

基于 ROS 的实时视觉/惯性/地磁等 多传感器融合定位系统研究

中图分类号： TP242.6

UDC分类号： 621.3

基于 ROS 的实时视觉/惯性/地磁等 多传感器融合定位系统研究

作 者 姓 名 _____

学 院 名 称 _____

指 导 教 师 _____

答 辩 委 员 会 主 席 _____

申 请 学 位 _____ 工程硕士

学 科 专 业 _____ 控制工程

学 位 授 予 单 位 _____

论 文 答 辩 日 期 _____

Research on real-time vision/inertia/geomagnetic multi-sensor fusion positioning system based on ROS

Candidate Name: _____

School or Department: _____

Faculty Mentor: _____

Chair, Thesis Committee: _____

Degree Applied: _____ Master of Engineering

Major: _____ Control Engineering

Degree by: _____

The Date of Defence: _____

基于ROS的实时视觉惯性地磁等多传感器融合定位系统研究

研究成果声明

本人郑重声明：所提交的学位论文是我本人在指导教师的指导下进行的研究工作获得的研究成果。尽我所知，文中除特别标注和致谢的地方外，学位论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京理工大学或其它教育机构的学位或证书所使用过的材料。与我一同工作的合作者对此研究工作所做的任何贡献均已在学位论文中作了明确的说明并表示了谢意。

特此申明。

作者签名: _____ 签字日期: _____

关于学位论文使用权的说明

本人完全了解北京理工大学有关保管、使用学位论文的规定，其中包括：**①**学校有权保管、并向有关部门送交学位论文的原件与复印件；**②**学校可以采用影印、缩印或其它复制手段复制并保存学位论文；**③**学校可允许学位论文被查阅或借阅；**④**学校可以学术交流为目的，复制赠送和交换学位论文；**⑤**学校可以公布学位论文的全部或部分内容（保密学位论文在解密后遵守此规定）。

作者签名: _____ 导师签名: _____

签字日期: _____ 签字日期: _____

摘要

视觉 SLAM 技术能够使机器人在未知的场景中，仅依靠相机进行定位和建图。然而单目视觉 SLAM 有众多的缺点，其中最显著的就是缺乏尺度信息，这使得纯粹的单目视觉 SLAM 无法在实际环境中使用。而 IMU 是一个在很多方面与相机互补的传感器，将 IMU 和单目视觉融合可构成一个具有绝对尺度的六自由度状态估计器。本文正是研究如何将视觉和 IMU 进行融合，获得高精度的状态估计。

本文首先对传感器进行数学建模与标定，使用高精度的工业级传感器搭建了一套独特的、简单的高精度硬件平台。然后针对硬件平台设计了多传感器融合算法，包括前端、初始化、后端和闭环。从前端提取图像的 FAST 特征，利用 LK 光流跟踪图像特征，预积分 IMU 数据。初始化将视觉与 IMU 对齐，得到状态估计的初始值。与其他初始化算法不同，本文研究的初始化算法能够在运动中实时初始化，并通过磁力计获得绝对的航向信息。后端采用一种基于紧耦合、非线性优化的方法，融合前端预积分的 IMU 数据和相机的特征观测数据。闭环采用词袋模型进行检测，并对检测到的闭环帧进行重定位，然后实施位姿优化，获得全局一致的定位轨迹。最后本文做了公开数据集实验和本地车载实验，并分析了每个实验的定位精度，验证了本系统硬件及软件算法的可行性和鲁棒性。

实验结果表明，本系统能够搭载在飞行器和汽车上，在室内或者室外进行高精度、鲁棒性的状态估计，从而定位自身的位置。

关键词： 视觉 SLAM；惯性测量单元；视觉-惯性里程计；非线性优化

Abstract

Visual SLAM technology enables the robot to locate and map in unknown scenes only depending on the camera. However, monocular visual SLAM has many disadvantages, the most significant of which is the lack of scale information, which makes the pure monocular visual SLAM unable to be used in the actual environment. The IMU is a sensor complementary to the camera in many aspects. The fusion of IMU and monocular vision can constitute a six-dof state estimator with absolute scale. This paper is to study how to integrate vision and IMU to obtain high-precision state estimation.

In this paper, the sensor is firstly modeled and calibrated, and a set of unique and simple high-precision hardware platform is built by using high-precision industrial-grade sensors. Then the multi-sensor fusion algorithm is designed for the hardware platform, including front end, initialization, back end and closed loop. FAST features of images were extracted from the front end, and LK optical flow was used to track image features, and pre-integration IMU data was obtained. The initialization aligns the vision with the IMU to obtain the initial value of the state estimation. Different from other initialization algorithms, the initialization algorithm studied in this paper can be initialized in real time in motion and obtain absolute yaw information through magnetometer. The back end adopts a method based on tight coupling and nonlinear optimization, and integrates the IMU data of the front end pre-integration with the feature observation data of the camera. The closed loop is detected by the word bag model, and the detected closed loop frame is repositioned, and then the pose optimization is carried out to obtain globally consistent positioning trajectory. Finally, this paper makes public data set experiment and local vehicle-mounted experiment, analyzes the positioning accuracy of each experiment, and verifies the feasibility and robustness of the system's hardware and software algorithm.

Experimental results show that the system can be mounted on aircraft and vehicles, indoor or outdoor for high-precision, robust state estimation, so as to locate its own position.

Key Words: VSLAM; IMU; VIO; Nonlinear Optimization

目 录

摘要	I
Abstract	II
第 1 章 绪论	1
1.1 研究背景及意义	1
1.2 国内外研究现状	2
1.2.1 视觉 SLAM 研究概述	2
1.2.2 视觉/惯性融合研究概述	5
1.3 本文主要工作及结构安排	7
第 2 章 硬件平台搭建	9
2.1 传感器模型与选型	9
2.1.1 相机及畸变模型	9
2.1.2 IMU 状态模型	13
2.1.3 磁力计数学模型	18
2.1.4 传感器选型	19
2.2 传感器的标定与同步	22
2.2.1 相机标定	23
2.2.2 IMU 标定	28
2.2.3 同步与联合标定	29
2.3 硬件组建与 ROS 信息采集	33
2.3.1 硬件组建	33
2.3.2 ROS 信息采集	34
2.4 本章小结	35
第 3 章 系统前端及初始化	36
3.1 视觉信息预处理	36
3.1.1 关键点和关键帧	36

3.1.2 特征跟踪	38
3.1.3 异常点剔除	40
3.2 IMU 预积分	43
3.2.1 当前帧的位置、速度和姿态	43
3.2.2 两帧之间的位置、速度和姿态增量	44
3.2.3 误差状态传播	46
3.3 系统初始化	53
3.3.1 多视图几何基础	54
3.3.2 纯视觉初始化	61
3.3.3 视觉/惯性联合初始化	61
3.4 本章小结	65
第 4 章 后端优化及闭环检测	66
4.1 后端非线性优化	66
4.1.1 状态向量和目标函数	66
4.1.2 IMU 约束	68
4.1.3 视觉约束	70
4.1.4 边缘化	71
4.2 闭环检测	75
4.2.1 词袋模型	75
4.2.2 字典构建	77
4.2.3 相似度计算	77
4.3 重定位和闭环优化	79
4.3.1 重定位	79
4.3.2 闭环优化	80
4.4 本章小结	83
第 5 章 实验	84
5.1 EuRoc 数据集实验	84
5.1.1 EuRoc 构成及硬件平台	84

5.1.2 EuRoc 实验一	85
5.1.3 EuRoc 实验二	88
5.2 本地实时实验	91
5.2.1 本地实验硬件及环境	91
5.2.2 本地实验一	92
5.2.3 本地实验二	94
5.3 本章小结	97
结论	98
参考文献	100
攻读学位期间发表论文与研究成果清单	105

第 1 章 绪论

1.1 研究背景及意义

科学技术总是能让我们生活变得更加美好，它的发展经常在我们的意料之外，特别是机器人这个行业，在过去几十年里发展迅速。银行大厅里可爱的机器人助手、现代化餐厅的送餐机器人、室内清扫机器人、儿童手中的无人机玩具等，这些移动机器人都是现代机器人技术发展的结晶。世界上很多研究机构，例如波士顿公司和 MTI 等，正在研究仿生机器人，并且取得了显著的成果。随着人工智能和 5G 技术的快速发展，机器人行业的一个分支——无人驾驶技术，正成为国际上一个重要的研究方向和热点。所有这些机器人都有一个共同点——自主性，拥有在不熟悉的环境中移动而不依赖外部信息和控制的能力。自主定位与导航技术为实现这一共性提供了基础。

SLAM 是 Simultaneous Localization and Mapping 的缩写，译为“同时定位与地图构建”^[1]。它是指携带有特定传感器的移动载体，在未知的陌生环境下，边运动边感知周围环境，并构建地图，同时估计出本体在地图中的位置^[2]。大多数机器人实现自主定位用的都是 SLAM 技术。SLAM 可以使用激光传感器，称为激光 SLAM。也可以使用视觉传感器，称为视觉 SLAM，简称 VSLAM (Visual SLAM)。早期研究者大都以激光 SLAM 研究为主，现今，随着计算机硬件和软件的极速发展，视觉 SLAM 凭借着信息量大、体积小、重量轻、适用范围广和成本低等优点受到广泛关注，逐渐成为研究者心中的热门研究领域。

然而视觉 SLAM 本身也存在着诸多问题，比如当相机运动过快时图像出现模糊或者前后两帧没有重叠区域导致特征匹配失败，无法应对特征缺失和大面积重复纹理场景，光照变化幅度大时跟踪容易失败等等。惯性测量单元 (Inertial Measurement Unit, IMU) 输出频率高，在短时间内对快速运动估计较为准确。大多数 IMU 读数漂移严重，而相机的漂移较小。相机属于外感受型传感器，当图像变化时我们无法知道是相机在运动还是周围环境发生了变化，所以纯视觉 SLAM 常常难以应对动态场景。而 IMU 属于内感受型器，能够感受到自身的 6-DoF (六自由度) 运动信息。因此相机和 IMU 被认为是一对具有互补性的传感器。在大多数情况下，最好的状态估计就是同时有效利用内感受型和外感受型传感器的观测值^[3]。因此视觉/惯性融合成为目前学术界的研究热点。

1.2 国内外研究现状

SLAM 技术已经有 30 多年的历史了，伴随着计算机硬件的巨大改进，研究热点从激光转移到视觉。有部分研究者提出了划时代的理论算法，为 SLAM 的发展奠定了理论基础。还有部分研究者实际研究并编程实现了 SLAM 的各种算法的完整框架，甚至开源了他们的源代码，为 SLAM 的发展做出了无私的贡献。正是因为有很多这样孜孜不倦、无私奉献的学者，SLAM 技术才能日渐成熟，走向大众。

下面，本文将首先对视觉 SLAM 的发展历程和有影响力的研究成果经行概述，然后对视觉/惯性融合的发展现状经行详细阐述。

1.2.1 视觉 SLAM 研究概述

SLAM 最早叫做概率 SLAM (probabilistic SLAM)。早期大多以激光传感器为主，SLAM 中存在一个基本问题——全局一致性。Cheesman^[4] 和 Durrant-Whyte^[5] 为了描述路标与路标之间的相互关系而建立了一个统计学框架，该框架主要用来解决几何上的不确定性。他们证明了估计的路标位置之间具有强关联性，并且这种关联性会随着观测的时间增加而增加。

Ayache 和 Faugeras 最先从事视觉 SLAM 的研究。Smith 等人提出移动机器人在陌生环境中获取的路标时，因为这些路标具有相同的估计误差，所以路标的估计值也是相关的^[6]。这样就加大了难度。因为状态向量中将包含机器人的位姿和每一个路标点的观测值，而且每观测一次都要实时更新，这导致状态向量的维数非常大，进而带来巨大的计算量。SLAM 中的地图有一个收敛的性质，因为 Smith 等人将其忽略，并假设地图的估计误差是发散的，因此学者们只能视路标点之间没有相关性或者相关性很小。这是因为这个原因，当时的 SLAM 大都是将定位和建图问题分开处理的。

1995 年文章 International Symposium on Robotics 提出了收敛性和同时定位建图的概念^[7]，此时研究者们才将定位与地图构建问题结合在一起，突破了新的理论。之后 SLAM 的研究集中在计算速度的提升以及多源信息融合上。在 ISRR99 上，Thrun 首次提出将基于卡尔曼滤波 (Kalman filter) 的 SLAM 与概率定位建图相结合^[8]。

进入二十一世纪后，因为计算机硬件性能的提升，研究热潮逐渐转向以相机为传感器的视觉 SLAM，并出现了很多可以实现的理论框架，甚至有一些理论框架已经落地市场化。

A. J. Davison 可以说是视觉 SLAM 领域中的领头人，他在单目视觉 SLAM 中做了奠基性的工作^{[2][9]}，极大的推动了视觉 SLAM 的发展。他在 2007 年的一篇文章中提出了 MonoSLAM^[2]，该系统有一个重大突破——可以实时运行。MonoSLAM 仅仅依靠单目相机，检测每一帧图像中的 Shi-Tomasi 角点^[10]，该角点是很稀疏的角点，然后在投影椭圆中实施特征匹配。经典的扩展卡尔曼滤波器（EKF）被用于 SLAM 后端，路标点的位置被建模为高斯分布，位姿及其关联的路标点构成状态向量，并实时更新其均值和协方差。

Klein 等人在 2007 年提出并开源了 PTAM（Parallel Tracking and Mapping）^[11]，并在两年后将 PTAM 算法移植到智能手机上^[12]。PTAM 首次提出并编程实现了前端特征跟踪和后端建图的并行化，将前端和后端分开来并行计算，这是一个值得称赞的突破。时至今日，这种思想仍在继续使用，效果极好。此外，PTAM 还创新地使用了关键帧机制，这一重大创新使得处理器不必处理每一帧图像，而是处理几个关键的图像，大大减少了计算量。而且，对比 MonoSLAM，PTAM 在后端不使用卡尔曼滤波器。取而代之的是非线性优化，这在历史上是第一次的。这一开创性突破直接导致了在 PTAM 之后，研究者开始以非线性优化为主来设计后端算法。并采取关键帧机制和多线程并行化思想。PTAM 对视觉 SLAM 研究提出了很多创新的算法和重大革新，对视觉 SLAM 的发展做出了巨大的贡献。

Raul Mur-Artal 等人在 2015 年提出了 ORB-SLAM 系统^[13]。如果说 MonoSLAM 和 PTAM 是在时间上距离我们较为久远的方案，那么 ORB-SLAM 则是距离我们很近的全新的方案。ORB-SLAM 是自 2015 年以来所有纯视觉 SLAM 算法中最为优秀的算法之一。ORB-SLAM 提出了自己独特的创新和改进，抛弃了传统的 SIFT（Scale Invariant Feature transform）^[14] 和 SURF（Speeded Up Robust Features）^[15] 特征，采用自己独特的 ORB 特征进行特征提取。相比于 SIFT 和 SURF，ORB 特征计算快，可以在处理器上实现实时计算。与 Harris 角点相比，ORB 特征点比 Harris 角点多了一个尺度不变特性以及描述子，可以快速的进行特征匹配、回环检测或者重定位。因此，ORB 特征是一种折衷了计算速度和计算精度的优秀特征方案。在回环检测中 ORB-SLAM 使用了 ORB 字典，大大减少了系统的累计误差，同时在特征丢失之后也能快速找回，大大提高了整个系统的鲁棒性。最后，该系统创新地使用了三线程并行方案。使得前端、后端和回环各自有条不紊的工作。ORB-SLAM 是主流特征点法的巅峰之作，极大的推动了视觉 SLAM 技术的发展。

MonoSLAM、PTAM 和 ORB-SLAM 的运行效果如图1.1所示。

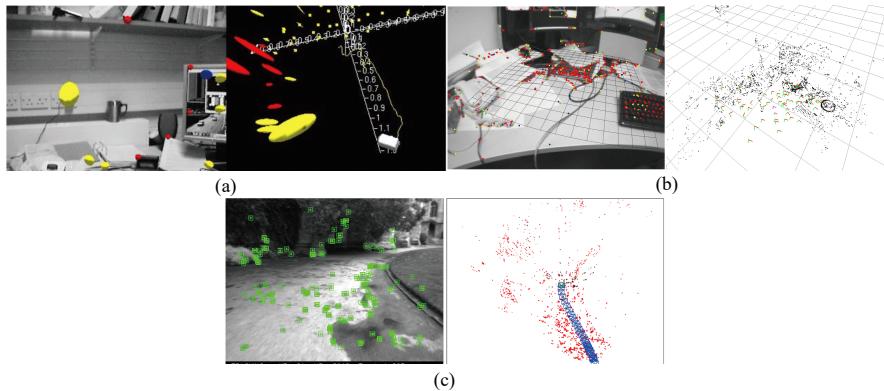


图 1.1 (a) MonoSLAM; (b) PTAM; (c) ORB-SLAM.

以上所介绍的视觉 SLAM 系统都是基于特征点的方案，称作特征点法或者间接法。特征点是人为从图像中提取和设计的特殊的角点，一般包括关键点和描述子。在视觉 SLAM 的前端跟踪中用来代替整张图片的纹理，在后端中将其视为路标点进行迭代优化。特征点的存在使得相机在运动的过程中，SLAM 系统前端跟踪也能取得良好的效果。万事有利必有弊，虽然特征点法应用广泛，鲁棒性高，但是特征点法仍然有缺点：提取关键点需要花费很长时间；提取特征点意味着主动丢掉了图像中的其他信息；在一些单色无纹理区域，往往很难提取特征点，在重复纹理区域，特征点普遍跟踪混乱，无法使用。

除了特征点法，近几年来研究者们也在研究一种基于像素的直接法。直接法与光流法非常相似，其灰度不变性的假设是应用的前提。该方法不需要提取图像特征点，而是通过最小化亮度误差直接估计像素深度，从而计算像素 3D 空间位置和相机的运动。因此，直接法相比于特征点法在计算速度上有很大提升，并且能适应特征缺失和图像模糊的场景。同时，直接法仅用 CPU 就能构建稀疏或者半稠密地图，这是特征法做不到的。

2014 年，TUM 计算机视觉组的 J. Engel 等人提出 LSD-SLAM^{[16][17]}，LSD-SLAM 采用直接法，可以构建半密集三维地图。这是单目直接法在视觉 SLAM 中的典型应用。与以往的方案不同，LSD-SLAM 仅使用 CPU 就可以完成半密集地图的三维重建。LSD-SLAM 有很多自己的创新之处，例如，LSD-SLAM 既不是利用像素，也不是利用块匹配去跟踪图相机，而是在极线上等间距的取五个点，计算其 SSD (Sum of Squared Distance)；LSD-SLAM 没有复杂的初始化过程，而是对深度随机赋值，然后在将计算出来的深度归一化，得到全局一致的尺度；使用光度不确定性项来计算深度协方差；使用李群和李代数来对关键帧进行约束，在后端优化中考虑不同的尺度，进而减小大

范围场景中的尺度漂移现象。

Forster 等人在 2014 年提出 SVO (Semi-direct Visual Odometry)^[18]。SVO 采用稀疏直接法，也称作“半直接法”。之所以成为“半直接法”，是因为 SVO 同时使用了特征点法和直接法，在前端提取角点，然后利用角点周围的信息通过直接法估计出相机的运动以及自身的位置。该方案对比于其他方案，显著的优势是速度极快，它甚至可以在低端处理器上实现实时计算，这使得 SVO 可以应用到无人机、智能手机、VR 等设备上。另外，SVO 在理论上也有一个重大的创新，提出了深度滤波器的概念。在估计角点的位置时，逆深度被用来设计参数，使之能够更好的计算特征点位置。

LSD-SLAM 和 SVO 的运行效果如图1.2所示。

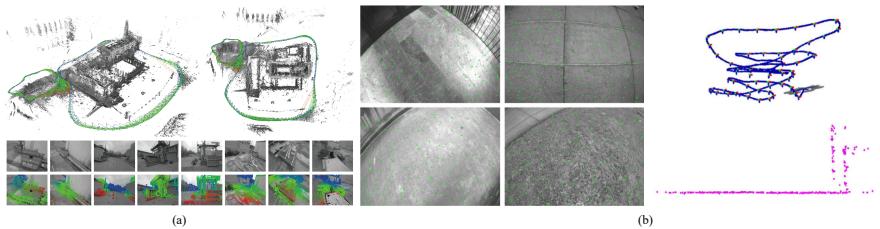


图 1.2 (a) LSD-SLAM; (b) SVO.

国内 SLAM 的研究因为起步较晚，所以一直落后于国外，但是，近几年来随着自动驾驶行业的发展，SLAM 研究越来越受到重视。很多高校都在开展 SLAM 相关的课题研究，也出现了一些详细解读 SLAM 技术的文章^{[19][20][21]}。上海交通大学的邹丹平等提出了基于线特征的 StructSLAM^[22] 以及基于多目相机的 Co-SLAM^[23]。浙江大学计算机学院的章国锋等人于 2013 年提出了 RDSLAM^[24]，RDSLAM 的创新之处是将传统的随机采样一致性（RANSAC）^[25] 算法改为一种基于时间序列和先验信息的自适应 RANSAC 算法，该算法弥补了之前算法在动态场景下表现欠佳的缺点。2016 年，他们又提出了改进版的基于移动设备的 RKSLAM^[26]，可以实时运行。2017 年，他们提出基于 RGB-D 相机的 RKD-SLAM 系统^[27]，RKD-SLAM 能够构建稠密地图。

1.2.2 视觉/惯性融合研究概述

尽管仅用相机就能完成定位和建图任务的视觉 SLAM。然而，纯视觉 SLAM 往往因为各种各样的问题导致其应用场景有限，鲁棒性低。广大学者和工程师往往希望算法能够得到实际的应用，而不是局限于在个人 PC 上跑 demo。在这种应用需求的背景下，视觉与惯性融合的定位方案逐渐成为学术界的研究热点。

目前的 VIO 算法可以分为松耦合和紧耦合^[28]。松耦合将相机和 IMU 分开处理，

各自单独估计，然后将二者估计出的结果进行融合。紧耦合是指将 IMU 和相机的状态量融合在一起，组成一个整体的状态方程。然后对状态方程进行求解。类比于纯视觉 SLAM，VIO 也可以分为基于滤波和基于优化两大类。下面将基于这几种分类对 VIO 的研究现状进行分析。

宾夕法尼亚大学 Kumar 实验室的 Mourikis 等人于 2007 年提出基于滤波的紧耦合 VIO 方案——MSCKF（Multi-State Constraint KF）^[29]，并于 2015 年提出了改进的 MSCKF^[30]。传统的 EKF-SLAM（Extended kalman filtering SLAM）^[31] 系统会将图像的特征信息加入到状态向量和协方差矩阵里，如果在初始化特征点深度和协方差矩阵的时候有误差，那么很容易造成全局轨迹不一致的情况。而 MSCKF 始终保持一个先进先出的滑动窗口序列，在滑窗内建立 EKF 的预测和更新。

苏黎世联邦理工学院自动控制系统实验室的 Stephen Weiss 等人于 2013 年提出基于滤波的松耦合 VIO 方案——SSF 和 MSF^[32]。松耦合系统没有紧耦合系统那么繁杂，而且 MSF 可以用来融合除相机和 IMU 之外的其他多种传感器。松耦合还有另一个优点：当一个传感器不能工作时，系统任然能够依靠其他传感器正常运行。紧耦合系统则是所有传感器相互依赖，缺一不可。

2014 年，苏黎世联邦理工学院的 Stefan Leutenegger 等人提出基于优化的紧耦合 VIO 方案——OKVIS^[33]。OKVIS 在一个滑窗内优化关键帧，带权重的视觉重投影误差和带权重的 IMU 误差组成残差项。前端提取尺度不变的 Harris 特征点，并计算 BRISK 描述子^[34]，后端使用非线性优化的方式进行状态估计。另外 OKVIS 同时支持单目和双目两种类型的相机，Leutenegger 的实验表明在双目情况下有更好的效果。

2015 年，苏黎世联邦理工学院的 Michael Bloesch 等人提出了基于滤波的紧耦合 VIO 方案——ROVIO^[31]。ROVIO 利用 EKF 进行状态估计，而且有许多创新的地方。首先，ROVIO 在提取角点时采用的是 FAST 角点^[35]，使用矢量坐标点方向，用标量表示坐标点大小；其次，图像块用于描述所有的角点，视频流用于多级表达；最后使用惯导计算出的位姿估计视觉重投影的光度误差，并将其用于后端优化。ROVIO 仅仅支持单目相机。

2017 年，香港科技大学的沈绍杰等人提出基于优化的紧耦合 VIO 方案——VINS-Mono^[36]。VINS-Mono 是一个基于滑动窗口的紧耦合非线性优化系统。在初始化阶段，通过松耦合的方式融合相机和 IMU，从而获得位置、姿态和尺度的初始值。在后端通过 IMU 残差、视觉残差和边缘化残差构成目标函数，用非线性优化进行计算状态量。

值得注意的是，VINS-Mono 还具有回环检测和重定位的功能，并在检测到回环时进行 4-DoF 全局位姿图优化。VINS-Mono 在精度和鲁棒性上都表现优异，是一个极具实用价值的 VIO 开源方案。

基于优化的松耦合方案不多，因为学术界普遍认为其效果没有紧耦合好。2016 年，Gabe Sibley 等人在其文章中提到了基于优化的松耦合方法^[37]。简而言之，就是将前端计算出的位姿进行坐标变换，然后放到 IMU 的优化框架中。

目前，VIO 算法已经应用在包括自动驾驶、飞行机器人和 VR/AR 等领域。视觉/惯性融合方案总结如图 1.3。本文将对基于优化的紧耦合 VIO 方案进行细致的研究，并基于单目相机、IMU 和磁力计实际搭建一套高精度、简单易用的视觉/惯性/地磁等多传感器融合定位系统。

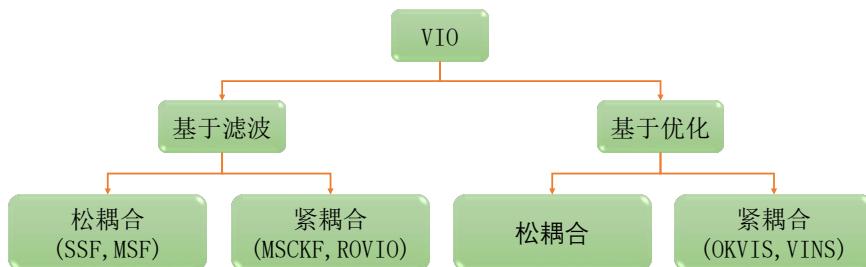


图 1.3 (a) LSD-SLAM; (b) SVO.

1.3 本文主要工作及结构安排

本文以视觉/惯性融合为主，磁力计为辅，研究多传感器融合定位算法，并实际搭建一套软硬件结合的多传感器融合定位系统。

本文主要工作如下：

- (1) 搜集并研读国内外视觉 SLAM 和 VIO 相关的论文，详细阐述了视觉 SLAM 和 VIO 的国内外发展现状；
- (2) 与市场上普通的消费级视惯套件不同，本文对高精度的工业级传感器进行硬件同步，搭建了一套高精度的多传感器融合定位系统硬件平台；
- (3) 研究了视觉 SLAM 算法、多传感器融合算法，针对自己的硬件系统设计前端、后端及闭环算法，并创新的实现了运动中在线实时初始化和绝对航向初始化；
- (4) 在 ROS 下编程实现所有算法，整个系统拥有良好的可视化功能，可扩展性强；

(5) 进行公开数据集实验和本地实验，分析了系统在不同环境下的定位精度，验证了系统的可行性。

本文各个章节安排如下：

第二章主要研究了系统的硬件平台。首先研究了传感器（相机和 IMU）的数学模型及选型；然后研究传感器的标定以及硬件同步原理；最后研究了基于 ROS 的传感器信息采集方法。

第三章主要研究系统的前端和初始化算法。首先研究视觉信息预处理算法；然后研究 IMU 预积分原理；最后研究了系统初始化流程，包括纯视觉初始化和视觉/惯性联合初始化。

第四章主要研究系统后端和闭环算法。首先研究了后端非线性优化的方法；然后研究了闭环检测算法；最后研究了重定位和闭环优化算法。

第五章主要是实验。验证算法有效性，并分析定位误差。包括公共数据集实验和本地校园环境实验。

第 2 章 硬件平台搭建

本章主要研究硬件平台的构成及各个传感器数学模型，为后续章节研究 VIO 算法提供硬件支撑。首先，研究传感器的数学模型以及选型；其次，研究传感器的标定原理及方法，传感器之间的硬件同步方案；最后，研究传感器信息采集方案。

2.1 传感器模型与选型

2.1.1 相机及畸变模型

(1) 针孔相机模型

众所周知，小孔成像能够将 3D 物体投影成 2D 图像。相机成像原理类似，称为针孔相机模型。如图 2.1 所示。其中 $O - x - y - z$ 是相机的坐标系， z 轴正方向是相

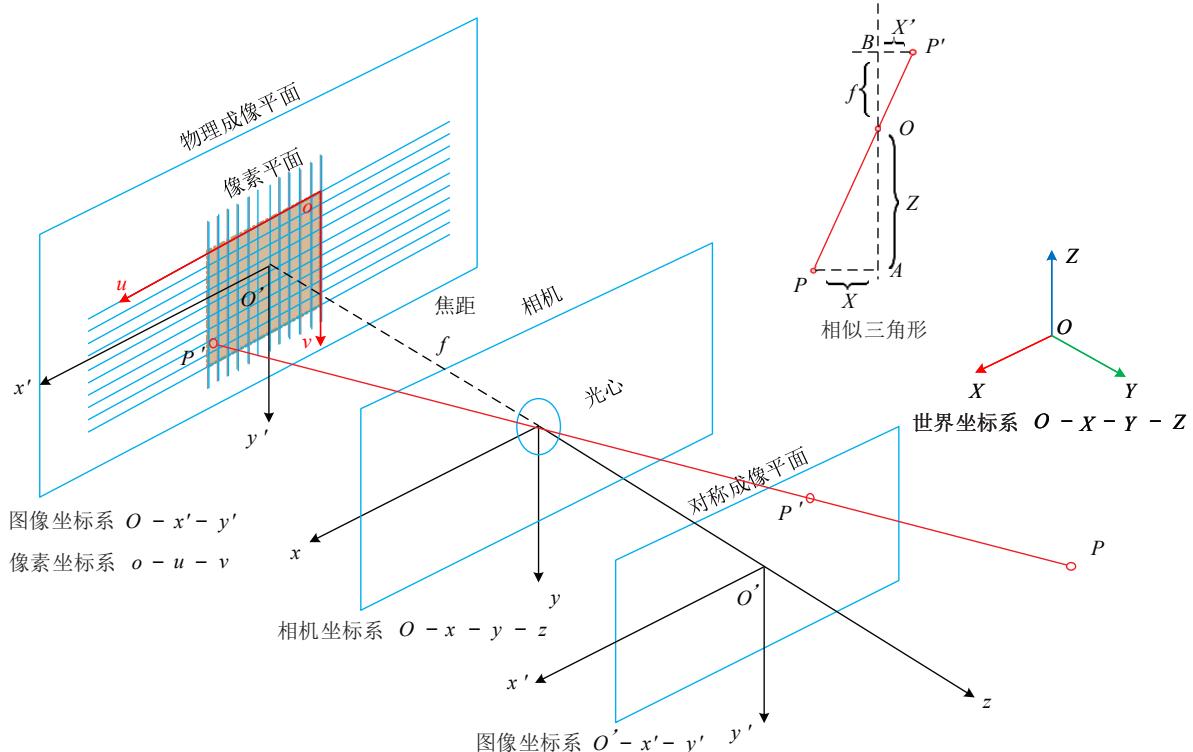


图 2.1 针孔相机模型

机前方，向右为 x 轴正方向，向下为 y 轴正方向。 O 为相机的光心，即针孔模型的针孔。真实世界中三维坐标点 P ，经过光心 O 投影到物理成像平面 $O' - x' - y'$ 上，即图像坐标系，投影坐标点是 P' 。设 P 在相机坐标系下的坐标为 $[X, Y, Z]^T$ ， P' 在图像

坐标系下的坐标为 $[X', Y']^T$ ，光心到物理成像平面的距离为焦距 f 。那么，根据相似三角形可得：

$$\frac{Z}{f} = -\frac{X}{X'} = -\frac{Y}{Y'} \quad (2.1)$$

其中的负号是因为这个成像看起来是上下颠倒的。为了简化模型，将物理成像平面对称平移到相机的前方，即与三维空间点 P 在同一侧。

这个时候就可以将式 (2.1) 中的负号去掉，得到更加简洁的式子：

$$\frac{Z}{f} = \frac{X}{X'} = \frac{Y}{Y'} \quad (2.2)$$

整理得到：

$$\begin{cases} X' = f \frac{X}{Z} \\ Y' = f \frac{Y}{Z} \end{cases} \quad (2.3)$$

将成像平面平移到相机前方只是为了简化成像模型，是一种数学手段，不会带来负面影响。所以，在不引起异议的情况下，将这种简化的成像模型定义为针孔模型。

式 (2.3) 建立了三维空间点和它的二维成像点之间的映射关系。然而，相机采集的图像是以像素为单位的，所以，为了描述光线到像素之间的映射关系，假设在物理成像平面上固连了一个像素平面，如图2.1所示。

图像的左上角 o 是像素坐标系的原点，向右为 u 轴正方向，向下为 v 轴正方向。设 P' 在像素坐标系下的坐标为 $[u, v]^T$ ，像素坐标在 u 轴上缩放了 α 倍，在 v 轴上缩放了 β 倍，原点平移了 $[c_x, c_y]^T$ 。那么， P' 的图像坐标与像素坐标之间的映射关系为：

$$\begin{cases} u = \alpha X' + c_x \\ v = \beta Y' + c_y \end{cases} \quad (2.4)$$

代入式 (2.3)，得：

$$\begin{cases} u = f_x \frac{X}{Z} + c_x \\ v = f_y \frac{Y}{Z} + c_y \end{cases} \quad (2.5)$$

其中， $f_x = \alpha f$ ， $f_y = \beta f$ 。 f 的单位为米， α, β 得单位为像素每米， f_x 和 f_y 的单位

为像素。通过齐次坐标，将式写成矩阵形式，得：

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \frac{1}{Z} \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ z \end{pmatrix} \triangleq \frac{1}{Z} \mathbf{K} \mathbf{P} \quad (2.6)$$

等式两边同时乘以 Z 得：

$$Z \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \triangleq \mathbf{K} \mathbf{P} \quad (2.7)$$

其中， \mathbf{K} 为相机的内参矩阵，下文中的相机标定需要标定此参数。有时候，为了不失一般性，可以在内参矩阵中添加一个扭曲参数 γ ，该参数用来表示像素坐标系的两个坐标轴之间的扭曲程度。此时，内参矩阵 \mathbf{K} 有 5 个参数：

$$\mathbf{K} = \begin{bmatrix} f_x & \gamma & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.8)$$

值得注意的是，式 (2.6) 中 \mathbf{P} 是点 P 在相机坐标系下的坐标。记点 P 在世界坐标系下的坐标为 \mathbf{P}_w ，则 \mathbf{P} 和 \mathbf{P}_w 之间存在一个坐标变换，这个变换矩阵由相机当前的位姿决定。记点 P 在像素坐标系下的坐标为 \mathbf{P}_{uv} ，设相机的旋转和平移为 \mathbf{R} ， \mathbf{t} ，那么有：

$$Z \mathbf{P}_{uv} = Z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} (\mathbf{R} \mathbf{P}_w + \mathbf{t}) = \mathbf{K} \mathbf{T} \mathbf{P}_w \quad (2.9)$$

其中， \mathbf{T} 为变换矩阵，又叫做特殊欧式群 (Special Euclidean Group)：

$$SE(3) = \left\{ \mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \in \mathbb{R}^{4 \times 4} \mid \mathbf{R} \in SO(3), \mathbf{t} \in \mathbb{R}^3 \right\} \quad (2.10)$$

式 (2.9) 两侧使用的都是齐次坐标，也就是对相机投影平面进行了归一化，所以

式(2.9)可以简化为:

$$\mathbf{P}_{uv} = \mathbf{K} \mathbf{T} \mathbf{P}_w \quad (2.11)$$

式(2.11)表达了世界坐标系下的坐标与像素坐标系下的坐标之间的映射关系,也就是所谓的针孔相机模型。

(2) 相机畸变模型

上一小节研究的针孔相机模型是理想情况下的模型,实际上,相机的镜头影响光线的传播方向,进而影响成像。影响分为两个方面:一是镜头的形状会对光线产生折射,二是镜头和相机在机械组装时镜头和感光平面不完全平行导致成像的位置发生变化。这两种影响都会使图形产生畸变,前者引起的畸变称为径向畸变,后者引起的畸变称为切向畸变。

图2.2展示了两种不同类型的径向畸变。

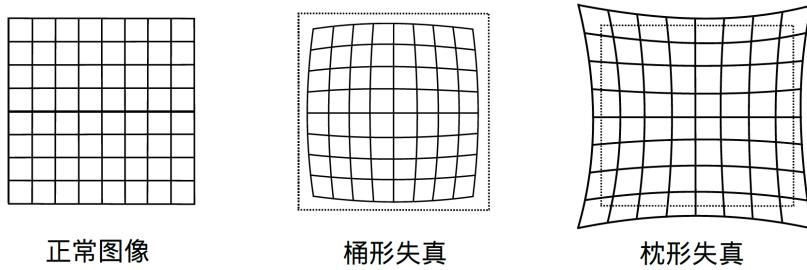


图 2.2 两种类型的径向畸变

切向畸变示意图如图2.3。设归一化成像平面上的点 p 的笛卡尔坐标为 $[x, y]^T$, 极坐标为 $[r, \theta]^T$ 。径向畸变使坐标点的长度 r 变化了 δr , 切向畸变使坐标点的水平夹角 θ 变化了 $\delta\theta$ 。

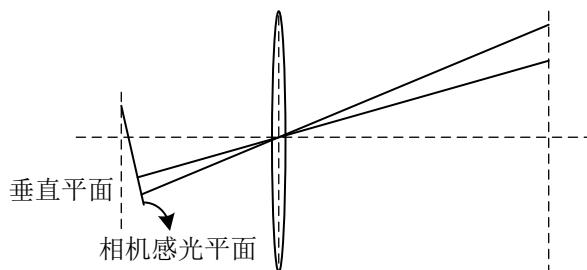


图 2.3 切向畸变示意图

使用一个函数来表达径向畸变前后坐标的变化:

$$\begin{cases} x_{\text{corrected}} = x (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \\ y_{\text{corrected}} = y (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \end{cases} \quad (2.12)$$

其中, $[x_{\text{corrected}}, y_{\text{corrected}}]^T$ 表示去畸变后的坐标。公式 (2.12) 中, 越靠近图像中心, 畸变越小, 畸变系数 k_1 起主导作用。越远离图像中心, 畸变越大, 畸变系数 k_2 起主导作用。一般的针孔相机用这两个系数就能去除径向畸变。对于像鱼眼镜头等畸变较大的相机, 可以加入 k_3 来去畸变。

对于切向畸变, 可以使用参数 p_1, p_2 来进行去畸变:

$$\begin{cases} x_{\text{corrected}} = x + 2p_1xy + p_2(r^2 + 2x^2) \\ y_{\text{corrected}} = y + p_1(r^2 + 2y^2) + 2p_2xy \end{cases} \quad (2.13)$$

由式 (2.12) 和式 (2.13) 可以得到坐标点在归一化成像平面上去除径向畸变和切向畸变后的正确坐标点:

$$\begin{cases} x_{\text{corrected}} = x(1 + k_1r^2 + k_2r^4 + k_3r^6) + 2p_1xy + p_2(r^2 + 2x^2) \\ y_{\text{corrected}} = y(1 + k_1r^2 + k_2r^4 + k_3r^6) + p_1(r^2 + 2y^2) + 2p_2xy \end{cases} \quad (2.14)$$

进一步, 通过内参矩阵可以得到去畸变后的像素坐标:

$$\begin{cases} u_{\text{corrected}} = f_x x_{\text{corrected}} + c_x \\ v_{\text{corrected}} = f_y y_{\text{corrected}} + c_y \end{cases} \quad (2.15)$$

2.1.2 IMU 状态模型

四元数可表示为:

$$\mathbf{q} = q_w + q_x i + q_y j + q_z k \Leftrightarrow \mathbf{q} = q_w + \mathbf{q}_v \quad (2.16)$$

其中, i, j, k 是四元数的虚部, 满足:

$$\begin{cases} i^2 = j^2 = k^2 = -1 \\ ij = k, ji = -k \\ jk = i, kj = -i \\ ki = j, ik = -j \end{cases} \quad (2.17)$$

也可以用一个四维向量来表示：

$$\mathbf{q} \triangleq \begin{bmatrix} q_w & \mathbf{q}_v \end{bmatrix} = \begin{bmatrix} q_w & q_x & q_y & q_z \end{bmatrix}^T \quad (2.18)$$

设有两个四元数 \mathbf{p}, \mathbf{q} ，那么四元数的运算可表示如下：

(1) 加法和减法：

$$\mathbf{p} \pm \mathbf{q} = \begin{bmatrix} p_w \\ \mathbf{p}_v \end{bmatrix} \pm \begin{bmatrix} q_w \\ \mathbf{q}_v \end{bmatrix} = \begin{bmatrix} p_w \pm q_w \\ \mathbf{p}_v \pm \mathbf{q}_v \end{bmatrix} \quad (2.19)$$

满足交换律和结合律，

$$\mathbf{p} + \mathbf{q} = \mathbf{q} + \mathbf{p} \quad (2.20)$$

$$\mathbf{p} + (\mathbf{q} + \mathbf{r}) = (\mathbf{p} + \mathbf{q}) + \mathbf{r} \quad (2.21)$$

(2) 乘法：

$$\mathbf{p} \otimes \mathbf{q} = \begin{bmatrix} p_w q_w - p_x q_x - p_y q_y - p_z q_z \\ p_w q_x + p_x q_w + p_y q_z - p_z q_y \\ p_w q_y - p_x q_z + p_y q_w + p_z q_x \\ p_w q_z + p_x q_y - p_y q_x + p_z q_w \end{bmatrix} \quad (2.22)$$

其中， \otimes 表示四元数相乘。写成标量和矢量部分，

$$\mathbf{p} \otimes \mathbf{q} = \begin{bmatrix} p_w q_w - \mathbf{p}_v^\top \mathbf{q}_v \\ p_w \mathbf{q}_v + q_w \mathbf{p}_v + \mathbf{p}_v \times \mathbf{q}_v \end{bmatrix} \quad (2.23)$$

