈处起飞, 在操场上以平均 1.5m/s 的速度飞行, 飞行距离为 230m。现场的环境如图 5-12 所示。

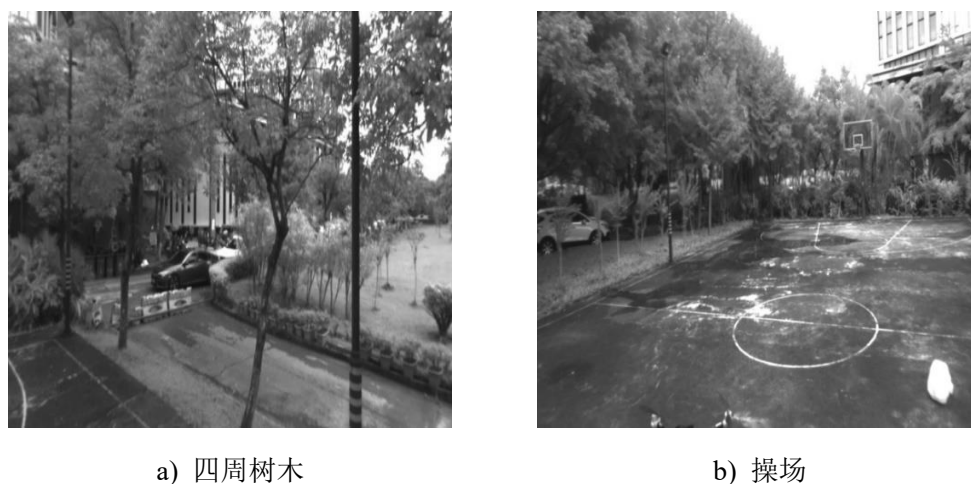


图 5-12 室外实验场景

本文 SLAM 系统在室外环境下提取的线特征如图 5-13 所示, 在太阳光强烈的场景下, 即使 AR0144 相机开启了自动曝光, 也可以看出整幅图像偏白, 此时的提取的点特征较为稀疏, 在大楼上很难提取到点特征。从图中可以看出, 由于线特征具有光照不变性和视角不变性, 在图中大楼处可以提取出较多的线段。此时单独使用点特征进行系统

位姿估计，由于点特征过于稀疏，系统容易跟丢，单独使用线特征进行系统位姿估计，由于线特征过于集中在图像的右上角，系统位姿估计不准确。本文将点特征与线特征进行融合，提供给 SLAM 系统更多的视觉信息，以保证系统的精度与鲁棒性。



图 5-13 室外场景线特征

本文以飞行器的 RTK 信息作为真实轨迹与 SLAM 算法轨迹进行对比，如表 5-2 所示，同样对比平移误差，总共飞行 230m，基于点线特征的双目视觉惯性 SLAM 系统的平移误差为 0.53m，VINS-Fusion 的平移误差达到了 0.76m，VINS-MONO 的平移误差为 0.61m，MSCKF 的平移误差为 0.72m。由于环境中光线较强，相机图像的上半部分有点过曝，提取的点特征较为稀疏，都在图像的下半部分，且操场四周的树上提取的点特征不稳定，而环境中存在大量稳定的线特征，比如楼房，篮球架，路灯，此时本文提出的基于点线特征的双目视觉惯性系统表现最恰，超过了其他三种算法。在本次室外实验中，本文提出的 SLAM 算法优于 VINS-Fusion 达到了 0.23m，提高了系统 30%的定位精度。由于本文搭建的硬件系统使用的 IMU 性能远低于 EuRoC 数据集使用的 ADIS16448，IMU 的加速度计噪声较大，因而室外实验的精度整体低于 EuRoC 数据集。

表 5-2 四种算法性能对比

数据集编号	MY-VIO	VINS-Fusion	VINS-MONO	MSCKF
室外实验	0.529654	0.763540	0.606235	0.720785

本文同样使用 EVO 对实验结果进行评估与显示，轨迹如图 5-14 所示。其中 MY-VIO 是蓝色实线，VINS-Fusion 是绿色实线，真实轨迹是灰色虚线，从图中可以看出，蓝色实线比绿色实线更贴近灰色虚线，即蓝色实线具有更好的追踪效果。由下图可以看出，起飞时平移误差较大，其后轨迹和 RTK 轨迹基本吻合，在 X，Y，Z 轴轨迹上，蓝色实线的追踪效果也比绿色实线表现要好。综上所述，本文搭建的基于点线特征的双目