不满足交换律，

$$\mathbf{p} \otimes \mathbf{q} \neq \mathbf{q} \otimes \mathbf{p} \quad (2.24)$$

满足结合律和分配律，

$$(\mathbf{p} \otimes \mathbf{q}) \otimes \mathbf{r} = \mathbf{p} \otimes (\mathbf{q} \otimes \mathbf{r}) \quad (2.25)$$

$$\mathbf{p} \otimes (\mathbf{q} + \mathbf{r}) = \mathbf{p} \otimes \mathbf{q} + \mathbf{p} \otimes \mathbf{r} \quad \text{and} \quad (\mathbf{p} + \mathbf{q}) \otimes \mathbf{r} = \mathbf{p} \otimes \mathbf{r} + \mathbf{q} \otimes \mathbf{r} \quad (2.26)$$

两个四元数的乘积可以等效的表示为两个矩阵的乘积，

$$\mathbf{q}_1 \otimes \mathbf{q}_2 = [\mathbf{q}_1]_L \mathbf{q}_2 \quad \text{and} \quad \mathbf{q}_1 \otimes \mathbf{q}_2 = [\mathbf{q}_2]_R \mathbf{q}_1 \quad (2.27)$$

其中 $[\mathbf{q}]_L$ 和 $[\mathbf{q}]_R$ 分别为,

$$[\mathbf{q}]_L = \begin{bmatrix} q_w & -q_x & -q_y & -q_z \\ q_x & q_w & -q_z & q_y \\ q_y & q_z & q_w & -q_x \\ q_z & -q_y & q_x & q_w \end{bmatrix}, \quad [\mathbf{q}]_R = \begin{bmatrix} q_w & -q_x & -q_y & -q_z \\ q_x & q_w & q_z & -q_y \\ q_y & -q_z & q_w & q_x \\ q_z & q_y & -q_x & q_w \end{bmatrix} \quad (2.28)$$

或者,

$$[\mathbf{q}]_L = q_w \mathbf{I} + \begin{bmatrix} 0 & -\mathbf{q}_v^\top \\ \mathbf{q}_v & [\mathbf{q}_v]_\times \end{bmatrix}, \quad [\mathbf{q}]_R = q_w \mathbf{I} + \begin{bmatrix} 0 & -\mathbf{q}_v^\top \\ \mathbf{q}_v & -[\mathbf{q}_v]_\times \end{bmatrix} \quad (2.29)$$

这里, 用运算符 $[\bullet]_\times$ 产生反对称矩阵,

$$[\mathbf{a}]_\times \triangleq \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \quad (2.30)$$

因为,

$$(\mathbf{q} \otimes \mathbf{x}) \otimes \mathbf{p} = [\mathbf{p}]_R [\mathbf{q}]_L \mathbf{x} \quad \text{and} \quad \mathbf{q} \otimes (\mathbf{x} \otimes \mathbf{p}) = [\mathbf{q}]_L [\mathbf{p}]_R \mathbf{x} \quad (2.31)$$

所以,

$$[\mathbf{p}]_R [\mathbf{q}]_L = [\mathbf{q}]_L [\mathbf{p}]_R \quad (2.32)$$

(3) 共轭:

$$\mathbf{q}^* \triangleq q_w - \mathbf{q}_v = \begin{bmatrix} q_w & -\mathbf{q}_v \end{bmatrix}^T \quad (2.33)$$

有以下特性,

$$\mathbf{q} \otimes \mathbf{q}^* = \mathbf{q}^* \otimes \mathbf{q} = q_w^2 + q_x^2 + q_y^2 + q_z^2 = \begin{bmatrix} q_w^2 + q_x^2 + q_y^2 + q_z^2 \\ \mathbf{0}_v \end{bmatrix} \quad (2.34)$$

$$(\mathbf{p} \otimes \mathbf{q})^* = \mathbf{q}^* \otimes \mathbf{p}^* \quad (2.35)$$

(4) 范数:

$$\|\mathbf{q}\| \triangleq \sqrt{\mathbf{q} \otimes \mathbf{q}^*} = \sqrt{\mathbf{q}^* \otimes \mathbf{q}} = \sqrt{q_w^2 + q_x^2 + q_y^2 + q_z^2} \in \mathbb{R} \quad (2.36)$$

(5) 逆:

$$\mathbf{q}^{-1} = \mathbf{q}^*/\|\mathbf{q}\|^2 \quad (2.37)$$

$$\mathbf{q} \otimes \mathbf{q}^{-1} = \mathbf{q}^{-1} \otimes \mathbf{q} = \mathbf{q}_1 \quad (2.38)$$

(6) 单位四元数:

对于单位四元数, 因为 $\|\mathbf{q}\| = 1$, 所以,

$$\mathbf{q}^{-1} = \mathbf{q}^* \quad (2.39)$$

当把单位四元数解释为方向变换或旋转算子时, 这个性质意味着可以用共轭四元数实现相反的旋转。单位四元数总是可以写成这种形式,

$$\mathbf{q} = \begin{bmatrix} \cos \theta & \mathbf{u} \sin \theta \end{bmatrix}^T \quad (2.40)$$

其中, $\mathbf{u} = u_x i + u_y j + u_z k$ 为单位向量, θ 为标量。IMU 状态模型主要由运动模型以及观测和噪声模型构成^[38], 状态量包括: $[\mathbf{p}_t, \mathbf{v}_t, \mathbf{q}_t, \mathbf{b}_{at}, \mathbf{b}_{\omega_t}, \mathbf{g}^w]$, 分别表示 t 时刻 IMU 的位置, 速度, 姿态(旋转), 加速度 bias, 角速度 bias 以及世界坐标系下的重力向量。根据式 (2.22)(2.27)(2.28)(2.29)(2.40) 可得,

$$\begin{aligned} \dot{\mathbf{q}}_t &= \lim_{\delta t \rightarrow 0} \frac{1}{\delta t} (\mathbf{q}_{t+\delta t} - \mathbf{q}_t) \\ &= \lim_{\delta t \rightarrow 0} \frac{1}{\delta t} \left(\mathbf{q}_t \otimes \delta \mathbf{q}_t - \mathbf{q}_t \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right) \\ &= \lim_{\delta t \rightarrow 0} \frac{1}{\delta t} \left(\mathbf{q}_t \otimes \begin{bmatrix} \mathbf{u} \sin \frac{\theta}{2} \\ \cos \frac{\theta}{2} \end{bmatrix} - \mathbf{q}_t \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right) \\ &\approx \lim_{\delta t \rightarrow 0} \frac{1}{\delta t} \left(\mathbf{q}_t \otimes \begin{bmatrix} \mathbf{u} \frac{\theta}{2} \\ 1 \end{bmatrix} - \mathbf{q}_t \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right) \quad (2.41) \\ &= \lim_{\delta t \rightarrow 0} \frac{1}{\delta t} \left(\begin{bmatrix} \mathbf{u} \frac{\theta}{2} \\ 1 \end{bmatrix}_R - \begin{bmatrix} 0 \\ 1 \end{bmatrix}_R \right) \mathbf{q}_t \\ &= \lim_{\delta t \rightarrow 0} \frac{1}{\delta t} \begin{bmatrix} -\frac{[\theta]_x}{2} & \frac{\theta}{2} \\ -\frac{\theta^T}{2} & 0 \end{bmatrix} \mathbf{q}_t \end{aligned}$$

其中 $\delta \mathbf{q}_t$ 表示在 t 时刻加的三维小扰动。又因为角速度为:

$$\omega = \lim_{\delta t \rightarrow 0} \frac{\theta}{\delta t}$$

所以式 (2.41) 可以化简为:

$$\begin{aligned}
 \dot{\mathbf{q}}_t &= \frac{1}{2} \begin{bmatrix} -[\omega]_{\times} & \omega \\ -\omega^T & 0 \end{bmatrix} \mathbf{q}_t \\
 &= \frac{1}{2} \boldsymbol{\Omega}(\omega) \mathbf{q}_t \\
 &= \frac{1}{2} \begin{bmatrix} \omega \\ 0 \end{bmatrix} R \mathbf{q}_t \\
 &= \frac{1}{2} [\mathbf{q}_t]_L \otimes \begin{bmatrix} \omega \\ 0 \end{bmatrix}
 \end{aligned} \tag{2.42}$$

假设, \mathbf{a}_t 和 ω_t 表示 IMU 真实的加速度和角速率。假设用 $\hat{\mathbf{a}}_t$ 和 $\hat{\omega}_t$ 表示加速度和角速率的观测值, 即传感器读数, 测量噪声为 \mathbf{n}_a 和 \mathbf{n}_w , 那么,

$$\begin{aligned}
 \hat{\mathbf{a}}_t &= \mathbf{a}_t + \mathbf{b}_{at} + \mathbf{R}_w^t \mathbf{g}^w + \mathbf{n}_a \\
 \hat{\omega}_t &= \omega_t + \mathbf{b}_{wt} + \mathbf{n}_w.
 \end{aligned} \tag{2.43}$$

其中 \mathbf{R}_w^t 的作用是将重力矢量从世界坐标系下变换到 IMU 坐标系下。需要注意的是, 式 (2.43) 忽略了地球自转角速率 $\omega_{\mathcal{E}}$, 否则将是: $\hat{\omega}_t = \omega_t + \mathbf{R}_w^t \omega_{\mathcal{E}} + \mathbf{b}_{wt} + \mathbf{n}_w$ 。在大多数情况下, 不需要考虑地球自转, 但是, 当使用的 IMU 传感器分辨率很高, 且具有非常小的噪声和 bias 时, $\omega_{\mathcal{E}}$ 可能会被传感器测量到 ($\omega_{\mathcal{E}} = 15^\circ/h \approx 7.3 \times 10^{-5} rad/s$)。在这种情况下, 为了保持 IMU 误差模型的有效性, 在公式中不应该忽略 $\omega_{\mathcal{E}}$ 。

由式 (2.43) 可以得到 IMU 真实的加速度 \mathbf{a}_t 和角速率 ω_t :

$$\begin{aligned}
 \mathbf{a}_t &= \hat{\mathbf{a}}_t - \mathbf{b}_{at} - \mathbf{R}_w^t \mathbf{g}^w - \mathbf{n}_a \\
 \omega_t &= \hat{\omega}_t - \mathbf{b}_{wt} - \mathbf{n}_w.
 \end{aligned} \tag{2.44}$$

IMU 的测量值是在 IMU 自身坐标系下测量得到的, 其中包含了重力矢量, 并且受到加速度 bias、陀螺仪 bias 以及附加测量噪声的影响。假设加速度计和陀螺仪附加的测量噪声为高斯白噪声, $\mathbf{n}_a \sim \mathcal{N}(0, \sigma_a^2)$, $\mathbf{n}_w \sim \mathcal{N}(0, \sigma_w^2)$ 。加速度计 bias 和陀螺仪 bias 被建模为随机游走, 随机游走是一个离散模型。将维纳过程 (Wiener Process)^[39] 离散化就是随机游走。而 Wiener Process 是高斯白噪声的积分, 故其导数为高斯分布模型, $\mathbf{n}_{b_a} \sim \mathcal{N}(0, \sigma_{b_a}^2)$, $\mathbf{n}_{b_w} \sim \mathcal{N}(0, \sigma_{b_w}^2)$ 。所以, 可得:

$$\dot{\mathbf{b}}_{at} = \mathbf{n}_{b_a}, \quad \dot{\mathbf{b}}_{wt} = \mathbf{n}_{b_w} \tag{2.45}$$

所以，IMU 的运动方程为：

$$\left\{ \begin{array}{l} \dot{\mathbf{p}}_t = \mathbf{v}_t \\ \dot{\mathbf{v}}_t = \mathbf{a}_t = \hat{\mathbf{a}}_t - \mathbf{b}_{at} - \mathbf{R}_w^t \mathbf{g}^w - \mathbf{n}_a \\ \dot{\mathbf{q}}_t = \frac{1}{2} \boldsymbol{\Omega}(\omega) \mathbf{q}_t = \frac{1}{2} \boldsymbol{\Omega} (\hat{\boldsymbol{\omega}}_t - \mathbf{b}_{wt} - \mathbf{n}_w) \mathbf{q}_t \\ \dot{\mathbf{b}}_{at} = \mathbf{n}_{b_a} \\ \dot{\mathbf{b}}_{wt} = \mathbf{n}_{b_w} \\ \dot{\mathbf{g}}^w = 0 \end{array} \right. \quad (2.46)$$

2.1.3 磁力计数学模型

磁力计也叫电子罗盘，能够测量地球磁场分布，进而计算载体的航向信息^[40]。如图2.4所示，是地球磁场分布图。地球是一个硕大的磁体，磁感线方向从南极到北极。在磁场的南北两极，磁场垂直于水平面，靠近地球赤道，磁场方向平行于水平面。值得注意的是地球磁场的两极和地球地理两极并不完全的重合，他们之间大约有 11° 的夹角。

地球上某一点的磁场可以用一个矢量 \mathbf{H} 来表示，矢量 \mathbf{H} 的大小和方向分别代表该点磁场的强度和磁场的方向。如图2.5所示，矢量 \mathbf{H} 可以被分解为两个水平分量 $\mathbf{H}_x, \mathbf{H}_y$ ，和一个垂直分量 \mathbf{H}_z 。 α 是要求的航向角，它是载体当前的正前方与磁北的夹角。航向角的变化值域是 $0^\circ \sim 360^\circ$ 。当磁力计水平放置时，磁场矢量 \mathbf{H} 在重力方向

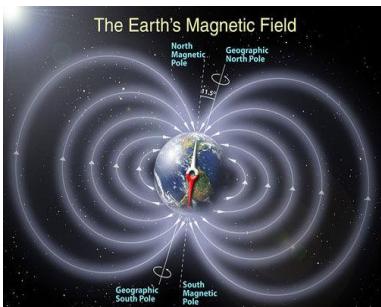


图 2.4 地球磁场分布图

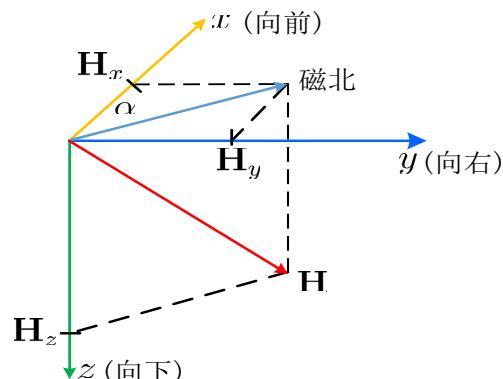


图 2.5 地磁场矢量分解示意图

z 轴上的分量 \mathbf{H}_z 为零，此时 α 为：

$$\alpha = \arctan \left(\frac{\mathbf{H}_y}{\mathbf{H}_x} \right) \quad (2.47)$$

但是，通常情况下磁力计不会和水平面完全平行，他们之间会有一个夹角，这个时候就不能通过式(2.47)来计算 α 的值。需要用加速度计来补偿这个夹角。假设磁力计和加速度计的坐标系重合，在静止或者匀速直线运动的情况下，加速度计测量的三个轴向的加速度是重力加速度在三个轴向的分量。假设在 x, y, z 三个轴的测得的重力分量分别为 $\mathbf{g}_x, \mathbf{g}_y, \mathbf{g}_z$ ，那么横滚角 γ 和俯仰角 θ 分别为：

$$\begin{aligned}\gamma &= \arctan\left(\frac{\mathbf{g}_y}{\mathbf{g}_z}\right) \\ \theta &= \arcsin\left(\frac{-\mathbf{g}_x}{\mathbf{g}}\right)\end{aligned}\quad (2.48)$$

假设磁力计输出的三轴磁分量为 $\mathbf{M}_x, \mathbf{M}_y, \mathbf{M}_z$ ，可以根据载体的横滚角和俯仰角计算出当地磁场的水平分量 $\mathbf{H}_x, \mathbf{H}_y$ ：

$$\begin{aligned}\mathbf{H}_x &= \mathbf{M}_x \cos\theta + \mathbf{M}_y \sin\theta \sin\gamma + \mathbf{M}_z \sin\theta \cos\gamma \\ \mathbf{H}_y &= \mathbf{M}_z \sin\gamma - \mathbf{M}_y \cos\gamma\end{aligned}\quad (2.49)$$

将式(2.49)代入式(2.48)可得航向角：

$$\alpha = \arctan\left(\frac{\mathbf{M}_z \sin\gamma - \mathbf{M}_y \cos\gamma}{\mathbf{M}_x \cos\theta + \mathbf{M}_y \sin\theta \sin\gamma + \mathbf{M}_z \sin\theta \cos\gamma}\right)\quad (2.50)$$

2.1.4 传感器选型

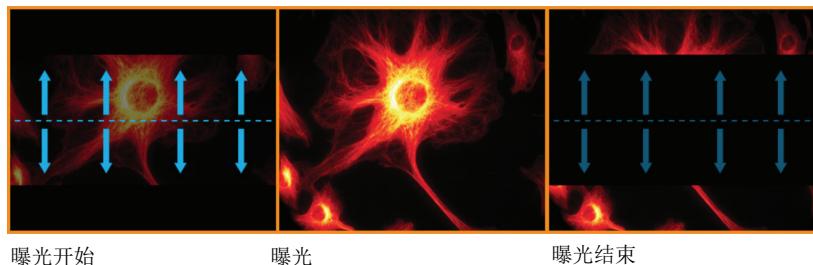
考虑到本系统的应用环境以及相机和IMU之间需要硬件同步等需求，所以需要慎重选择满足需求的相机和IMU。

(1) 单目相机及镜头选型

首先，考虑使用全局快门相机还是卷帘快门相机，这两种相机的成像方式有区别。如图2.6，全局相机是一次性成像，所有像素同时开始和停止曝光，而卷帘相机是逐行成像，每个像素曝光时间不同。因此当拍摄物体运动速度越快，卷帘相机的“果冻效应”就越明显，如图2.7左图所示。图2.7右图是的全局快门相机拍摄的图片，可以看到虽然出现了图像模糊问题，但是并没有出现“果冻效应”。对于SLAM系统，高速运动是普遍存在的，因此为了提高系统的鲁棒性，本文选用全局快门相机。其次，对于视觉SLAM系统来说，不需要彩色的图像，利用黑白的图像就可以进行特征点提取与匹配，所以选取灰色相机即可。而且，相机的视角不能太小，太小的话容易导致特征

跟踪失败。考虑到室外的环境，相机焦距也不能太短，否则远处的场景会模糊。最后，为了让相机能够在不同的季节和地区正常工作，应该选取具有宽温特性的工业相机。

(a) 卷帘快门相机曝光（单帧）



(b) 全局快门相机曝光（单帧）

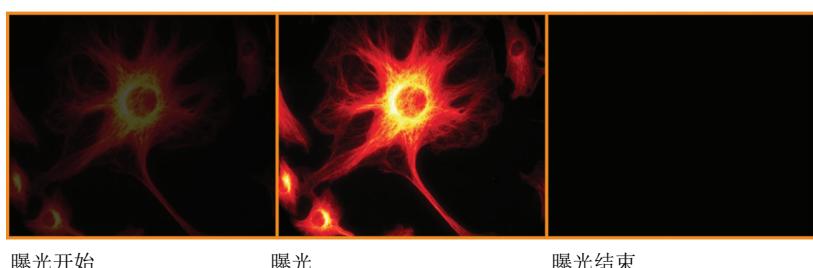


图 2.6 全局相机和卷帘快门相机成像方式

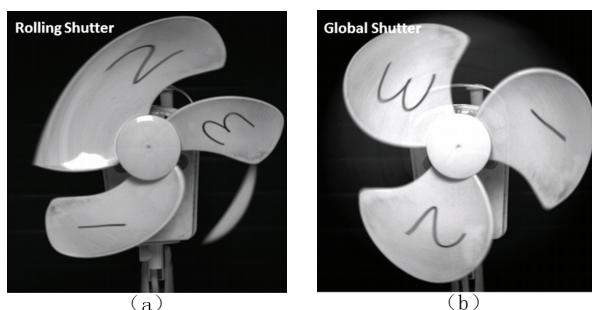


图 2.7 高速运动下的 (a) 卷帘快门相机图像; (b) 全局相机图像

基于以上分析，本系统选择灰点公司的 Grasshopper3-USB3 单目相机，如图2.8所示。采用快速、高灵敏度的 CMOS 芯片——IMX174，可提供最高 1920×1200 的图像分辨率，最高帧率达到 163FPS。该相机采用 USB 3.0 接口，非常适合于处理由传感器产生的远远超过 360MByte/s 的高速数据传输率，并且比其他高速数字接口更具成本效益。该相机的参数如表2.1所示。

相机镜头选用沃乐斯 WL1412-1K-1 工业定焦镜头，其外观和尺寸如图2.9所示。表2.2是沃乐斯 WL1412-1K-1 镜头参数。

(2) IMU 选型。

考虑到本系统主要在汽车或无人机上使用，所以光纤陀螺和激光陀螺等大型的



图 2.8 Grasshopper3-USB3 相机及其三视图

表 2.1 相机参数

型 号	GS3-U3-123S6M-C
分 辨 率	1920 x 1200
帧 率	163 FPS
传 感 器	Sony IMX174, CMOS, 1/1.2"
快门方式	全局快门
模数转换	10 位/12 位
曝光时间	0.005 ms ~ 31.9 s
触发方式	Standard, bulb, overlapped, multi-shot
输出格式	Mono8, Mono12, Mono16
数据接口	USB 3.0
传输速率	5 Gbit/s
存 储	128MB 帧缓存, 2MB 永久闪存
外 观 尺 寸	44 mm×29 mm×58 mm
重 量	90g
功 耗	≤4.5 W
镜 头 接 口	C
工 作 温 度	0°C ~ +50°C

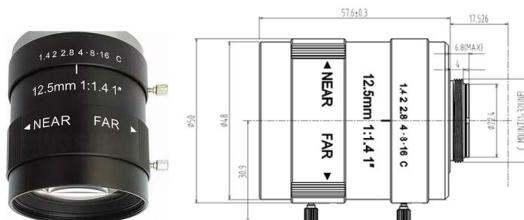


图 2.9 沃乐斯 WL1412-1K-1 镜头

高精度惯性器件虽然精度很高，但是通常体积大、重量大，所以不适用于本系统。而 MEMS 体积小，重量轻，非常适合车载和机载。但是普通的低价 MEMS 往往精度低，噪声大，会严重影响系统的输出精度。所以，需要尽可能的在成本可接受范围内选择一款相对来说精度高，噪声小，而且具备宽温特性的 MEMS。

通过以上分析以及多方对比，本系统的 IMU 选用挪威 SensoNOR 公司的 STIM300。

表 2.2 沃乐斯 WL1412-1K-1 镜头参数

型 号	GS3-U3-123S6M-C
焦 距	1920 x 1200
接 口	163 FPS
光圈范围	Sony IMX174, CMOS, 1/1.2"
视 场 角	全局快门
后 焦 距	10 位/12 位
畸 变 率	0.005 ms 31.9 s
分 辨 率	Standard, bulb, overlapped, multi-shot
外 观 尺 寸	Mono8, Mono12, Mono16
工 作 温 度	USB 3.0
重 量	5 Gbit/s

STIM300 提供了一个可供外部同步信号输入的 IO 口和一个与自身采样频率相同的同步方波输出 IO 口。STIM300 及其尺寸图如图2.10。表2.3是 STIM300 的性能参数。

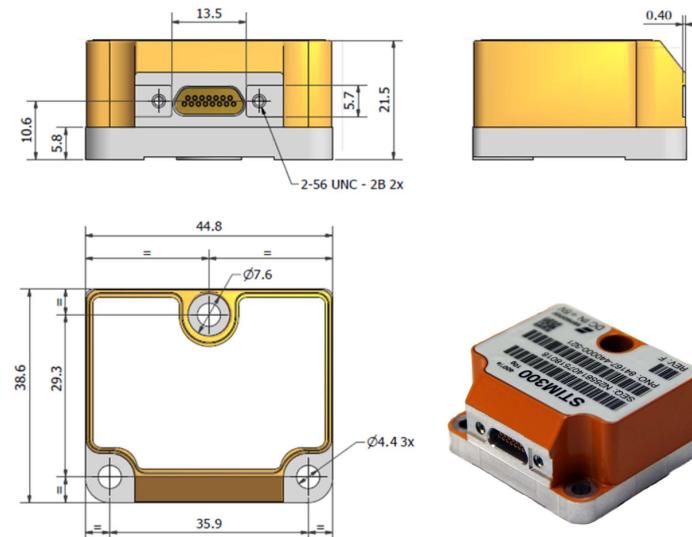


图 2.10 STIM300 及其尺寸

值得注意的是，由于本文中视觉/惯性融合系统是紧耦合系统，需要尽可能准确的时间同步，所以需要相机和 IMU 在硬件上进行同步。这就要求传感器具有触发或者被触发的功能，可以是相机触发 IMU，也可以是 IMU 触发相机。

2.2 传感器的标定与同步

尽管传感器在出厂时已经确定了具体的参数和指标，但是鉴于出厂时不可避免的有安装误差，以及传感器精度会随着使用环境的变化而变化，在使用传感器时有必要

表 2.3 STIM300 性能参数

整体指标		陀螺仪性能		加速度计性能	
重 量	55g	测量范围	$\pm 400^\circ/\text{s}$	测量范围	$\pm 10\text{g}$
输出形式	RS422	零 偏	$0.3^\circ/\text{h}$	零 偏	0.05mg
工作温度	-20°C +85°C	随机游走	$0.15^\circ/\sqrt{\text{h}}$	随机游走	$0.07\text{m/s}^\circ/\sqrt{\text{h}}$
工作电压	$5\text{V}\pm 0.5\text{V}$	分 辨 率	$0.22^\circ/\text{h}$	分 辨 率	$1.9\mu\text{g}$
功 耗	< 1.5W	全温零偏	$\pm 10^\circ/\text{h rms}$	全温零偏	$\pm 2\text{mg rms}$
采样频率	$\leq 2\text{kHz}$				

对其进行标定。

2.2.1 相机标定

表2.4 总结了常见的相机标定方法及其优缺点。其中使用最广泛的“张正友标定

表 2.4 常见的相机标定方法及其优缺点

算法分类	算法	优点	缺点	标定物
传统 标定法	直接线性变换 (DLT)	模型简单， 计算量小。	误差较大	3D 立体 靶标
	径向一致约束 (RAC)	标定精度高， 计算量适中。	-	3D 立体 靶标
	张正友标定法	计算量较小，精度 较高，应用广泛。	-	2D 棋盘格 标定板
主动视觉 标定法	基于纯旋转的标定	方法简单， 鲁棒性高。	成本高， 要求高。	无标定物
自标定法	Kruppa 方程标定法	方法简单， 灵活性强。	精度较差， 计算量大	无标定物

法”^[41]，该方法优点多，所以采用此方法标定相机参数。

“张正友标定法”使用的标定物是 2D 棋盘格，如图2.11所示。由2.1.1小节的针孔相机模型可知，标定用的棋盘格平面到像素平面的单应性关系为：

$$\mathbf{P}_{uv} = \mathbf{K} \mathbf{T} \mathbf{P}_w$$

其中， \mathbf{P}_w 为世界坐标点， $\mathbf{P}_w = [X, Y, Z]^T$ ， \mathbf{P}_{uv} 为像素坐标点， $\mathbf{P}_{uv} = [u, v]^T$ 。令 $\mathbf{T} = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{t}]$ ，棋盘格所在平面为 $Z = 0$ 的平面，则，

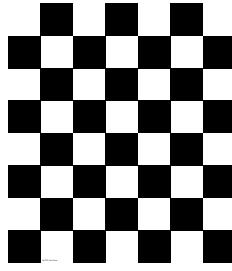


图 2.11 7×6 棋盘格

$$\begin{aligned}
 s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} &= \mathbf{K} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \\
 &= \mathbf{H} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}
 \end{aligned} \tag{2.51}$$

其中 s 是尺度因子， \mathbf{H} 是单应矩阵 $\mathbf{H} = \mathbf{K} [\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{t}]$ 。

\mathbf{H} 矩阵有八个未知数，棋盘格的每对对应角点能提供两个方程，因此至少需要四个对应角点就可以算出世界平面到像素平面的映射矩阵，即单应矩阵 \mathbf{H} 。

设 $\mathbf{H} = [\mathbf{h}_1 \quad \mathbf{h}_2 \quad \mathbf{h}_3] = \lambda \mathbf{K} [\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{t}]$ ，由式 (2.51) 得，

$$\begin{cases} \lambda = \frac{1}{s} \\ \mathbf{r}_1 = \frac{1}{\lambda} \mathbf{K}^{-1} \mathbf{h}_1 \\ \mathbf{r}_2 = \frac{1}{\lambda} \mathbf{K}^{-1} \mathbf{h}_2 \end{cases} \tag{2.52}$$

因为 \mathbf{r}_1 和 \mathbf{r}_2 是旋转矩阵，所以 \mathbf{r}_1 和 \mathbf{r}_2 正交，则

$$\begin{cases} \mathbf{r}_1^T \mathbf{r}_2 = 0 \\ \|\mathbf{r}_1\| = \|\mathbf{r}_2\| = 1 \end{cases} \tag{2.53}$$

由式(2.52)和(2.53)可得,

$$\begin{cases} \mathbf{h}_1^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{h}_2 = 0 \\ \mathbf{h}_1^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{h}_1 = \mathbf{h}_2^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{h}_2 = 1 \end{cases} \quad (2.54)$$

即一个单应性矩阵可以提供两个方程, 内参矩阵中有 5 个参数, 因此至少需要 3 个单应性矩阵才能求解内参矩阵 \mathbf{K} 。所以至少需要在 3 个不同的位置拍 3 张图片才能完成标定任务。

为了便于计算, 令

$$\begin{aligned} \mathbf{B} &= \mathbf{K}^{-T} \mathbf{K}^{-1} = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{\alpha^2} & -\frac{\gamma}{\alpha^2 \beta} & \frac{v_0 \gamma - u_0 \beta}{\alpha^2 \beta} \\ -\frac{\gamma}{\alpha^2 \beta} & \frac{\gamma^2}{\alpha^2 \beta^2} + \frac{1}{\beta^2} & -\frac{\gamma(v_0 \gamma - u_0 \beta)}{\alpha^2 \beta^2} - \frac{v_0}{\beta^2} \\ \frac{v_0 \gamma - u_0 \beta}{\alpha^2 \beta} & -\frac{\gamma(v_0 \gamma - u_0 \beta)}{\alpha^2 \beta^2} - \frac{v_0}{\beta^2} & \frac{(v_0 \gamma - u_0 \beta)^2}{\alpha^2 \beta^2} + \frac{v_0}{\beta^2} + 1 \end{bmatrix} \end{aligned} \quad (2.55)$$

注意, \mathbf{B} 是一个对称矩阵, 其有效元素为 6 个, 将这 6 个元素写成向量的形式,

$$\mathbf{b} = [B_{11}, B_{12}, B_{22}, B_{13}, B_{23}, B_{33}] \quad (2.56)$$

令 \mathbf{h}_i 为 \mathbf{H} 的第 i 个列向量, 则

$$\mathbf{h}_i = [h_{i1}, h_{i2}, h_{i3}]^T \quad (2.57)$$

所以,

$$\mathbf{h}_i \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{h}_j = \mathbf{h}_i \mathbf{B} \mathbf{h}_j = \mathbf{v}_{ij}^T \mathbf{b} \quad (2.58)$$

其中,

$$\mathbf{v}_{ij} = [h_{i1}h_{j1} \ h_{i1}h_{j2} + h_{i2}h_{j1} \ h_{i2}h_{j2} \ h_{i3}h_{j1} + h_{i1}h_{j3} \ h_{i3}h_{j2} + h_{i2}h_{j3} \ h_{i3}h_{j3}]^T$$

进一步, 由式(2.51)可推得,

$$\begin{cases} \mathbf{v}_{22}^T \mathbf{b} = 0 \\ \mathbf{v}_{11} \mathbf{b} = \mathbf{v}_{12} \mathbf{b} \end{cases} \quad (2.59)$$

写成矩阵形式,

$$\begin{bmatrix} \mathbf{v}_{12}^T \\ \mathbf{v}_{11} - \mathbf{v}_{22} \end{bmatrix} \mathbf{b} = 0 \quad (2.60)$$

这是一张棋盘格图像得到的约束等式, 当有 n 张图像时,

$$\mathbf{V}\mathbf{b} = 0 \quad (2.61)$$

其中, \mathbf{V} 是一个 $2n \times 6$ 的矩阵, \mathbf{b} 是一个 6 维列向量, 所以, 当有三张或者三张以上棋盘格图像时, 就可以计算得到 \mathbf{B} , 进而得到相机的内参数矩阵 \mathbf{K} 。

以上结果是在理想情况下推导得出的, 但是现实中往往存在高斯噪声, 所以为了解增加标定结果的可靠性, 使用 MLE (Maximum Likelihood Estimation) 来优化上面计算的结果。假设相机从不同的角度采集了 n 张包含棋盘格的图像, 每张图像里有 m 个棋盘格角点。设第 i 张图像上的第 j 个角点的三维点为 \mathbf{M}_{ij} , 则

$$\hat{m}(\mathbf{K}, \mathbf{R}_i, \mathbf{t}_i, \mathbf{M}_{ij}) = \mathbf{K}\mathbf{T}\mathbf{M}_{ij} \quad (2.62)$$

其中, $\hat{m}(\mathbf{K}, \mathbf{R}_i, \mathbf{t}_i, \mathbf{M}_{ij})$ 表示 \mathbf{M}_{ij} 对应的图像角点, $\mathbf{R}_i, \mathbf{t}_i$ 表示相机拍摄第 i 张图像时的旋转和平移, \mathbf{K} 是相机内参矩阵, 角点 m_{ij} 的概率密度函数是:

$$f(m_{ij}) = \frac{1}{\sqrt{2\pi}} e^{-\frac{-(\hat{m}(\mathbf{K}, \mathbf{R}_i, \mathbf{t}_i, \mathbf{M}_{ij}))^2}{\sigma^2}} \quad (2.63)$$

构造似然函数:

$$L(\mathbf{K}, \mathbf{R}_i, \mathbf{t}_i, \mathbf{M}_{ij}) = \prod_{i=1, j=1}^{n, m} f(m_{ij}) = \frac{1}{\sqrt{2\pi}} e^{-\frac{\sum_{i=1}^n \sum_{j=1}^m (\hat{m}(\mathbf{K}, \mathbf{R}_i, \mathbf{t}_i, \mathbf{M}_{ij}))^2}{\sigma^2}} \quad (2.64)$$

让 L 取得最大值, 需要最小化下面的式子:

$$\sum_{i=1}^n \sum_{j=1}^m \|\hat{m}(\mathbf{K}, \mathbf{R}_i, \mathbf{t}_i, \mathbf{M}_{ij}) - m_{ij}\|^2 \quad (2.65)$$

针对这个最优解问题, 可以使用 LM (Levenberg-Marquardt) 方法^[42] 求。

值得注意的是, “张正友标定法”只考虑了影响最大的径向畸变, 且只用了前两

个畸变参数，即式(2.12)中的 k_1, k_2 ，

$$\begin{cases} \hat{x} = x + x[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \\ \hat{y} = y + y[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \end{cases} \quad (2.66)$$

其中， (x, y) 和 (\hat{x}, \hat{y}) 分别是畸变前和畸变后归一化成像平面的坐标。

设 (u, v) 和 (\hat{u}, \hat{v}) 分别是畸变前和畸变后的像素坐标，式(2.8)中的 $\gamma = 0$ ，则，

$$\begin{cases} \hat{u} = u + (u - c_x)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \\ \hat{v} = v + (v - c_y)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \end{cases} \quad (2.67)$$

将式(2.67)写成矩阵形式：

$$\begin{bmatrix} (u - c_x)(x^2 + y^2) & (u - c_x)(x^2 + y^2)^2 \\ (v - c_y)(x^2 + y^2) & (v - c_y)(x^2 + y^2)^2 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} \hat{u} - u \\ \hat{v} - v \end{bmatrix} \quad (2.68)$$

式(2.68)是从一张图像上的一个角点得到的，假设相机从不同的角度采集了 n 张包含棋盘格的图像，每张图像里有 m 个棋盘格角点，那么总共可以得到 $2mn$ 个等式，记作，

$$\mathbf{D}\mathbf{k} = \mathbf{d} \quad (2.69)$$

可以得到，

$$\mathbf{k} = [k_1 \ k_2]^T = (\mathbf{D}^T \mathbf{D})^{-1} \mathbf{D}^T \mathbf{d} \quad (2.70)$$

同样，使用最大似然估计求最优解，使用LM方法最小化下面的式子求取参数，

$$\sum_{i=1}^n \sum_{j=1}^m \|\hat{m}(\mathbf{K}, \mathbf{k}, \mathbf{R}_i, \mathbf{t}_i, \mathbf{M}_{ij}) - m_{ij}\|^2 \quad (2.71)$$

本系统使用的灰点相机标定出的内参和畸变结果如下：

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1066.015725 & 0 & 474.867131 \\ 0 & 1066.986974 & 300.515111 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{k} = \begin{bmatrix} k_1 & k_2 \end{bmatrix}^T = \begin{bmatrix} -0.171420 & 0.210863 \end{bmatrix}^T$$

2.2.2 IMU 标定

在 2.1.3 小节研究了 IMU 的噪声模型，也就是加速度计和陀螺仪的高斯白噪声 \mathbf{n}_a \mathbf{n}_w 和，以及 bias \mathbf{b}_{at} 和 \mathbf{b}_{wt} 。需要标定的就是这四个值。

通常用 Allan 方差来分析 IMU 的噪声模型^[43]，以陀螺仪为例，Allan 方差可以分析如图2.12所示的五种典型误差。加速度噪声模型分析同理，不再累述。

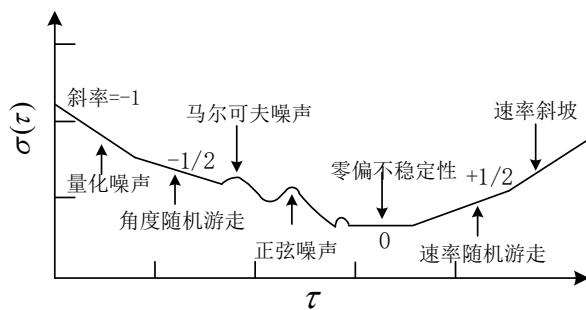


图 2.12 Allan 方差分析图

首先区分几个比较复杂的概念，在 2.1.3 小节中说的白噪声 $\mathbf{n}_w, \mathbf{n}_a$ 通常在 IMU 使用手册中被描述为角度随机游走和速度随机游走，这是因为对角速度（加速度）的白噪声进行积分就是角度随机游走（速度随机游走）。也有的 IMU 手册中叫角速度噪声密度和加速度噪声密度，甚至直接简称为噪声密度。

角速率随机游走就是 2.1.3 小节中说的 bias（随机游走） \mathbf{b}_w 。一般 IMU 手册上不直接给出 bias，而是给出 bias (in)stability，即零偏（不）稳定性。因为在实践中，bias 在长时间的积分中不会表现出真正的随机游走特性。bias (in)stability 的值可以近似地表示 bias 的精度。在实践中，可以根据白噪声的大小以及 bias (in)stability 的值来确定 bias 的合理值。

Allan 方差可以分析任何信号，以确定潜在噪声过程的特征。信号的 Allan 方差是平均时间的函数。对于平均时间 t ，Allan 方差计算如下：

- (1) 获取一长串数据并将其按时间分成 n 个长度为 t 的区间。 $n \geq 9$ （否则得到的结果开始失去其意义）。
- (2) 求取每个区间中的数据的平均值得到平均值列表 $(a(t)_1, a(t)_2, \dots, a(t)_n)$ 。

(3) 通过下面的式子计算 Allan 方差:

$$\text{AVAR}(t) = \frac{1}{2 \cdot (n - 1)} \sum_i (a(t)_{i+1} - a(t)_i)^2 \quad (2.72)$$

为了分析噪声特性, 需要计算 Allan 标准差:

$$\text{AD}(t) = \sqrt{\text{AVAR}(t)} \quad (2.73)$$

在对数标度上绘制 Allan 标准差曲线 (对数-对数 AD 曲线)。不同类型的随机噪声过程导致具有不同梯度的斜率。此外, 不同的随机噪声过程通常出现在 t 的不同区域中, 从而可以容易地分辨出他们。确定随机噪声过程之后, 可以直接从图中读取其数值参数。对于需要的白噪声, bias (随机游走) 以及 bias instability (零偏稳定性) 可以通过下面的方式读取:

(1) 对数-对数 AD 曲线上出现白噪声的地方的斜率为-1/2, 通过在斜率上拟合直线并在 $t = 1$ 处读取。

(2) 对数-对数 AD 曲线上出现 bias (随机游走) 的地方的斜率为 +1/2, 通过在斜率上拟合直线并在 $t = 3$ 处读取。

(3) 对数-对数 AD 曲线上出现 bias instability (零偏稳定性) 的地方是曲线最小值附近的平坦区域, 通过读取曲线上的最低点得到。

在 IMU 静止状态下采集四个小时的数据, 然后绘制陀螺仪和加速度计的 Allan 方差曲线, 如图2.13和图2.14所示。并将分析结果展示在表2.5中, 其中 Gyr_n 和 Acc_n 表示噪声项, Gyr_w 和 Acc_w 表示随机游走。

表 2.5 IMU 标定结果

	X 轴	Y 轴	Z 轴	平均值
Gyr_n (rad/s)	1.25251e-03	1.44058e-03	1.23255e-03	1.30854e-03
Gyr_w (rad/s)	4.15778e-06	6.66752e-06	1.79872e-06	4.20800e-06
Acc_n (m/s^2)	1.70161e-02	1.74575e-02	1.56289e-02	1.67009e-02
Acc_w (m/s^2)	4.58804e-04	4.14023e-04	4.63646e-04	4.45491e-04

2.2.3 同步与联合标定

在视觉和 IMU 融合系统中, 保证二者在时间上的同步非常重要。因为在初始化的时候需要将 IMU 预积分的值与视觉的观测值进行对齐, 从而恢复单目视觉的尺度、重力的方向以及其它初始化参数。只有保证了正确的初始值, 才能确保后端非线性优