VIO 系统在硬件方面具有很高的可靠性、稳定性，在软件方面具有很高的精度和鲁棒性。

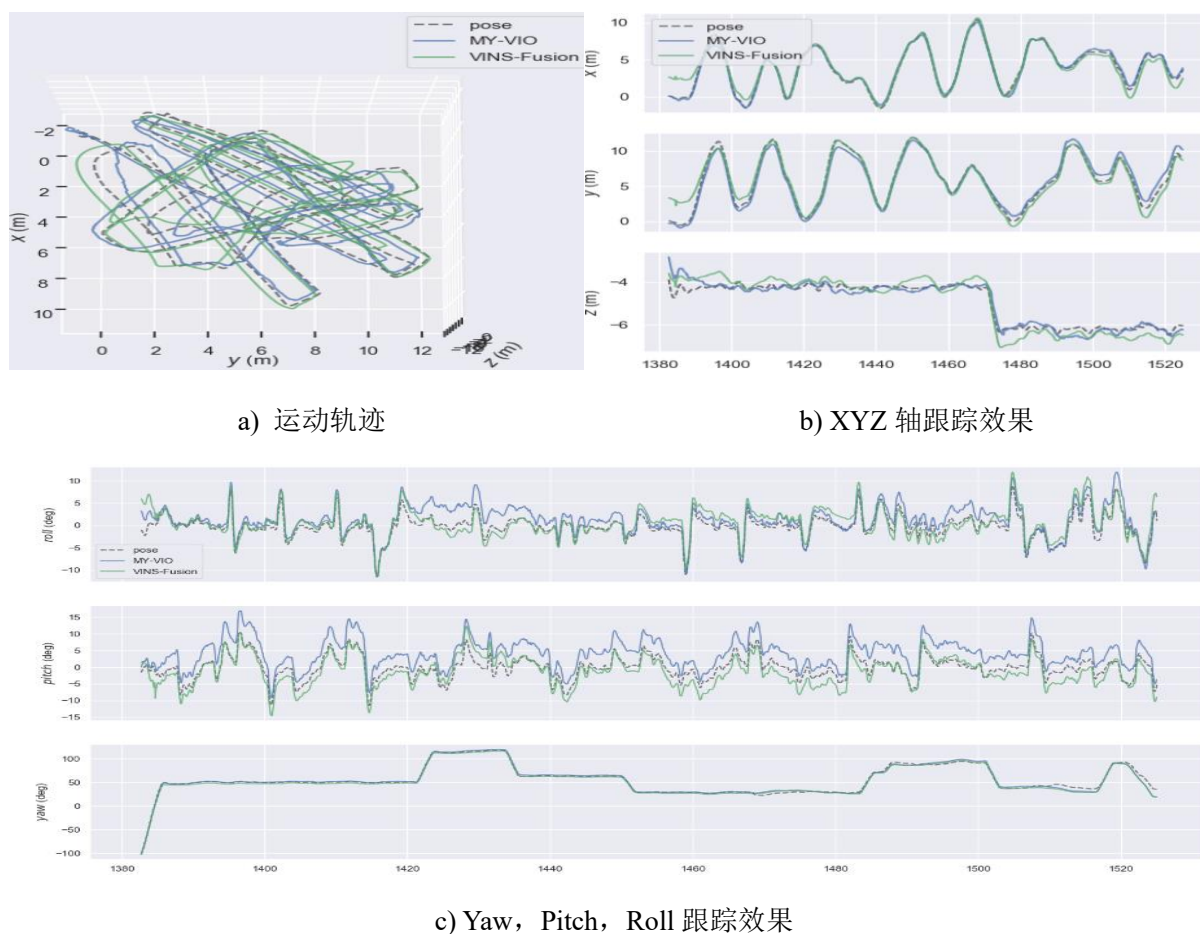


图 5-14 室外场景轨迹对比

5.4 本章小结

本章首先介绍了基于点线特征的双目视觉惯性 SLAM 系统的硬件平台搭建和软件系统的组成，为了验证本文提出的基于点线特征的双目视觉惯性 SLAM 算法效果和硬件平台的性能，分别进行了基于 EuRoC 数据集的实验和基于室外环境的实验。在基于 EuRoC 数据集的实验中，本文对比了 VINS-MONO, VINS-Fusion 和 MSCKF_VIO 三种主流 VIO 算法，表明了本文提出的基于点线特征的双目 VIO 系统具有较高的定位精度和系统鲁棒性。在室外环境的实验中，将飞行器的 RTK 信息作为基准信息，同样对比了 VINS-MONO, VINS-Fusion 和 MSCKF_VIO 三种主流 VIO 算法，表明了本文搭建的基于点线特征的双目 VIO 系统在硬件方面具有很高的可靠性、稳定性，在软件方面具有很高的精度和鲁棒性。

总结与展望

1、工作总结

本文搭建了基于点线特征的双目视觉惯性 SLAM 系统，对点特征、线特征和 IMU 信息融合进行深入研究，通过联合优化点特征，线特征和 IMU 三者的约束进行相机位姿估计，用来研究结构化环境中点特征、线特征和 IMU 信息融合对 SLAM 系统定位精度的影响。

本文完成的工作如下：

(1) 搭建了双目 IMU 视觉惯性模组。描述了模组的性能要求，对模组选型与设计，从元件选型、电路图设计到硬件框架对模组做一个整体介绍；然后就双目 IMU 视觉惯性模组驱动设计部分，本文讲解了 FPGA 驱动和图像 IMU 数据采集驱动的功能与设计实现，仔细分析了图像 IMU 数据采集驱动的 UDP 数据采集线程和图像 IMU 数据解析线程的设计思路，最后对双目 IMU 视觉惯性模组的性能进行测试，验证了双目 IMU 视觉惯性模组具有较高的时间同步性与成像稳定性。

(2) 分析了 SLAM 中的线特征原理，描述了图像中线段的提取方法，描述子的生成方法和空间直线的参数化方法。其中详细介绍了使用 LSD 方法提取图像中线段的步骤和 LBD 描述子的生成方法，然后仔细分析了空间直线的普吕克表示方法以及正交表示法的原理和优缺点。

(3) 本章主要介绍了基于点线特征的双目视觉 IMU 紧耦合优化。分析了滑动窗口优化算法、边缘化与关键帧筛选策略，IMU 观测残差，两帧间同相机的点特征观测残差，两帧间两相机的点特征观测残差，一帧间两相机的点特征观测残差和线特征观测残差原理。其中详细介绍了优化目标的构成，优化变量的更新方式，边缘化原理，关键帧的筛选策略，IMU 预积分和 IMU 观测残差的推导，两帧间同相机的点特征重投影误差定义，两帧间两相机的点特征重投影误差定义，一帧间两相机的点特征重投影误差定义，线特征投影图像平面推导、线特征的重投影误差定义与线特征的雅可比推导。

(4) 本章首先介绍了基于点线特征的双目视觉惯性 SLAM 系统的硬件平台搭建和软件系统的组成，为了验证本文提出的基于点线特征的双目视觉惯性 SLAM 算法效果和硬件平台的性能，分别进行了基于 EuRoC 数据集的实验和基于室外环境的实验。在

基于 EuRoC 数据集的实验中，本文对比了三种主流 VIO 算法，在室外环境的实验中，将飞行器的 RTK 信息作为基准信息，同样对比了三种主流 VIO 算法，表明了本文搭建的基于点线特征的双目 VIO 系统在硬件方面具有很高的可靠性、稳定性，在软件方面具有很高的精度和鲁棒性。

2、工作展望

本文搭建的基于点线特征的双目视觉惯性 SLAM 系统还有待进一步完善，可以从以下几个方面进行展开研究：

（1）本文使用的线特征 LSD 提取方法和 LBD 描述子匹配方法的耗时较高，考虑使用 FPGA 加速线特征的提取和描述子的提取。

（2）采用环境中人造结构上的直线代替环境中的普通直线，在曼哈顿世界^[47]中，存在着大量人造的水平竖直直线，例如建筑物上的纹理直线，地板上的纹理直线等等，他们基本上都是平行的，筛选出这些直线，根据这些结构化的直线进行位姿估计。

（3）本文构建了带有环境结构信息的线特征地图，在此基础上构建语义地图，有利于提高机器人对环境的感知能力，同时在回环时，构建的语义地图，进一步帮助系统提高定位精度。