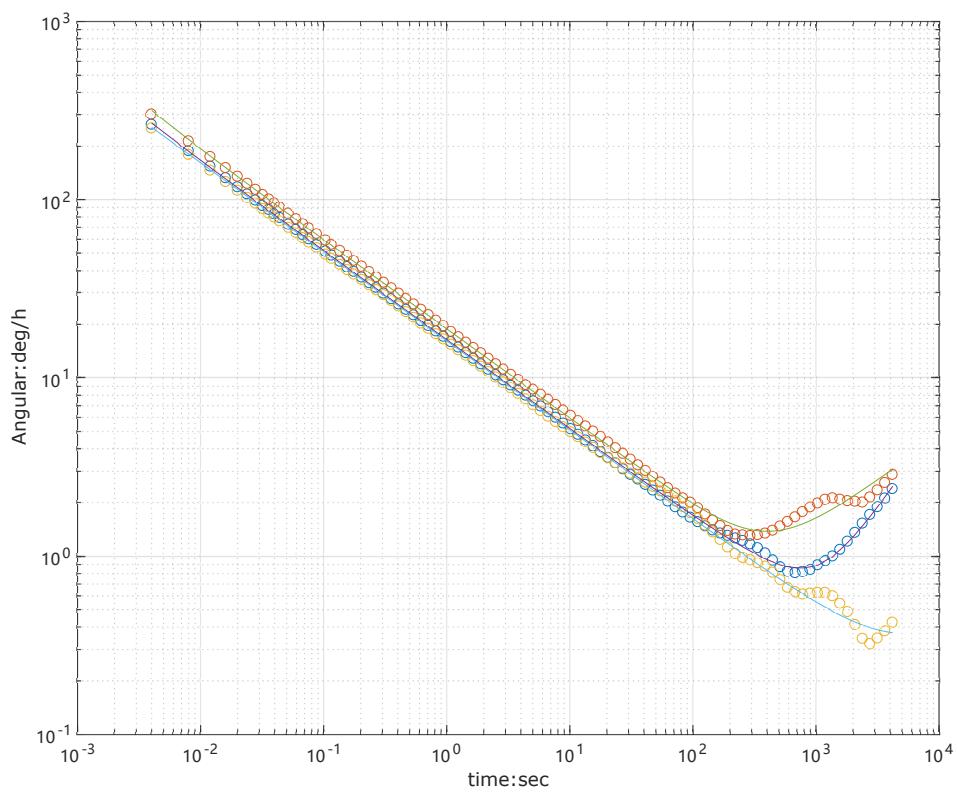


图 2.13 陀螺仪 Allan 方差曲线

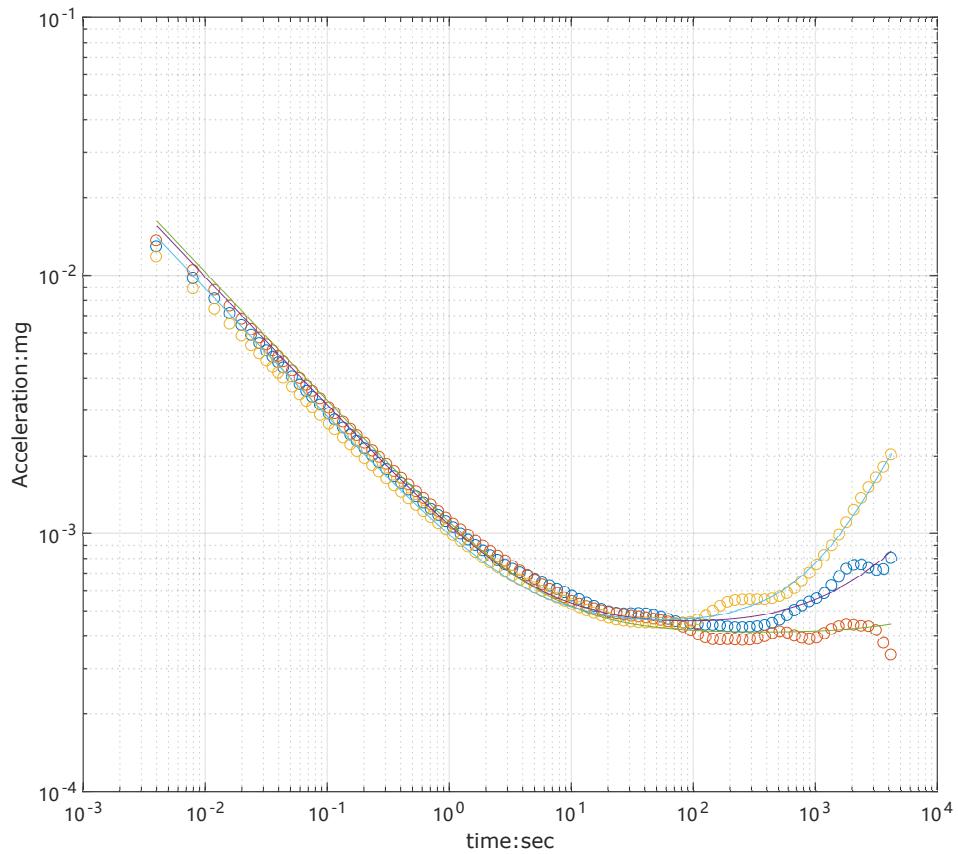


图 2.14 加速度计 Allan 方差曲线

化能够收敛到一个最优值。而 IMU 和视觉对齐的程度完全取决于时间戳的匹配精准度。为了尽可能的对齐相机和 IMU 的时间戳，对传感器进行硬件同步。

本文选用的灰点相机拥有外部触发模式，也就是能够通过外部脉冲来触发相机进行曝光。选用的 IMU 能够向外输出与自己等采样频率的方波。如图2.15所示，进行硬件同步的思路是：将 IMU 输出的方波（250Hz）通过硬件分频电路十分频，然后送到相机的触发端口（25Hz）进行触发曝光。

硬件同步为接下来的相机/IMU 联合标定奠定了基础。

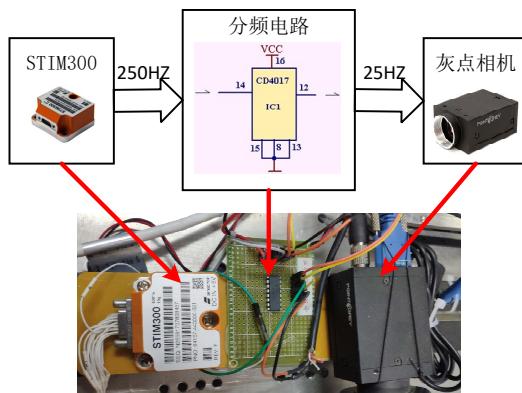


图 2.15 硬件同步示意图

使用开源的视觉/惯性标定工具：kalibr^{[44] [45]} 进行联合标定。联合标定需要动态标定，需要的输入文件包括：相机内参及畸变参数，IMU 噪声参数，带 IMU 信息的视频以及标定板信息。其中，相机内参和 IMU 噪声参数是2.2.1和2.2.2小节标定出来的结果。视频是在不同角度下拍摄的标定板图像，需要注意的是，在拍摄视频时需要激活 IMU 所有的轴向信息（包括偏航、横滚和俯仰）。采用的标定板和传统标定相机的棋盘格标定板（图2.11）不一样，是 Aprilgrid 标定板，如图2.16所示。Aprilgrid 标定板能够给出序号信息，可以避免姿态计算出现跳跃的情况。视 IMU 坐标系和载体坐标系重合，相机和 IMU 的坐标系定义如图2.17所示。

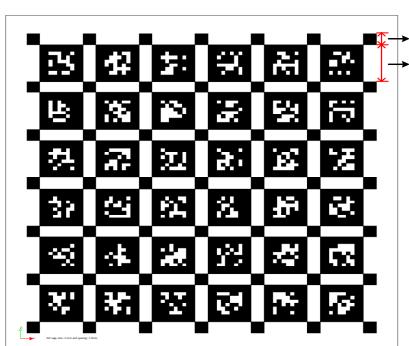


图 2.16 Aprilgrid 标定板

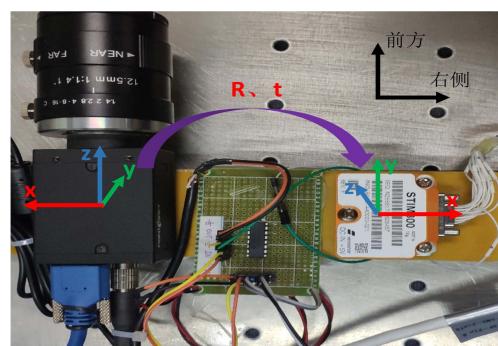


图 2.17 相机和 IMU 的坐标系定义

联合标定主要标定如下参数：

- (1) T_{ic} : 与 IMU 的外参。即相机到 IMU 的变换矩阵。
- (2) $t_{meshift}$: 相机与 IMU 之间的时间偏移。这个值可以反应出时间同步的精度， $t_{meshift}$ 越小表明同步效果越好。
- (3) Gravity vector: 重力矢量。重力的大小能够反映出标定的精度。

另外，kalibr 还会给出加速度和角速度的误差图，用来评估 IMU 的精度，以及相机重投影误差，可以用来评估的标定结果。

表2.6是的相机和 IMU 联合标定的结果。可见相机和 IMU 之间的平移在 10cm 左右，这符合本系统实际安装距离。时间偏移约为 9ms。重力大小是 $9.8m/s^2$ 。可见，标定出的结果较为可靠。

表 2.6 联合标定结果

T_{ic}	$\begin{bmatrix} 0.99956 & 0.02956 & 0.00015 & 0.10048 \\ 0.00028 & -0.00458 & -0.99999 & 0.01208 \\ -0.02955 & 0.99955 & -0.00459 & -0.07922 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Timeshift(s)	-0.009307066448628993
Gravity vector(m/s^2)	[0.12538627 -9.80181878 -0.27757847]

图2.18是 IMU 角速度误差和加速误差。可见，角速度误差在 $\pm 0.04rad/s$ 范围内波动，加速度误差在 $0.5m/s^2$ 范围内波动。

图2.19是相机的重投影误差。可见，重投影误差在 ± 0.8 个像素单位波动。这表明相机内参和畸变参数标定的很准确。

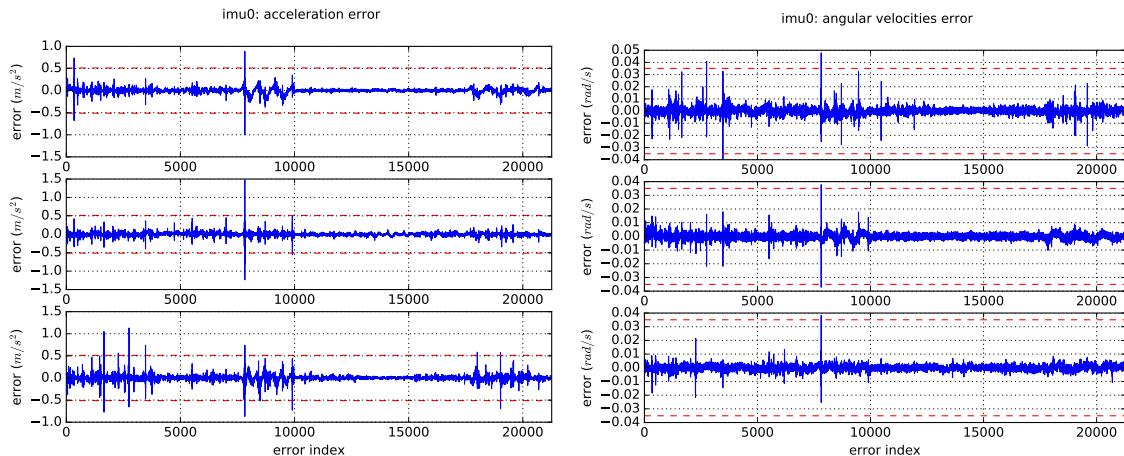


图 2.18 IMU 角速度误差和加速误差

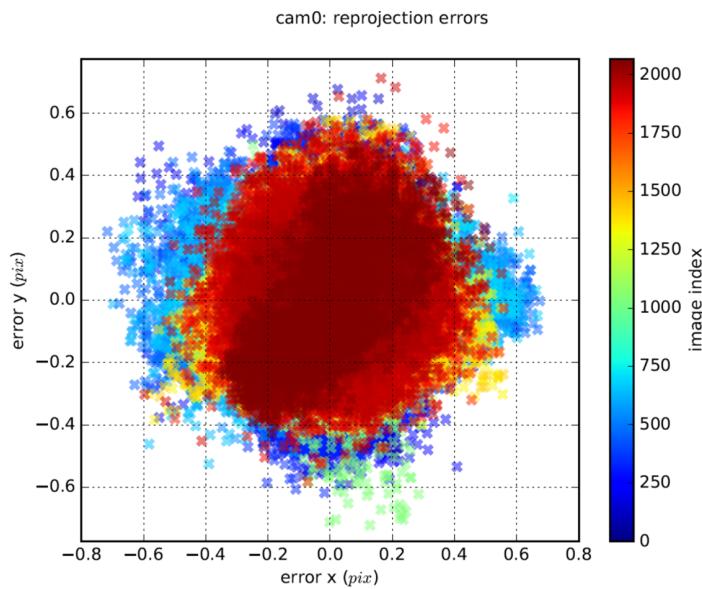


图 2.19 相机重投影误差

2.3 硬件组建与 ROS 信息采集

2.3.1 硬件组建

本系统的硬件系统构成如图2.20所示。

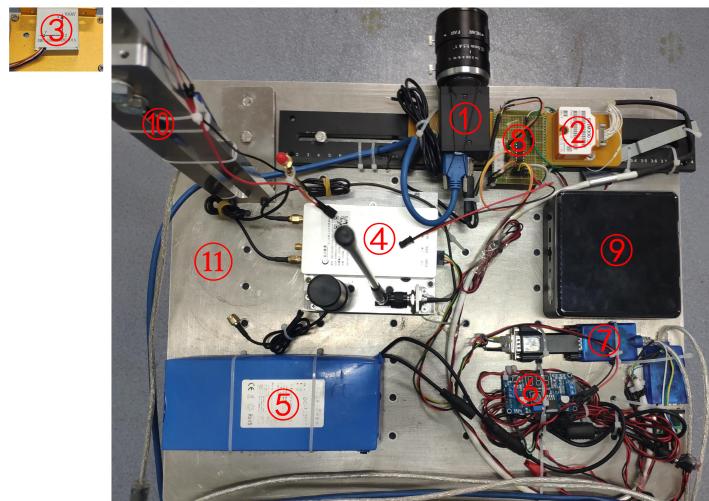


图 2.20 硬件系统构成

图中 ① 是灰点单目相机(配镜头), ② 是 IMU(STIM300), ③ 是磁力计, ④ 是 RTK, ⑤ 是 12v 锂电池, ⑥ 是稳压模块, ⑦ 是 RS232 串口转 USB, ⑧ 是相机和 IMU 的同步电路, ⑨ 是 Intel NUC 处理器, ⑩ 是磁力计和底座连接的连接杆, ⑪ 是底座。其中, RTK 不属于多传感器融合系统, 主要用他来采集差分 GPS 信号, 将 GPS 的轨迹当作真实值并和本系统输出轨迹作比较, 方便分析系统的定位精度。为了避免载体(汽车)本身对磁力计的干扰, 将磁力计用 ⑩ 架高, 相对于底座的高度为 60cm。

而且为了尽量避免铁磁性物体对磁场的干扰，硬件系统的底座以及连接杆都是铝制的。

2.3.2 ROS 信息采集

ROS (Robot Operating System) 译作机器人操作系统^[46]，通常应用在机器人上，它操作简单，功能强大，尤其适用于向机器人这种多任务、多节点的复杂环境。ROS 的功能非常丰富，本文不打算一一介绍，只对其通信机制中的“话题”作简要研究。

在 ROS 中，一个节点 (Node) 就是一个可执行文件。由于机器人的模块众多，通常不会将所有的模块放在一个节点中实现，而是一个节点实现一个模块。比如，节点 1 用来驱动相机获取图像，节点 2 用来驱动电机控制机器人移动，节点 3 用来接收图像并进行定位……这样以来可以降低系统的崩溃频率，使程序便于维护。在 ROS 的众多通信方式中，话题 (topic) 是最常用的一种。话题适合处理类似传感器信息这样的实时性、周期性的消息。话题是一种节点与节点之间的单向异步通信方式。发布节点 (Publisher Node) 发布话题到节点管理器 Master，订阅节点从 Master 订阅话题。订阅节点接收到话题消息后会触发回调函数，在回调函数里对话题消息进行处理。

如2.21所示，使用节点 1 采集相机图像，话题为/camera/image_raw；节点 2 采集 IMU 数据，话题为 imu_data；节点 3 采集磁力计数据，话题为/dmc_data；节点 4 采集 RTK 数据，话题为/rtk_data。节点 5, 6, 7 分别代表前端、后端和回环。节点 1~4 采集到的传感器数据将被以话题的形式发布到节点管理器 Matser，然后节点 5, 6, 7 都可以独立的订阅所有话题，并在自己的回调函数进行处理，各个节点互不干扰，独立运行，这就是话题单向异步通信的优势。值得称赞的是，使用 ROS 采集传感器信息的时候可以将所有话题记录下来，之后可以像回放视频那样重复播放，达到重复使用数据的效果。

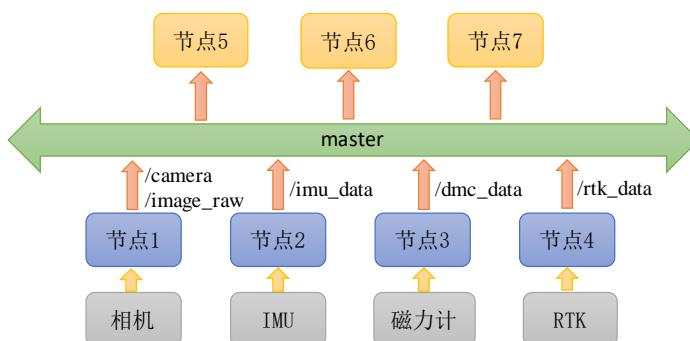


图 2.21 ROS 采集传感器信息及同通信示意图

2.4 本章小结

本章主要研究系统的硬件部分，包括传感器模型及其选型，传感器标定与同步，以及硬件系统搭建和传感器信息采集。对传感器的数学模型进行了详细的推导，并详细研究了传感器的标定原理和步骤。完善可靠的硬件系统是后续进行算法研究和实验验证的基础。

第3章 系统前端及初始化

本章主要研究系统的前端和初始化相关内容，其中前端主要包括视觉信息预处理和 IMU 预积分，视觉信息预处理包括特征提取和跟踪，异常点剔除。初始化主要包括多视图几何基础，visual 初始化，以及 Visual-IMU 联合初始化。

3.1 视觉信息预处理

在视觉 SLAM 中，前端也称为视觉里程计（Visual Odometer，VO）^[47]。VO 的实现方法有很多，包括特征点法，直接法和光流法。特征点法需要提取特征点，这包括提取关键点和计算描述子，然后进行特征匹配；直接法是从光流法继承而来的，不需特征提取及匹配；光流法不进行特征点匹配，取而代之的是利用光流跟踪特征点。表3.1列出了三种方法的优缺点。综合考虑，本文选用光流法作为视觉前端算法。

表 3.1 特征点法、直接法和光流法对比

优缺点	特征点法	直接法	光流法
优点	(1) 对比直接法和光流法，鲁棒性高；(2) 不受照影响。	(1) 不需要特征点提取和匹配，计算量小；(2) 可以在无纹理或者重复纹理场景下工作；(3) 可以构建半密集或密集地图。	(1) 不需要计算描述子和匹配特征，计算量小；(2) 可以在无纹理或者重复纹理场景下工作；(3) 容易移植到嵌入式系统中，方便 IMU 进行融合；(4) 可以构建半密集或者密集的地图。
缺点	(1) 在无纹理或者重复纹理场景下无法工作；(2) 计算量大，不能保证实时性；(3) 仅用 CPU 无法构建半稠密或稠密地图。	(1) 基于灰度不变假设，对光照变化敏感；(2) 单个像素没有区分度，需要计算像素块；(3) 图像是非凸的，容易产生局部极值。	(1) 基于灰度不变假设，对光照变化敏感；

3.1.1 关键点和关键帧

关键点也可以称为角点，它与特征点的区别是它没有描述子。常用的角点检测算法有 Harris 角点^[48] 和 FAST 角点^[49]。相比于 Harris 角点，FAST 角点检测速度更快，因为 FAST 角点不需要计算像素的梯度，仅仅通过比较周围像素的亮度来确定是否为角点。如图3.1，它的算法流程如下：

- (1) 选取图像中的一个像素 p ，设其亮度为 I_p ；
- (2) 设定一个合适的亮度阈值 T （例如设为 I_p 的 20%）；
- (3) 以像素 p 为圆心，画半径为 3 像素的圆，并选取圆上的所有像素（16 个）；
- (4) 如果在这 16 个像素点中，有连续 N 个点的亮度都大于 $I_p + T$ 或者小于 $I_p - T$ ，那么像素 p 就是角点。通常 N 取 12, 9 和 11，分别称为 FAST-12, FAST-9 和 FAST-11；
- (5) 对图像中的每一个像素执行以上步骤，提取所有的角点。

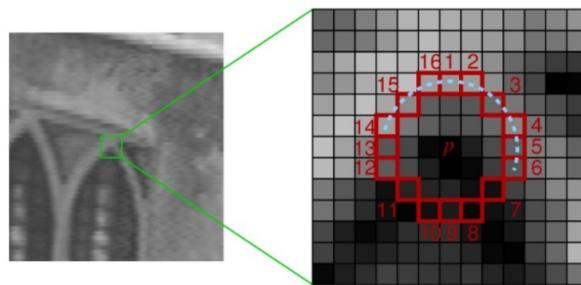


图 3.1 FAST 角点

针对 FAST-12 算法，为了提高检测效率，可以添加一个预判别的操作，用来排除掉大部分不是角点的像素点。方法为：直接检测像素 p 周围第 1, 5, 9 和 13 个像素点的亮度。当这四个像素中有三个像素的亮度同时大于 $I_p + T$ 或者小于 $I_p - T$ 时，该像素才有可能是角点，才会继续利用 FAST-12 进行角点检测，否则直接排除该像素。这样的预判别操作大幅度提升了角点检测的速度。

增加预判操作确实可以大大提高算法效率，但是到此为止算法还存在一个缺点，检测出来的很多相邻的角点挤在一起。为了消除这种邻近位置有多个角点重合的情况，需要进行非大极值抑制（Non-Maximal Suppression, NMS）^[50]。方法如下：

- (1) 计算每一个检测到的角点的响应值（score function） V ，这里 V 定义为角点点 p 和它邻域 16 个像素点的绝对差值之和，即，

$$V = \sum_{x \in S} |I_{p \rightarrow x} - I_p|$$

其中 S 表示 16 的邻域像素点的集合。

(2) 考虑两个相邻的角点，并比较它们的 V 值，并删除 V 值较小的角点。图3.2展示了 FAST 角点提取。其中，(a) 是原始图像，(b) 是不加 NMS 的 FAST 角点提取，(c) 是加入 NMS 的 FAST 角点提取。

考虑到系统的实时性，不会对所有的图像都进行关键点提取，而是引入关键帧机制，即只对关键帧提取关键点，非关键帧则直接忽略。本文依据两个标准来选择关键

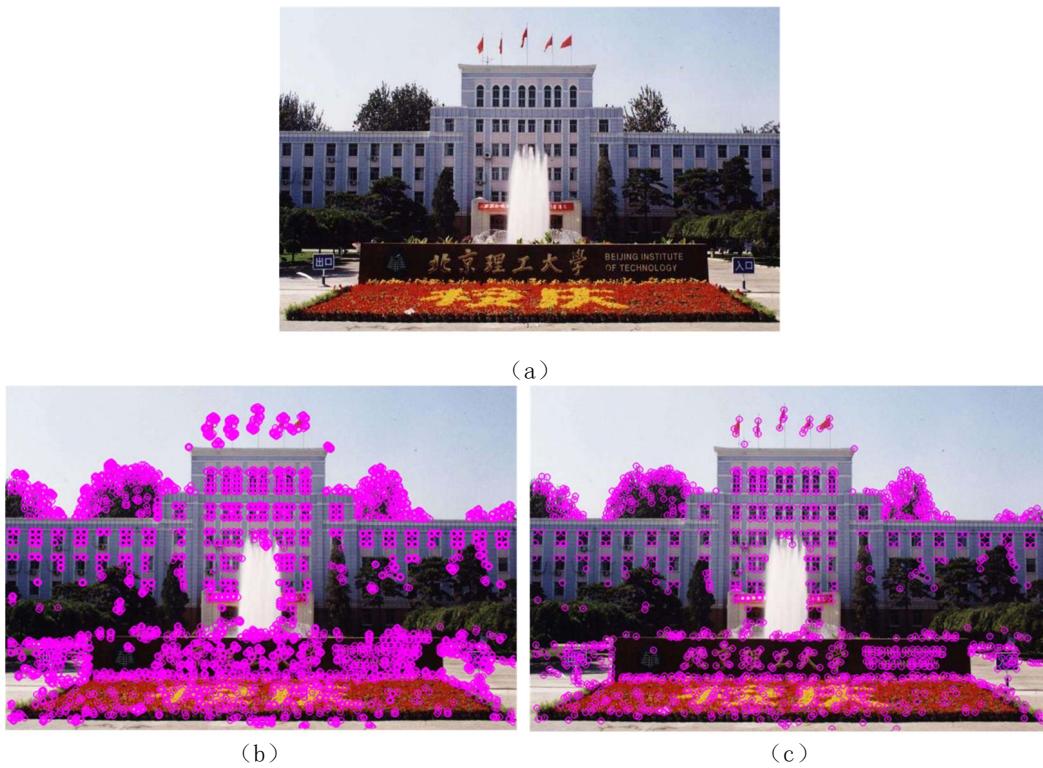


图 3.2 (a) 原始图像; (b) 不加 NMS 的 FAST 角点提取; (c) 加入 NMS 的 FAST 角点提取。

帧。第一，两个关键帧之间要有足够的视差。当跟踪到的特征的平均视差介于当前帧和最近的关键帧视差之间，则将当前帧视为新的关键帧。第二，每张图像跟踪到的关键点要大于某一阈值，否则，将该帧视为关键帧。这样做是为了防止图像关键点完全跟丢，导致系统崩溃。

3.1.2 特征跟踪

关键点检测出来后就需要对其进行跟踪，本文采用光流跟踪关键点。光流是用来描述像素在图像中随着时间运动的方法，如图3.3所示。跟踪部分像素的运动称为稀疏光流，跟踪所有像素的运动称为稠密光流。在 SLAM 中通常用 LK (Lucas-Kanade) 光流^[51] 来跟踪关键点的位置。

在 LK 光流中，相机是运动的，也就是说相机的图像随着时间的变化而变化。所以，图像和时间之间存在一个映射关系： $I(t)$ 。那么，在 t 时刻，图像中 (x, y) 处的像素的灰度可以记作： $I(x, y, t)$ 。这样，图像就是关于位置和时间的函数，该函数的值域就是图像的灰度。假设三维空间中的一个点 p 在 t 时刻时的像素坐标为 (x, y) 。随着时间的变化， p 的像素坐标将发生变化。要估计出每一时刻该点的像素坐标。首先，需要提出一个基本的假设：灰度不变假设。即三维空间中的点在每张图像中的像素灰

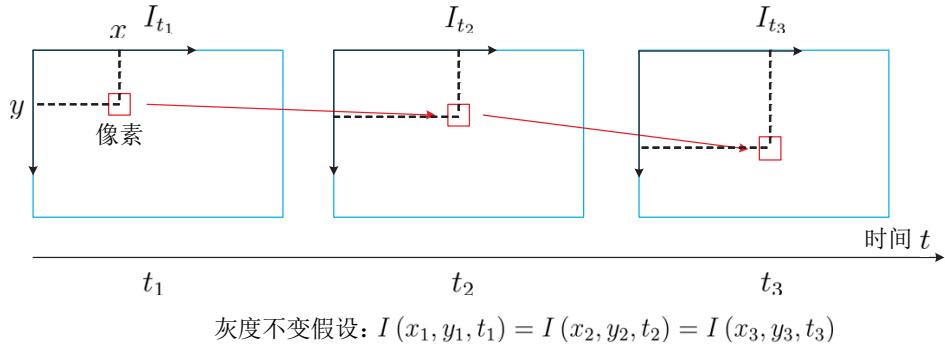


图 3.3 LK 光流法示意图

度值是固定不变的。例如，在 $t + dt$ 时刻点 p 的像素坐标为 $(x + dx, y + dy)$ ，则，

$$\mathbf{I}(x + dx, y + dy, t + dt) = \mathbf{I}(x, y, t) \quad (3.1)$$

对式(3.1)左边用泰勒公式展开并保留一阶项，得，

$$\mathbf{I}(x + dx, y + dy, t + dt) \approx \mathbf{I}(x, y, t) + \frac{\partial \mathbf{I}}{\partial x} dx + \frac{\partial \mathbf{I}}{\partial y} dy + \frac{\partial \mathbf{I}}{\partial t} dt \quad (3.2)$$

由式(3.1)和(3.2)得，

$$\frac{\partial \mathbf{I}}{\partial x} dx + \frac{\partial \mathbf{I}}{\partial y} dy + \frac{\partial \mathbf{I}}{\partial t} dt = 0 \quad (3.3)$$

$$\frac{\partial \mathbf{I}}{\partial x} \frac{dx}{dt} + \frac{\partial \mathbf{I}}{\partial y} \frac{dy}{dt} = -\frac{\partial \mathbf{I}}{\partial t} \quad (3.4)$$

其中， dx/dt 和 dy/dt 分别表示像素在 x 轴方向和 y 轴方向上的速度分量，将其分别记为 u, v 。 $\partial \mathbf{I}/\partial x$ 和 $\partial \mathbf{I}/\partial y$ 表示 x 轴方向和 y 轴方向上的像素梯度，分别记为 $\mathbf{I}_x, \mathbf{I}_y$ 。 $\partial \mathbf{I}/\partial t$ 表示图像灰度随时间的变化量，记为 \mathbf{I}_t 。这样可以将式(3.4)写成如下矩阵形式，

$$\begin{bmatrix} \mathbf{I}_x & \mathbf{I}_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -\mathbf{I}_t \quad (3.5)$$

假设在图像中的某一个 $w \times w$ 大小的窗口内所有的像素（共有 w^2 个）具有相同的运动，那么可以得到 w^2 个方程，

$$\begin{bmatrix} \mathbf{I}_x & \mathbf{I}_y \end{bmatrix}_k \begin{bmatrix} u \\ v \end{bmatrix} = -\mathbf{I}_{tk}, \quad k = 1, \dots, w^2 \quad (3.6)$$

记,

$$A = \begin{bmatrix} [\mathbf{I}_x, \mathbf{I}_y]_1 \\ \vdots \\ [\mathbf{I}_x, \mathbf{I}_y]_k \end{bmatrix}, \mathbf{b} = \begin{bmatrix} \mathbf{I}_{t1} \\ \vdots \\ \mathbf{I}_{tk} \end{bmatrix} \quad (3.7)$$

所以, 式 (3.6) 可以写为,

$$\mathbf{A} \begin{bmatrix} u \\ v \end{bmatrix} = -\mathbf{b} \quad (3.8)$$

使用最小二乘法求解式 (3.8),

$$\begin{bmatrix} u \\ v \end{bmatrix}^* = -(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} \quad (3.9)$$

这样就可以计算处像素的运动速度 u, v , 从而计算出像素在图像中的位置。也就实现了使用 LK 光流来跟踪角点的目的。图3.4是光流跟踪 FAST 角点示意图。其中, 左图 (a) 中的点是检测到的 FAST 角点, 右图 (b) 是相机向左平移一段距离拍摄的图像, 蓝线是 LK 光流跟踪角点的轨迹。



图 3.4 光流跟踪 FAST 角点

3.1.3 异常点剔除

在视觉 SLAM 中, 不管是特征点匹配还是光流跟踪都会出现误匹配或误跟踪的情况, 那些匹配正确或者跟踪正确的点称为“局内点 (inlier)”, 而匹配错误或者跟踪错误的点称为“局外点 (outlier)”或者“异常点”。这些“异常点”会影响 SLAM 的定位精度, 因此需要对其进行剔除。

在视觉 SLAM 中剔除“outlier”常用的方法是 RANSAC (Random Sample Consensus)，该算法是 Fischler 和 Bolles 等人于 1981 年最先提出的^[52]。它可以得到有效的数据，即“局内点”，剔除无用数据，即“局外点”。在实践中，“异常点”通常是由于测量错误、假设错误和计算错误等原因造成，他们通常与正确数据“相距”很远。

该算法假设：

- (1) 样本数据集包含“局内点”和“局外点”；
- (2) “局外点”不满足计算假设的数学模型；
- (3) “局内点”满足某一数学模型，并可以计算其参数；
- (4) 其余数据是噪声数据。

该算法步骤如下：

- (1) 从样本数据集 S 中随机选取一小组点集 s (共有 n 个)，假设 s 的所有点为“局内点”，通过该点集得到一个数学模型 F ；
- (2) 用该数学模型 F 去测试 S 中除 s 外的其他点，并假设满足该数学模型的点为“局内点”；
- (3) 如果被假设为“局内点”的数量 n 大于设定的阈值 T ，那么就认为数学模型 F 是合理的；
- (4) 用所有假设的“局内点”重新估计数学模型；
- (5) 如果迭代次数大于等于 k ，则停止，否则重复上述步骤。

关键在于如何确定 k 值。假设在样本数据集中“局内点”的概率是 w ，则，

$$w = \frac{n_{inliers}}{n_{inliers} + n_{outliers}} \quad (3.10)$$

最少需要假设 n 个“局内点”确定模型 F ， n 视具体问题而定。 n 个假设的“局内点”都是实际的“局内点”的概率为 w^n ， n 个假设的“局内点”中至少有一个是实际的“局外点”的概率为 $1 - w^n$ 。 k 次随机选取的点集 s 中没有一次全部是“局内点”的概率是 $(1 - w^n)^k$ ，用 P 表示 k 次随机选取的点集 s 中全部都是“局内点”的概率， P 也表示取到正确解的概率或者置信度，则，

$$P = 1 - (1 - w^n)^k \quad (3.11)$$

“局内点”的概率 w 是一个先验值，可以根据先验假设一个初值。此时，可以计算出

迭代次数 k ,

$$k = \frac{\log(1 - P)}{\log(1 - w^n)} \quad (3.12)$$

以上是“异常点”剔除的通用步骤，下面研究 RANSAC 是如何剔除关键点的。以计算基本矩阵来剔除“异常点”为例，则上面研究的数学模型 F 为基本矩阵。使用 8 点法计算 F ，则 $n = 8$ 。图3.5展示了使用基本矩阵剔除异常点的步骤。

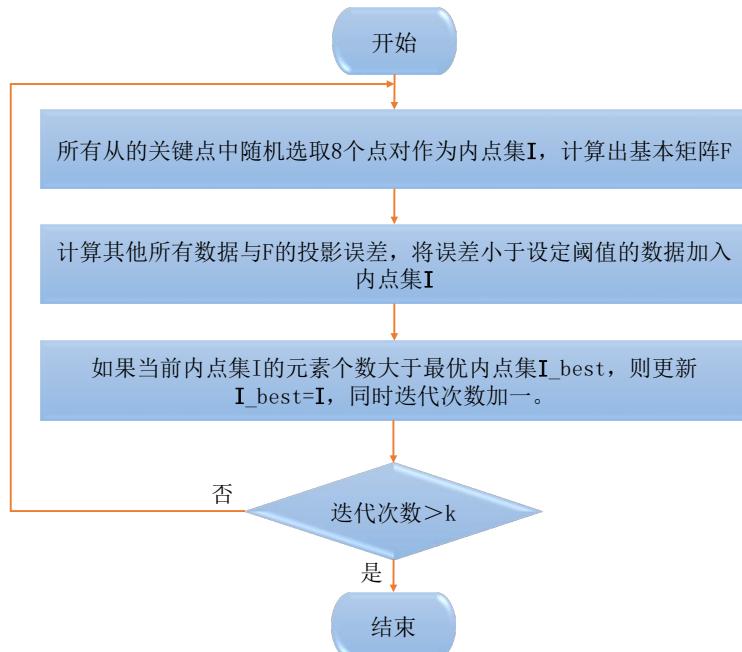


图 3.5 用基本矩阵剔除异常点

图3.6是异常点剔除前后对比图。其中，图 (a) 是 LK 光流跟踪到的 FAST 角点（没有进行异常点剔除），总共 1270 个点；图 (b) 是经过异常点剔除后的跟踪的角点，总共 775 个点。

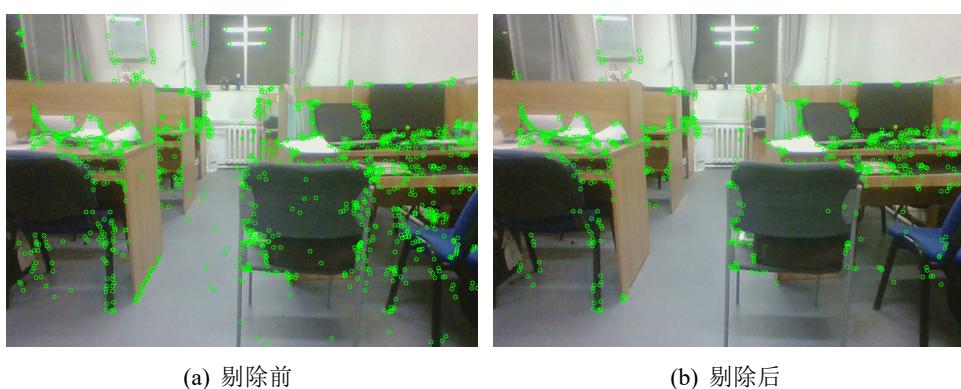


图 3.6 异常点剔除

3.2 IMU 预积分

通常情况下, IMU 的数据频率要远大于相机的数据频率, 不可能计算每一个 IMU 时刻状态量, 为了与视觉进行融合, 需要提前对相邻视觉帧之间的 IMU 数据进行积分, 这个步骤称为预积分^{[53][54]}。图3.7是预积分示意图。

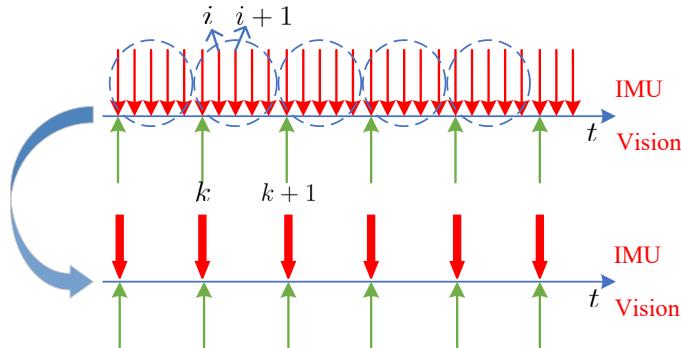


图 3.7 预积分示意图

首先定义本节使用得符号和坐标系。定义 $(\cdot)^w$ 是世界坐标系。重力的方向与世界坐标系的 z 轴对齐。 $(\cdot)^b$ 是本体坐标系, 本文把它定义为与 IMU 坐标系相同。 $(\cdot)^c$ 是相机坐标系。 \mathbf{q}_b^w , \mathbf{p}_b^w 是从本体坐标系到世界坐标系的旋转和平移。 b_k 是第 k 个图像对应的本体帧 (IMU 帧)。 c_k 是第 k 个相机帧。 \otimes 表示两个四元数之间的乘法运算。在世界坐标系下 $\mathbf{g}^w = [0, 0, g]^T$ 。 $(\hat{\cdot})$ 表示为某一量的噪声测量值或估计值。

3.2.1 当前帧的位置、速度和姿态

假设图像的第 $k + 1$ 帧为当前帧, 根据 2.1.3 小结研究的 IMU 状态模型, 将图像的第 k 帧和第 $k + 1$ 帧之间的 IMU 进行积分, 得到第 $k + 1$ 帧的 IMU 位置、速度和姿态。

$$\begin{aligned}\mathbf{p}_{b_{k+1}}^w &= \mathbf{p}_{b_k}^w + \mathbf{v}_{b_k}^w \Delta t_k + \iint_{t \in [t_k, t_{k+1}]} (\mathbf{R}_t^w (\hat{\mathbf{a}}_t - \mathbf{b}_{at} - \mathbf{n}_a) - \mathbf{g}^w) dt^2 \\ \mathbf{v}_{b_{k+1}}^w &= \mathbf{v}_{b_k}^w + \int_{t \in [t_k, t_{k+1}]} (\mathbf{R}_t^w (\hat{\mathbf{a}}_t - \mathbf{b}_{at} - \mathbf{n}_a) - \mathbf{g}^w) dt \\ \mathbf{q}_{b_{k+1}}^w &= \mathbf{q}_{b_k}^w \otimes \int_{t \in [t_k, t_{k+1}]} \frac{1}{2} \boldsymbol{\Omega} (\hat{\boldsymbol{\omega}}_t - \mathbf{b}_{wt} - \mathbf{n}_w) \mathbf{q}_t^{b_k} dt,\end{aligned}\quad (3.13)$$

其中, \mathbf{R}_t^w 表示从 IMU 坐标系到世界坐标系的变换, 与式 (2.43) 中的 \mathbf{R}_w^t 正好相反。

式 (3.13) 是当前帧的位置、速度和姿态的连续表达形式, 在实际工程中, 为了方

便编程，需要离散的形式。使用中值法对式(3.13)进行离散化，得，

$$\begin{aligned}\mathbf{p}_{i+1}^w &= \mathbf{p}_i^w + \mathbf{v}_i^w \delta t + \frac{1}{2} \bar{\hat{\mathbf{a}}}_l \delta t^2 \\ \mathbf{v}_{i+1}^w &= \mathbf{v}_i^w + \bar{\hat{\mathbf{a}}}_l \delta t^2 \\ \mathbf{q}_{i+1}^w &= \mathbf{q}_i^w \otimes \begin{bmatrix} 1 \\ \frac{1}{2} \bar{\hat{\omega}}_l \delta t \end{bmatrix},\end{aligned}\tag{3.14}$$