参考文献

- [1] Durrant-Whyte H, Bailey T. Simultaneous localization and mapping: part I[J]. IEEE robotics & automation magazine, 2006, 13(2): 99-110.
- [2] Bailey T, Durrant-Whyte H. Simultaneous localization and mapping (SLAM): Part II[J]. IEEE Robotics & Automation Magazine, 2006, 13(3): 108-117.
- [3] Durrant-Whyte H. Where am I? A tutorial on mobile vehicle localization[J]. Industrial Robot: An International Journal, 1994, 21(2): 11-16.
- [4] 刘浩敏, 章国锋, 鲍虎军. 基于单目视觉的同时定位与地图构建方法综述[J]. 计算机辅助设计与图形学学报, 2016, 28(06): 855-868.
- [5] Smith R, Self M, Cheeseman P. Estimating uncertain spatial relationships in Robotics[C]/Proceedings of the Second Conference on Uncertainty in Artificial Intelligence. AUAI Press, 1986: 267-288.
- [6] Cesar Cadena, Luca Carlone, Henry Carrillo. Past, Present, and Future of Simultaneous Localization And Mapping: Towards the Robust-Perception Age. 2016.
- [7] Davison A J. SLAM with a single camera[C]/Workshop on Concurrent Mapping and Localization for Autonomous Mobile Robots, in conjunction with ICRA. 2002.
- [8] Davison A J. Real-time simultaneous localization and mapping with a single camera[C]//null. IEEE, 2003: 1403.
- [9] Klein G, Murray D. Parallel tracking and mapping for small AR workspaces[C]/Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on. IEEE, 2007: 225-234.
- [10] Mur-Artal R, Montiel J M M, Tardos J D. ORB-SLAM: a versatile and accurate monocular SLAM system[J]. IEEE Transactions on Robotics, 2015, 31(5): 1147-1163.
- [11] Engel J, Schöps T, Cremers D. LSD-SLAM: Large-scale direct monocular SLAM[C]/European Conference on Computer Vision. Springer, Cham, 2014: 834-849.
- [12] Montemerlo M, Thrun S, Koller D, et al. FastSLAM: A factored solution to the simultaneous localization and mapping problem[C]//Aai/iaai. 2002: 593-598.

-
- [13] Montemerlo M, Thrun S. FastSLAM 2.0[J]. FastSLAM: A scalable method for the simultaneous localization and mapping problem in robotics, 2007: 63-90.
- [14] Mourikis A I, Roumeliotis S I. A multi-state constraint Kalman filter for vision-aided inertial navigation[C]//Robotics and automation, 2007 IEEE international conference on.IEEE, 2007: 3565-3572.
- [15] 高翔, 张涛. 视觉 SLAM 十四讲.电子工业出版社, 2017.
- [16] Christian Forster, Zichao Zhang, Michael Gassner. SVO: Semi-Direct Visual Odometry for Monocular and Multi-Camera Systems, 2016.
- [17] Tong Qin, Peiliang Li, Shaojie Shen. VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. 2017.
- [18] Stefan Leutenegger, Paul Furgale, Vincent Rabaud. Keyframe-Based Visual-Inertial SLAM Using Nonlinear Optimization. 2014.
- [19] Neira J E, Mar I, Ribeiro J, et al. Mobile Robot Localization and Map Building using Monocular Vision[J]. Springer Berlin, 1997,39(6):275-284.
- [20] Sol Joan, Vidal-Calleja T, et al. Impact of Landmark Parametrization on Monocular EKF-SLAM with Points and Lines[J]. International Journal of Computer Vision, 2012,97(3):339-368.]
- [21] Bartoli A, Sturm P. Structure-from-motion using lines: Representation, triangulation, and bundle adjustment[J]. Computer Vision & Image Understanding, 2005,100(3):416-441.
- [22] Strasdat H, Montiel J M M, Davison A J. Real-time monocular SLAM: Why filter?[J]. 2010,58(8):2657-2664.
- [23] Klein G, Murray D. Improving the Agility of Keyframe-Based SLAM: Computer Vision – ECCV 2008, European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, 2008[C].
- [24] Lu Y, Song D. Robust RGB-D Odometry Using Point and Line Features: IEEE International Conference on Computer Vision, 2015[C].
- [25] Gioi R G V, Jakubowicz J, Morel J M, et al. LSD: A Fast Line Segment Detector with a False Detection Control[J]. IEEE Transactions on Pattern Analysis & Machine

Intelligence,2010,32(4):722-732.

[26] 谢晓佳. 基于点线综合特征的双目视觉 SLAM 方法[D]. 杭州: 浙江大学, 2017.

[27] Wang Z, Liu H, Wu F. MSLD: A robust descriptor for line matching[J]. Pattern Recognition,2009,42(5):941-953.

[28] Zhang L, Koch R. An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency[J]. Journal of Visual Communication & Image Representation, 2013,24(7):794-805.

[29] Zhang G, Lee J H, Lim J, et al. Building a 3-D Line-Based Map Using Stereo SLAM. IEEE Trans. Robot. 2015, 31, 1364–1377.

[30] Kaess M, Johannsson H, Roberts R, et al. Incremental smoothing and mapping using the Bayes tree. Int. J. Robot. Res. 2012, 31, 216–235.

[31] Shen S, Michael N, Kumar V. Tightly-coupled monocular visual-inertial fusion for autonomous flight of rotorcraft MAVs. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 5303–5310.

[32] Forster, Christian, Carlone, Luca, et al. On-Manifold Preintegration Theory for Fast and Accurate Visual-Inertial Navigation[J]. 2015.

[33] Forster, Christian Carlone, Luca Dellaert. IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation. 2015.

[34] Forster, Christian, Carlone, et al. Supplementary Material to: IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation. 2015.

[35] Richard Hartley, Andrew Zisserman. multiple view geometry in computer vision. 2004.

[36] Bartoli A, Sturm P. The 3D line motion matrix and alignment of line reconstructions. Int. J. Comput. Vis. 2004, 57, 159–178.

[37] Nistér D, Naroditsky O, Bergen J. Visual odometry: Computer Vision and Pattern Recognition,2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, 2004[C].

[38] Bartoli A, Sturm P. The 3D line motion matrix and alignment of line reconstructions:

- Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, 2001[C].
- [39] Yijia He, Ji Zhao, Yue Guo. PL-VIO: Tightly-Coupled Monocular Visual-Inertial Odometry Using Point and Line Features. 2018.
- [40] Lucas B D, Kanade T. An iterative image registration technique with an application to stereo vision. In Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI), Vancouver, BC, Canada, 24–28 August 1981.
- [41] Rosten E, Porter R, Drummond T. Faster and better: A machine learning approach to corner detection. IEEE Trans. Pattern Anal. Mach. Intell. 2010, 32, 105–119.
- [42] Kaehler A, Bradski G. Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library. O'Reilly Media, Inc.: Sebastopol, CA, USA, 2016.
- [43] Agarwal S, Mierle K. Ceres Solver. Available online: <http://ceres-solver.org> (accessed on 9 April 2018).
- [44] Burri M, Nikolic J, Gohl P, et al. The EuRoC micro aerial vehicle datasets. Int. J. Robot. Res. 2016, 35, 1157–1163.
- [45] Quigley M, Conley K, Gerkey B, et al. An Open-Source Robot Operating System. ICRA Workshop on Open Source Software. ICRA: Kobe, Japan, 2009. p. 5.
- [46] Sturm J, Engelhard N, Endres F, et al. A benchmark for the evaluation of RGB-D SLAM systems. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vilamoura, Portugal, 7–12 October 2012. pp. 573–580.
- [47] Lu Y, Song D, Yi J. High level landmark-based visual navigation using unsupervised geometric constraints in local bundle adjustment. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014. pp. 1540–1545.

攻读硕士学位期间取得的研究成果

一、已发表（包括已接受待发表）的论文，以及已投稿、或已成文打算投稿、或拟成文投稿的论文情况（只填写与学位论文内容相关的部分）：

序号	作者（全体作者，按顺序排列）	题 目	发表或投稿刊物名称、级别	发表的卷期、年月、页码	相当于学位论文的哪一部分（章、节）	被索引收录情况

注：在“发表的卷期、年月、页码”栏：

- 1 如果论文已发表，请填写发表的卷期、年月、页码；
 - 2 如果论文已被接受，填写将要发表的卷期、年月；
 - 3 以上都不是，请据实填写“已投稿”，“拟投稿”。
- 不够请另加页。