其中，

$$\begin{aligned}\bar{\hat{\mathbf{a}}}_l &= \frac{1}{2} [\mathbf{q}_i (\hat{\mathbf{a}}_i - \mathbf{b}_{a_i}) - \mathbf{g}^w + \mathbf{q}_{i+1} (\hat{\mathbf{a}}_{i+1} - \mathbf{b}_{a_i}) - \mathbf{g}^w] \\ \bar{\hat{\omega}}_l &= \frac{1}{2} (\hat{\omega}_i + \hat{\omega}_{i+1}) - \mathbf{b}_{\omega_i}\end{aligned}\tag{3.15}$$

i 是与 $[t_k, t_{k+1}]$ 内的 IMU 时刻， δt 是时间间隔。

3.2.2 两帧之间的位置、速度和姿态增量

观察式(3.13)，发现积分项中包含第 k 帧的 v 和 \mathbf{R}_t^w ，也就是第 k 帧的速度和相机相对世界坐标系的旋转，这样将导致三个问题：

- (1) 这样就相当于将相机位姿耦合到了积分中，如果需要对 $k+1$ 帧的相机的位姿求导，那么需要通过链式法则对 $k+1$ 帧之前的所有位姿进行求导，这样将极大的增加计算量；
- (2) 在进行非线性优化的时候，每一次优化都会改变状态量，这样就需要重新进行状态传播，同样会增加计算量；
- (3) \mathbf{R}_t^w 实际上是需要求解的未知量。

为了解决上述问题，考虑将优化的状态量从第 k 帧到第 $k+1$ 帧的积分项中提取出来，并将参考坐标系从世界坐标系变化为 IMU 坐标系。在公式(3.13)的等式两边同时乘以 $\mathbf{R}_w^{b_k}$ ，得，

$$\begin{aligned}\mathbf{R}_w^{b_k} \mathbf{p}_{b_{k+1}}^w &= \mathbf{R}_w^{b_k} (\mathbf{p}_{b_k}^w + \mathbf{v}_{b_k}^w \Delta t_k - \frac{1}{2} \mathbf{g}^w \Delta t_k^2) + \boldsymbol{\alpha}_{b_{k+1}}^{b_k} \\ \mathbf{R}_w^{b_k} \mathbf{v}_{b_{k+1}}^w &= \mathbf{R}_w^{b_k} (\mathbf{v}_{b_k}^w - \mathbf{g}^w \Delta t_k) + \boldsymbol{\beta}_{b_{k+1}}^{b_k} \\ \mathbf{q}_w^{b_k} \otimes \mathbf{q}_{b_{k+1}}^w &= \boldsymbol{\gamma}_{b_{k+1}}^{b_k},\end{aligned}\tag{3.16}$$

其中，

$$\begin{aligned}\boldsymbol{\alpha}_{b_{k+1}}^{b_k} &= \iint_{t \in [t_k, t_{k+1}]} \mathbf{R}_t^{b_k} (\hat{\mathbf{a}}_t - \mathbf{b}_{a_t} - \mathbf{n}_a) dt^2 \\ \boldsymbol{\beta}_{b_{k+1}}^{b_k} &= \int_{t \in [t_k, t_{k+1}]} \mathbf{R}_t^{b_k} (\hat{\mathbf{a}}_t - \mathbf{b}_{a_t} - \mathbf{n}_a) dt \\ \boldsymbol{\gamma}_{b_{k+1}}^{b_k} &= \iint_{t \in [t_k, t_{k+1}]} \frac{1}{2} \boldsymbol{\Omega} (\hat{\boldsymbol{\omega}}_t - \mathbf{b}_{w_t} - \mathbf{n}_w) \boldsymbol{\gamma}_t^{b_k} dt.\end{aligned}\quad (3.17)$$

与式 (3.13) 不同的是，式 (3.17) 积分出来的是两帧图像之间 IMU 的增量信息，而式 (3.13) 给出的是当前帧的位置、速度和姿态信息。称式 (3.17) 为预积分公式，可以看到预积分项只通过 IMU 测量值就能得到， b_k 变为参考帧。 $\boldsymbol{\alpha}_{b_{k+1}}^{b_k}, \boldsymbol{\beta}_{b_{k+1}}^{b_k}, \boldsymbol{\gamma}_{b_{k+1}}^{b_k}$ 只与 b_k 和 b_{k+1} 中的 IMU 偏置 bias 有关，与其他状态量无关。当 bias 估计变化很小时，假设 $\boldsymbol{\alpha}_{b_{k+1}}^{b_k}, \boldsymbol{\beta}_{b_{k+1}}^{b_k}, \boldsymbol{\gamma}_{b_{k+1}}^{b_k}$ 与 bias 是线性关系，可以按其对 bias 的一阶近似来调整，

$$\begin{aligned}\boldsymbol{\alpha}_{b_{k+1}}^{b_k} &\approx \hat{\boldsymbol{\alpha}}_{b_{k+1}}^{b_k} + \mathbf{J}_{b_a}^\alpha \delta \mathbf{b}_a + \mathbf{J}_{b_\omega}^\alpha \delta \mathbf{b}_\omega \\ \boldsymbol{\beta}_{b_{k+1}}^{b_k} &\approx \hat{\boldsymbol{\beta}}_{b_{k+1}}^{b_k} + \mathbf{J}_{b_a}^\beta \delta \mathbf{b}_a + \mathbf{J}_{b_\omega}^\beta \delta \mathbf{b}_\omega \\ \boldsymbol{\gamma}_{b_{k+1}}^{b_k} &\approx \hat{\boldsymbol{\gamma}}_{b_{k+1}}^{b_k} \otimes \left[\frac{1}{2} \mathbf{J}_{b_\omega}^\gamma \delta \mathbf{b}_\omega \right]\end{aligned}\quad (3.18)$$

其中得 Jacobian 矩阵计算及含义参考3.2.3小节中的递推公式 (3.38) 和 (3.40)。否则就进行重新传播。这种方法为基于非线性优化的算法节省了大量的计算资源。

在开始时， $\boldsymbol{\alpha}_{b_k}^{b_k}, \boldsymbol{\beta}_{b_k}^{b_k}$ 是 0， $\boldsymbol{\gamma}_{b_k}^{b_k}$ 是单位四元数。注意，噪声项 $\mathbf{n}_a, \mathbf{n}_w$ 是未知的，在实现中被视为零，所以这里使用预积分的估计值，记为 $(\hat{\cdot})$ 。仍然使用中值法对式 (3.18) 进行离散化，得，

$$\begin{aligned}\hat{\boldsymbol{\alpha}}_{i+1}^{b_k} &= \hat{\boldsymbol{\alpha}}_i^{b_k} + \hat{\boldsymbol{\beta}}_i^{b_k} \delta t + \frac{1}{2} \bar{\mathbf{a}}_l \delta t^2 \\ \hat{\boldsymbol{\beta}}_{i+1}^{b_k} &= \hat{\boldsymbol{\beta}}_i^{b_k} + \bar{\mathbf{a}}_l \delta t \\ \hat{\boldsymbol{\gamma}}_{i+1}^{b_k} &= \hat{\boldsymbol{\gamma}}_i^{b_k} \otimes \hat{\boldsymbol{\gamma}}_{i+1}^i = \hat{\boldsymbol{\gamma}}_i^{b_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2} \bar{\boldsymbol{\omega}}_l \delta t \end{bmatrix},\end{aligned}\quad (3.19)$$

其中，

$$\begin{aligned}\bar{\mathbf{a}}_l &= \frac{1}{2} [\mathbf{q}_i (\hat{\mathbf{a}}_i - \mathbf{b}_{a_i}) + \mathbf{q}_{i+1} (\hat{\mathbf{a}}_{i+1} - \mathbf{b}_{a_i})] \\ \bar{\boldsymbol{\omega}}_l &= \frac{1}{2} (\hat{\boldsymbol{\omega}}_i + \hat{\boldsymbol{\omega}}_{i+1}) - \mathbf{b}_{\omega_i}\end{aligned}\quad (3.20)$$

3.2.3 误差状态传播

IMU 在两个图像帧之间积分出来的值是有误差的，然后研究 IMU 误差传播过程。IMU 的误差状态量为 $[\delta\boldsymbol{\alpha}_t^{b_k}, \delta\boldsymbol{\beta}_t^{b_k}, \delta\boldsymbol{\theta}_t^{b_k}, \delta\mathbf{b}_{a_t}, \delta\mathbf{b}_{w_t}]^T$ ，易得：

$$\delta\dot{\boldsymbol{\alpha}}_t^{b_k} = \delta\boldsymbol{\beta}_t^{b_k} \quad (3.21)$$

由式(2.45)得，

$$\begin{aligned} \delta\dot{\mathbf{b}}_{a_t} &= \mathbf{n}_{b_a} \\ \delta\dot{\mathbf{b}}_{w_t} &= \mathbf{n}_{b_w} \end{aligned} \quad (3.22)$$

用 $(\cdot)_{true}$ 表示包含噪声的真实测量值， $(\cdot)_{nominal}$ 表示无噪声的理论值。为了方便书写和表达，下面将 $\delta\dot{\boldsymbol{\beta}}_t^{b_k}$ 和 $\delta\dot{\boldsymbol{\theta}}_t^{b_k}$ 分别简写成 $\delta\dot{\boldsymbol{\beta}}$ 和 $\delta\dot{\boldsymbol{\theta}}$ 则，

$$\delta\dot{\boldsymbol{\beta}} = \dot{\boldsymbol{\beta}}_{true} - \dot{\boldsymbol{\beta}}_{nominal} \quad (3.23)$$

其中，

$$\begin{aligned} \dot{\boldsymbol{\beta}}_{true} &= \hat{\mathbf{R}}_{t_{true}}^{b_k} (\hat{\mathbf{a}}_{t_{true}} - \mathbf{b}_{a_{t_{true}}}) \\ &= \mathbf{R}_t^{b_k} \text{Exp}([\delta\boldsymbol{\theta}]_\times) (\hat{\mathbf{a}}_t - \mathbf{n}_a - \mathbf{b}_{a_t} - \delta\mathbf{b}_{a_t}) \\ &= \mathbf{R}_t^{b_k} (\mathbf{I} + [\delta\boldsymbol{\theta}]_\times) (\hat{\mathbf{a}}_t - \mathbf{n}_a - \mathbf{b}_{a_t} - \delta\mathbf{b}_{a_t}) \\ &= \mathbf{R}_t^{b_k} (\hat{\mathbf{a}}_t - \mathbf{n}_a - \mathbf{b}_{a_t} - \delta\mathbf{b}_{a_t} + [\delta\boldsymbol{\theta}]_\times (\hat{\mathbf{a}}_t - \mathbf{b}_{a_t})) \\ &= \mathbf{R}_t^{b_k} (\hat{\mathbf{a}}_t - \mathbf{n}_a - \mathbf{b}_{a_t} - \delta\mathbf{b}_{a_t} - [(\hat{\mathbf{a}}_t - \mathbf{b}_{a_t})]_\times \delta\boldsymbol{\theta}) \end{aligned} \quad (3.24)$$

$$\dot{\boldsymbol{\beta}}_{nominal} = \mathbf{R}_t^{b_k} (\hat{\mathbf{a}}_t - \mathbf{b}_{a_t}) \quad (3.25)$$

将式(3.23)、(3.24)代入式(3.25)得，

$$\begin{aligned} \delta\dot{\boldsymbol{\beta}} &= \dot{\boldsymbol{\beta}}_{true} - \dot{\boldsymbol{\beta}}_{nominal} \\ &= -\mathbf{R}_t^{b_k} [(\hat{\mathbf{a}}_t - \mathbf{b}_{a_t})]_\times \delta\boldsymbol{\theta} - \mathbf{R}_t^{b_k} \delta\mathbf{b}_{a_t} - \mathbf{R}_t^{b_k} \mathbf{n}_a \end{aligned} \quad (3.26)$$

由式(2.42)可得,

$$\begin{aligned}\dot{\mathbf{q}}_{true} &= \frac{1}{2} \mathbf{q}_{t_{true}} \otimes \begin{bmatrix} \boldsymbol{\omega}_{true} \\ 0 \end{bmatrix} \\ &= \frac{1}{2} \mathbf{q}_t \otimes \delta \mathbf{q} \otimes \begin{bmatrix} \hat{\boldsymbol{\omega}}_t - \mathbf{b}_{\omega_t} - \mathbf{n}_\omega - \delta \mathbf{b}_{\omega_t} \\ 0 \end{bmatrix}\end{aligned}\quad (3.27)$$

由四元数导数性质可得,

$$\begin{aligned}\dot{\mathbf{q}}_{true} &= (\mathbf{q}_t \otimes \dot{\delta \mathbf{q}}) \\ &= \dot{\mathbf{q}}_t \otimes \delta \mathbf{q} + \mathbf{q}_t \otimes \dot{\delta \mathbf{q}} \\ &= \frac{1}{2} \mathbf{q}_t \otimes \begin{bmatrix} \hat{\boldsymbol{\omega}}_t - \mathbf{b}_{\omega_t} \\ 0 \end{bmatrix} \otimes \delta \mathbf{q} + \mathbf{q}_t \otimes \dot{\delta \mathbf{q}}\end{aligned}\quad (3.28)$$

联立式(3.27)、(3.28)可得,

$$\begin{aligned}\frac{1}{2} \mathbf{q}_t \otimes \delta \mathbf{q} \otimes \begin{bmatrix} \hat{\boldsymbol{\omega}}_t - \mathbf{b}_{\omega_t} - \mathbf{n}_\omega - \delta \mathbf{b}_{\omega_t} \\ 0 \end{bmatrix} &= \frac{1}{2} \mathbf{q}_t \otimes \begin{bmatrix} \hat{\boldsymbol{\omega}}_t - \mathbf{b}_{\omega_t} \\ 0 \end{bmatrix} \otimes \delta \mathbf{q} + \mathbf{q}_t \otimes \dot{\delta \mathbf{q}} \\ \Leftrightarrow \frac{1}{2} \delta \mathbf{q} \otimes \begin{bmatrix} \hat{\boldsymbol{\omega}}_t - \mathbf{b}_{\omega_t} - \mathbf{n}_\omega - \delta \mathbf{b}_{\omega_t} \\ 0 \end{bmatrix} &= \frac{1}{2} \begin{bmatrix} \hat{\boldsymbol{\omega}}_t - \mathbf{b}_{\omega_t} \\ 0 \end{bmatrix} \otimes \delta \mathbf{q} + \mathbf{q}_t \otimes \dot{\delta \mathbf{q}} \\ \Leftrightarrow 2\dot{\delta \mathbf{q}} &= \delta \mathbf{q} \otimes \begin{bmatrix} \hat{\boldsymbol{\omega}}_t - \mathbf{b}_{\omega_t} - \mathbf{n}_\omega - \delta \mathbf{b}_{\omega_t} \\ 0 \end{bmatrix} - \begin{bmatrix} \hat{\boldsymbol{\omega}}_t - \mathbf{b}_{\omega_t} \\ 0 \end{bmatrix} \otimes \delta \mathbf{q} \\ \Leftrightarrow 2\dot{\delta \mathbf{q}} &= \left[\begin{bmatrix} \hat{\boldsymbol{\omega}}_t - \mathbf{b}_{\omega_t} - \mathbf{n}_\omega - \delta \mathbf{b}_{\omega_t} \\ 0 \end{bmatrix} \right]_R \delta \mathbf{q} - \left[\begin{bmatrix} \hat{\boldsymbol{\omega}}_t - \mathbf{b}_{\omega_t} \\ 0 \end{bmatrix} \right]_L \delta \mathbf{q} \\ \Leftrightarrow 2\dot{\delta \mathbf{q}} &= \begin{bmatrix} -[2\hat{\boldsymbol{\omega}}_t - 2\mathbf{b}_{\omega_t} - \mathbf{n}_\omega - \delta \mathbf{b}_{\omega_t}]_\times & -\mathbf{n}_\omega - \delta \mathbf{b}_{\omega_t} \\ (\mathbf{n}_\omega + \delta \mathbf{b}_{\omega_t})^T & 0 \end{bmatrix} \begin{bmatrix} \frac{\delta \theta}{2} \\ 1 \end{bmatrix},\end{aligned}\quad (3.29)$$

又因为,

$$2\dot{\delta \mathbf{q}} = \begin{bmatrix} \dot{\delta \boldsymbol{\theta}} \\ 0 \end{bmatrix}\quad (3.30)$$

所以，

$$\begin{bmatrix} \dot{\delta\theta} \\ 0 \end{bmatrix} = \begin{bmatrix} -[2\hat{\omega}_t - 2\mathbf{b}_{\omega_t} - \mathbf{n}_\omega - \delta\mathbf{b}_{\omega_t}]_\times & -\mathbf{n}_\omega - \delta\mathbf{b}_{\omega_t} \\ (\mathbf{n}_\omega + \delta\mathbf{b}_{\omega_t})^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \frac{\delta\theta}{2} \\ 1 \end{bmatrix}, \quad (3.31)$$

从而得到 $\dot{\delta\theta}$ 的表达式，

$$\begin{aligned} \dot{\delta\theta} &= -[2\hat{\omega}_t - 2\mathbf{b}_{\omega_t} - \mathbf{n}_\omega - \delta\mathbf{b}_{\omega_t}]_\times \frac{\delta\theta}{2} - \mathbf{n}_\omega - \delta\mathbf{b}_{\omega_t} \\ &\approx -[\hat{\omega}_t - \mathbf{b}_{\omega_t}]_\times \delta\theta - \mathbf{n}_\omega - \delta\mathbf{b}_{\omega_t} \end{aligned} \quad (3.32)$$

由式 (3.21)、(3.22)、(3.26) 和 (3.32) 可得 IMU 预积分在 t 时刻误差项得导数为：

$$\begin{aligned} \begin{bmatrix} \dot{\delta\alpha}_t^{b_k} \\ \dot{\delta\beta}_t^{b_k} \\ \dot{\delta\theta}_t^{b_k} \\ \dot{\delta\mathbf{b}}_{a_t} \\ \dot{\delta\mathbf{b}}_{w_t} \end{bmatrix} &= \begin{bmatrix} 0 & \mathbf{I} & 0 & 0 & 0 \\ 0 & 0 & -\mathbf{R}_t^{b_k} [\hat{\mathbf{a}}_t - \mathbf{b}_{a_t}]_\times & -\mathbf{R}_t^{b_k} & 0 \\ 0 & 0 & -[\hat{\omega}_t - \mathbf{b}_{w_t}]_\times & 0 & -\mathbf{I} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta\alpha_t^{b_k} \\ \delta\beta_t^{b_k} \\ \delta\theta_t^{b_k} \\ \delta\mathbf{b}_{a_t} \\ \delta\mathbf{b}_{w_t} \end{bmatrix} \\ &+ \begin{bmatrix} 0 & 0 & 0 & 0 \\ -\mathbf{R}_t^{b_k} & 0 & 0 & 0 \\ 0 & -\mathbf{I} & 0 & 0 \\ 0 & 0 & \mathbf{I} & 0 \\ 0 & 0 & 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{n}_a \\ \mathbf{n}_w \\ \mathbf{n}_{b_a} \\ \mathbf{n}_{b_w} \end{bmatrix} = \mathbf{F}_t \delta\mathbf{z}_t^{b_k} + \mathbf{G}_t \mathbf{n}_t, \end{aligned} \quad (3.33)$$

式 (3.33) 是式 (3.17) 在连续时间下的 IMU 误差状态方程，其中， \mathbf{F}_t 是 15×15 维， \mathbf{G}_t 是 15×12 维， $\delta\mathbf{z}_t^{b_k}$ 是 15×1 维， \mathbf{n}_t 是 12×1 维。

将式 (3.33) 简写为：

$$\dot{\delta\mathbf{z}}_t^{b_k} = \mathbf{F}_t \delta\mathbf{z}_t^{b_k} + \mathbf{G}_t \mathbf{n}_t \quad (3.34)$$

由导数定义知，

$$\dot{\delta\mathbf{z}}_t^{b_k} = \lim_{\delta t \rightarrow 0} \frac{\delta\mathbf{z}_{t+\delta t}^{b_k} - \delta\mathbf{z}_t^{b_k}}{\delta t} \quad (3.35)$$

所以，

$$\delta\mathbf{z}_{t+\delta t}^{b_k} = \delta\mathbf{z}_t^{b_k} + \dot{\delta\mathbf{z}}_t^{b_k} \delta t \quad (3.36)$$

将式(3.34)代入式(3.36)得,

$$\delta \mathbf{z}_{t+\delta t}^{b_k} = (\mathbf{I} + \mathbf{F}_t \delta t) \delta \mathbf{z}_t^{b_k} + (\mathbf{G}_t \delta t) \mathbf{n}_t \quad (3.37)$$

式(3.37)表示当前时刻IMU的预积分误差与上一时刻的IMU的预积分误差成线性关系, 将式(3.37)与扩展卡尔曼滤波的运动方程对比发现, 其与EKF对非线性系统线性化的过程如出一辙。所以, 可以根据当前时刻IMU预积分的误差预测出下一时刻IMU预积分的误差的均值和协方差。均值预测可以通过式(3.37)计算, 而协方差预测公式可以类比EKF协方差更新公式得到,

$$\mathbf{P}_{t+\delta t}^{b_k} = (\mathbf{I} + \mathbf{F}_t \delta t) \mathbf{P}_t^{b_k} (\mathbf{I} + \mathbf{F}_t \delta t)^T + (\mathbf{G}_t \delta t) \mathbf{Q} (\mathbf{G}_t \delta t)^T \quad (3.38)$$

协方差初始值 $\mathbf{P}_{b_k}^{b_k} = 0$, 其中 \mathbf{Q} 是噪声的对角线协方差矩阵, 维数为 12×12 ,

$$\mathbf{Q} = \begin{bmatrix} \sigma_a^2 & 0 & 0 & 0 \\ 0 & \sigma_w^2 & 0 & 0 \\ 0 & 0 & \sigma_{ba}^2 & 0 \\ 0 & 0 & 0 & \sigma_{bw}^2 \end{bmatrix} \quad (3.39)$$

由公式(3.37)可以得到误差项的Jacobian递归计算公式,

$$\mathbf{J}_{t+\delta t} = (\mathbf{I} + \mathbf{F}_t \delta t) \mathbf{J}_t \quad (3.40)$$

其中, \mathbf{J} 的初始值为 $\mathbf{J}_{b_k} = \mathbf{I}$ 。

利用递推公式(3.38)和(3.40), 可以得到协方差矩阵 $\mathbf{P}_{b_{k+1}}^{b_k}$ 和雅可比矩阵 $\mathbf{J}_{b_{k+1}}$ 。
 $\alpha_{b_{k+1}}^{b_k}, \beta_{b_{k+1}}^{b_k}, \gamma_{b_{k+1}}^{b_k}$ 关于偏置的一阶近似可以写成公式(3.18)的形式。公式(3.18)中 $\mathbf{J}_{b_a}^\alpha$ 是 $\mathbf{J}_{b_{k+1}}$ 中的分块矩阵, 其位置对应于 $\delta \alpha_{b_{k+1}}^{b_k} / \delta b_{a_k}$, $\mathbf{J}_{b_w}^\alpha, \mathbf{J}_{b_a}^\beta, \mathbf{J}_{b_w}^\beta, \mathbf{J}_{b_w}^\gamma$ 。类似, 不再累述。当bias估计发生轻微变化时, 使用公式(3.18)近似校正预积分结果, 而不是重新传播。

下面写出IMU的预积分测量模型:

$$\begin{bmatrix} \hat{\alpha}_{b_{k+1}}^{b_k} \\ \hat{\beta}_{b_{k+1}}^{b_k} \\ \hat{\gamma}_{b_{k+1}}^{b_k} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_w^{b_k} (\mathbf{p}_{b_{k+1}}^w - \mathbf{p}_{b_k}^w + \frac{1}{2} \mathbf{g}^w \Delta t_k^2 - \mathbf{v}_{b_k}^w \Delta t_k) \\ \mathbf{R}_w^{b_k} (\mathbf{v}_{b_{k+1}}^w + \mathbf{g}^w \Delta t_k - \mathbf{v}_{b_k}^w) \\ \mathbf{q}_{b_k}^{w^{-1}} \otimes \mathbf{q}_{b_{k+1}}^w \\ \mathbf{b}_{ab_{k+1}} - \mathbf{b}_{ab_k} \\ \mathbf{b}_{wb_{k+1}} - \mathbf{b}_{wb_k} \end{bmatrix}. \quad (3.41)$$

考虑到实际编程需求，接下来对连续时间下的 IMU 误差状态方程 (3.33) 进行离散化。设 IMU 离散化的误差状态量为 $[\delta\alpha_k, \delta\theta_k, \delta\beta_k, \delta\mathbf{b}_{a_k}, \delta\mathbf{b}_{w_k}]^T$ 。

式 (3.32) 是角度增量误差导数的连续形式，对其进行离散化，得，

$$\delta\dot{\theta}_k = - \left[\frac{\hat{\omega}_k + \hat{\omega}_{k+1}}{2} - \mathbf{b}_{\omega_k} \right]_{\times} \delta\theta_k - \frac{\mathbf{n}_{\omega_k} + \mathbf{n}_{\omega_{k+1}}}{2} - \delta\mathbf{b}_{\omega_k} \quad (3.42)$$

根据导数的定义，得到下一时刻的角度增量误差为，

$$\delta\theta_{k+1} = \left(\mathbf{I} - \left[\frac{\hat{\omega}_k + \hat{\omega}_{k+1}}{2} - \mathbf{b}_{\omega_k} \right]_{\times} \delta t \right) \delta\theta_k - \frac{\mathbf{n}_{\omega_k} + \mathbf{n}_{\omega_{k+1}}}{2} \delta t - \delta t \delta\mathbf{b}_{\omega_k} \quad (3.43)$$

式 (3.26) 是速度增量误差导数的连续形式，对其进行离散化，得，

$$\begin{aligned} \delta\dot{\beta}_k &= -\frac{1}{2}\mathbf{R}_k [\hat{\mathbf{a}}_k - \mathbf{b}_{a_k}]_{\times} \delta\theta_k - \frac{1}{2}\mathbf{R}_{k+1} [\hat{\mathbf{a}}_{k+1} - \mathbf{b}_{a_k}]_{\times} \delta\theta_{k+1} \\ &\quad - \frac{1}{2}(\mathbf{R}_k + \mathbf{R}_{k+1}) \delta\mathbf{b}_{a_k} - \frac{1}{2}\mathbf{R}_k \mathbf{n}_{a_k} - \frac{1}{2}\mathbf{R}_{k+1} \mathbf{n}_{a_{k+1}} \end{aligned} \quad (3.44)$$

将式 (3.43) 代入式 (3.44) 得，

$$\begin{aligned} \delta\dot{\beta}_k &= -\frac{1}{2}\mathbf{R}_k [\hat{\mathbf{a}}_k - \mathbf{b}_{a_k}]_{\times} \delta\theta_k - \frac{1}{2}\mathbf{R}_{k+1} [\hat{\mathbf{a}}_{k+1} - \mathbf{b}_{a_k}]_{\times} \\ &\quad \left\{ \left[\mathbf{I} - \left[\frac{\hat{\omega}_k + \hat{\omega}_{k+1}}{2} - \mathbf{b}_{\omega_k} \right]_{\times} \delta t \right] \delta\theta_k - \frac{\mathbf{n}_{\omega_k} + \mathbf{n}_{\omega_{k+1}}}{2} \delta t - \delta t \delta\mathbf{b}_{\omega_k} \right\} \\ &\quad - \frac{1}{2}(\mathbf{R}_k + \mathbf{R}_{k+1}) \delta\mathbf{b}_{a_k} - \frac{1}{2}\mathbf{R}_k \mathbf{n}_{a_k} - \frac{1}{2}\mathbf{R}_{k+1} \mathbf{n}_{a_{k+1}} \\ &= \left\{ -\frac{1}{2}\mathbf{R}_k [\hat{\mathbf{a}}_k - \mathbf{b}_{a_k}]_{\times} - \frac{1}{2}\mathbf{R}_{k+1} [\hat{\mathbf{a}}_{k+1} - \mathbf{b}_{a_k}]_{\times} \right. \\ &\quad \left. \left[\mathbf{I} - \left[\frac{\hat{\omega}_k + \hat{\omega}_{k+1}}{2} - \mathbf{b}_{\omega_k} \right]_{\times} \delta t \right] \delta\theta_k \right\} \\ &\quad + \frac{\delta t}{4}\mathbf{R}_{k+1} [\hat{\mathbf{a}}_{k+1} - \mathbf{b}_{a_k}]_{\times} \mathbf{n}_{\omega_k} + \frac{\delta t}{4}\mathbf{R}_{k+1} [\hat{\mathbf{a}}_{k+1} - \mathbf{b}_{a_k}]_{\times} \mathbf{n}_{\omega_{k+1}} \\ &\quad + \frac{\delta t}{2}\mathbf{R}_{k+1} [\hat{\mathbf{a}}_{k+1} - \mathbf{b}_{a_k}]_{\times} \delta\mathbf{b}_{\omega_k} - \frac{1}{2}(\mathbf{R}_k + \mathbf{R}_{k+1}) \delta\mathbf{b}_{a_k} \\ &\quad - \frac{1}{2}\mathbf{R}_k \mathbf{n}_{a_k} - \frac{1}{2}\mathbf{R}_{k+1} \mathbf{n}_{a_{k+1}} \end{aligned} \quad (3.45)$$

根据导数的定义，得到下一时刻的速度增量误差为，

$$\begin{aligned}\delta\beta_{k+1} = & f_{21}\delta\theta_k + \delta\beta_k - \frac{1}{2}(\mathbf{R}_k + \mathbf{R}_{k+1})\delta t\delta\mathbf{b}_{a_k} + f_{24}\delta\mathbf{b}_{\omega_k} \\ & - \frac{1}{2}\mathbf{R}_k\delta t\mathbf{n}_{a_k} + v_{21}\mathbf{n}_{\omega_k} - \frac{1}{2}\mathbf{R}_{k+1}\delta t\mathbf{n}_{a_{k+1}} + v_{23}\mathbf{n}_{\omega_{k+1}}\end{aligned}\quad (3.46)$$

其中，

$$\begin{aligned}f_{21} = & -\frac{1}{2}\mathbf{R}_k[\hat{\mathbf{a}}_k - \mathbf{b}_{a_k}]_\times\delta t \\ & - \frac{1}{2}\mathbf{R}_{k+1}[\hat{\mathbf{a}}_{k+1} - \mathbf{b}_{a_k}]_\times\left[\mathbf{I} - \left[\frac{\hat{\omega}_k + \hat{\omega}_{k+1}}{2} - \mathbf{b}_{\omega_k}\right]_\times\delta t\right]\delta t \\ f_{24} = & \frac{1}{2}\mathbf{R}_{k+1}[\hat{\mathbf{a}}_{k+1} - \mathbf{b}_{a_k}]_\times\delta t^2 \\ v_{21} = v_{23} = & \frac{1}{4}\mathbf{R}_{k+1}[\hat{\mathbf{a}}_{k+1} - \mathbf{b}_{a_k}]_\times\delta t^2\end{aligned}\quad (3.47)$$

由式(3.21)知，位置增量误差导数的连续时间形式为，

$$\delta\dot{\alpha}_k = \delta\beta_k \quad (3.48)$$

对其离散化得，

$$\begin{aligned}\delta\dot{\alpha}_k = & \frac{1}{2}\delta\beta_k + \frac{1}{2}\delta\beta_{k+1} \\ = & \delta\beta_k + \frac{1}{2}f_{21}\delta\theta_k - \frac{1}{4}(\mathbf{R}_k + \mathbf{R}_{k+1})\delta t\delta\mathbf{b}_{a_k} + \frac{1}{2}f_{24}\delta\mathbf{b}_{\omega_k} \\ & - \frac{1}{4}\mathbf{R}_k\delta t\mathbf{n}_{a_k} + \frac{1}{2}v_{21}\mathbf{n}_{\omega_k} - \frac{1}{4}\mathbf{R}_{k+1}\delta t\mathbf{n}_{a_{k+1}} + \frac{1}{2}v_{23}\mathbf{n}_{\omega_{k+1}}\end{aligned}\quad (3.49)$$

根据导数的定义，得到下一时刻的位置增量误差为，

$$\begin{aligned}\delta\alpha_{k+1} = & \delta\alpha_k + \frac{\delta t}{2}f_{21}\delta\theta_k + \delta t\delta\beta_k - \frac{1}{4}(\mathbf{R}_k + \mathbf{R}_{k+1})\delta t^2\delta\mathbf{b}_{a_k} + \frac{\delta t}{2}f_{24}\delta\mathbf{b}_{\omega_k} \\ & - \frac{1}{4}\mathbf{R}_k\delta t^2\mathbf{n}_{a_k} + \frac{\delta t}{2}v_{21}\mathbf{n}_{\omega_k} - \frac{1}{4}\mathbf{R}_{k+1}\delta t^2\mathbf{n}_{a_{k+1}} + \frac{\delta t}{2}v_{23}\mathbf{n}_{\omega_{k+1}}\end{aligned}\quad (3.50)$$

由式(2.43)、(3.46)和(3.50)可得离散形式的IMU增量误差状态方程,

$$\begin{bmatrix} \delta\boldsymbol{\alpha}_{k+1} \\ \delta\boldsymbol{\theta}_{k+1} \\ \delta\boldsymbol{\beta}_{k+1} \\ \delta\mathbf{b}_{a_{k+1}} \\ \delta\mathbf{b}_{w_{k+1}} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & f_{01} & \delta t & f_{03} & f_{04} \\ 0 & f_{11} & 0 & 0 & -\delta t \\ 0 & f_{21} & I & f_{23} & f_{24} \\ 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & I \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{\alpha}_k \\ \delta\boldsymbol{\theta}_k \\ \delta\boldsymbol{\beta}_k \\ \delta\mathbf{b}_{a_k} \\ \delta\mathbf{b}_{w_k} \end{bmatrix} + \begin{bmatrix} v_{00} & v_{01} & v_{02} & v_{03} & 0 & 0 \\ 0 & \frac{-\delta t}{2} & 0 & \frac{-\delta t}{2} & 0 & 0 \\ -\frac{\mathbf{R}_k \delta t}{2} & v_{21} & -\frac{\mathbf{R}_{k+1} \delta t}{2} & v_{23} & 0 & 0 \\ 0 & 0 & 0 & 0 & \delta t & 0 \\ 0 & 0 & 0 & 0 & 0 & \delta t \end{bmatrix} \begin{bmatrix} \mathbf{n}_{a_k} \\ \mathbf{n}_{w_k} \\ \mathbf{n}_{a_{k+1}} \\ \mathbf{n}_{w_{k+1}} \\ \mathbf{n}_{b_a} \\ \mathbf{n}_{b_w} \end{bmatrix} \quad (3.51)$$

其中,

$$\left\{ \begin{array}{l} f_{01} = \frac{\delta t}{2} f_{21} \\ = -\frac{1}{4} \mathbf{R}_k [\hat{\mathbf{a}}_k - \mathbf{b}_{a_k}]_\times \delta t^2 - \frac{1}{4} \mathbf{R}_{k+1} [\hat{\mathbf{a}}_{k+1} - \mathbf{b}_{a_k}]_\times \\ \left[\mathbf{I} - \left[\frac{\hat{\boldsymbol{\omega}}_k + \hat{\boldsymbol{\omega}}_{k+1}}{2} - \mathbf{b}_{\omega_k} \right]_\times \right] \delta t^2 \\ f_{03} = -\frac{1}{4} (\mathbf{R}_k + \mathbf{R}_{k+1}) \delta t^2 \\ f_{04} = \frac{\delta t}{2} f_{24} = \frac{1}{4} \mathbf{R}_{k+1} [\hat{\mathbf{a}}_{k+1} - \mathbf{b}_{a_k}]_\times \delta t^3 \\ f_{11} = I - \left[\frac{\hat{\boldsymbol{\omega}}_k + \hat{\boldsymbol{\omega}}_{k+1}}{2} - \mathbf{b}_{\omega_k} \right]_\times \delta t \\ f_{21} = -\frac{1}{2} \mathbf{R}_k [\hat{\mathbf{a}}_k - \mathbf{b}_{a_k}]_\times \delta t - \frac{1}{2} \mathbf{R}_{k+1} [\hat{\mathbf{a}}_{k+1} - \mathbf{b}_{a_k}]_\times \\ \left[\mathbf{I} - \left[\frac{\hat{\boldsymbol{\omega}}_k + \hat{\boldsymbol{\omega}}_{k+1}}{2} - \mathbf{b}_{\omega_k} \right]_\times \right] \delta t \\ f_{23} = -\frac{1}{2} (\mathbf{R}_k + \mathbf{R}_{k+1}) \delta t \\ f_{24} = \frac{1}{2} \mathbf{R}_{k+1} [\hat{\mathbf{a}}_{k+1} - \mathbf{b}_{a_k}]_\times \delta t^2 \end{array} \right. \quad (3.52)$$

$$\begin{cases} v_{00} = -\frac{1}{4}\mathbf{R}_k \delta t^2 \\ v_{01} = v_{03} = \frac{\delta t}{2} v_{21} = \frac{1}{4} \mathbf{R}_{k+1} [\hat{\mathbf{a}}_{k+1} - \mathbf{b}_{a_k}]_\times \delta t^2 \frac{\delta t}{2} \\ v_{02} = -\frac{1}{4} \mathbf{R}_{k+1} \delta t^2 \\ v_{21} = v_{23} = \frac{1}{4} \mathbf{R}_{k+1} [\hat{\mathbf{a}}_{k+1} - \mathbf{b}_{a_k}]_\times \delta t^2 \end{cases} \quad (3.53)$$

将式 (3.51) 简写为:

$$\delta \mathbf{z}_{k+1} = \mathbf{F} \delta \mathbf{Z}_k + \mathbf{V} \mathbf{Q} \quad (3.54)$$

其中, $\delta \mathbf{z}_{k+1}$ 为 15×1 维, \mathbf{F} 为 15×15 维, $\delta \mathbf{Z}_k$ 为 15×1 维, \mathbf{V} 为 15×18 维, \mathbf{Q} 为 18×1 维。雅可比迭代公式为:

$$\mathbf{J}_{k+1} = \mathbf{F} \mathbf{J}_k \quad (3.55)$$

其中 \mathbf{J} 的初始值为 $\mathbf{J}_k = \mathbf{I}$ 。有了 \mathbf{J}_{k+1} 就可以对 bias 进行线性化。

协方差的迭代公式为:

$$\mathbf{P}_{k+1} = \mathbf{F} \mathbf{P}_k \mathbf{F}^T + \mathbf{V} \mathbf{Q} \mathbf{V}^T \quad (3.56)$$

其中, 协方差矩阵的初始值为 $\mathbf{P}_k = \mathbf{0}$ 。 \mathbf{Q} 为噪声项协方差矩阵, 是一个对角矩阵:

$$\mathbf{Q} = \begin{bmatrix} \sigma_a^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_w^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_a^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_w^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{ba}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{bw}^2 \end{bmatrix} \quad (3.57)$$

3.3 系统初始化

单目视觉/惯性紧耦合融合系统是一个高度非线性系统, 在后端采用非线性优化的方法来计算各个状态量, 因此, 需要在优化前对各个状态量进行初始化。本文采用视觉和 IMU 松耦合的方式将视觉信息和 IMU 信息进行对齐, 从而初始化一些必要参数, 包括: 单目视觉不能算出的绝对尺度信息、IMU 的 bias、重力矢量和速度。系统初始化分两步: Visual 初始化和 Visual-IMU 对齐。

3.3.1 多视图几何基础

在纯视觉初始化的时候需要用到一些多视图几何的基础知识，因此首先研究初始化用到的多视图几何知识，主要包括：对极几何、三角化、PnP 和 BA（Bundle Adjustment）。

1. 对极几何求位姿

对于单目相机来说，仅有 2D 的像素坐标信息，需要通过两幅单目图像来估计相机的运动。对极几何用来解决此问题。如图3.8所示，假设有两幅图像，其对应的成像平面为 I_1 和 I_2 ，已经有了 I_1 和 I_2 匹配好的所有关键点。假设 I_1 到 I_2 的旋转和平移分别为 \mathbf{R}, \mathbf{t} ， O_1, O_2 是相机的光心， p_1 和 p_2 是一对匹配好的关键点，其对应的 3D 空间点是点 P 。 $\overrightarrow{O_1 p_1}$ 和 $\overrightarrow{O_2 p_2}$ 相交于点 P 。此时 O_1, O_2, P 三个点构成一个平面，称为极平面。光心 O_1, O_2 的连线与成像平面 I_1, I_2 相交于点 e_1, e_2 ， e_1, e_2 称为极点，线段 $O_1 O_2$ 称为基线。极平面与成像平面相交于线段 l_1, l_2 ， l_1, l_2 称为极线。

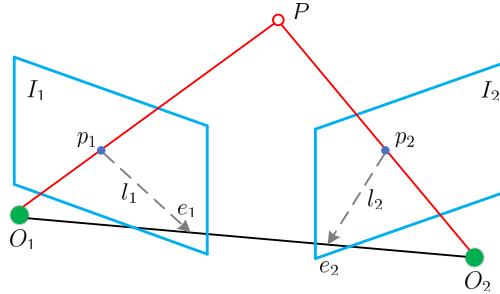


图 3.8 对极几何约束示意图

假设 P 的 3D 空间坐标为 $\mathbf{P} = [X, Y, Z]^T$ ，根据 2.1.1 小节研究的针孔相机模型，可以得到 P 与 p_1, p_2 的关系为：

$$\begin{aligned} s_1 \mathbf{p}_1 &= \mathbf{K} \mathbf{P}, \\ s_2 \mathbf{p}_2 &= \mathbf{K}(\mathbf{R} \mathbf{P} + \mathbf{t}) \end{aligned} \quad (3.58)$$

其中， \mathbf{K} 为相机内参， s_1, s_2 为尺度因子。如果使用齐次坐标，式 (3.58) 可以写为：

$$\begin{aligned} \mathbf{p}_1 &= \mathbf{K} \mathbf{P}, \\ \mathbf{p}_2 &= \mathbf{K}(\mathbf{R} \mathbf{P} + \mathbf{t}) \end{aligned} \quad (3.59)$$

设 p_1, p_2 的归一化平面坐标为 $\mathbf{x}_1, \mathbf{x}_2$ ，易知：

$$\begin{aligned} \mathbf{x}_1 &= \mathbf{K}^{-1} \mathbf{p}_1 \\ \mathbf{x}_2 &= \mathbf{K}^{-1} \mathbf{p}_2 \end{aligned} \quad (3.60)$$

联立式 (3.59) 和 (3.60) 可得:

$$\mathbf{x}_2 = \mathbf{R}\mathbf{x}_1 + \mathbf{t} \quad (3.61)$$

等式 (3.61) 两边同时左乘 $[\mathbf{t}]_{\times}$ 可得:

$$[\mathbf{t}]_{\times} \mathbf{x}_2 = [\mathbf{t}]_{\times} \mathbf{R}\mathbf{x}_1 \quad (3.62)$$