二、与学位内容相关的其它成果（包括专利、著作、获奖项目等）

1、发明专利已受理

名称：基于 FPGA 的数据高速采集装置及方法

发明人：第二发明人（导师第一发明人）

专利号：202010115663.3

2、实用新型专利已受理

名称：一种双目图像和 IMU 数据高速采集装置

发明人：第二发明人（导师第一发明人）

专利号：202020205636.0

致谢

接近三年的研究生生涯就快要结束了，在论文即将完成之际，我谨向所有关心过、帮助过、支持过我的老师、同学和亲人表示由衷的感谢。

在此，我首先要感谢我的导师裴海龙教授，裴老师的项目实践应用性强，给我提供了一个可以研究本课题的机会，除此之外，裴老师在研究期间的一直都很关心我的研究进展，经常询问项目中的难点并提供解决方法，在裴老师的帮助下，我顺利地完成了本课题的研究，项目也取得了很大的进展。裴老师是一位非常敬业的老师，在实验室的研究课题上倾注了大量的精力和心血，他的学术思想、严谨踏实的科学态度、勤奋刻苦的钻研精神和朴素的工作作风都给我留下了深刻的印象，将在我未来的工作和学习中鞭策和指导着我。

感谢实验室的同届其他同学：杨鑫、李明辉、杨少基、伍期哲、蒙超恒、简昱颖。很幸运能跟你们一同学习，共同进步，在这三年里，从你们身上学到的学习思维和学习方式将使我获益终生。

感谢实验室的各位师兄：夏义道、姚土才、陈宏润、吴伟坊、谢俊文、李浩森。感谢师兄们在学习和科研上给我的耐心指导和无私帮助，感谢林裕峻、宋逸凡、黄霖杰等师弟师妹，与你们交流使我受益匪浅。

感谢实验室为我提供的优秀的平台：自主系统与网络控制教育部重点实验室（华南理工大学），广东省无人机系统工程技术研究中心（华南理工大学）。在这样优秀的平台下学习，我收获颇丰，衷心祝福实验室越办越好。

感谢 2017 级的所有同学们，正因为你们，使我的研究生生活变得丰富多彩，感谢我的室友王猛飞、李明辉和肖瑞群，感谢你们在生活上对我的关心和带给我的快乐。

感谢我的家人，在我不顺利的时候一直支持和鼓励着我，陪我度过一个个难关。

感谢各位评审专家、教授在百忙中抽出时间对本论文进行审阅并提出指正。

韩世豪

2020 年 5 月于华南理工大学

3.答辩委员会对论文的评语

（主要内容包括：1.对论文的综合评价；2.对论文主要工作和创造性成果的简要介绍；3.对作者掌握基础理论、专业知识程度、独立从事科研工作能力以及在答辩中表现的评价；4.存在的不足之处和建议；5.答辩委员会结论意见等）

该论文围绕基于点线特征的双目视觉惯性 SLAM 系统开展了深入研究，选题新颖，具有重要的理论研究意义和应用价值。

论文的主要工作如下：1. 论文主要提出了一种基于点线特征的双目视觉惯性 SLAM 系统，首先根据所需功能进行平台的选型与设计，选择合适的视觉惯性模组并完成驱动设计。2. 在基于点特征的视觉惯性 VIO 的基础上，提出了一种线特征的提取及描述方法，并设计相应的线特征观测残差加入整体的状态优化方程，实现结构化环境中点线特征的位姿估计，同时恢复了线特征的空间地图。3. 搭建了基于点线特征的双目视觉惯性 SLAM 系统进行实验，包括硬件平台搭建和软件系统，对提出的方法进行了 EuRoC 数据集及实际环境的实验。

该论文表明作者具有较强的独立从事科研工作能力。论文结构合理，撰写规范，达到硕士学位论文水平。答辩委员会一致同意该生通过论文答辩，建议授予硕士学位。

论文答辩日期：2020 年 6 月 10 日

答辩委员会委员 3 人

表决票数：同意毕业及授予学位（3）票

同意毕业，但不同意授予学位（0）票

不同意毕业（0）票

表决结果（打“√”）：同意毕业及授予学位（√）

同意毕业，但不同意授予学位（ ）

不同意毕业（ ）

答辩成员 签名	傅才颖 (主席)	张3	黄成 (因被答辩人请假)
答辩秘书 签名	李慧良		