等式 (3.62) 两边同时左乘 \mathbf{x}_2^T 可得:

$$\mathbf{x}_2^T [\mathbf{t}]_{\times} \mathbf{x}_2 = \mathbf{x}_2^T [\mathbf{t}]_{\times} \mathbf{R}\mathbf{x}_1 \quad (3.63)$$

由于向量 $[\mathbf{t}]_{\times} \mathbf{x}_2$ 和 \mathbf{x}_2 垂直, 所以等式 (3.63) 左侧等于 0, 所以:

$$\mathbf{x}_2^T [\mathbf{t}]_{\times} \mathbf{R}\mathbf{x}_1 = 0 \quad (3.64)$$

联立公式 (3.60) 和 (3.64) 可得:

$$\mathbf{p}_2^T \mathbf{K}^{-T} [\mathbf{t}]_{\times} \mathbf{R} \mathbf{K}^{-1} \mathbf{p}_1 = 0 \quad (3.65)$$

公式 (3.64) 和 (3.65) 称为对极约束, 其几何意义是 O_1, P, O_2 三点共面。记

$$\begin{aligned} \mathbf{E} &= [\mathbf{t}]_{\times} \mathbf{R} \\ \mathbf{F} &= \mathbf{K}^{-T} [\mathbf{t}]_{\times} \mathbf{R} \mathbf{K}^{-1} \\ &= \mathbf{K}^{-T} \mathbf{E} \mathbf{K}^{-1} \end{aligned} \quad (3.66)$$

称 \mathbf{F} 为基本矩阵, \mathbf{E} 为本质矩阵。可见 \mathbf{F} 和 \mathbf{E} 之间只相差了一个相机内参。这样以来, 就可以根据匹配的关键点, 通过对极约束求出 \mathbf{F} 或者 \mathbf{E} , 进而求出 \mathbf{R}, \mathbf{t} 。

矩阵 $[\mathbf{t}]_{\times} \mathbf{R}$ 有六个自由度, 由对极约束 (3.64) 可知 \mathbf{E} 具有尺度不确定性, 所以 \mathbf{E} 只有五个自由度。这表明, 最少使用五对匹配的关键点就可以求出 \mathbf{E} ^[55]。由于本质矩阵 \mathbf{E} 的奇异值形式是 $[\sigma, \sigma, 0]^T$ ^[56], 这种非线性性质使得求解线性方程变得困难。因此, 通常使用八对匹配的关键点来求解 \mathbf{E} , 这种方法称为八点法 (Eight-point-algorithm)^{[56] [57] [58]}。

假设有一对匹配好的关键点, 其归一化的齐次坐标为: $\mathbf{x}_1 = [u_1, v_1, 1]^T$, $\mathbf{x}_2 = [u_2, v_2, 1]^T$ 。由对极约束 (3.64) 可得,

$$(u_1, v_1, 1) \begin{pmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & e_9 \end{pmatrix} \begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} = 0 \quad (3.67)$$

中间的矩阵是本质矩阵 \mathbf{E} 。改变 \mathbf{E} 的形式，令 $\mathbf{e} = [e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9]^T$ ，则式(3.67)可变形为：

$$[u_1u_2, u_1v_2, u_1, v_1u_2, v_1v_2, v_1, u_2, v_2, 1] \cdot \mathbf{e} = 0 \quad (3.68)$$

对于其他七个点可以得到类似的方程，将这八个方程放在一起写成矩阵的形式，可得：

$$\begin{pmatrix} u_1^1u_2^1 & u_1^1v_2^1 & u_1^1 & v_1^1u_2^1 & v_1^1v_2^1 & v_1^1 & u_2^1 & v_2^1 & 1 \\ u_1^2u_2^2 & u_1^2v_2^2 & u_1^2 & v_1^2u_2^2 & v_1^2v_2^2 & v_1^2 & u_2^2 & v_2^2 & 1 \\ \vdots & \vdots \\ u_1^8u_2^8 & u_1^8v_2^8 & u_1^8 & v_1^8u_2^8 & v_1^8v_2^8 & v_1^8 & u_2^8 & v_2^8 & 1 \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_8 \\ e_8 \\ e_9 \end{pmatrix} = 0 \quad (3.69)$$

通过求解这个方程组就可以得到本质矩阵 \mathbf{E} ，然后通过奇异值分解(SVD)来计算 \mathbf{R}, \mathbf{t} 。假设 \mathbf{E} 的奇异值分解为：

$$\mathbf{E} = \mathbf{U}\Sigma\mathbf{V}^T \quad (3.70)$$

其中， \mathbf{U}, \mathbf{V} 是正交矩阵， Σ 是奇异值矩阵。由 \mathbf{E} 的非线性性质可得 $\Sigma = \text{diag}(\sigma, \sigma, 0)$ 。在 SVD 分解过程中可以发现，一个本质矩阵对应两组可能的 \mathbf{R}, \mathbf{t} ：

$$\begin{aligned} [\mathbf{t}_1]_\times &= \mathbf{U}\mathbf{R}_Z\left(\frac{\pi}{2}\right)\Sigma\mathbf{U}^T, & \mathbf{R}_1 &= \mathbf{U}\mathbf{R}_Z^T\left(\frac{\pi}{2}\right)\mathbf{V}^T \\ [\mathbf{t}_2]_\times &= \mathbf{U}\mathbf{R}_Z\left(-\frac{\pi}{2}\right)\Sigma\mathbf{U}^T, & \mathbf{R}_2 &= \mathbf{U}\mathbf{R}_Z^T\left(-\frac{\pi}{2}\right)\mathbf{V}^T \end{aligned} \quad (3.71)$$

其中， $\mathbf{R}_Z\left(\frac{\pi}{2}\right)$ 表示绕轴旋转 90° 的旋转矩阵。 \mathbf{E} 的尺度不确定性导致 \mathbf{E} 和 $-\mathbf{E}$ 是等价的，故对于 $-t$ 也会得到式(3.71)的结果。所以总共有四种可能的情况，如图3.9所示。根据实际情况，分解出来的 \mathbf{R}, \mathbf{t} 应该满足使得路标点的深度为正值，也就是说关键点对应的 3D 空间点 P 应该和相机成像平面在同一侧，可见在图3.9的 4 种情形中，只有第一种符合要求，从而确定该解为正确的解。

2. 三角化求路标点深度

在单目视觉 SLAM 中，无法直接获取路标点的深度信息，需要利用相机位姿通

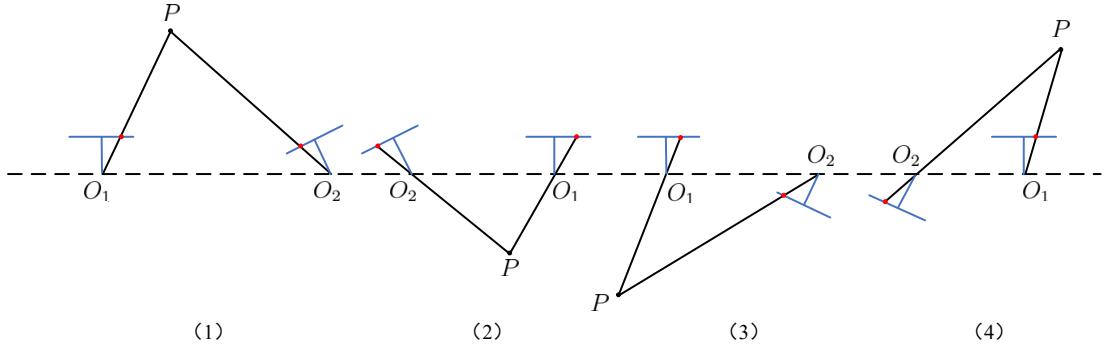


图 3.9 分解本质矩阵得到的四种解

过三角测量 (Triangulation, 也称三角化) 的方法来计算路标点深度。三角测量是指，在不同的方向观察目标点，通过两次观察的方向与目标点的夹角来确定目标的距离。三角测量由高斯最先提出，在地理学和天文学中都有广泛的应用。与图3.8类似，图3.10展示了三角化计算路标点深度的示意图。

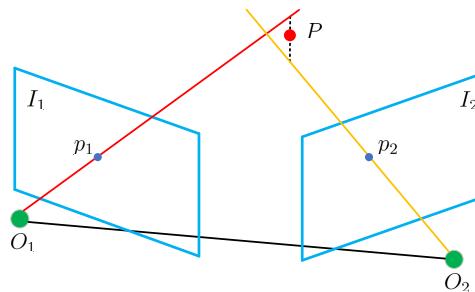


图 3.10 三角化计算路标点深度

以左边图像 I_1 为参考帧，假设右边图像 I_2 的旋转和平移为 \mathbf{R}, \mathbf{t} 。理想情况下，射线 O_1p_1 和射线 O_2p_2 会相交于点 P ，点 P 是 p_1, p_2 对应的三维空间坐标点。但是，实际情况下，由于传感器有噪声，射线 O_1p_1 和射线 O_2p_2 通常不会严格的相交于点 P ，甚至在三维空间中都不能相交。这种情况下通常利用最小二乘法来求解。

设 $\mathbf{x}_1, \mathbf{x}_2$ 是 p_1, p_2 的归一化坐标，由对极约束可得：

$$s_1 \mathbf{x}_1 = s_2 \mathbf{R} \mathbf{x}_2 + \mathbf{t} \quad (3.72)$$

其中 s_1, s_2 分别为 p_1, p_2 的深度。将等式 (3.72) 两边同时乘以 $[\mathbf{x}_1]_\times$ 得，

$$s_1 [\mathbf{x}_1]_\times \mathbf{x}_1 = 0 = s_2 [\mathbf{x}_1]_\times \mathbf{R} \mathbf{x}_2 + [\mathbf{x}_1]_\times \mathbf{t} \quad (3.73)$$

因为 \mathbf{R}, \mathbf{t} 已知，故可以直接算出深度 s_2 。同理可得 s_1 。当然，根据刚才的分析，噪声的存在会使等式 (3.73) 不严格的等于 0。所以需使用最小二乘法求解。

值得注意的是，由于本质矩阵 \mathbf{E} 的尺度不确定性，计算出的路标点深度是没有尺度的，只有大小，也就是说此时得到的深度没有单位度量。在后面的联合初始化中，

通过将视觉和 IMU 对齐来恢复出单目视觉的尺度信息。

三角测量是通过在不同的方向上观察路标点计算而得到的深度，因此相机必须要有平移，在纯旋转的情况下三角测量失效。在有平移的情况下，三角测量还存在一个矛盾的问题。这个矛盾是由像素的不确定性导致的。

如图3.11所示，当相机的平移 t 很小时，像素由于相机测量噪声产生的微小不确定性 $\delta\theta$ ，将导致计算的深度值有较大的不确定性 δd 。对比图3.11的左右两图可以发现，当像素不确定性一样的情况下，较大的平移 t 会产生更小的深度不确定性 δd 。所以，在相机分辨率一样的情况下，较大的平移会使得三角测量更准确。但是，并不是平移量越大越好，因为太大的平移量会图像特征跟踪（匹配）失败。因此就会产生一个三角化矛盾：增大平移会导致特征跟踪（匹配）失败，而减小平移又会降低三角测量的精度。

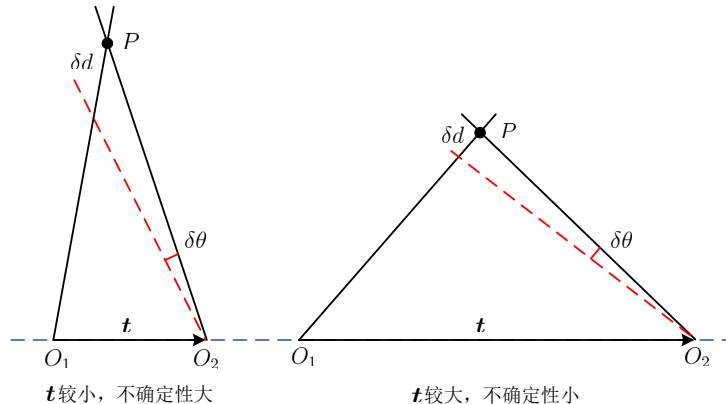


图 3.11 三角测量的矛盾

3. PnP 求位姿

通过对极约束和三角测量已经可以计算出相机的姿态以及路标点的 3D 空间位置（没有尺度信息），那么能否根据已经计算出来的 3D 路标点，以及该路标点在其他图像帧中的 2D 投影坐标，来计算其他图像帧对应的相机姿态呢？当然可以，PnP 正是求解此问题的方法，也就是说，当知道 n 个 3D 空间点以及他们在图像中的投影位置时，可以用 PnP 来求解相机的位姿。

求解 PnP 有很多种方法，比如使用 3 对点的 P3P^[59]，直接线性变换（Direct Linear Transform, DLT），EPnP（Efficient PnP）^[60]，UPnP^[61] 等等。下面研究 DLT 方法。

假设 3D 空间中的一点 P ，用齐次坐标表示为 $\mathbf{P} = (X, Y, Z, 1)^T$ ，其在某一幅图像中的投影点为 \mathbf{x}_1 ，用齐次坐标表示其在归一化平面上的坐标 $\mathbf{x}_1 = (u_1, v_1, 1)^T$ 。该图像对应的相机位姿为 \mathbf{R}, \mathbf{t} ，将 \mathbf{R}, \mathbf{t} 放在一起构成一个增广矩阵 $\mathbf{T} = [\mathbf{R}|\mathbf{t}]$ ， \mathbf{T} 为

3×4 矩阵。根据 3D 空间点和归一化平面坐标点之间的映射关系可得，

$$s \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} = \begin{pmatrix} t_1 & t_2 & t_3 & t_4 \\ t_5 & t_6 & t_7 & t_8 \\ t_9 & t_{10} & t_{11} & t_{12} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (3.74)$$

其中，中间得矩阵为 \mathbf{T} 的展开式。可以利用最后一行等式将式 (3.74) 中的尺度 s 消去，得到两个约束等式：

$$\begin{aligned} u_1 &= \frac{t_1X + t_2Y + t_3Z + t_4}{t_9X + t_{10}Y + t_{11}Z + t_{12}} \\ v_1 &= \frac{t_5X + t_6Y + t_7Z + t_8}{t_9X + t_{10}Y + t_{11}Z + t_{12}} \end{aligned} \quad (3.75)$$

令 \mathbf{T} 的行向量为：

$$\begin{aligned} \mathbf{t}_1 &= (t_1, t_2, t_3, t_4)^T \\ \mathbf{t}_2 &= (t_5, t_6, t_7, t_8)^T \\ \mathbf{t}_3 &= (t_9, t_{10}, t_{11}, t_{12})^T \end{aligned} \quad (3.76)$$

则式 (3.75) 可以写为：

$$\begin{aligned} \mathbf{t}_1^T \mathbf{P} - \mathbf{t}_3^T \mathbf{P} u_1 &= 0 \\ \mathbf{t}_2^T \mathbf{P} - \mathbf{t}_3^T \mathbf{P} v_1 &= 0 \end{aligned} \quad (3.77)$$

可以观察到，一对 3D 路标点和 2D 关键点可以提供两个关于 t 的线性约束项等式。那么 N 对 3D 路标点和 2D 关键点可以提供 $2N$ 个约束项，写成矩阵形式为：

$$\begin{pmatrix} \mathbf{P}_1^T & 0 & -u_1 \mathbf{P}_1^T \\ 0 & \mathbf{P}_1^T & -v_1 \mathbf{P}_1^T \\ \vdots & \vdots & \vdots \\ \mathbf{P}_N^T & 0 & -u_N \mathbf{P}_N^T \\ 0 & \mathbf{P}_N^T & -v_N \mathbf{P}_N^T \end{pmatrix} \begin{pmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \\ \mathbf{t}_3 \end{pmatrix} = 0 \quad (3.78)$$

t 的维数为 12，因此最少可以通过 6 对 3D 路标点和 2D 关键点就可以求解出矩阵 \mathbf{T} 。当匹配或跟踪的点对大于 6 对时，可以使用 SVD 方法求解。

4. Bundle Adjustment

BA (Bundle Adjustment) 中文可译为“捆集调整”^{[62][63]}。通过最小化重投影误差 (Reprojection Error) 来迭代优化求解问题。对于单目相机来说，使用 PnP 需要先求相机位姿，在求 3D 路标点位置，而 BA 将相机位姿和路标点位置看作优化变量，放在一起进行迭代优化。通常情况下，用 BA 来优化 PnP 估计的结果。

假设有 n 个 3D 空间点 P ，在某一幅图像中对应的投影坐标点为 p ，第 i 个 3D 空间点的坐标为 $\mathbf{P}_i = [X_i, Y_i, Z_i]^T$ ，对应的投影像素坐标为 $\mathbf{u}_i = [u_i, v_i]^T$ 。该图像对应的相机位姿为 \mathbf{R}, \mathbf{t} ，其李代数为 $\exp([\boldsymbol{\xi}]_\times)$ 。根据针孔相机模型可得：

$$s_i \mathbf{u}_i = \mathbf{K} \exp([\boldsymbol{\xi}]_\times) \mathbf{P}_i$$

$$s_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \mathbf{K} \exp([\boldsymbol{\xi}]_\times) \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} \quad (3.79)$$

其中， s_i 是尺度， \mathbf{K} 是相机内参。然而，实际情况中相机是有噪声的，噪声使得等式 (3.79) 不相等，有一个误差。将每个观测点噪声带来的误差相加构成一个代价函数，建模为最小二乘问题，通过最小化代价函数来寻找最优的 \mathbf{P}_i 和 $[\boldsymbol{\xi}]_\times$ ：

$$e = \arg \min_{\boldsymbol{\xi}} \frac{1}{2} \sum_{i=1}^n \left\| \mathbf{u}_i - \frac{1}{s_i} \mathbf{K} \exp([\boldsymbol{\xi}]_\times) \mathbf{P}_i \right\|_2^2 \quad (3.80)$$

对于最小二乘问题，通常使用高斯-牛顿法或者 LM 方法求解。在使用 G-N 或者 L-M 方法的时候，需要知道代价函数关于误差项的导数，也就是雅可比矩阵，这里不再详细推导，直接给出代价函数 e 关于相机位姿 $\boldsymbol{\xi}$ 和 3D 空间点 P 的雅可比矩阵：

$$\frac{\partial e}{\partial \delta \boldsymbol{\xi}} = - \begin{bmatrix} \frac{f_x}{Z'} & 0 & -\frac{f_x X'}{Z'^2} & -\frac{f_x X' Y'}{Z'^2} & f_x + \frac{f_x X^2}{Z'^2} & -\frac{f_x Y'}{Z'} \\ 0 & \frac{f_y}{Z'} & -\frac{f_y Y'}{Z'^2} & -f_y - \frac{f_y Y'^2}{Z'^2} & \frac{f_y X' Y'}{Z'^2} & \frac{f_y X'}{Z'} \end{bmatrix} \quad (3.81)$$

$$\frac{\partial e}{\partial \mathbf{P}} = - \begin{bmatrix} \frac{f_x}{Z'} & 0 & -\frac{f_x X'}{Z'^2} \\ 0 & \frac{f_y}{Z'} & -\frac{f_y Y'}{Z'^2} \end{bmatrix} \mathbf{R} \quad (3.82)$$

3.3.2 纯视觉初始化

为了减少计算量，初始化在一个滑动窗口内进行。首先检查窗口内当前的图像帧与其他帧的关键点对应关系，找到与当前帧匹配关键点数量最多的关键帧作为参考帧。然后使用对极几何求解基础矩阵，从而计算出当前帧与参考帧帧之间的变换矩阵。然后用 SFM (Structure-From-Motion)^[64] 计算处滑窗内的所有关键帧的位姿以及路标点的 3D 空间坐标，SFM 具体步骤包括：

- (1) 通过三角化得到当前帧和参考帧的路标点；
- (2) 对于当前帧和参考帧之间的所有帧用 PnP 算法求解其位姿；
- (3) 通过三角化得到当前帧和参考帧之间的所有帧的路标点；
- (4) 对于当前帧和第一帧之间的所有帧用 PnP 算法求解其位姿；
- (5) 通过三角化得到当前帧和第一帧之间的所有帧的路标点；
- (6) 对于其他没有被三角化的关键点，对其三角化得到剩余的路标点；
- (7) 通过 BA 对滑动窗口内的所有关键帧位姿进行优化。

在完成了 SFM 后，已经可以估计出滑动窗口内所有关键帧的位姿，然而，视觉初始化可能不是一次就能成功，这个时候，关键帧的总数可能会超过窗口大小，还需要再次使用 PnP 算法估计出那些不被包括在窗口内的关键帧的位姿。

至此，纯视觉初始化就完成了。值得注意的是，在纯视觉初始化的时候，将第一帧图像 $(\cdot)^{c_0}$ 作为参考帧，因此需要将位姿从第一帧图像坐标系转换到 IMU 坐标系，方法如下：

$$\begin{aligned} \mathbf{q}_{b_k}^{c_0} &= \mathbf{q}_{c_k}^{c_0} \otimes (\mathbf{q}_c^b)^{-1} \\ s\bar{\mathbf{p}}_{b_k}^{c_0} &= s\bar{\mathbf{p}}_{c_k}^{c_0} - \mathbf{R}_{b_k}^{c_0} \mathbf{p}_c^b, \end{aligned} \quad (3.83)$$

其中， $\mathbf{p}_c^b, \mathbf{q}_c^b$ 是外参 (Visual 到 IMU)。

3.3.3 视觉/惯性联合初始化

如图3.12所示，视觉/惯性联合初始化是将 IMU 预积分的值和出视觉初始化的值进行对齐，从而计算出绝对尺度 s 、陀螺仪 bias、重力矢量 \mathbf{g} 和速度 \mathbf{v} 。注意，不计算加速度计的 bias，因为重力加速度的值要远远大于加速度计的 bias，这导致加速度计的 bias 很难被观测到^[65]，因此忽略加速度计 bias 的影响，只计算陀螺仪的 bias。

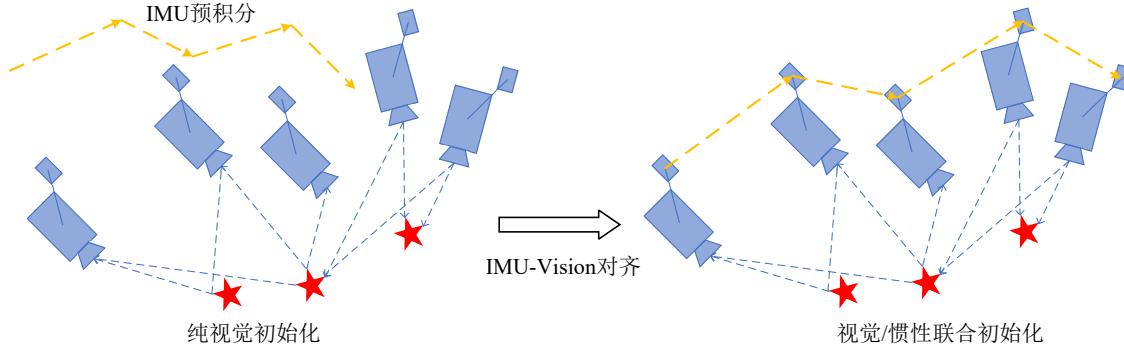


图 3.12 视觉/惯性联合初始化示意图

(1) 矫正陀螺仪 bias

考虑滑动窗口内连续两帧图像 b_k 和 b_{k+1} ，从纯视觉初始化中得到的两帧之间的旋转 $\mathbf{q}_{b_{k+1}}^{c_0} \mathbf{q}_{b_k}^{c_0}$ 应该和 IMU 预积分得到旋转 $\gamma_{b_{k+1}}^{b_k}$ 相等。通过式(3.18)线性化 IMU 预积分项，并定义代价函数：

$$\min_{\delta \mathbf{b}_w} \sum_{k \in \mathcal{B}} \left\| \mathbf{q}_{b_{k+1}}^{c_0} \mathbf{q}_{b_k}^{c_0} \otimes \gamma_{b_{k+1}}^{b_k} \right\|^2 \quad (3.84)$$

其中， \mathcal{B} 代表窗口中的所有帧，

$$\gamma_{b_{k+1}}^{b_k} \approx \hat{\gamma}_{b_{k+1}}^{b_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2} \mathbf{J}_{b_w}^\gamma \delta \mathbf{b}_w \end{bmatrix} \quad (3.85)$$

代价函数(3.84)的最小值为单位四元数，所以可以将式(3.84)写为：

$$\mathbf{q}_{b_{k+1}}^{c_0} \mathbf{q}_{b_k}^{c_0} \otimes \gamma_{b_{k+1}}^{b_k} = \begin{bmatrix} \mathbf{1} \\ 0 \end{bmatrix} \quad (3.86)$$

将式(3.85)代入式(3.86)得，

$$\begin{aligned} \hat{\gamma}_{b_{k+1}}^{b_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2} \mathbf{J}_{b_w}^\gamma \delta \mathbf{b}_w \end{bmatrix} &= \mathbf{q}_{b_k}^{c_0-1} \otimes \mathbf{q}_{b_{k+1}}^{c_0} \otimes \begin{bmatrix} \mathbf{1} \\ 0 \end{bmatrix} \\ \Rightarrow \quad \begin{bmatrix} 1 \\ \frac{1}{2} \mathbf{J}_{b_w}^\gamma \delta \mathbf{b}_w \end{bmatrix} &= \hat{\gamma}_{b_{k+1}}^{b_k-1} \otimes \mathbf{q}_{b_k}^{c_0-1} \otimes \mathbf{q}_{b_{k+1}}^{c_0} \otimes \begin{bmatrix} \mathbf{1} \\ 0 \end{bmatrix} \\ \Rightarrow \quad \mathbf{J}_{b_w}^\gamma \delta \mathbf{b}_w &= 2 \operatorname{Im} \left((\hat{\gamma}_{b_{k+1}}^{b_k})^{-1} \otimes \mathbf{q}_{b_k}^{c_0-1} \otimes \mathbf{q}_{b_{k+1}}^{c_0} \right) \end{aligned} \quad (3.87)$$

其中 \mathbf{Im} 表示虚部。将式(3.87)两边同时乘以 $\mathbf{J}_{b_\omega}^{\gamma T}$, 使等式左边变为正定矩阵,

$$\mathbf{J}_{b_\omega}^{\gamma T} \mathbf{J}_{b_\omega}^{\gamma} \delta \mathbf{b}_\omega = 2 \mathbf{J}_{b_\omega}^{\gamma T} \mathbf{Im} \left((\hat{\boldsymbol{\gamma}}_{b_{k+1}}^{b_k})^{-1} \otimes \mathbf{q}_{b_k}^{c_0-1} \otimes \mathbf{q}_{b_{k+1}}^{c_0} \right) \quad (3.88)$$

这样就可以直接通过 Cholesky 分解法得到陀螺仪的 bias。然后再用计算出的陀螺仪 bias 再次重新积分 IMU。

(2) 速度、重力矢量和尺度因子初始化

将速度、重力和尺度因子放在一起作为优化变量,

$$\mathcal{X}_I = [\mathbf{v}_{b_0}^{b_0}, \mathbf{v}_{b_1}^{b_1}, \dots, \mathbf{v}_{b_n}^{b_n}, \mathbf{g}^{c_0}, s] \quad (3.89)$$

其中, 当取第 k 帧图像时, $\mathbf{v}_{b_k}^{b_k}$ 是 IMU 坐标系下的速度, \mathbf{g}^{c_0} 是第 0 帧图像对应得相机坐标系下的重力向量, s 是尺度因子。

考虑窗口中连续的两帧图像 b_k 和 b_{k+1} , 那么式(3.16)可以写成:

$$\begin{aligned} \boldsymbol{\alpha}_{b_{k+1}}^{b_k} &= \mathbf{R}_{c_0}^{b_k} (s(\bar{\mathbf{p}}_{b_{k+1}}^{c_0} - \bar{\mathbf{p}}_{b_k}^{c_0}) + \frac{1}{2} \mathbf{g}^{c_0} \Delta t_k^2 - \mathbf{R}_{b_k}^{c_0} \mathbf{v}_{b_k}^{b_k} \Delta t_k) \\ \boldsymbol{\beta}_{b_{k+1}}^{b_k} &= \mathbf{R}_{c_0}^{b_k} (\mathbf{R}_{b_{k+1}}^{c_0} \mathbf{v}_{b_{k+1}}^{b_{k+1}} + \mathbf{g}^{c_0} \Delta t_k - \mathbf{R}_{b_k}^{c_0} \mathbf{v}_{b_k}^{b_k}). \end{aligned} \quad (3.90)$$

定义残差项为 b_k 和 b_{k+1} 帧图像间的 IMU 预积分量 $\hat{\boldsymbol{\alpha}}_{b_{k+1}}^{b_k}, \hat{\boldsymbol{\beta}}_{b_{k+1}}^{b_k}$ 与预测值 $\boldsymbol{\alpha}_{b_{k+1}}^{b_k}, \boldsymbol{\beta}_{b_{k+1}}^{b_k}$ 之间的误差,

$$\begin{aligned} \mathbf{r} \left(\hat{\mathbf{Z}}_{b_{k+1}}^{b_k}, \mathcal{X}_I \right) &= \begin{bmatrix} \delta \boldsymbol{\alpha}_{b_{k+1}}^{b_k} \\ \delta \boldsymbol{\beta}_{b_{k+1}}^{b_k} \end{bmatrix} \\ &= \begin{bmatrix} \hat{\boldsymbol{\alpha}}_{b_{k+1}}^{b_k} - \mathbf{R}_{c_0}^{b_k} (s(\bar{\mathbf{p}}_{b_{k+1}}^{c_0} - \bar{\mathbf{p}}_{b_k}^{c_0}) + \frac{1}{2} \mathbf{g}^{c_0} \Delta t_k^2 - \mathbf{R}_{b_k}^{c_0} \mathbf{v}_{b_k}^{b_k} \Delta t_k) \\ \hat{\boldsymbol{\beta}}_{b_{k+1}}^{b_k} - \mathbf{R}_{c_0}^{b_k} (\mathbf{R}_{b_{k+1}}^{c_0} \mathbf{v}_{b_{k+1}}^{b_{k+1}} + \mathbf{g}^{c_0} \Delta t_k - \mathbf{R}_{b_k}^{c_0} \mathbf{v}_{b_k}^{b_k}) \end{bmatrix} \end{aligned} \quad (3.91)$$

将式(3.83)代入式(3.91)中的 $\delta \hat{\boldsymbol{\alpha}}_{b_{k+1}}^{b_k}$ 并写成 $\mathbf{Hx} = \mathbf{b}$ 的形式,

$$\mathbf{R}_{c_0}^{b_k} (\bar{\mathbf{p}}_{c_{k+1}}^{c_0} - \bar{\mathbf{p}}_{c_k}^{c_0})_S - \Delta t_k \mathbf{v}_{b_k}^{b_k} + \frac{1}{2} \mathbf{R}_{c_0}^{b_k} \Delta t_k^2 \mathbf{g}^{c_0} = \boldsymbol{\alpha}_{b_{k+1}}^{b_k} - \mathbf{p}_c^{b_k} + \mathbf{R}_{c_0}^{b_k} \mathbf{R}_{b_{k+1}}^{c_0} \mathbf{p}_c^b \quad (3.92)$$

写成矩阵形式,

$$\begin{bmatrix} -\mathbf{I}\Delta t_k & 0 & \frac{1}{2}\mathbf{R}_{c_0}^{b_k}\Delta t_k^2 & \mathbf{R}_{c_0}^{b_k}(\bar{\mathbf{p}}_{c_{k+1}}^{c_0} - \bar{\mathbf{p}}_{c_k}^{c_0}) \end{bmatrix} \begin{bmatrix} \mathbf{v}_{b_k}^{b_k} \\ \mathbf{v}_{b_{k+1}}^{b_{k+1}} \\ \mathbf{g}^{c_0} \\ S \end{bmatrix} = \boldsymbol{\alpha}_{b_{k+1}}^{b_k} - \mathbf{p}_{c_0}^{b_k} + \mathbf{R}_{c_0}^{b_k}\mathbf{R}_{b_{k+1}}^{c_0}\mathbf{p}_c^b \quad (3.93)$$

同理, 将 $\hat{\boldsymbol{\beta}}_{b_{k+1}}^{b_k}$ 也可以写成类似式 (3.93) 的形式, 并与式 (3.93) 合并写成如下矩阵形式:

$$\begin{bmatrix} -\mathbf{I}\Delta t_k & 0 & \frac{1}{2}\mathbf{R}_{c_0}^{b_k}\Delta t_k^2 & \mathbf{R}_{c_0}^{b_k}(\bar{\mathbf{p}}_{c_{k+1}}^{c_0} - \bar{\mathbf{p}}_{c_k}^{c_0}) \\ -\mathbf{I} & \mathbf{R}_{c_0}^{b_k}\mathbf{R}_{b_{k+1}}^{c_0} & \mathbf{R}_{c_0}^{b_k}\Delta t_k & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v}_{b_k}^{b_k} \\ \mathbf{v}_{b_{k+1}}^{b_{k+1}} \\ \mathbf{g}^{c_0} \\ S \end{bmatrix} = \begin{bmatrix} \boldsymbol{\alpha}_{b_{k+1}}^{b_k} - \mathbf{p}_{c_0}^{b_k} + \mathbf{R}_{c_0}^{b_k}\mathbf{R}_{b_{k+1}}^{c_0}\mathbf{p}_c^b \\ \boldsymbol{\beta}_{b_{k+1}}^{b_k} \end{bmatrix} \quad (3.94)$$

式 (3.94) 通过 Cholesky 分解可以解出 \mathcal{X}_I 。

(3) 重力矢量修正

由于噪声等原因, 在上一步求出来的重力 \mathbf{g}^{c_0} 是有误差的。通常情况下, 重力大小是已知的, $g \approx 9.81 \text{m/s}^2$ 。因此, 可以根据这个先验信息来修正上一步估计出来的 \mathbf{g}^{c_0} 。

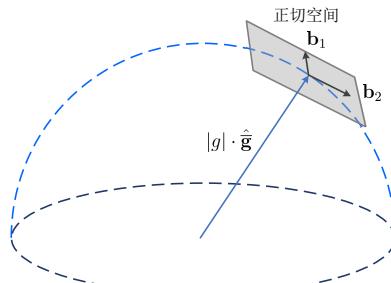


图 3.13 两自由度重力矢量参数化示意图

因为重力的大小已知, 所以重力的自由度由 3 个变为 2 个。如图3.13, 对重力重新参数化为:

$$\hat{\mathbf{g}} = g \cdot \bar{\mathbf{g}} + w_1 \mathbf{b}_1 + w_2 \mathbf{b}_2 \quad (3.95)$$

其中， $\hat{\mathbf{g}}$ 为上一步估计出来的重力的方向向量， $\mathbf{b}_1, \mathbf{b}_2$ 是垂直于 $\hat{\mathbf{g}}$ 的两个正交基。

将式 (3.95) 代入式 (3.94) 可得，

$$\begin{aligned} & \begin{bmatrix} -\mathbf{I}\Delta t_k & 0 & \frac{1}{2}\mathbf{R}_{c_0}^{b_k}\Delta t_k^2\mathbf{b} & \mathbf{R}_{c_0}^{b_k}(\bar{\mathbf{p}}_{c_{k+1}}^{c_0} - \bar{\mathbf{p}}_{c_k}^{c_0}) \\ -\mathbf{I} & \mathbf{R}_{c_0}^{b_k}\mathbf{R}_{b_{k+1}}^{c_0} & \mathbf{R}_{c_0}^{b_k}\Delta t_k\mathbf{b} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v}_{b_k}^{b_k} \\ \mathbf{v}_{b_{k+1}}^{b_{k+1}} \\ \boldsymbol{\omega} \\ s \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{\alpha}_{b_{k+1}}^{b_k} - \mathbf{p}_{c_0}^{b_k} + \mathbf{R}_{c_0}^{b_k}\mathbf{R}_{b_{k+1}}^{c_0}\mathbf{p}_c^b - \frac{1}{2}\mathbf{R}_{c_0}^{b_k}\Delta t_k^2\mathbf{g} \cdot \hat{\mathbf{g}} \\ \boldsymbol{\beta}_{b_{k+1}}^{b_k} - \mathbf{R}_{c_0}^{b_k}\Delta t_k\mathbf{g} \cdot \hat{\mathbf{g}} \end{bmatrix} \end{aligned} \quad (3.96)$$

其中， $\boldsymbol{\omega} = [\omega_1, \omega_2]^T$ ， $\mathbf{b} = [b_1, b_2]^T$ 。通过求解式 (3.96)，可以得到 $\boldsymbol{\omega}$ ，从而可以得到修正后的 \mathbf{g}^{c_0} 。通过 \mathbf{g}^{c_0} 和世界坐标系的 Z 轴方向，可以得到第一帧图像坐标系到世界坐标系的旋转 $\mathbf{q}_{C_0}^w$ ，从而可以将所有的状态变量变换到世界坐标系下。至此，初始化完成。

(4) 初始航向修正

在联合初始化完成之后，使用磁力计测量的航向角对初始化过程中估计出的每一个关键帧的航向角进行修正，从而得到无偏的初始航向。

3.4 本章小结

本章对系统的前端和初始化算法进行了详细的阐述。其中，前端部分详细研究了特征的提取、跟踪和异常点剔除算法，并对 IMU 预积分理论的相关公式进行了详细的推导。系统初始化部分详细研究了 visual 初始化算法，以及 Visual-IMU 对齐方法。系统的前端和初始化在整个系统中占了很大的比重，前端和初始化的好坏影响后端非线性优化的精度，从而影响整个系统的定位精度。因此，一个良好的前端和初始化是实现系统高精度定位的基础。

第4章 后端优化及闭环检测

在初始化完成后，采用基于滑动窗口的紧耦合单目 VIO 进行高精度和鲁棒性的状态估计。后端非线性优化是实现视觉/惯性紧耦合的关键步骤，闭环检测能够实现系统的重定位，提高系统的鲁棒性。本章将对后端非线性优化算法、闭环检测算法以及闭环优化算法进行详细分析与研究。

4.1 后端非线性优化

4.1.1 状态向量和目标函数

滑动窗口中的所有状态向量定义为：

$$\begin{aligned}\mathcal{X} &= [\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{x}_c^b, \lambda_0, \lambda_1, \dots, \lambda_m] \\ \mathbf{x}_k &= [\mathbf{p}_{b_k}^w, \mathbf{v}_{b_k}^w, \mathbf{q}_{b_k}^w, \mathbf{b}_a, \mathbf{b}_g], k \in [0, n] \\ \mathbf{x}_c^b &= [\mathbf{p}_c^b, \mathbf{q}_c^b],\end{aligned}\tag{4.1}$$

其中 \mathbf{x}_k 是第 k 帧图像关联的 IMU 状态，它包含了 IMU 在世界坐标系中的位置、速度和方向，以及在 IMU 坐标系中的加速度计 bias 和陀螺仪 bias， n 是关键帧的总数， m 是滑动窗口中的路标点总数， λ_l 是第一次观测到的第 l 个路标的逆深度（深度的倒数）。 \mathbf{x}_c^b 是外参。

定义目标函数为所有先验项和测量残差的马氏范数（Mahalanobis Norm）之和，从而得到最大后验概率估计：

$$\min_{\mathcal{X}} \left\{ \|\mathbf{r}_p - \mathbf{H}_p \mathcal{X}\|^2 + \sum_{k \in \mathcal{B}} \left\| \mathbf{r}_{\mathcal{B}}(\hat{\mathbf{z}}_{b_{k+1}}^{b_k}, \mathcal{X}) \right\|_{\mathbf{P}_{b_{k+1}}}^2 + \sum_{(l,j) \in \mathcal{C}} \rho(\|\mathbf{r}_{\mathcal{C}}(\hat{\mathbf{z}}_l^{c_j}, \mathcal{X})\|_{\mathbf{P}_l}^2) \right\} \tag{4.2}$$

其中， $\mathbf{r}_{\mathcal{B}}(\hat{\mathbf{z}}_{b_{k+1}}^{b_k}, \mathcal{X})$ 和 $\mathbf{r}_{\mathcal{C}}(\hat{\mathbf{z}}_l^{c_j}, \mathcal{X})$ 分别是 IMU 和视觉测量的残差。 \mathcal{B} 是所有 IMU 测量的集合， \mathcal{C} 是在当前滑动窗口中至少观察到两次的一组特征。 $\{\mathbf{r}_p, \mathbf{H}_p\}$ 是来自边缘化的先验信息。 $\rho(\cdot)$ 为：

$$\rho(s) = \begin{cases} s & s \leq 1, \\ 2\sqrt{s} - 1 & s > 1. \end{cases} \tag{4.3}$$

将 $\|\mathbf{r}_c\|^2$ 代入式 (4.3) 得,

$$\frac{1}{2}\rho(\|\mathbf{r}_c\|) = \begin{cases} \frac{1}{2}\|\mathbf{r}_c\|^2 & |\mathbf{r}_c| \leq 1, \\ \|\mathbf{r}_c\| - \frac{1}{2} & |\mathbf{r}_c| > 1. \end{cases} \quad (4.4)$$

令 $e = \|\mathbf{r}_c\|$, 则式 (4.4) 可变形为,

$$H(e) = \begin{cases} \frac{1}{2}e^2, & e \leq 1, \\ |e| - \frac{1}{2}, & e > 1. \end{cases} \quad (4.5)$$

$H(e)$ 是胡贝尔 (Huber) 核函数^[66]。Huber 核函数是鲁棒核函数得一种, 使用核函数是为了避免异常的误差值对整个目标函数的影响。因为当误差很大时, 二范数 (或马氏范数) 增长太快, 导致这种异常的误差项掩盖掉其他正确得误差项。图 4.1 展示了 Huber 核函数与二次函数的图像对比, 可见当误差较大时, Huber 核函数的增长速度要低于二次函数。

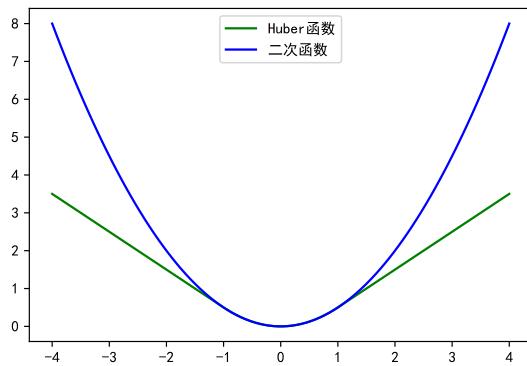


图 4.1 Huber 函数与二次函数图像

需要注意的是, 使用马氏范数而不是二范数因为马氏范数消除了不同残差项的尺度不一致性 (单位/量纲不一致性), 或者说马氏范数对残差项的概率模型进行了归一化, 从而使得不同的残差项可以直接相加。

要计算目标函数的最小值, 使用迭代优化的思想, 每次给优化变量一个增量, 使得目标函数最小。对于 IMU 残差项,

$$\begin{aligned} & \min_{\delta X} \left\| \mathbf{r}_B(\hat{\mathbf{z}}_{b_{k+1}}^{b_k}, \mathcal{X} + \Delta \mathcal{X}) \right\|_{P_{b_{k+1}}^{b_k}}^2 \\ &= \min_{\delta X} \left\| \mathbf{r}_B(\hat{\mathbf{z}}_{b_{k+1}}^{b_k}, \mathcal{X}) + \mathbf{J}_{b_{k+1}}^{b_k} \Delta \mathcal{X} \right\|_{P_{b_{k+1}}^{b_k}}^2 \end{aligned} \quad (4.6)$$

其中, $\mathbf{J}_{b_{k+1}}^{b_k}$ 是 IMU 残差项 \mathbf{r}_B 关于状态向量 \mathcal{X} 的雅可比矩阵。上式可以展开成 $\mathbf{H}\Delta\mathbf{x} = \mathbf{b}$ 的形式:

$$(\mathbf{J}_{b_{k+1}}^{b_k})^T (\mathbf{P}_{b_{k+1}}^{b_k})^{-1} \mathbf{J}_{b_{k+1}}^{b_k} \Delta\mathcal{X} = -(\mathbf{J}_{b_{k+1}}^{b_k})^T (\mathbf{P}_{b_{k+1}}^{b_k})^{-1} \mathbf{r}_B$$

其中, $\mathbf{P}_{b_{k+1}}^{b_k}$ 是 IMU 预积分中噪声项的协方差矩阵。

同理, 可以写出视觉残差项的增量方程。整体目标函数的增量方程为:

$$\begin{aligned} & \left(\mathbf{H}_p + \sum (\mathbf{J}_{b_{k+1}}^{b_k})^T (\mathbf{P}_{b_{k+1}}^{b_k})^{-1} \mathbf{J}_{b_{k+1}}^{b_k} + \sum \mathbf{J}_l^{C_j T} (\mathbf{P}_l^{C_j})^{-1} \mathbf{J}_l^{C_j} \right) \Delta\mathcal{X} \\ &= \mathbf{b}_p + \sum \mathbf{J}_{b_{k+1}}^{b_k} (\mathbf{P}_{b_{k+1}}^{b_k})^{-1} \mathbf{r}_B + \sum \mathbf{J}_l^{C_j T} (\mathbf{P}_l^{C_j})^{-1} \mathbf{r}_C \end{aligned} \quad (4.7)$$

其中, $\mathbf{P}_l^{C_j}$ 是视觉噪声的协方差。暂时不考虑先验信息, 用 $(\mathbf{H}_p, \mathbf{b}_p)$ 表示。称 $\mathbf{P}_{b_{k+1}}^{b_k}^{-1}$ 为信息矩阵, 其大小可以代表 IMU 测量数据的可靠程度, 越大代表越可靠, 越小代表越不可靠。可以观察到, IMU 噪声越大, 信息矩阵就越小, 此时 IMU 观测量就越不可靠, 可以直观的理解为目标函数“更加相信”视觉的观测量。

将增量方程 (4.7) 简化为:

$$(\Lambda_p + \Lambda_B + \Lambda_C) \Delta\mathcal{X} = \mathbf{b}_p + \mathbf{b}_B + \mathbf{b}_C \quad (4.8)$$

其中, $\Lambda_p, \Lambda_B, \Lambda_C$ 分别为先验残差项, IMU 残差项和视觉残差项的海森 (Hessian) 矩阵。

4.1.2 IMU 约束

考虑滑动窗口中连续两帧图像 (b_k 和 b_{k+1}) 内的 IMU 测量, 根据式 (3.41) 中定义的 IMU 测量模型, 预积分残差可以定义为:

$$\mathbf{r}_B(\hat{\mathbf{z}}_{b_{k+1}}^{b_k}, \mathcal{X}) = \begin{bmatrix} \delta\boldsymbol{\alpha}_{b_{k+1}}^{b_k} \\ \delta\boldsymbol{\beta}_{b_{k+1}}^{b_k} \\ \delta\boldsymbol{\theta}_{b_{k+1}}^{b_k} \\ \delta\mathbf{b}_a \\ \delta\mathbf{b}_g \end{bmatrix} = \begin{bmatrix} \mathbf{R}_w^{b_k} (\mathbf{p}_{b_{k+1}}^w - \mathbf{p}_{b_k}^w + \frac{1}{2} \mathbf{g}^w \Delta t_k^2 - \mathbf{v}_{b_k}^w \Delta t_k) - \hat{\boldsymbol{\alpha}}_{b_{k+1}}^{b_k} \\ \mathbf{R}_w^{b_k} (\mathbf{v}_{b_{k+1}}^w + \mathbf{g}^w \Delta t_k^2 - \mathbf{v}_{b_k}^w) - \hat{\boldsymbol{\beta}}_{b_{k+1}}^{b_k} \\ 2 \left[\mathbf{q}_{b_k}^{w-1} \otimes \mathbf{q}_{b_{k+1}}^w \otimes (\hat{\boldsymbol{\gamma}}_{b_{k+1}}^{b_k})^{-1} \right]_{xyz} \\ \mathbf{b}_{ab_{k+1}} - \mathbf{b}_{ab_k} \\ \mathbf{b}_{wb_{k+1}} - \mathbf{b}_{wb_k} \end{bmatrix} \quad (4.9)$$

其中, $[\cdot]_{xyz}$ 表示四元数的向量。 $\delta\boldsymbol{\theta}_{b_{k+1}}^{b_k}$ 是四元数对应的三维误差状态。 $[\hat{\boldsymbol{\alpha}}_{b_{k+1}}^{b_k}, \hat{\boldsymbol{\beta}}_{b_{k+1}}^{b_k}, \hat{\boldsymbol{\gamma}}_{b_{k+1}}^{b_k}]^T$ 是在两个连续图像帧之间的间隔时间内带有噪声的预积分 IMU 测量项。 $\delta\mathbf{b}_a, \delta\mathbf{b}_g$ 是加

速度计 bias 和陀螺仪 bias 的误差。

IMU 残差项的优化变量为:

$$[\mathbf{p}_{b_k}^w, \mathbf{q}_{b_k}^w], [\mathbf{v}_{b_k}^w, \mathbf{b}_{a_k}, \mathbf{b}_{\omega_k}], [\mathbf{p}_{b_{k+1}}^w, \mathbf{q}_{b_{k+1}}^w], [\mathbf{v}_{b_{k+1}}^w, \mathbf{b}_{a_{k+1}}, \mathbf{b}_{\omega_{k+1}}]$$

在进行迭代优化时需要知道残差项对优化变量的一阶导数, 也就是雅可比矩阵, 以及协方差矩阵。仍旧采用扰动模型求导, 扰动为:

$$[\delta \mathbf{p}_{b_k}^w, \delta \theta_{b_k}^w], [\delta \mathbf{v}_{b_k}^w, \delta \mathbf{b}_{a_k}, \delta \mathbf{b}_{\omega_k}], [\delta \mathbf{p}_{b_{k+1}}^w, \delta \theta_{b_{k+1}}^w], [\delta \mathbf{v}_{b_{k+1}}^w, \delta \mathbf{b}_{a_{k+1}}, \delta \mathbf{b}_{\omega_{k+1}}]$$

这里直接给出通过扰动模型求导后的雅可比矩阵:

$$\begin{aligned} \mathbf{J}[0]^{15 \times 7} &= \left[\frac{\partial \mathbf{r}_B}{\partial \mathbf{p}_{b_k}^w}, \frac{\partial \mathbf{r}_B}{\partial \mathbf{q}_{b_k}^w} \right] \\ &= \begin{bmatrix} -\mathbf{R}_w^{b_k} & \left[\mathbf{R}_w^{b_k} \left(\mathbf{p}_{b_{k+1}}^w - \mathbf{p}_{b_k}^w - \mathbf{v}_{b_k}^w \Delta t_k + \frac{1}{2} \mathbf{g}^w \Delta t_k^2 \right) \right]_x \\ 0 & -\left[(\mathbf{q}_{b_{k+1}}^w)^{-1} \otimes \mathbf{q}_{b_k}^w \right]_L \left[\boldsymbol{\gamma}_{b_{k+1}}^{b_k} \right]_R \\ 0 & \left[\mathbf{R}_w^{b_k} \left(\mathbf{v}_{b_{k+1}}^w - \mathbf{v}_{b_k}^w + \mathbf{g}^w \Delta t_k \right) \right]_x \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (4.10) \end{aligned}$$

$$\begin{aligned} \mathbf{J}[1]^{15 \times 9} &= \left[\frac{\partial \mathbf{r}_B}{\partial \mathbf{v}_{b_k}^w}, \frac{\partial \mathbf{r}_B}{\partial \mathbf{b}_k}, \frac{\partial \mathbf{r}_B}{\partial \mathbf{b}_{w_k}} \right] \\ &= \begin{bmatrix} -\mathbf{R}_w^{b_k} \Delta t & -\mathbf{J}_{b_a}^\alpha & -\mathbf{J}_{b_\omega}^\alpha \\ 0 & 0 & \left[\mathbf{q}_{b_{k+1}}^{w-1} \otimes \mathbf{q}_{b_k}^w \otimes \boldsymbol{\gamma}_{b_{k+1}}^{b_k} \right]_L \mathbf{J}_{b_\omega}^\gamma \\ -\mathbf{R}_w^{b_k} & -\mathbf{J}_{b_a}^\beta & -\mathbf{J}_{b_\omega}^\beta \\ 0 & -\mathbf{I} & 0 \\ 0 & 0 & -I \end{bmatrix} \quad (4.11) \end{aligned}$$

$$\mathbf{J}[2]^{15 \times 7} = \left[\frac{\partial \mathbf{r}_B}{\partial \mathbf{p}_{b_{k+1}}^w}, \frac{\partial \mathbf{r}_B}{\partial \mathbf{q}_{b_{k+1}}^w} \right] = \begin{bmatrix} \mathbf{R}_w^{b_k} & 0 \\ 0 & \left[\boldsymbol{\gamma}_{b_{k+1}}^{b_k}^{-1} \otimes \mathbf{q}_{b_k}^{w-1} \otimes \mathbf{q}_{b_{k+1}}^w \right]_L \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (4.12)$$

$$\mathbf{J}[3]^{15 \times 9} = \left[\frac{\partial \mathbf{r}_B}{\partial \mathbf{v}_{b_{k+1}}^w}, \frac{\partial \mathbf{r}_B}{\partial \mathbf{b}_{a_{k+1}}}, \frac{\partial \mathbf{r}_B}{\partial \mathbf{b}_{w_{k+1}}} \right] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \mathbf{R}_w^{b_k} & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & \mathbf{I} \end{bmatrix} \quad (4.13)$$

协方差矩阵是3.2.3小节中研究的预积分中的IMU增量误差的协方差矩阵，即公式(3.56)。

4.1.3 视觉约束

考虑第 l 个特征，其对应的3D空间点为 \mathbf{P}_l^w ，该特征在第 i 幅图像中被第一次观测到，观测到的像素坐标为 $\mathbf{P}_{uv_i} = [u_l^{c_i}, u_l^{c_j}]^T$ 。

$$\mathbf{P}_{uv_i} = \lambda_l \pi_c (\mathbf{T}_{b \leftarrow c}^{-1} \mathbf{T}_{w \leftarrow b_i}^{-1} \mathbf{P}_l^w) \quad (4.14)$$

其中， π_c 将世界坐标点映射为像素坐标点， $\mathbf{T}_{b \leftarrow c}^{-1}$ 将坐标从IMU系变换到相机系， $\mathbf{T}_{w \leftarrow b_i}^{-1}$ 将世界坐标系下的坐标变换到IMU坐标系下。 λ_l 是第 l 个特征在第 i 幅图像下的逆深度。由式(4.14)可得，

$$\begin{aligned} \mathbf{P}_l^w &= \mathbf{T}_{w \leftarrow b_i} \mathbf{T}_{b \leftarrow c} \frac{1}{\lambda_l} \pi_c^{-1} (\mathbf{P}_{uv_i}) \\ &= \mathbf{R}_{b_i}^w \left(\mathbf{R}_c^b \frac{1}{\lambda_l} \pi_c^{-1} \left(\begin{bmatrix} u_l^{c_i} \\ v_l^{c_i} \end{bmatrix} \right) + \mathbf{p}_c^b \right) + \mathbf{p}_{b_i}^w \end{aligned} \quad (4.15)$$

\mathbf{P}_l^w 在第 j 副图像对应的相机坐标系下的坐标为，

$$\mathcal{P}_l^{c_j} = \mathbf{T}_{b \leftarrow c}^{-1} \mathbf{T}_{w \leftarrow b_j}^{-1} \mathbf{P}_l^w \quad (4.16)$$

进而得到，

$$\begin{aligned} \mathbf{P}_l^w &= \mathbf{T}_{w \leftarrow b_j} \mathbf{T}_{b \leftarrow c} \mathcal{P}_l^{c_j} \\ &= \mathbf{R}_{b_j}^w (\mathbf{R}_c^b \mathcal{P}_l^{c_j} + \mathbf{p}_c^b) + \mathbf{p}_{b_j}^w \end{aligned} \quad (4.17)$$

联立式(4.15)和(4.17)可得，

$$\mathcal{P}_l^{c_j} = \mathbf{R}_b^c (\mathbf{R}_{b_i}^{b_j} (\mathbf{R}_c^b \frac{1}{\lambda_l} \pi_c^{-1} \left(\begin{bmatrix} u_l^{c_j} \\ v_l^{c_j} \end{bmatrix} \right) + \mathbf{p}_c^b) + \mathbf{p}_{b_i}^w - \mathbf{p}_{b_j}^w) - \mathbf{p}_c^b \quad (4.18)$$

此时，可以定义该特征在第 j 幅图像中的观测点和重投影点之间的残差定义为：

$$\mathbf{r}_c(\hat{\mathbf{z}}_l^{c_j}, \mathcal{X}) = \hat{\mathcal{P}}_l^{c_j} - \frac{\mathcal{P}_l^{c_j}}{\|\mathcal{P}_l^{c_j}\|} \quad (4.19)$$

其中， $\mathcal{P}_l^{c_j}$ 为 \mathbf{P}_l^w 在第 j 副图像上对应的归一化平面坐标，

$$\hat{\mathcal{P}}_l^{c_j} = \pi_c^{-1} \left(\begin{bmatrix} \hat{u}_l^{c_j} \\ \hat{v}_l^{c_j} \end{bmatrix} \right) \quad (4.20)$$

视觉残差项的优化变量为：

$$[\mathbf{p}_{b_i}^w, \mathbf{q}_{b_i}^w], [\mathbf{p}_{b_j}^w, \mathbf{q}_{b_j}^w], [\mathbf{p}_c^b, \mathbf{q}_c^b], \lambda_l$$

同样需要计算残差项关于这些优化变量的一阶导数，得到其雅可比矩阵。这里直接给出各优化变量的雅可比矩阵：

$$\left\{ \begin{array}{l} \mathbf{J}[0]^{3 \times 7} = \left[\frac{\partial \mathbf{r}_c}{\partial \mathbf{p}_{b_i}^w}, \frac{\partial \mathbf{r}_c}{\partial \mathbf{q}_{b_i}^w} \right] = \left[\mathbf{R}_b^c \mathbf{R}_w^{b_j} - \mathbf{R}_b^c \mathbf{R}_w^{b_j} \mathbf{R}_{b_i}^w \left[\mathbf{R}_c^b \frac{1}{\lambda_l} \hat{\mathcal{P}}_l^{c_i} + \mathbf{p}_c^b \right]_{\times} \right] \\ \mathbf{J}[1]^{3 \times 7} = \left[\frac{\partial \mathbf{r}_c}{\partial \mathbf{p}_{b_j}^w}, \frac{\partial \mathbf{r}_c}{\partial \mathbf{q}_{b_j}^w} \right] \\ = \left[-\mathbf{R}_b^c \mathbf{R}_w^{b_j} \mathbf{R}_b^c \left[\mathbf{R}_w^{b_j} \left[\mathbf{R}_{b_i}^w \left(\mathbf{R}_c^b \frac{\hat{P}_l^{c_i}}{\lambda_l} + \mathbf{p}_c^b \right) + \mathbf{p}_{b_i}^w - \mathbf{p}_{b_j}^w \right] \right]_{\times} \right] \\ \mathbf{J}[2]^{3 \times 7} = \left[\frac{\partial \mathbf{r}_c}{\partial \mathbf{p}_c^b}, \frac{\partial \mathbf{r}_c}{\partial \mathbf{q}_c^b} \right] \\ = \left[\begin{array}{c} \mathbf{R}_b^c \left(\mathbf{R}_w^{b_j} \mathbf{R}_{b_i}^w - \mathbf{I}_{3 \times 3} \right) \\ -\mathbf{R}_b^c \mathbf{R}_w^{b_j} \mathbf{R}_{b_i}^w \mathbf{R}_c^b \left[\frac{\hat{\mathcal{P}}_l^{c_i}}{\lambda_l} \right]_{\times} + \left[\mathbf{R}_b^c \mathbf{R}_w^{b_j} \mathbf{R}_{b_i}^w \mathbf{R}_c^b \frac{\hat{\mathcal{P}}_l^{c_i}}{\lambda_l} \right]_{\times} \\ + \left[\mathbf{R}_b^c \left[\mathbf{R}_w^{b_j} \left(\mathbf{R}_{b_i}^w \mathbf{p}_c^b + \mathbf{p}_{b_i}^w - \mathbf{p}_{b_j}^w \right) - \mathbf{p}_c^b \right] \right]_{\times} \end{array} \right]^T \\ \mathbf{J}[3]^{3 \times 1} = \frac{\partial \mathbf{r}_c}{\partial \lambda_l} = -\mathbf{R}_b^c \mathbf{R}_w^{b_j} \mathbf{R}_{b_i}^w \mathbf{R}_c^b \frac{\hat{P}_l^{c_i}}{\lambda_l^2} \end{array} \right. \quad (4.21)$$

4.1.4 边缘化

前面在研究残差项的时候，忽略了先验残差项，这个先验残差项就是通过边缘化 (Marginalization)^[67] 来的。整个后端是基于滑动窗口来进行非线性优化的，滑动窗口

内的图像帧数量是固定的（在实际工程中取 10 帧），这样会带来一个问题：窗口在向前滑动的过程中，总会有新的图像帧和路标点进来，而窗口中最旧的图像帧和路标点会滑出窗口。要对滑窗内的所有图像位姿及其路标点进行优化，那么不能直接将最旧的图像帧及其观测到的路标点直接删除，因为这样会丢失观测信息，降维约束项。那么怎么才能在不减少约束项的情况下丢掉最旧的图像帧及其路标点呢？边缘化就是用来解决此问题的。

(1) 舒尔补消元

采用高斯牛顿或者 LM 方法进行非线性优化时，每次迭代都要求解增量方程 $\mathbf{H}\delta\mathbf{x} = \mathbf{b}$ ，将 $\mathbf{H}\delta\mathbf{x} = \mathbf{b}$ 写为：

$$\begin{bmatrix} \Lambda_a & \Lambda_b \\ \Lambda_b^T & \Lambda_c \end{bmatrix} \begin{bmatrix} \delta\mathbf{x}_1 \\ \delta\mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix} \quad (4.22)$$

其中， $\delta\mathbf{x}_1$ 是想要边缘化掉的优化变量， $\delta\mathbf{x}_2$ 是与 $\delta\mathbf{x}_1$ 有约束关系的想要保留的优化变量。定义舒尔项^[68]：

$$\begin{bmatrix} \mathbf{I} & 0 \\ -\Lambda_b^T \Lambda_a^{-1} & \mathbf{I} \end{bmatrix} \quad (4.23)$$

将等式 (4.22) 两边同时乘以式 (4.23)，

$$\begin{aligned} & \begin{bmatrix} \mathbf{I} & 0 \\ -\Lambda_b^T \Lambda_a^{-1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \Lambda_a & \Lambda_b \\ \Lambda_b^T & \Lambda_c \end{bmatrix} \begin{bmatrix} \delta\mathbf{x}_1 \\ \delta\mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{I} & 0 \\ -\Lambda_b^T \Lambda_a^{-1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix} \\ \Rightarrow & \begin{bmatrix} \Lambda_a & \Lambda_b \\ 0 & \Lambda_c - \Lambda_b^T \Lambda_a^{-1} \Lambda_b \end{bmatrix} \begin{bmatrix} \delta\mathbf{x}_1 \\ \delta\mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 - \Lambda_b^T \Lambda_a^{-1} \mathbf{b}_1 \end{bmatrix} \end{aligned} \quad (4.24)$$

通过式 (4.24)，可以忽略 $\delta\mathbf{x}_1$ 单独求解 $\delta\mathbf{x}_2$ ，

$$\begin{aligned} & (\Lambda_c - \Lambda_b^T \Lambda_a^{-1} \Lambda_b) \delta\mathbf{x}_2 = \mathbf{b}_2 - \Lambda_b^T \Lambda_a^{-1} \mathbf{b}_1 \\ \Rightarrow & \mathbf{H}^* \delta\mathbf{x}_2^* = \mathbf{b}^* \end{aligned} \quad (4.25)$$

式 (4.25) 中， \mathbf{b}^* 是 $\Lambda_b^T \Lambda_a^{-1} \mathbf{b}_1$ 有关的函数，所以，通过舒尔补，在保留了变量 $\delta\mathbf{x}_1$ 的约束信息的情况下消掉了 $\delta\mathbf{x}_1$ 。因此可以说式 (4.25) 保留了变量的先验信息。此时可以

构建先验残差项:

$$\mathbf{r}_p - \mathbf{H}_p \boldsymbol{\chi} \iff \hat{\mathbf{b}}^* - \mathbf{H}^* \delta \mathbf{x}_2^* \quad (4.26)$$

(2) 边缘化过程

假设相机在四个不同的地方拍摄了四张图像，相机的位姿分别为 $\mathbf{x}_{p_1}, \mathbf{x}_{p_2}, \mathbf{x}_{p_3}, \mathbf{x}_{p_4}$ ，四张图像总共观测到了六个路标点 $\mathbf{x}_{m_1}, \mathbf{x}_{m_2}, \mathbf{x}_{m_3}, \mathbf{x}_{m_4}, \mathbf{x}_{m_5}, \mathbf{x}_{m_6}$ ，他们之间的关系如图4.2所示。图4.2中相机和路标点之间的连线表示相机观测到了该路标点，相机和相

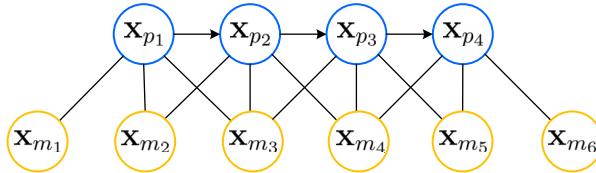


图 4.2 位姿和路标点的关系

机之间的箭头连接表示他们之间的 IMU 约束。

用图4.5 (a) 表示原始的 \mathbf{H} 矩阵，图中， m 表示路标， p 表示位姿，颜色块代表相互关联。图 4.3 (b) 表示将 \mathbf{H} 矩阵中跟位姿 \mathbf{x}_{p_1} 有关的元素移至左上角。图 4.3 (c) 表示将位姿 \mathbf{x}_{p_1} 边缘化掉后的 \mathbf{H} 矩阵，也就是式 (4.25) 中的 $(\Lambda_c - \Lambda_b^T \Lambda_a^{-1} \Lambda_b)$ 。

观察图4.5 (c) 可以发现，相对于图 (b)，图 (c) 更加稠密，或者说边缘化掉一个位姿后，余下的 \mathbf{H} 矩阵会多出三部分颜色块（图中的黄色块）。图4.3表示图4.5 (c) 对应的关系图。

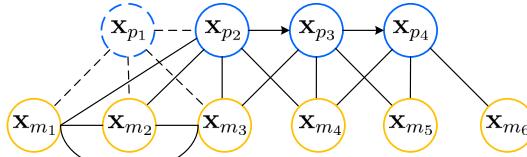
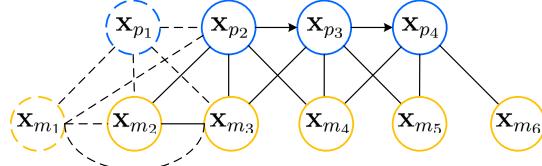
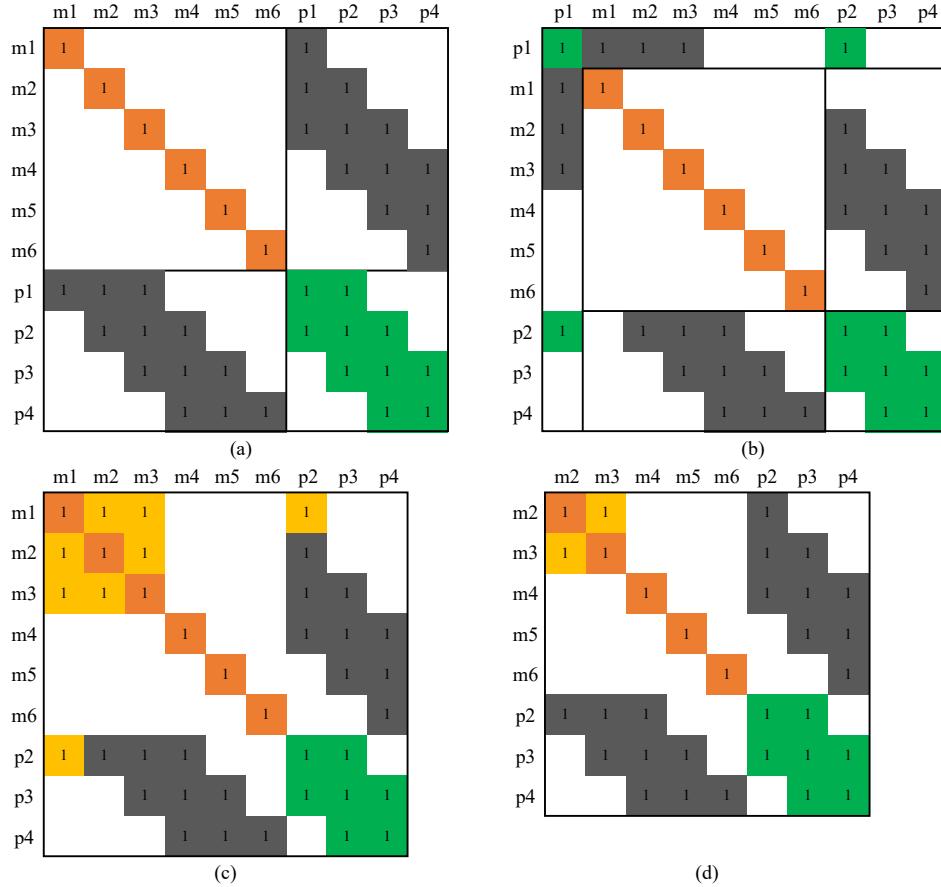


图 4.3 边缘化掉 \mathbf{x}_{p_1} 后位姿和路标点的关系

观察图4.3可以发现，路标点 $\mathbf{x}_{m_1}, \mathbf{x}_{m_2}, \mathbf{x}_{m_3}$ 之间已经相互有了新的联系，并且，位姿 \mathbf{x}_{p_2} 与路标点 \mathbf{x}_{m_2} 也有了新的联系。这些新的联系就是新的约束项，表现在图4.5 (c) 中的黄色区域。

图4.5 (d) 是表示再次边缘化掉路标点 \mathbf{x}_{m_1} 后的 \mathbf{H} 矩阵，其对应的关系图如图4.4所示。观察图4.5 (d)，发现 \mathbf{H} 矩阵并没有变得更稠密，这是因为路标点 \mathbf{x}_{m_1} 本身就没有被其他路标点观测到（从图4.3中可以看出），也就不会与其他位姿产生约束关系。因此，可以得出一个结论：当边缘化掉不被其他帧所观测到的路标点时， \mathbf{H} 矩阵不会进一步稠密。而对于被其他帧所观测到的路标点，通常不进行边缘化。

图 4.4 边缘化掉 \mathbf{x}_{p_1} 和 \mathbf{x}_{m_1} 后位姿和路标点的关系图 4.5 边缘化过程中 \mathbf{H} 矩阵变化图

3. 滑窗策略

如图4.6所示，整个后端非线性优化采用滑动窗口机制，窗口内维持固定的图像帧数量，根据新到来的图像帧是否为关键帧，使用不同的处理机制。图中 \mathbf{x}_n 为最新的图像帧， \mathbf{x}_{n-1} 为次新帧图像， \mathbf{x}_{n-2} 为第一帧旧的图像帧， \mathbf{x}_0 为最旧的图像帧。

- (1) 如果 \mathbf{x}_{n-1} 是关键帧，对 \mathbf{x}_0 进行边缘化处理，与 \mathbf{x}_0 及其关联的路标点的约束信息将转变为先验信息并整合到目标函数中；
- (2) 如果 \mathbf{x}_{n-1} 不是关键帧，将 \mathbf{x}_{n-1} 及其观测到的路标点直接丢弃。如果 \mathbf{x}_{n-1} 不是关键帧，说明它跟 \mathbf{x}_{n-2} 很相似（通过视差来判断是否为关键帧），那么 \mathbf{x}_{n-1} 及其路标点具有的约束信息和 \mathbf{x}_{n-2} 几乎一样，所以直接丢掉 \mathbf{x}_{n-1} 也不会丢失很多的约束信息。

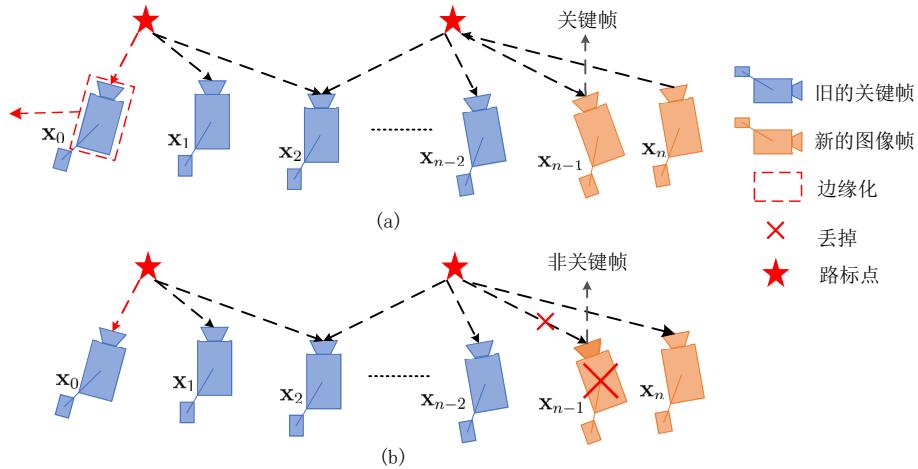


图 4.6 滑动窗口示意图

需要注意的是，不管是边缘化还是直接丢弃图像帧，都会完整的保留图像帧所对应的 IMU 预积分信息，从而确保 IMU 预积分不中断。

4.2 闭环检测

上一章研究了前端、初始化和后端优化，此时是视觉和 IMU 已经构成了紧耦合的 VIO 系统，能够估计出自身的位姿和路标点的位置。此时的 VIO 系统虽然可以对自身进行定位，但是由于视觉和 IMU 都有不可避免的累计误差，所以在融合之后的系统还是有误差的。这就导致了系统长时间估计的结果是不可靠的，无法构建全局一致（Global Consistent）的轨迹和地图。

闭环检测算法可以检测相机是否回到了之前的位置，如果检测到闭环，算法会将之前的累计误差消除掉，将轨迹拉回正确的位置，从而构建全局一致的轨迹和地图。而且，闭环检测能够给出当前图像与之前所有图像之间的关联信息，所以，如果在前端特征跟丢的情况下，可以通过闭环检测对当前的位置进行重定位，从而不至于使系统崩溃。

4.2.1 词袋模型

闭环检测有两种思路：基于里程计的几何关系和基于外观。基于里程计的几何关系是指，当相机运动到以前走过的某个位置附近时，检测是否存在闭环^[69]。这种方法有一个致命问题，长时间的视觉里程计的累计误差会导致轨迹严重漂移，相机自身是无法分辨出是否回到之前的某个位置，也就无法判断是否存在闭环关系。因此这种方

法仅仅适用于短距离，短时间的闭环运动。

第二种方法是基于外观的。它仅仅依靠图像间的相似程度来确定是否存在闭环关系。这种方法不受前端累计误差的影响，使得闭环检测可以独立的作为系统的一个模块。基于外观的方法是视觉 SLAM 中检测闭环的主流做法，并且已经应用于多个实际系统中^{[70] [71] [13]}。

如果使用第二种方法，那么闭环检测的关键就在于如何计算图像之间的相似程度。对于任意两张图像 \mathbf{A} 和 \mathbf{B} ，会用一种方法计算出他们之间的相似性评分： $s(\mathbf{A}, \mathbf{B})$ 。设定一个阈值，当评分大于这个阈值的时候，就认为他们之间存在闭环关系。通常，采用词袋模型（Bag-of-Words，BoW）^{[72] [73]} 来计算相似性评分。

词袋是用来描述图像内容特征的方法。图像中出现的一类物体（比如车，房子，人等）会对应于词袋中的一个“单词”，所有的单词构成一个“字典”。通常找出图像中出现了那些“单词”，出现单词的地方用“1”表示，没有出现的用“0”表示，所有的单词出现情况会构成一个向量，然后用这个向量来描述这张图像。最后，用向量的相似程度来表示图像间的相似程度。

假如有一个“字典”，字典里面的单词包括：“汽车”，“行人”和“房子”，分别记为 w_1, w_2, w_3 。有一副图像 \mathbf{A} 包含了“汽车”和“行人”，没有“房子”，图像 \mathbf{B} 包含了“汽车”和“房子”，没有“行人”那么 \mathbf{A}, \mathbf{B} 可以表示为：

$$\begin{aligned}\mathbf{A} &= 1 \cdot w_1 + 1 \cdot w_2 + 0 \cdot w_3, \\ \mathbf{B} &= 1 \cdot w_1 + 0 \cdot w_2 + 1 \cdot w_3\end{aligned}\tag{4.27}$$

所以，可以用向量 $[1, 1, 0]^T$ 来表示图像 \mathbf{A} ，向量 $[1, 0, 1]^T$ 来表示图像 \mathbf{B} 。可以发现，词袋模型描述的是图像中是否存在某个物体，而不管他们出现在图像的哪个地方。根据这两个向量来计算相似性：

$$s(\mathbf{A}, \mathbf{B}) = 1 - \frac{1}{W} \|\mathbf{A} - \mathbf{B}\|_1\tag{4.28}$$

其中， W 是词袋中“单词”的个数，这里 $W = 3$ ，当然，实际中的 W 会大得多。 $\|\cdot\|$ 代表 L_1 范数，也可以选用其他范数，依据具体情况而定。可以发现，当向量 \mathbf{A}, \mathbf{B} 完全一样时， $s = 1$ ；完全相反时， $s = 0$ 。这样就可以用 s 的大小来描述图像的相似性了。

4.2.2 字典构建

字典是由“单词”构成，而单词是从图像中抽象而来的。一个“单词”往往是一类特征的集合，所以可以通过聚类（Clustering）来确定“单词”。

聚类属于无监督机器学习（Unsupervised ML），用来寻找大量数据中的规律。假设在前端提取了 N 个特征，想要从这 N 个特征中抽象出来 k 个“单词”。使用 K 均值（K-means）^[74] 算法来解决该问题：

- (1) 在 N 个特征中随机选取 k 个特征作为中心点： c_1, \dots, c_k ；
- (2) 对其他的每个特征，计算它和每个中心点之间的距离，将它与距离最小的那个中心点归为一类；
- (3) 再次计算每个类的中心点；
- (4) 如果重新计算出来的中心点和之前的中心点相差不大，则说明算法收敛，聚类成功。否则，返回到步骤(1)。

至此，就可以得到一个字典了。接下来要考虑的是如何根据图像中提取的特征和字典中的“单词”进行匹配。最简单的方法是采取暴力匹配，将特征和字典中的每个“单词”进行匹配，取相似度最高的那个。然而，字典一般是通用性的，这就意味着字典中含有许多的单词，暴力匹配会很耗时。

为了解决这个问题，将字典表示称 k 叉树。 k 叉树将聚类层次化，是 k 均值的一种扩展。如图4.7所示，用刚才的 N 个特征构建一个深度为 d ，树权为 k 的树：

- (1) 用 k 均值法将根节点（所有的 N 个特征）聚成 k 类，得到第一层聚类；
- (2) 对第一层聚类的每个节点，再使用 k 均值法聚成 k 类，得到第二层聚类；
- (3) 以此类推，直到第 $k - 1$ 层聚类，也就是叶子节点，此时的每个叶子节点就是一个“单词”。

可以发现深度为 d ，分支为 k 的树，可以最终聚类成 k^d 个单词。在查找“单词”时，只需要查找每一个中间层聚类的聚类中心，总共需要查找 d 次。

4.2.3 相似度计算

在研究词袋模型的时候简单研究了一种计算相似度的方法，那是一种简单的为了帮助理解的方法。实际中，会用到另一种方法。在之前研究的计算方法中，对所有的“单词”都同等对待，有就是“1”，没有就是“0”。但是，实际中，不同的“单词”在

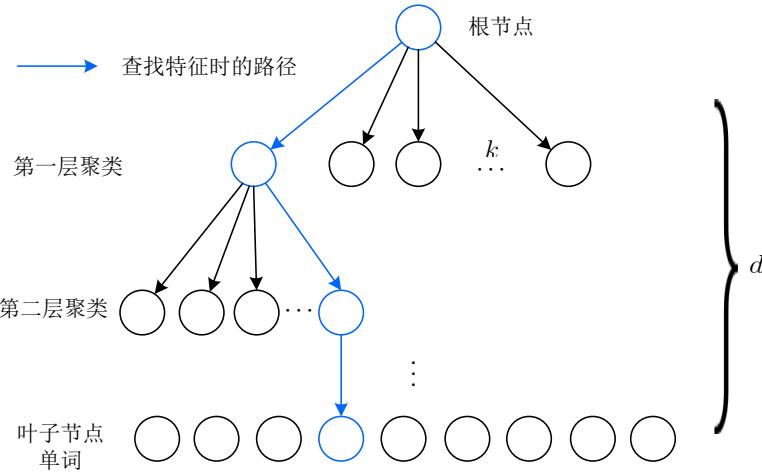


图 4.7 K 叉树字典示意图

区分性上不相同。比如，在空旷的野外，像“树木”，“花草”，“高山”等类的“单词”会在图像中经常出现，无法根据他们判断该图像跟其他图像的区别。但是如果有“房屋”，“汽车”等类的“单词”出现时，将更容易的判断出该图像相对于其他图像的独特之处。可以说，他们提供了区分度更大的信息。所以，需要对每个单词的区分度有一个量化的评估，通常使用加权的方法来区别不同单词的重要性。权重越大，区分度就越高。

在文本检索领域，TF-IDF（Term Frequency–Inverse Document Frequency）是一种常见的方法^{[75][76]}。TF 表示，如果文本中经常出现一个单词，则它的区分度相对较好。IDF 表示，文本中出现的相似单词越少，与其他文本的区别就越大。在闭环检测中可以借鉴此方法。

对于词袋模型，在构建词典时，借鉴 IDF 思想。计算某个“单词” w_i 中的特征数量与所有特征总数的比值，将这个比值作为 IDF。设 w_i 的特征数量为 n_i ，特征总数为 n ，那么 w_i 的 IDF 为：

$$\text{IDF}_i = \log \frac{n}{n_i} \quad (4.29)$$

另一方面，对于单个图像，借鉴 TF 思想。设图像 A 中，“单词” w_i 出现了 m_i 次，出现的单词总数为 m ，那么 w_i 的 TF 为：

$$\text{TF}_i = \frac{m_i}{m} \quad (4.30)$$

定义 w_i 的权重为：

$$\eta_i = \text{TF}_i \times \text{IDF}_i \quad (4.31)$$

有了每个“单词”的权重，可以计算该图像的词袋（用向量 \mathbf{v}_A 表示）：

$$\mathbf{v}_A \triangleq \{(w_1, \eta_1), (w_2, \eta_2), \dots, (w_N, \eta_N)\} \quad (4.32)$$

\mathbf{v}_A 是一个稀疏向量，其非零部分（TF-IDF 的值）表示图像中包括哪些“单词”。

假设计算出了图像 \mathbf{A} 和 \mathbf{B} 的词袋： \mathbf{v}_A 和 \mathbf{v}_B ，那么可以定义他们之间的相似度为：

$$s(\mathbf{v}_A - \mathbf{v}_B) = 2 \sum_{i=1}^N |\mathbf{v}_{Ai}| + |\mathbf{v}_{Bi}| - |\mathbf{v}_{Ai} - \mathbf{v}_{Bi}| \quad (4.33)$$

当然相似度的计算方法有很多种，这里采用 L_1 范数形式。

在滑动窗口策略中，会实时计算当前帧与字典里词袋的相似度分数，并与之前的所有关键帧进行比较，并对其实施闭环一致性检验，从而得到闭环候选帧。

4.3 重定位和闭环优化

4.3.1 重定位

在后端采用的滑动窗口和边缘化方案极大的降低了计算的复杂性，但也给系统带来了累积误差。更确切地说，累积误差主要发生在全局三维位置 (x, y, z) 和与重力方向垂直的偏航角上。为了减少误差，提出了一种利用闭环检测来矫正误差图像位置的重定位方法。

如图4.8所示，通过闭环检测建立闭环候选帧（蓝色的帧）和当前滑动窗口中的图像帧（绿色的帧）之间的特征关系。当检测到闭环时，将当前帧的位姿及其对应的路标点作为视觉约束项，并添加到目标函数中。所以式 (4.2) 可以重新写为：

$$\begin{aligned} \min_{\mathcal{X}} & \left\{ \|\mathbf{r}_p - \mathbf{H}_p \mathcal{X}\|^2 + \sum_{k \in \mathcal{B}} \left\| \mathbf{r}_B(\hat{\mathbf{z}}_{b_{k+1}}^{b_k}, \mathcal{X}) \right\|_{\mathbf{P}_{b_{k+1}}}^2 + \sum_{(l,j) \in \mathcal{C}} \rho(\|\mathbf{r}_C(\hat{\mathbf{z}}_l^{c_j}, \mathcal{X})\|_{\mathbf{P}_l^{c_j}}^2) \right. \\ & \left. + \sum_{(l,v) \in \mathcal{L}} \rho(\|\mathbf{r}_C(\hat{\mathbf{z}}_l^v, \mathcal{X}, \hat{\mathbf{q}}_v^w, \hat{\mathbf{p}}_v^w)\|_{\mathbf{P}_l^{c_v}}^2) \right\} \end{aligned} \quad (4.34)$$

其中， $(\hat{\mathbf{q}}_v^w, \hat{\mathbf{p}}_v^w)$ 是回环帧的姿态，被视为常数。 \mathcal{L} 是闭环帧中检索到的路标点的集合。 (l, v) 是指在闭环帧 v 中观察到的第 l 个特征。注意，虽然目标函数与之前略有不同，但待解状态量的维数保持不变，因为回环帧的姿态被视为常数。当滑动窗口建立多个回环时，使用来自所有闭环帧及其对应的路标点进行优化。这就为重新定位提供了多

视角的约束，从而提高了定位的精度和平滑性。

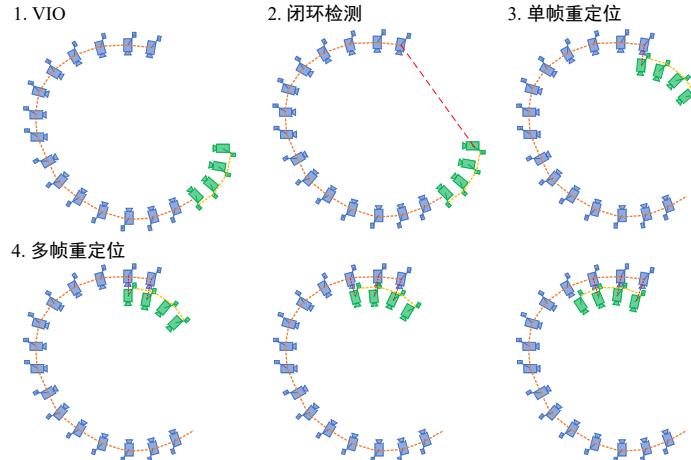


图 4.8 重定位过程示意图

4.3.2 闭环优化

之前讨论的 BA 优化，往往是同时优化位姿及其 3D 路标点，这样会存在一个问题：随着时间的累计，相机的运动距离越来越远，路标点也在不断增加，如果不使用滑窗优化，而是优化过去所有的位姿和路标点，那么计算量会逐渐增加，计算效率会逐渐下降。当然，本系统中使用了滑窗策略，避免了这种情况。但是，为了达到全局一致性的目的，当检测到闭环的时候，希望进行一次全局优化。为了避免极低的计算效率，采用位姿图优化，而不是全局 BA 优化。位姿图优化，顾名思义，只对位姿进行优化，而忽略路标点。这样做的原因是，通常路标点经过多次观测优化后会收敛到一个固定值，而那些发散的异常点会消失不见。对于那些收敛的点，不需一直进行优化。在闭环之前，路标点肯定都经过了多次优化，已经收敛，所以闭环的时候没有必要再次优化，而只需要优化位姿即可。

图优化是由节点和边构成的，节点是待优化的变量，边是约束关系。那么位姿图呢？位姿图的节点表示相机位姿，用李代数 ξ_1, \dots, ξ_n 表示，位姿图的边是位姿间相机的运动，用 $\Delta\xi_{ij}$ 表示位姿 ξ_i 和 ξ_j 之间的相机运动，则，

$$\Delta\xi_{ij} = \xi_i^{-1} \circ \xi_j = \ln(\exp([-\xi_i]_\times) \exp([\xi_j]_\times))^\vee \quad (4.35)$$

或者可以用李群表示为：

$$\Delta\mathbf{T}_{ij} = \mathbf{T}_i^{-1}\mathbf{T}_j \quad (4.36)$$

上式中的“=”不完全成立，将利用这个误差来构建残差项：

$$\begin{aligned} \boldsymbol{e}_{ij} &= \ln (\Delta \mathbf{T}_{ij}^{-1} \mathbf{T}_i^{-1} \mathbf{T}_j)^\vee \\ &= \ln (\exp ([-\boldsymbol{\xi}_{ij}]_\times) \exp ([-\boldsymbol{\xi}_i]_\times) \exp ([\boldsymbol{\xi}_j]_\times))^\vee \end{aligned} \quad (4.37)$$

按照前面研究优化的思路，需要求残差对优化变量的一阶导数，仍旧用扰动求导的方式：

$$\hat{\boldsymbol{e}}_{ij} = \ln (\mathbf{T}_{ij}^{-1} \mathbf{T}_i^{-1} \exp ([-\delta \boldsymbol{\xi}_i]_\times) \exp ([\delta \boldsymbol{\xi}_j]_\times) \mathbf{T}_j)^\vee \quad (4.38)$$

其中， $\delta \boldsymbol{\xi}_i$ 和 $\delta \boldsymbol{\xi}_j$ 为 $\boldsymbol{\xi}_i$ 和 $\boldsymbol{\xi}_j$ 的左扰动。根据李群上的伴随性质，即

$$\exp ([\text{Ad}(\mathbf{T}) \boldsymbol{\xi}]_\times) = \mathbf{T} \exp ([\boldsymbol{\xi}]_\times) \mathbf{T}^{-1} \quad (4.39)$$

其中，

$$\text{Ad}(\mathbf{T}) = \begin{bmatrix} \mathbf{R} & [\mathbf{t}]_\times \mathbf{R} \\ 0 & \mathbf{R} \end{bmatrix}$$

可得，

$$\exp ([\boldsymbol{\xi}]_\times) \mathbf{T} = \mathbf{T} \exp \left([\text{Ad}(\mathbf{T}^{-1}) \boldsymbol{\xi}]_\times \right) \quad (4.40)$$

利用式(4.40)，将扰动移到最右侧，从而推导出右乘雅可比矩阵：

$$\begin{aligned} \hat{\boldsymbol{e}}_{ij} &= \ln \left\{ \mathbf{T}_{ij}^{-1} \mathbf{T}_i^{-1} \exp ([-\delta \boldsymbol{\xi}_i]_\times) \exp ([\delta \boldsymbol{\xi}_j]_\times) \mathbf{T}_j \right\}^\vee \\ &= \ln \left\{ \mathbf{T}_{ij}^{-1} \mathbf{T}_i^{-1} \mathbf{T}_j \exp \left([-\text{Ad}(\mathbf{T}_j^{-1}) \delta \boldsymbol{\xi}_i]_\times \right) \exp \left([\text{Ad}(\mathbf{T}_j^{-1}) \delta \boldsymbol{\xi}_j]_\times \right) \right\}^\vee \\ &\approx \boldsymbol{e}_{ij} + \frac{\partial \boldsymbol{e}_{ij}}{\partial \delta \boldsymbol{\xi}_i} \delta \boldsymbol{\xi}_i + \frac{\partial \boldsymbol{e}_{ij}}{\partial \delta \boldsymbol{\xi}_j} \delta \boldsymbol{\xi}_j \end{aligned} \quad (4.41)$$

其中，

$$\begin{aligned} \frac{\partial \boldsymbol{e}_{ij}}{\partial \delta \boldsymbol{\xi}_i} &= -\mathcal{J}_r^{-1}(\boldsymbol{e}_{ij}) \text{Ad}(\mathbf{T}_j^{-1}) \\ \frac{\partial \boldsymbol{e}_{ij}}{\partial \delta \boldsymbol{\xi}_j} &= \mathcal{J}_r^{-1}(\boldsymbol{e}_{ij}) \text{Ad}(\mathbf{T}_j^{-1}) \end{aligned} \quad (4.42)$$

一般李代数上的雅可比矩阵比较复杂，通常近似取为：

$$\mathcal{J}_r^{-1}(\boldsymbol{e}_{ij}) \approx \mathbf{I} + \frac{1}{2} \begin{bmatrix} [\boldsymbol{\phi}_e]_\times & [\boldsymbol{\rho}_e]_\times \\ \mathbf{0} & [\boldsymbol{\phi}_e]_\times \end{bmatrix} \quad (4.43)$$

到此为止，求的仅仅是位姿图中的一条边的残差，记所有边的集合为，那么所有边的

残差就构成了目标函数：

$$\min_{\xi} \frac{1}{2} \sum_{i,j \in \mathcal{E}} e_{ij}^T \Sigma_{ij}^{-1} e_{ij} \quad (4.44)$$

然后用高斯-牛顿或者 LM 方法迭代求解最优值。

在本系统中，当检测到闭环的时候，进行全局位姿优化。这里的位姿指 3 自由度位置 (x, y, z) 和一个偏航角 $\hat{\psi}_{ij}$ ，总共四自由度。忽略横滚角和俯仰角是因为在单目/视觉惯性融合系统中，这两个角度是可以根据重力的方向观测到的，不会产生累计漂移。

当滑动窗口中最旧的那一帧关键帧滑出窗口时，它将被添加到位姿图中。该关键帧是位姿图的顶点，它通过两条边和其他顶点相连：

(1) 第一条边是滑动窗口中的两个关键帧之间的相对变换（或者叫相对位姿），称为序列边。假设关键帧 i 是最新进行边缘化的帧，其之前的某一个关键帧为 j ，那么序列边为包括 i 和 j 的相对位移 $\hat{\mathbf{p}}_{ij}^i$ 以及偏航角 $\hat{\psi}_{ij}$ ：

$$\begin{aligned} \hat{\mathbf{p}}_{ij}^i &= \hat{\mathbf{R}}_i^{w^{-1}} (\hat{\mathbf{p}}_j^w - \hat{\mathbf{p}}_i^w) \\ \hat{\psi}_{ij} &= \hat{\psi}_j - \hat{\psi}_i. \end{aligned} \quad (4.45)$$

(2) 第二条边是检测到闭环后形成的闭环边。其定义和式 (4.45) 一样，通过重定位获得闭环边的值。

有了顶点和边，就可以得到图优化的残差项及目标函数。定义关键帧 i 和 j 之间的残差为：

$$\mathbf{r}_{i,j}(\mathbf{p}_i^w, \psi_i, \mathbf{p}_j^w, \psi_j) = \begin{bmatrix} \mathbf{R}(\hat{\phi}_i, \hat{\theta}_i, \psi_i)^{-1} (\mathbf{p}_j^w - \mathbf{P}_i^w) - \hat{\mathbf{p}}_{ij}^i \\ \psi_j - \psi_i - \hat{\psi}_{ij} \end{bmatrix} \quad (4.46)$$

其中， $\hat{\phi}_i, \hat{\theta}_i$ 是单目 VIO 估计出的横滚角和俯仰角。

所有的序列边和闭环边的残差项之和构成了整体的目标函数：

$$\min_{\mathbf{p}, \psi} \left\{ \sum_{(i,j) \in \mathcal{S}} \|\mathbf{r}_{i,j}\|^2 + \sum_{(i,j) \in \mathcal{L}} \rho(\|\mathbf{r}_{i,j}\|^2) \right\} \quad (4.47)$$

其中 \mathcal{S} 是所有序列边的集合， \mathcal{L} 是所有闭环边的集合。 $\rho(\cdot)$ 是 Huber 范数。对闭环边使用 Huber 范数是为了减少错误闭环的影响，而对于序列边不使用 Huber 范数，因为序列边是 VIO 给出的，VIO 已经行了“异常点”剔除。

位姿图优化和重新定位将异步运行在两个独立的线程中，以便在进行重定位时，能立即使用最优化的位姿图。同样，即使当前的姿态图优化尚未完成，仍然可以使用

现有的姿态图进行重定位。这一过程如图4.9所示。

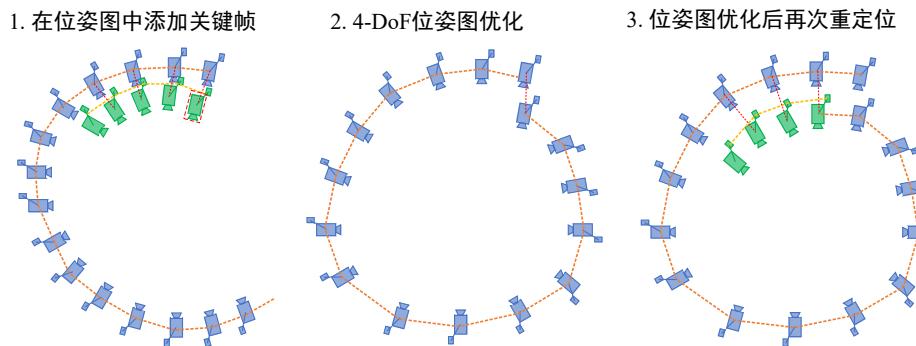


图 4.9 位姿优化示意图

4.4 本章小结

本章对系统的后端优化和闭环检测算法进行了详细的阐述。首先研究了后端优化的优化变量、目标函数以及构成目标函数的各个残差项，并解释了边缘化过程。然后研究了基于词袋模型的闭环检测算法。最后分析了重定位的实现过程，以及基于图优化的闭环优化算法。本章研究的优化算法体现了视觉和 IMU 之间的紧耦合过程，二者的状态量放在一起进行优化，紧密结合，相辅相成。

第 5 章 实验

在本章中，主要研究实验方案，以及分析实验结果。首先使用公开数据集 EuRoc 进行实验验证，EuRoc 是室内飞行数据集，提供 6-DoF 真值。然后使用本系统进行在线实时验证，本系统放在车顶，围绕北京理工大学中关村校区进行车载实验验证。EuRoc 数据集是室内环境，飞行路程短，包含 6-DoF 位姿真值。本地实验是室外环境，汽车行程远，包含 2-DoF 位置真值。

5.1 EuRoc 数据集实验

5.1.1 EuRoc 构成及硬件平台

EuRoc 飞行数据集是苏黎世联邦理工学院自动系统实验室于 2016 年发布的公开数据集^[77]，主要用于视觉 SLAM 和视觉/惯性融合的研究上。数据集中包含如下可用数据：

(1) 视觉/惯性数据

- 双目图像。使用 Aptina MT9V034 全局快门相机拍摄，单色图像，帧率为 20FPS；
- IMU 数据。使用 ADIS16448 测量角速率和加速度，采集频率为 200 Hz；
- 对齐的时间戳。

(2) 地面真值数据

- 6-DoF 位姿真值。通过 Vicon 系统得到；
- 3-DoF 位置真值。通过 Leica MS50 激光跟踪仪测量得到；
- 3D 空间结构。通过徕卡 MS50 3D 结构扫描仪器扫描得到。

(3) 标定数据

- 相机内参；
- 相机-IMU 外参；
- 时空对齐的地面真值。

如图5.1所示是 EuRoc 的硬件平台。图5.1 (a) 是无人机，图5.1 (b) 是无人机上搭载的视觉/惯性传感器。

使用 EuRoc 数据集中的 MH_01_easy 和 MH_05_difficult 两组数据，这两组数据是在机器大厅采集的，代表了“容易”和“困难”两种环境。图5.2（a）是机器大厅环境图，图5.2（b）是传感器和地面真值测量仪器之间的位置示意图。

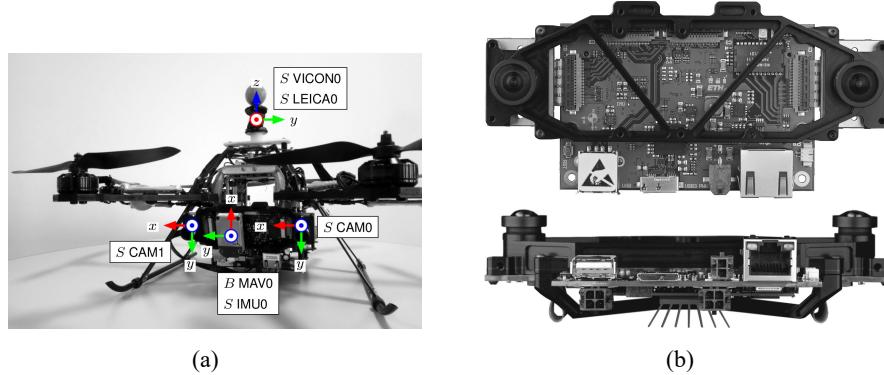


图 5.1 EuRoc 硬件平台

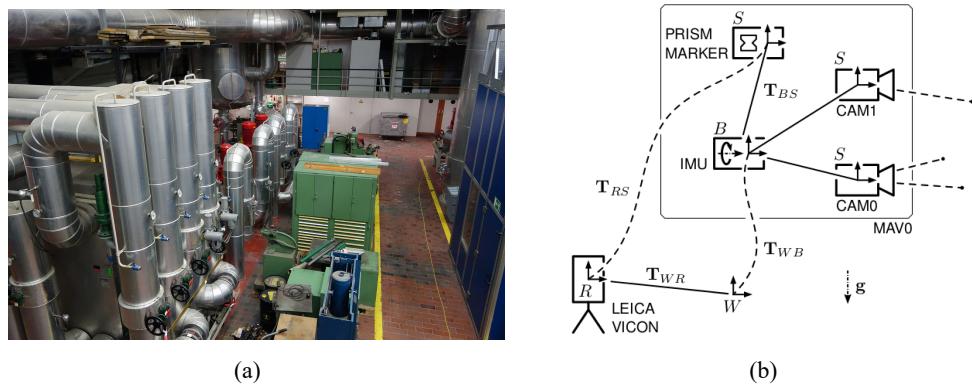


图 5.2 (a) 机器大厅环境；(b) 传感器和地面真值测量仪器之间的位置示意图

5.1.2 EuRoc 实验一

首先使用 MH_01_easy 进行实验。图5.3是三维定位轨迹图对比图，图5.4是三轴位置（ x , y , z ）对比图，图5.5是三轴姿态（roll, pitch, yaw）对比图。其中 VI_SLAM 是本系统输出轨迹，GroundTruth 是数据集中的真值轨迹。通过图5.3，图5.4和图5.5，可以定性分析系统的定位误差。

然后定量分析 EuRoc 实验一中系统的定位误差。图5.6是三维定位绝对误差图。计算出系统定位的绝对误差（APE），均方根误差（RMSE），标准差（STD），平均误差（MEAN）和中值误差（MEDIAN），将计算结果展示在表5.1中。为了更加直观，将这些误差以曲线图的形式展示出来，如图5.7所示。

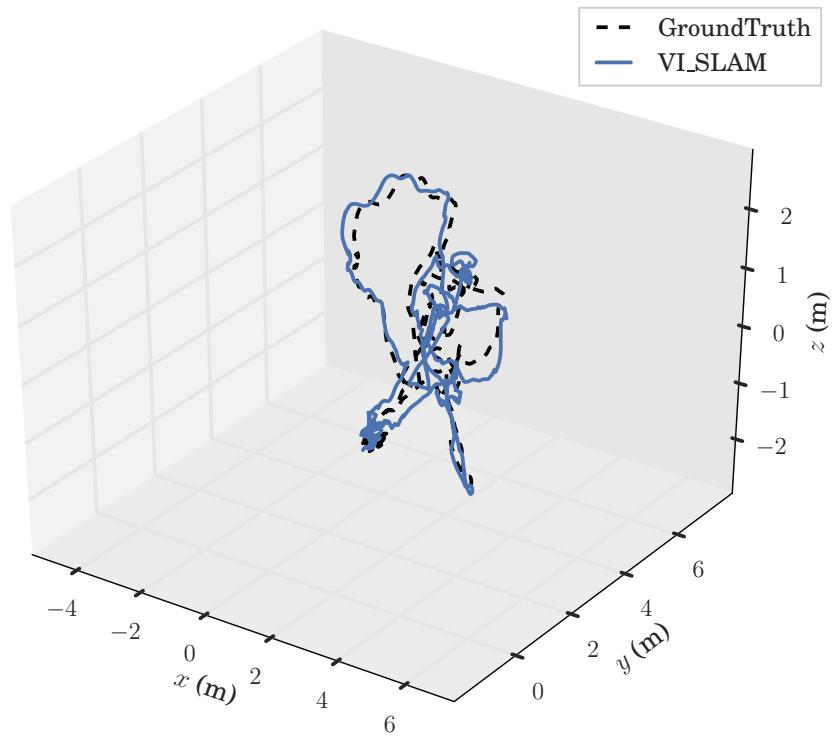


图 5.3 EuRoc 实验一的三维定位轨迹图

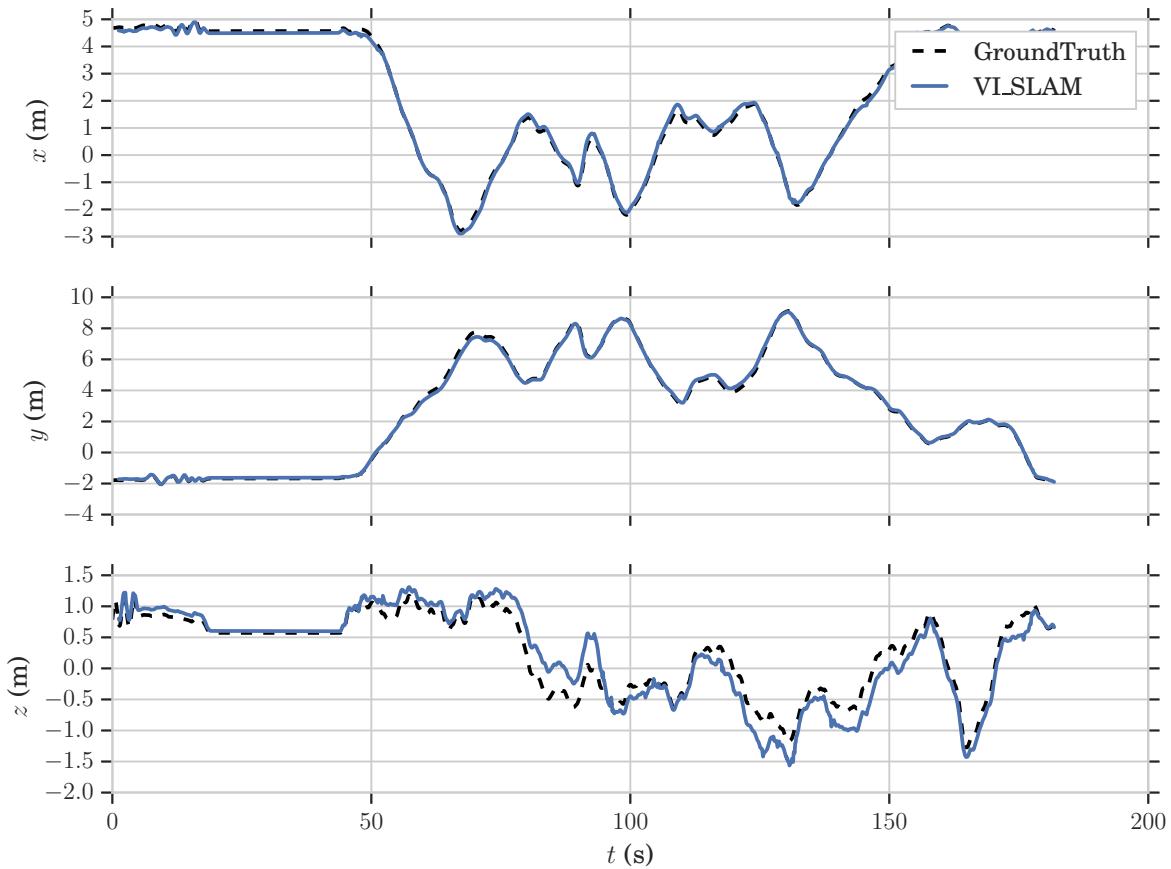


图 5.4 EuRoc 实验一的三轴位置 (x , y , z) 对比图

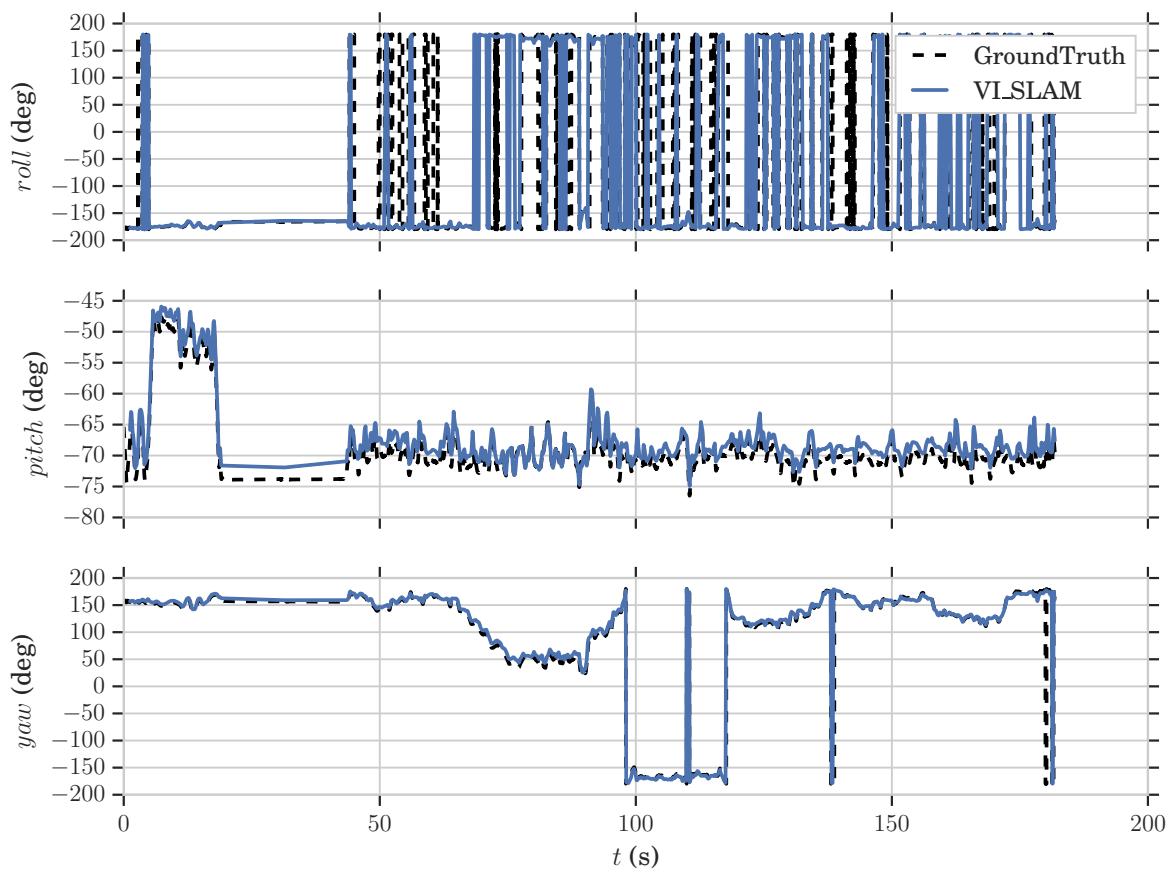


图 5.5 EuRoC 实验一的三轴姿态 (roll, pitch, yaw) 对比图

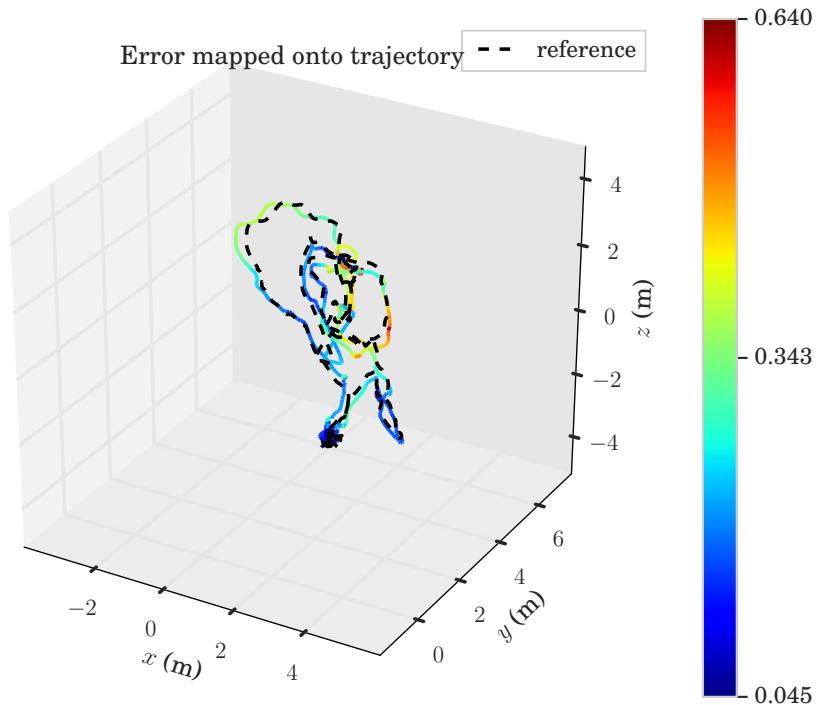


图 5.6 EuRoC 实验一的三维定位绝对误差图

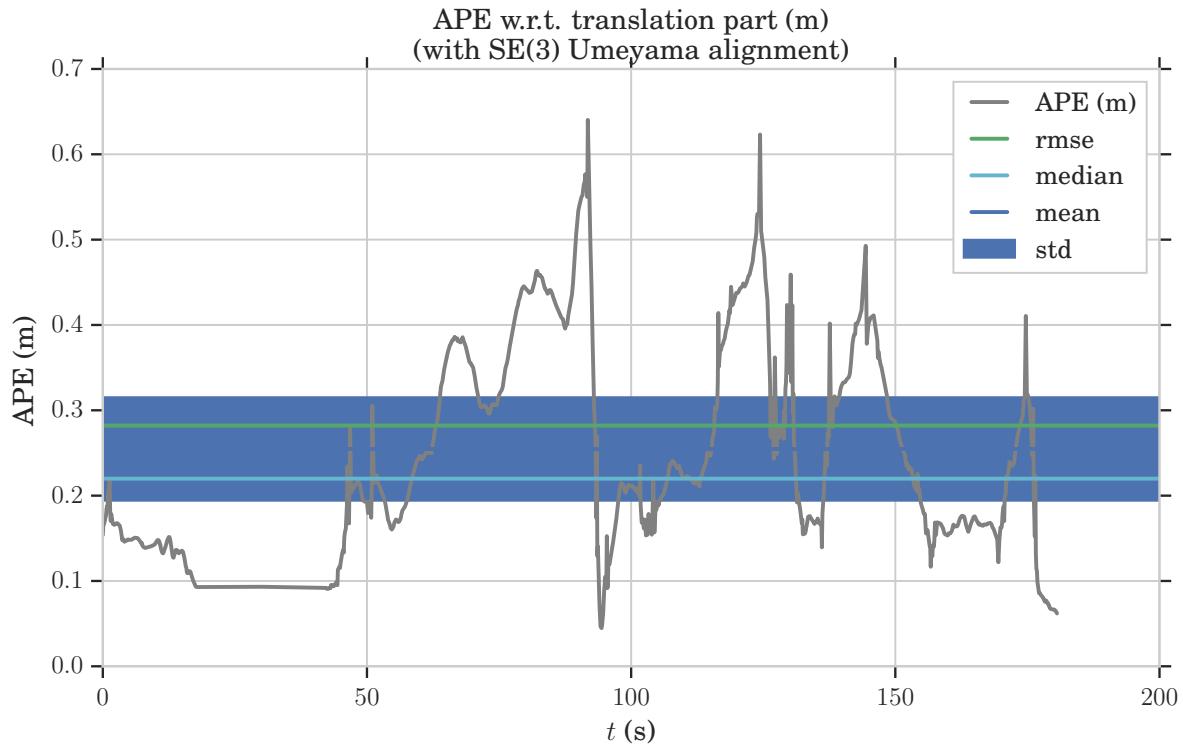


图 5.7 EuRoc 实验一的系统定位误差指标图

表 5.1 EuRoc 实验一的系统定位误差指标

APE_MAX(m)	APE_MIN(m)	RMSE	STD	MEAN(m)	MEDIAN(m)
0.6403	0.0447	0.2820	0.1212	0.2546	0.2199

对比其他开源 VIO 方案在 MH_01_easy 上的误差的 RMSE^[78]，MSCKF 为 0.43，OKVIS 为 0.16，ROVIO 为 0.21，可见本系统在 MH_01_easy 上的定位精度介于 MSCKF 和 ROVIO 之间。

5.1.3 EuRoc 实验二

使用 MH_05_difficult 数据集进行实验。图5.8是三维定位轨迹图对比图，图5.9是三轴位置(x, y, z)对比图，图5.10是三轴姿态(roll, pitch, yaw)对比图。其中 VI_SLAM 是本系统输出轨迹，GroundTruth 是数据集中的真值轨迹。通过图5.8，图5.9和图5.10，可以定性分析系统的定位误差。

然后定量分析 EuRoc 实验二中系统的定位误差。图5.11是三维定位绝对误差图。同样，计算出 APE，RMSE，STD，MEAN 和 MEDIAN，将计算结果展示在表5.2中。为了更加直观，将这些误差以曲线图的形式展示出来，如图5.12所示。

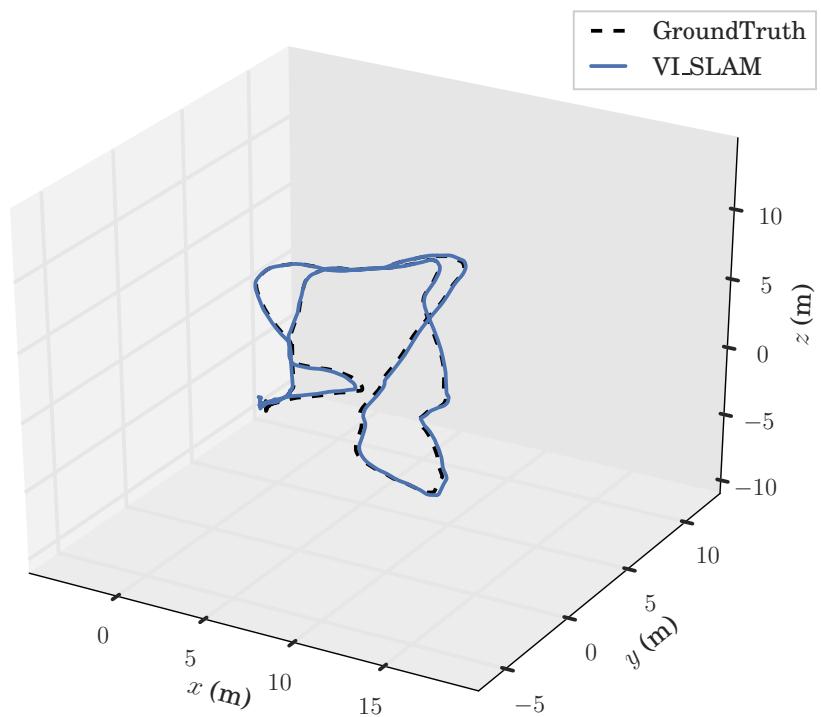


图 5.8 EuRoc 实验二的三维定位轨迹图

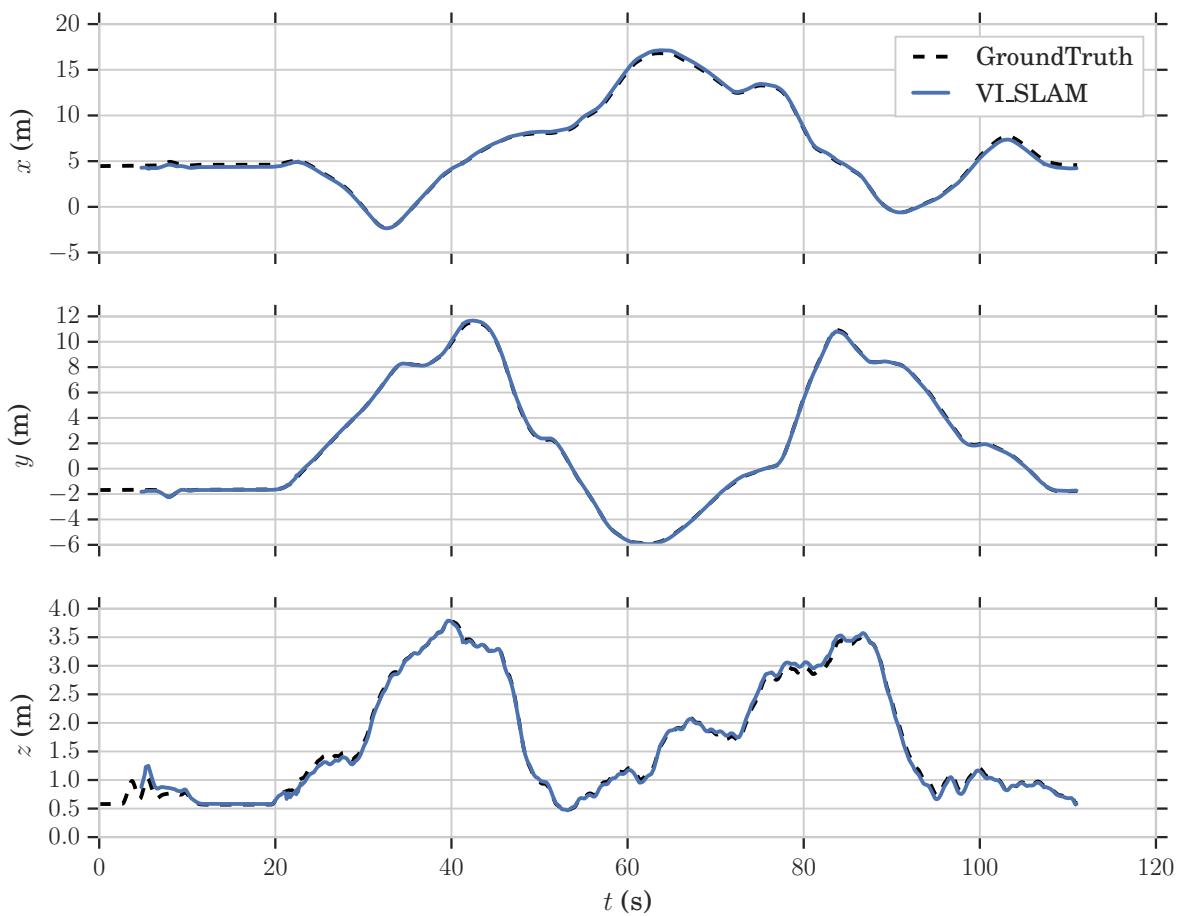


图 5.9 EuRoc 实验二的三轴位置 (x , y , z) 对比图

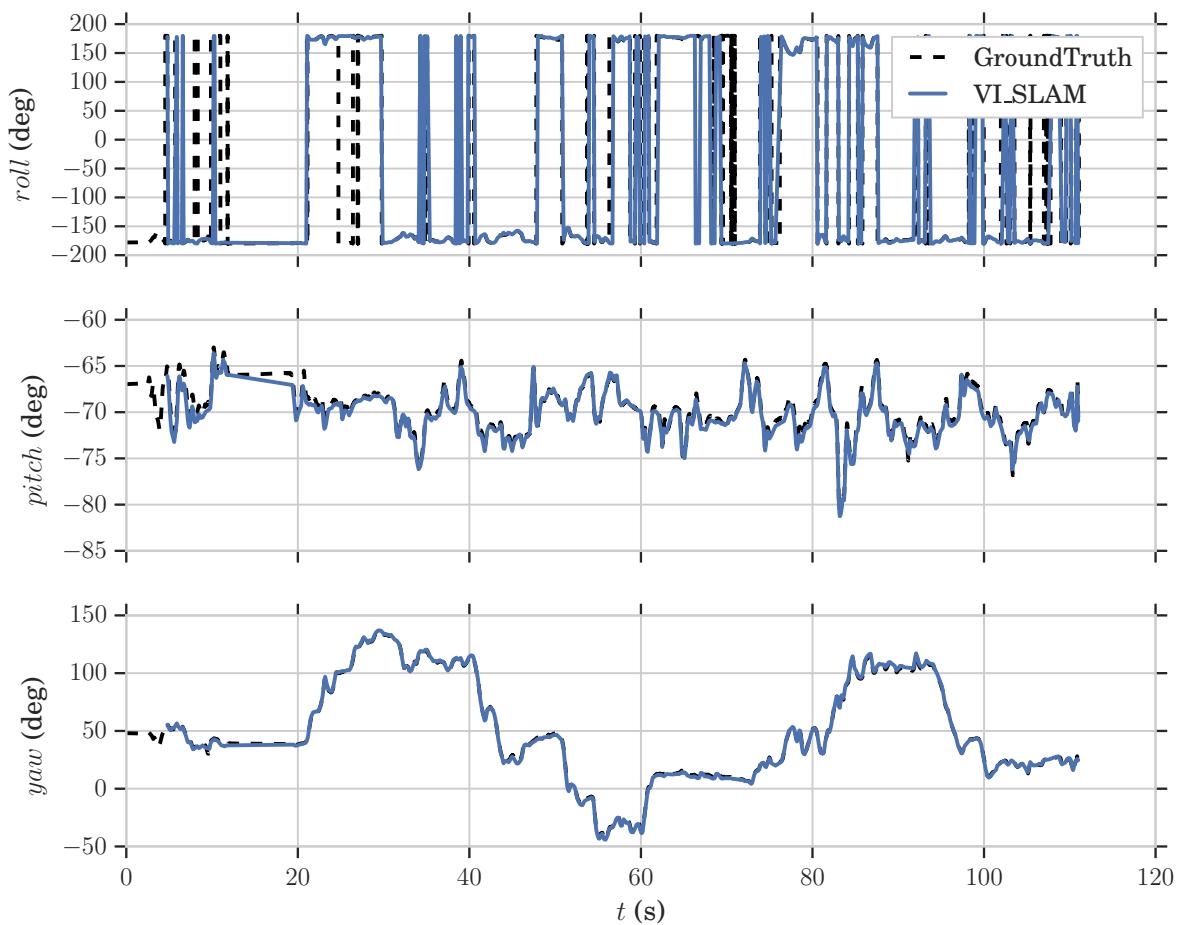


图 5.10 EuRoC 实验二的三轴姿态 (roll, pitch, yaw) 对比图

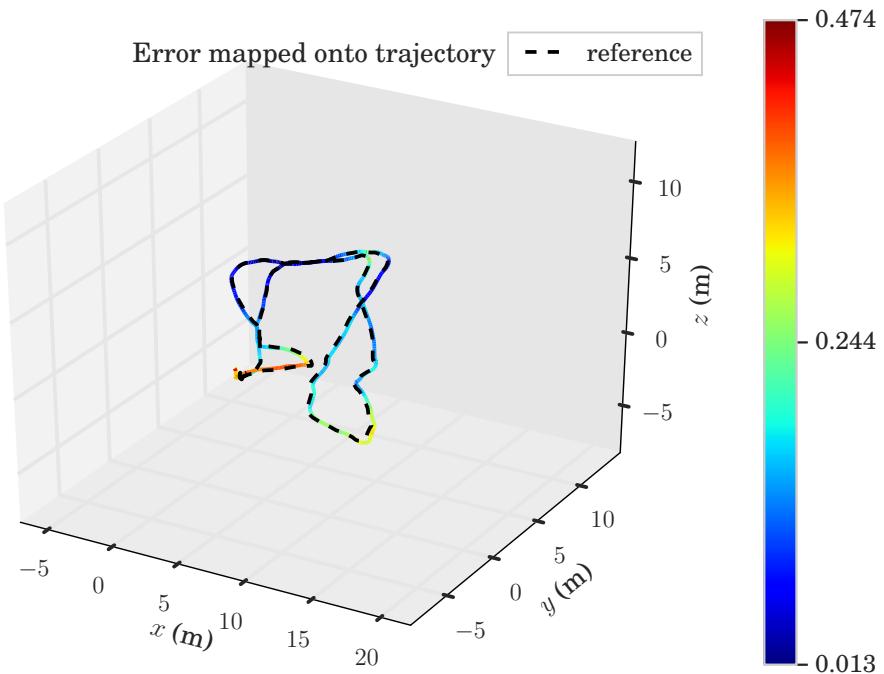


图 5.11 EuRoC 实验二的三维定位绝对误差图

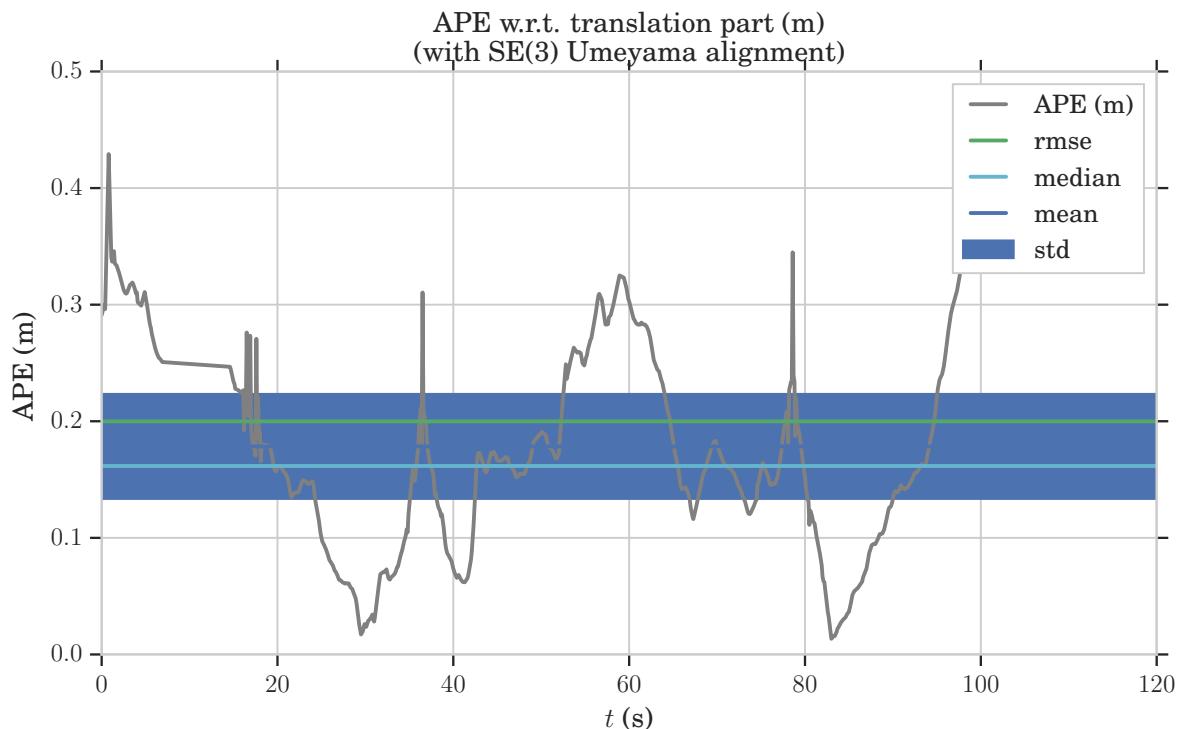


图 5.12 EuRoc 实验二的系统定位误差指标图

表 5.2 EuRoc 实验二的系统定位误差指标

APE_MAX(m)	APE_MIN(m)	RMSE	STD	MEAN(m)	MEDIAN(m)
0.4739	0.0134	0.1999	0.0901	0.1784	0.1616

对比其他开源 VIO 方案在 MH_05_difficult 上的误差的 RMSE^[78]，MSCKF 为 0.48，OKVIS 为 0.56，ROVIO 为 0.52，可见本系统在 MH_01_difficult 上的定位精度要远远高于其他开源方案。这也说明了本系统更加适应复杂环境，鲁棒性更高。

5.2 本地实时实验

5.2.1 本地实验硬件及环境

使用实验室的丰田 SUV 进行车载实验，如图5.13（a）所示。图5.13（b）是搭载在车顶上的多传感器硬件系统，其具体构成已经在 2.3.1 小节中介绍。

本地车载实验的路线有两个，一个是环绕半个北京理工大学中关村校区，大约 2.8km，如图5.14黄色 GPS 轨迹线所示。另一个是环绕整个北京理工大学中关村校区一周，大约 4.3km，如图5.14红色 GPS 轨迹线所示。由于是车载实验，所以本系统只关注 2 自由度（x, y）位置，忽略高度。本系统使用 RTK 采集经纬度信息，经过高斯

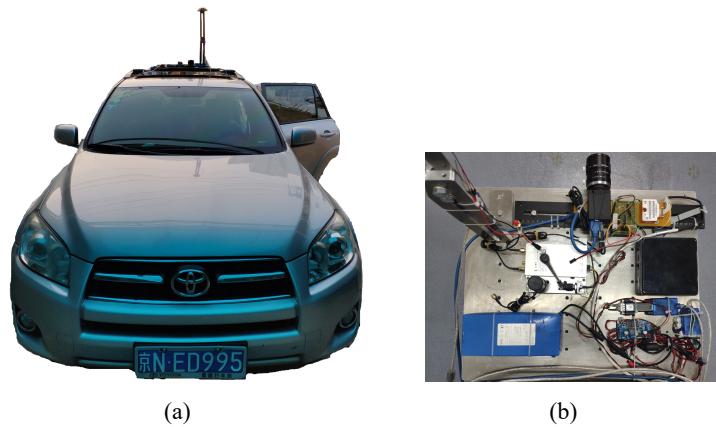


图 5.13 本地实验设备



图 5.14 车载实验路线

投影后转化为 WGS-84 坐标，用这个坐标点轨迹作为真值数据。所以，本系统的真值数据也只有 2 自由度位置信息，没有姿态真值。

5.2.2 本地实验一

实验一的行车轨迹如图5.14黄线所示。图5.15是二维定位轨迹图对比图，图5.16是二轴位置（ x, y ）对比图。其中 VI_SLAM 是本系统输出轨迹，RTK 是差分 GPS 轨迹。通过图5.15和图5.16，可以定性分析系统的定位误差。

然后定量分析本地实验一中系统的定位误差。图5.17是二维定位绝对误差图。同样，计算出 APE，RMSE，STD，MEAN 和 MEDIAN，将计算结果展示在表5.3中。为了更加直观，将这些误差以曲线图的形式展示出来，如图5.18所示。

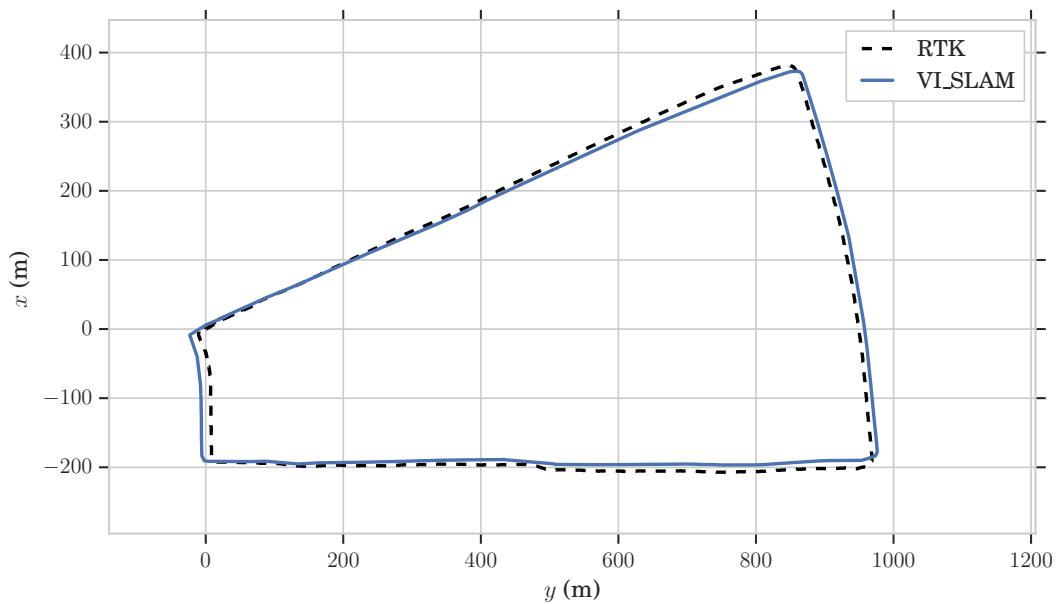


图 5.15 本地实验一的二维定位轨迹图对比图

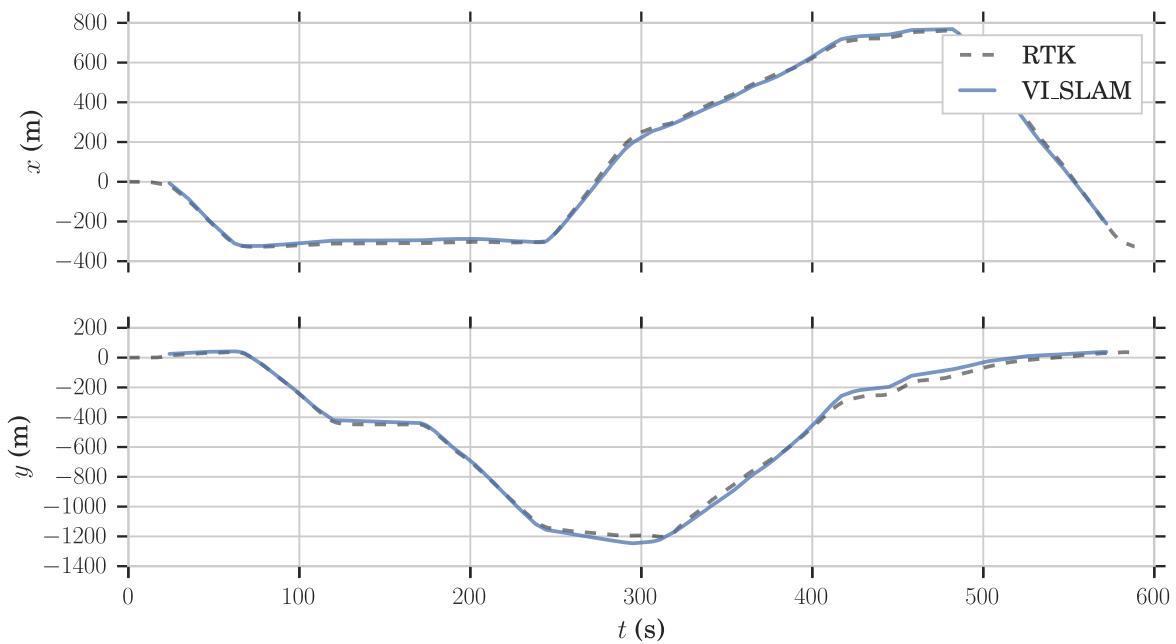


图 5.16 本地实验一的二轴位置 (x, y) 对比图

表 5.3 本地实验一的系统定位误差指标

APE_MAX(m)	APE_MIN(m)	RMSE	STD	MEAN(m)	MEDIAN(m)
19.5912	5.9771	13.0965	2.5359	12.8487	12.6319

可见，本地实验一的最大定位误差是 19.5912 米，定位精度 $\leq 0.6\%$ 。

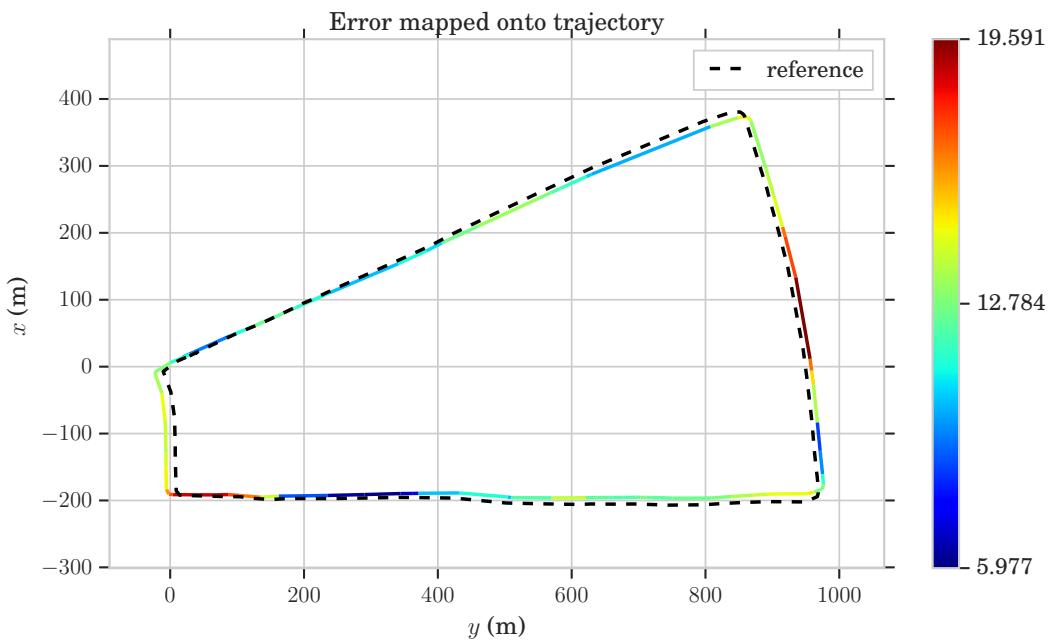


图 5.17 本地实验一的二维定位绝对误差图
APE w.r.t. translation part (m)
(with SE(3) Umeyama alignment)

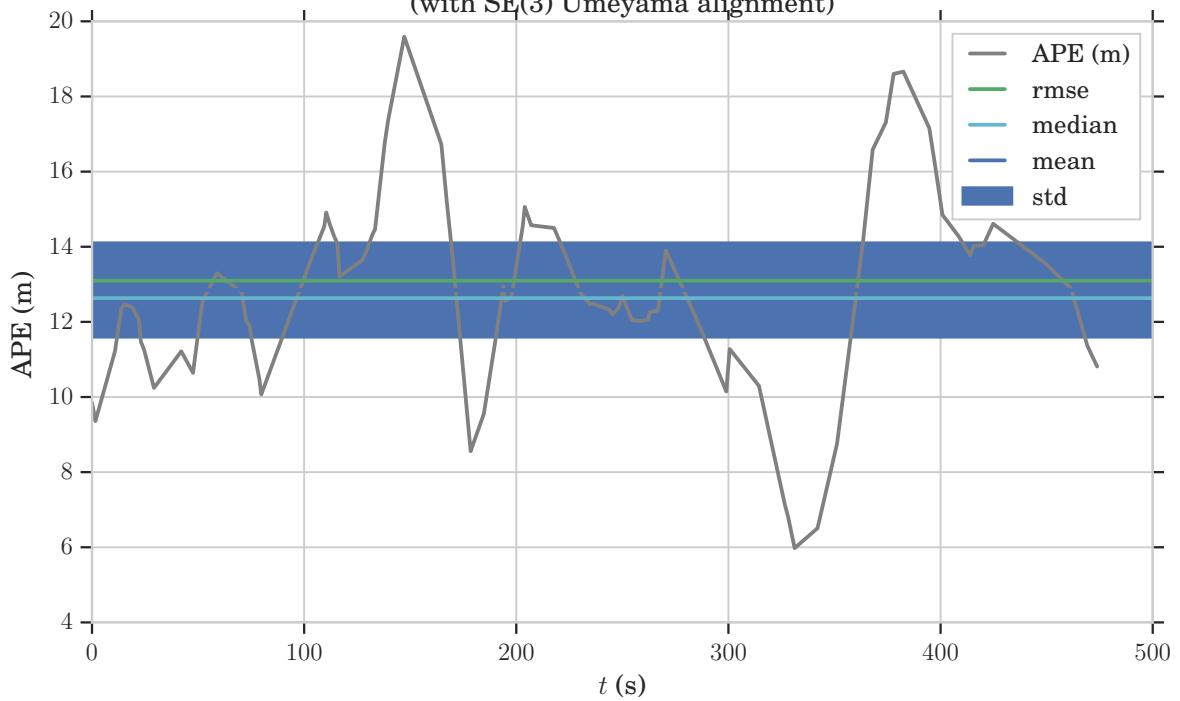


图 5.18 本地实验一的系统定位误差指标图

5.2.3 本地实验二

实验二的行车轨迹如图5.14红线所示。图5.19是二维定位轨迹图对比图，图5.20是二轴位置(x, y)对比图。其中 VI_SLAM 是本系统输出轨迹，RTK 是差分 GPS 轨迹。通过图5.19和图5.20，可以定性分析系统的定位误差。

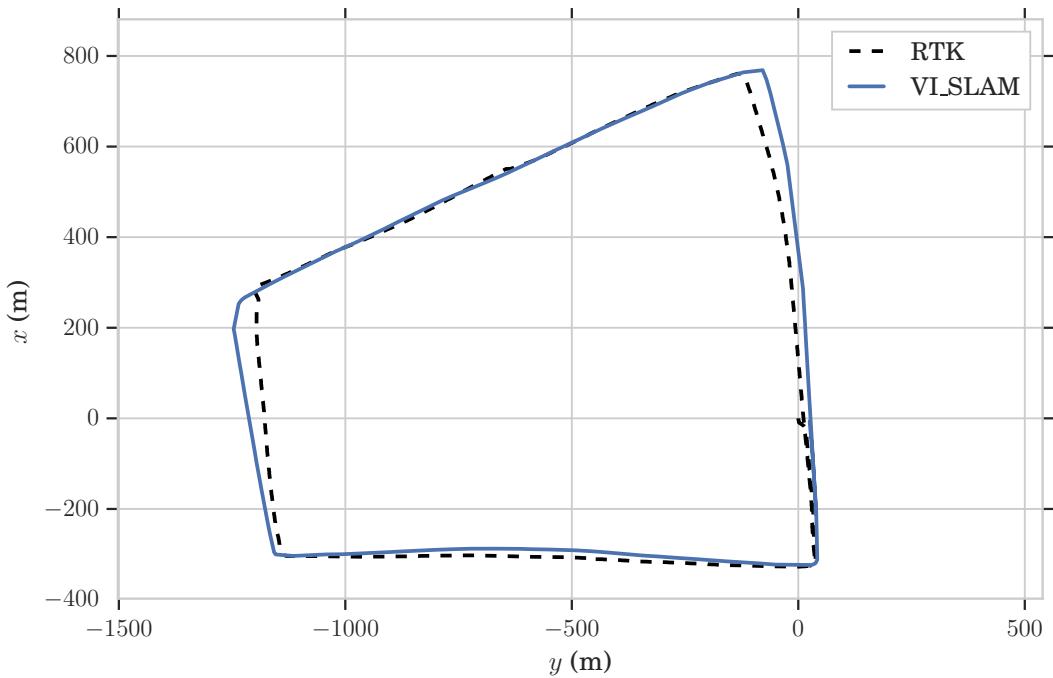


图 5.19 本地实验二的二维定位轨迹图对比图

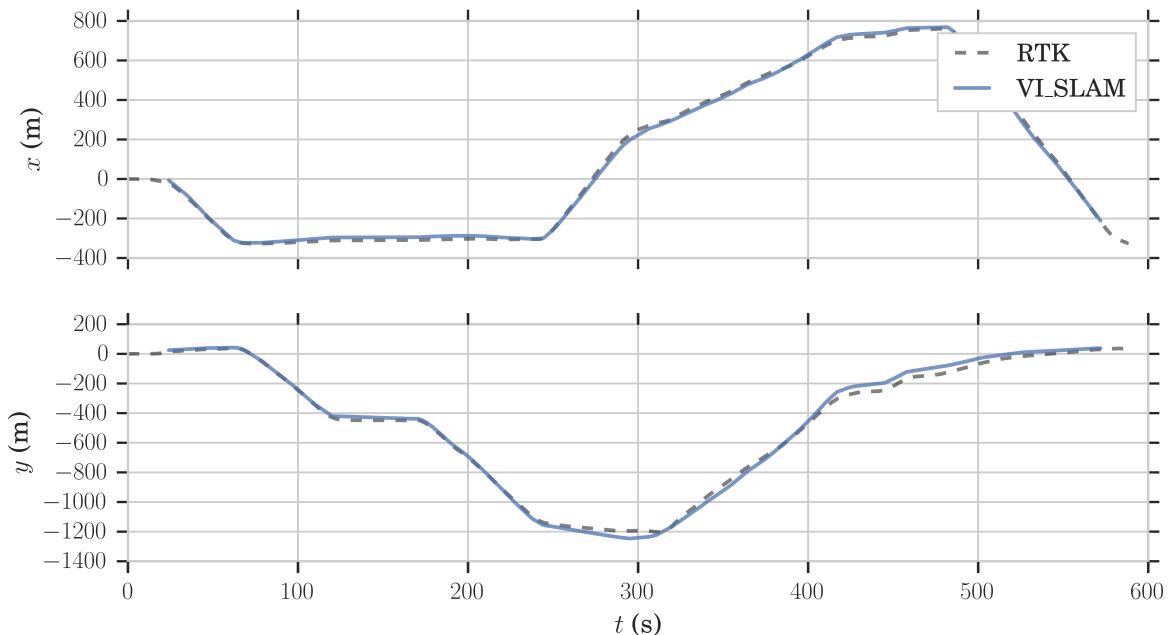


图 5.20 本地实验二的二维位置(x, y)对比图

然后定量分析本地实验二中系统的定位误差。图5.21是二维定位绝对误差图。同样，计算出 APE，RMSE，STD，MEAN 和 MEDIAN，将计算结果展示在表5.4中。为了更加直观，将这些误差以曲线图的形式展示出来，如图5.22所示。

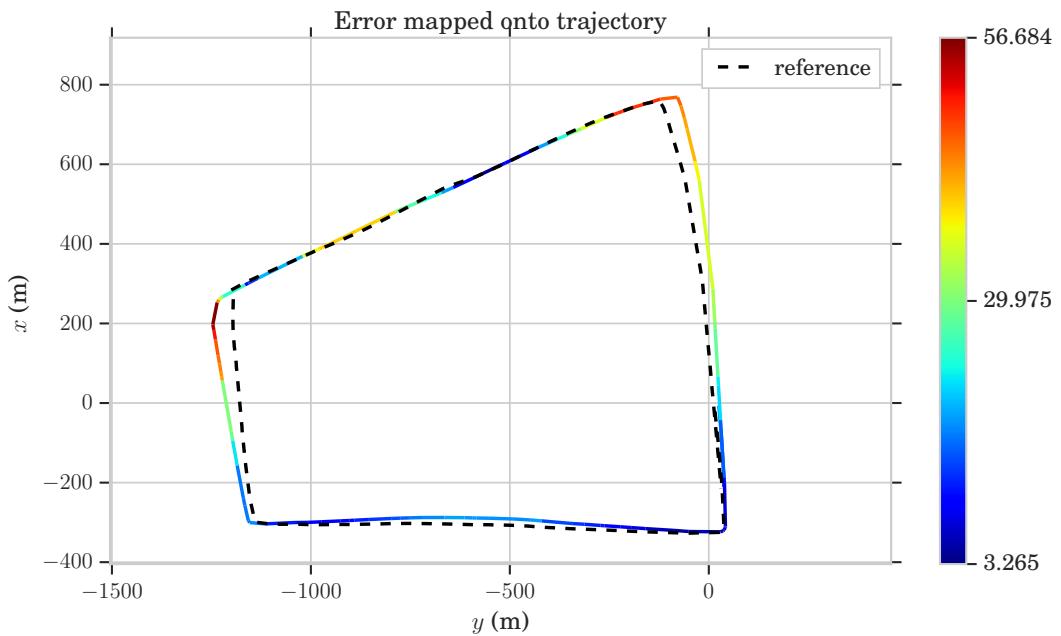


图 5.21 本地实验二的二维定位绝对误差图

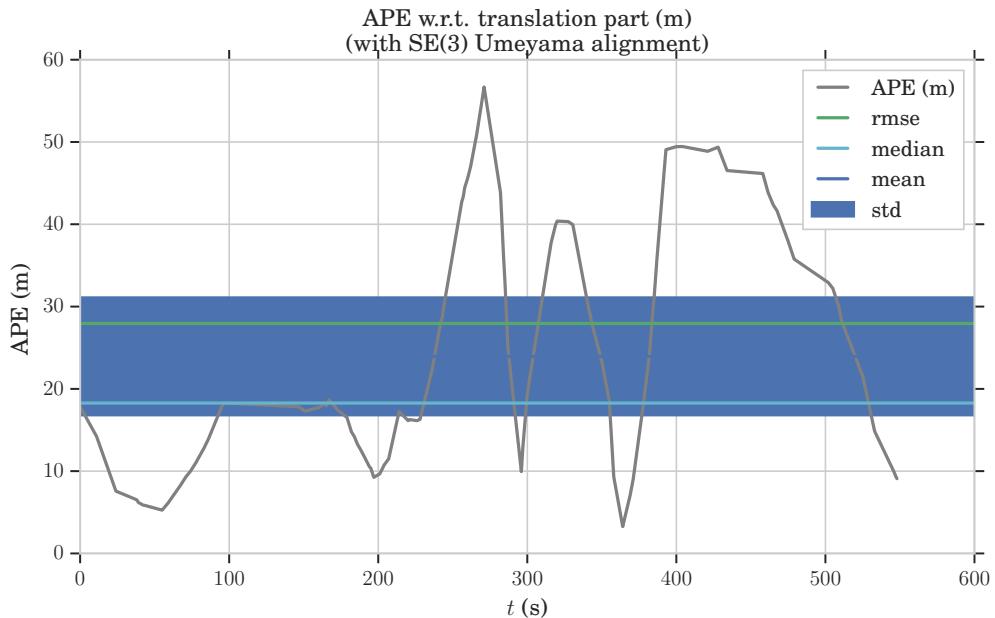


图 5.22 本地实验二的系统定位误差指标图

表 5.4 本地实验二的系统定位误差指标

APE_MAX(m)	APE_MIN(m)	RMSE	STD	MEAN(m)	MEDIAN(m)
56.6838	3.2654	27.9445	14.3974	23.9501	18.2750

可见，本地实验二的最大定位误差是 56.6838 米，定位精度 $\leq 1.3\%$ 。相比于本地实验一，定位精度有所下降。

5.3 本章小结

本章通过实验验证了本系统的定位效果。首先使用 EuRoc 数据集，验证了本系统在三维空间中的定位效果。然后进行本地校园车载实验，验证了本系统在大地平面上的定位效果。其中，EuRoc 是室内环境，飞行距离短，环境简单。校园车载实验是室外环境，行驶路线长，环境复杂多变。两种环境下的实验数据说明本系统在不同的环境下都能正常工作，在短距离情况下定位精度较高，但是随着距离的增加，绝对定位精度会逐渐下降。

结论

如今，飞速发展人工智能（AI）正在各个领域展露锋芒，AI 工作者期望利用 AI 改善人类生活。移动机器人、无人车和无人机等领域也属于人工智能的一部分。无人载体要实现无人驾驶的目标，通常需要具有自主定位、地图构建、目标识别、智能避障和路径规划等功能。单一传感器不能完成所有的这些功能，所以无人设备上往往搭载多种传感器，以便弥补单一传感器的不足之处。其中，Visual 和 IMU 一般情况下是无人载体上的必备传感器。Visual 和 IMU 在功能上有很多的互补性，将二者融合能够获得高精度的位姿估计。本文正是以视觉和 IMU 为对象，总结国内外视觉 SLAM 以及 VIO 的研究现状，深入研究 Visual-IMU 的融合算法，同时搭建硬件设备，将硬件设备和软件算法结合起来构建一套多传感融合定位系统。

本文首先详细研究了传感器的数学模型，以及如何选择相机和 IMU。然后对相机和 IMU 的标定原理和方法进行了分析总结，并给出了标定结果。同时研究了硬件同步的方法。之后研究了基于 ROS 的传感器信息采集方法。然后进入算法部分，首先研究前端算法，包括图像特征的提取与跟踪，特征的异常点剔除方法，IMU 预积分理论。然后研究初始化算法，包括对极几何、三角测量、PnP 等理论基础，纯视觉初始化流程，以及视觉/IMU 联合初始化方法。前端和初始化为后端非线性优化提供初始值，是后端优化的基础。

本文的后续部分对后端优化和闭环进行了深入研究。首先研究了紧耦合非线性滑窗优化方法，定义了优化的状态量以及目标函数，并对各个残差项进行详细的分析。接着为了使滑窗策略不丢失历史信息，研究了边缘化理论，并研究了本系统的边缘化方法。然后本文研究了闭环检测算法，对词袋模型和字典构建进行了细致的研究。最后，针对检测到的闭环帧，研究了如何对闭环帧进行重定位，如何在检测到闭环后对所有的关键帧进行位姿优化，获得全局一致的定位轨迹。

在文章的最后，针对本系统做了两组实验。一组是公开数据集 EuRoc 实验，另一组是本地校园车载实时实验。实验环境从简单到复杂，从短距离到长距离。实验数据表明了本系统的可靠性和有效性。

然而本系统仍旧有一些不足之处，故提出以下需要改进的地方：

1. 在硬件方面，本系统的硬件同步做的还不完善，目前 IMU 和相机同步的时间

差在毫秒级，而数据集可以达到微秒级。原因在于本系统的硬件同步电路仅仅采用了一片 CD4017 计数器，没有添加复杂的滤波电路，可能导致触发脉冲的精度不够高和响应时间不够短。

2. 算法方面，本系统算法复杂，代码繁多，计算量大，目前只能在 Intel NUC 或者个人 PC 上进行计算，无法移植到小型的嵌入式设备。后续研究需要对算法和代码进行优化，降低整体计算量。

3. 对于磁力计，鉴于磁力计容易受外界环境干扰，测量数据不稳定，外参标定复杂的情况，本系统仅仅使用磁力计进行航向初始化，并未在后端优化中使用。后续将研究如何对磁力计进行动态补偿、滤波，以获得稳定可用的数据，并将其融合在后端优化中。

参考文献

- [1] GU Z, LIU H. A survey of monocular simultaneous localization and mapping[J]. CAAI Transactions on Intelligent Systems, 2015, 10(4): 499–507.
- [2] Davison A J, Reid I D, Molton N D, et al. Monoslam: Real-time single camera slam[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2007(6): 1052–1067.
- [3] Barfoot T D. State estimation for robotics Cambridge University Press, 2017.
- [4] Smith R C, Cheeseman P. On the representation and estimation of spatial uncertainty[J]. The international journal of Robotics Research, 1986, 5(4): 56–68.
- [5] Durrant-Whyte H F. Uncertain geometry in robotics[J]. IEEE Journal on Robotics and Automation, 1988, 4(1): 23–31.
- [6] Smith R, Self M, Cheeseman P. Estimating uncertain spatial relationships in robotics[M]. Springer, 1990: 167–193.
- [7] Durrant-Whyte H, Rye D, Nebot E. Localization of autonomous guided vehicles[M]. Springer, 1996: 613–625.
- [8] Thrun S, Burgard W, Fox D. A probabilistic approach to concurrent mapping and localization for mobile robots[J]. Autonomous Robots, 1998, 5(3-4): 253–271.
- [9] Davison A J. Real-time simultaneous localisation and mapping with a single camera , 2003: 1403–1410.
- [10] Shi J, Tomasi C. Good features to track [M], 1994.
- [11] Klein G, Murray D. Parallel tracking and mapping for small ar workspaces[C]. IEEE Computer Society, 2007: 1–10.
- [12] Klein G, Murray D. Parallel tracking and mapping on a camera phone[C]. IEEE, 2009: 83–86.
- [13] 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.
- [14] Lowe D G. Distinctive image features from scale-invariant keypoints[J]. International journal of computer vision, 2004, 60(2): 91–110.
- [15] Bay H, Ess A, Tuytelaars T, et al. Speeded-up robust features (surf)[J]. Computer vision and image understanding, 2008, 110(3): 346–359.
- [16] Engel J, Schöps T, Cremers D. Lsd-slam: Large-scale direct monocular slam[C]. Springer, 2014: 834–849.

- [17] Engel J, Sturm J, Cremers D. Semi-dense visual odometry for a monocular camera[C]. Proceedings of the IEEE international conference on computer vision, 2013: 1449–1456.
- [18] Forster C, Pizzoli M, Scaramuzza D. Svo: Fast semi-direct monocular visual odometry[C]. IEEE, 2014: 15–22.
- [19] 权美香, 朴松昊, 李国. 视觉 slam 综述[M]. 北京: 智能系统学报, 2016.
- [20] 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(6): 855–868.
- [21] 何俊学, 李战明. 基于视觉的同时定位与地图构建方法综述[J]. 計算機應用研究, 2010, 27(8): 2839–2844.
- [22] Zhou H, Zou D, Pei L, et al. Structslam: Visual slam with building structure lines[J]. IEEE Transactions on Vehicular Technology, 2015, 64(4): 1364–1375.
- [23] Zou D, Tan P. Coslam: Collaborative visual slam in dynamic environments[J]. IEEE transactions on pattern analysis and machine intelligence, 2013, 35(2): 354–366.
- [24] Tan W, Liu H, Dong Z, et al. Robust monocular slam in dynamic environments[C]. IEEE, 2013: 209–218.
- [25] Fischler M A, Bolles R C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography[J]. Communications of the ACM, 1981, 24(6): 381–395.
- [26] Liu H, Zhang G, Bao H. Robust keyframe-based monocular slam for augmented reality[C]. IEEE, 2016: 1–10.
- [27] Liu H, Li C, Chen G, et al. Robust keyframe-based dense slam with an rgb-d camera[J]. arXiv preprint arXiv:1711.05166, 2017.
- [28] Martinelli A. Closed-form solution of visual-inertial structure from motion[J]. International journal of computer vision, 2014, 106(2): 138–152.
- [29] Mourikis A I, Roumeliotis S I. A multi-state constraint kalman filter for vision-aided inertial navigation[C]. IEEE, 2007: 3565–3572.
- [30] Li M, Mourikis A I. High-precision, consistent ekf-based visual-inertial odometry[J]. The International Journal of Robotics Research, 2013, 32(6): 690–711.
- [31] Bloesch M, Omari S, Hutter M, et al. Robust visual inertial odometry using a direct ekf-based approach[C]. IEEE, 2015: 298–304.
- [32] Lynen S, Achtelik M W, Weiss S, et al. A robust and modular multi-sensor fusion approach applied to mav navigation[C]. IEEE, 2013: 3923–3929.

- [33] Leutenegger S, Furgale P, Rabaud V, et al. Keyframe-based visual-inertial slam using nonlinear optimization [M].
- [34] Leutenegger S, Chli M, Siegwart R. Brisk: Binary robust invariant scalable keypoints[C]. Ieee, 2011: 2548–2555.
- [35] Smith S M. Fast robust automated brain extraction[J]. Human brain mapping, 2002, 17(3): 143–155.
- [36] Qin T, Li P, Shen S. Vins-mono: A robust and versatile monocular visual-inertial state estimator[J]. IEEE Transactions on Robotics, 2018, 34(4): 1004–1020.
- [37] Falquez J M, Kasper M, Sibley G. Inertial aided dense & semi-dense methods for robust direct visual odometry[C]. IEEE, 2016: 3601–3607.
- [38] Sola J. Quaternion kinematics for the error-state kalman filter[J]. arXiv preprint arXiv:1711.02508, 2017.
- [39] Orey S, Pruitt W E. Sample functions of the n-parameter wiener process[J]. The Annals of Probability, 1973: 138–163.
- [40] Butala M D, Kazerooni M, Makela J J, et al. Modeling geomagnetically induced currents from magnetometer measurements: Spatial scale assessed with reference measurements[J]. Space Weather, 2017, 15(10): 1357–1372.
- [41] Zhang Z. A flexible new technique for camera calibration[J]. IEEE Transactions on pattern analysis and machine intelligence, 2000, 22.
- [42] Moré J J. The levenberg-marquardt algorithm: implementation and theory[M]. Springer, 1978: 105–116.
- [43] Vagner M, Beneš P, Havránek Z. Experience with allan variance method for mems gyroscope performance characterization[C]. IEEE, 2012: 1343–1347.
- [44] Furgale P, Barfoot T D, Sibley G. Continuous-time batch estimation using temporal basis functions [C]. IEEE, 2012: 2088–2095.
- [45] Furgale P, Rehder J, Siegwart R. Unified temporal and spatial calibration for multi-sensor systems [C]. IEEE, 2013: 1280–1286.
- [46] Fairchild C, Harman T L. Ros robotics by example [M], 2016.
- [47] 高翔. 视觉 slam 十四讲: 从理论到实践[M]. 北京: 电子工业出版社, 2017.
- [48] Harris C G, Stephens M, et al. A combined corner and edge detector.[C]. Citeseer, 1988: 10–5244.
- [49] Rosten E, Drummond T. Machine learning for high-speed corner detection[C]. Springer, 2006: 430–443.
- [50] Blaschko M B. Branch and bound strategies for non-maximal suppression in object detection[C].

Springer, 2011: 385–398.

- [51] Bouguet J Y, et al. Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm[J]. Intel Corporation, 2001, 5(1-10): 4.
- [52] Fischler M A, Bolles R C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography[J]. Communications of the ACM, 1981, 24(6): 381–395.
- [53] Forster C, Carlone L, Dellaert F, et al. Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation Georgia Institute of Technology, 2015.
- [54] Shen S, Michael N, Kumar V. Tightly-coupled monocular visual-inertial fusion for autonomous flight of rotorcraft mavs[C]. IEEE, 2015: 5303–5310.
- [55] Nistér D. An efficient solution to the five-point relative pose problem[J]. IEEE transactions on pattern analysis and machine intelligence, 2004, 26(6): 0756–777.
- [56] Hartley R, Zisserman A. Multiple view geometry in computer vision Cambridge university press, 2003.
- [57] Hartley R I. In defense of the eight-point algorithm[J]. IEEE Pami, 1997, 19(6): 580–593.
- [58] Longuet-Higgins H C. A computer algorithm for reconstructing a scene from two projections[J]. Nature, 1981, 293(5828): 133.
- [59] Gao X S, Hou X R, Tang J, et al. Complete solution classification for the perspective-three-point problem[J]. IEEE transactions on pattern analysis and machine intelligence, 2003, 25(8): 930–943.
- [60] Lepetit V, Moreno-Noguer F, Fua P. Epnp: An accurate o (n) solution to the pnp problem[J]. International journal of computer vision, 2009, 81(2): 155.
- [61] Penate-Sanchez A, Andrade-Cetto J, Moreno-Noguer F. Exhaustive linearization for robust camera pose and focal length estimation[J]. IEEE transactions on pattern analysis and machine intelligence, 2013, 35(10): 2387–2400.
- [62] Triggs B, McLauchlan P F, Hartley R I, et al. Bundle adjustment—a modern synthesis[C]. Springer, 1999: 298–372.
- [63] Granshaw S. Bundle adjustment methods in engineering photogrammetry[J]. The Photogrammetric Record, 1980, 10(56): 181–207.
- [64] Ullman S. The interpretation of structure from motion[J]. Proceedings of the Royal Society of London. Series B. Biological Sciences, 1979, 203(1153): 405–426.
- [65] Mur-Artal R, Tardós J D. Visual-inertial monocular slam with map reuse[J]. IEEE Robotics and Automation Letters, 2017, 2(2): 796–803.

- [66] Huber P J. Robust estimation of a location parameter[M]. Springer, 1992: 492–518.
- [67] Paz L M, Tardos J D, Neira J. Divide and conquer: Ekf slam in $O(n)$ [J]. IEEE Transactions on Robotics, 2008, 24(5): 1107–1120.
- [68] Sibley G, Matthies L, Sukhatme G. Sliding window filter with application to planetary landing[J]. Journal of Field Robotics, 2010, 27(5): 587–608.
- [69] Hahnel D, Burgard W, Fox D, et al. An efficient fastslam algorithm for generating maps of large-scale cyclic environments from raw laser range measurements[C]. IEEE, 2003: 206–211.
- [70] Ulrich I, Nourbakhsh I. Appearance-based place recognition for topological localization[C]. Ieee, 2000: 1023–1029.
- [71] Latif Y, Cadena C, Neira J. Robust loop closing over time for pose graph slam[J]. The International Journal of Robotics Research, 2013, 32(14): 1611–1626.
- [72] Filliat D. A visual bag of words method for interactive qualitative localization and mapping[C]. IEEE, 2007: 3921–3926.
- [73] Gálvez-López D, Tardos J D. Bags of binary words for fast place recognition in image sequences[J]. IEEE Transactions on Robotics, 2012, 28(5): 1188–1197.
- [74] Jain A K. Data clustering: 50 years beyond k-means[J]. Pattern recognition letters, 2010, 31(8): 651–666.
- [75] Sivic J, Zisserman A. Video google: A text retrieval approach to object matching in videos[C]. IEEE, 2003: 1470.
- [76] Robertson S. Understanding inverse document frequency: on theoretical arguments for idf[J]. Journal of documentation, 2004, 60(5): 503–520.
- [77] Burri M, Nikolic J, Gohl P, et al. The euroc micro aerial vehicle datasets[J]. The International Journal of Robotics Research, 2016, 35(10): 1157–1163.
- [78] Delmerico J, Scaramuzza D. A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots[C]. IEEE, 2018: 2502–2509.

攻读学位期间发表论文与研究成果清单