硕士学位论文

基于单目视觉与 IMU 结合的 SLAM 技术研究

RESEARCH ON SLAM TECHNOLOGY BASED MONOCULAR VISION AND IMU

李建禹

哈尔滨工业大学

2018 年 6 月

国内图书分类号：TP242.6　　学校代码：10213
国际图书分类号：621.3　　密级：公开

工学硕士学位论文

基于单目视觉与 IMU 结合的 SLAM 技术研究

硕士研究生：李建禹
导　　师：刘国栋教授
申请学位：工学硕士
学　　科：仪器科学与技术
所在单位：电气工程及自动化学院
答辩日期：2018 年 6 月
授予学位单位：哈尔滨工业大学

Classified Index: TP242.6
U.D.C: 621.3

Dissertation for the Master Degree in Engineering

RESEARCH ON SLAM TECHNOLOGY BASED MONOCULAR VISION AND IMU

Candidate: Li Jianyu
Supervisor: Prof. Liu Guodong
Academic Degree Applied for: Master of Engineering
Speciality: Instrumentation Science and Technology
Affiliation: School of Electrical Engineering and Automation
Date of Defence: June，2018
Degree-Conferring-Institution: Harbin Institute of Technology

摘　要

同时定位与地图构建(Simultaneous Localization and Mapping, SLAM)是机器人领域中的一项关键技术，是实现移动机器人完全自主控制和智能化的核心和基础。随着三维计算机视觉算法的不断发展，基于视觉的 SLAM 方法成为近些年来的研究热点。然而，视觉 SLAM 方法过于依赖周围环境的特征信息，无法处理场景纹理缺失及动态场景的情况，且视觉传感器帧率较低，无法处理快速运动的情况。惯性测量元件（Inertial Measurement Unit, IMU）能够测量传感器本身的角速度和加速度，与视觉传感器具有明显的互补性，有潜力构建出鲁棒性更强的 SLAM 系统。基于此，本文提出了一种基于单目视觉与 IMU 结合的 SLAM 方案。本文的内容主要包括以下部分：

第一，对视觉与 IMU 结合的 SLAM 系统进行深入研究，概述了其研究背景，发展及研究现状。分析了 SLAM 系统的运动模型和观测模型与位姿表示方法，为构建视觉惯性 SLAM 系统提供了理论基础。

第二，重点研究了基于单目视觉和基于 IMU 信息的位姿估计方法，在多视图几何与 IMU 预积分理论的基础上，提出了一种视觉惯性里程计方法(VIO, Visual Inertial Odometry)，它的前端采用 Shi-Tomas 特征点提取与 KLT 光流法进行特征跟踪。它的后端以位置，姿态，速度以及传感器偏置作为状态变量，以紧耦合的方式同时优化视觉误差项与 IMU 误差项。并采用滑动窗口的方式控制优化变量的数量，将距离目前较远的帧边缘化，增强系统的实时性。

第三，在上述 VIO 的基础上，本文提出了一套完整的视觉惯性 SLAM 系统方案。该方案采用基于词袋模型的回环检测与全局位姿图优化，进一步优化 VIO 输出的位姿信息，以减小系统的累计误差。特别地，本文以回环检测和位姿图优化为基础，对系统增加了重定位和地图复用的功能，进一步增强了系统的鲁棒性与实用性。

最后对所构建的视觉惯性 SLAM 系统进行了实验和精度分析，验证了本文提出视觉惯性 SLAM 系统方案的有效性。分析了系统的相对位姿误差，绝对位姿误差以及系统运行时间，并与主流视觉惯性方案 OKVIS 进行对比。结果表明，本文所提出的系统精度优于 OKVIS 方法，绝对位姿误差均方根值最小可达 0.073m，相对位姿误差均方根值最小可达 0.0026m，且系统具有良好的实时性和鲁棒性。

关键词：SLAM；视觉惯性里程计；IMU；位姿估计；回环检测

Abstract

SLAM (Simultaneous Localization and Mapping) is a key technology in the field of robotics, which is considered as the core and foundation for realizing the complete autonomous control and the real intelligence of the mobile robot. Because of the unique advantages of visual sensors and the continuous development of 3D computer vision algorithms, vision based SLAM methods is known as a research hotpot in recent years. However, the visual SLAM method is too dependent on the texture information of the peripheral environment, and cannot handle the absence of the scene texture and the situation of the dynamic scene. The frame rate of the visual sensor is low, so it can not handle the fast motion situation. IMU (Inertial Measurement Unit) can measure the angular velocity and acceleration of the sensor itself. It can deal with the visual failure. It has the obvious complementarity with the visual sensor, and has the potential to build a more robust SLAM system. Based on this, we propose a SLAM method based on the combination of monocular vision and IMU, which can be used for robot positioning and attitude estimation under different environments. The main contents of this article include the following parts:

Firstly, this paper makes an in-depth study of SLAM system combining vision and IMU, and summarizes its research background, development and research status. The analysis of the motion model of the SLAM system and the observation model and position representation method, a mathematical model for the SLAM problem, provides a theoretical basis for the construction of the visual inertia SLAM system.

Secondly, we focus on the monocular vision combined with IMU based pose estimation method. Based on multi-view geometry and IMU pre-integration theory, we proposes a visual inertial odometry（VIO） method. Its front-end uses Shi-Tomas feature point extraction and KLT optical flow method for feature tracking. Its back-end takes position, posture, speed and sensor bias as state variables and optimizes visual error and IMU error simultaneously in tightly coupled mode. The sliding window method is adopted to control the number of optimization variables, and the edge of the far distance frame is marginalized, so that the method is real-time.

Thirdly, based on the above VIO, this paper proposes a complete scheme of visual inertial SLAM system. This scheme combines bag-of-word model based loop closure and the optimization of global pose map to further optimize the pose information of VIO output, so as to reduce the cumulative error of the system.

This paper based on loop closure and pose graph optimization, in particular adds the function of relocation and map reuse to the system, and further enhances the robustness and practicability of the system. Under the existing open source monocular vision SLAM framework, the computer vision library opencv and the nonlinear optimization library are used in the robot. On the ROS of operation system, we have completed the construction of the system proposed in this paper.

Finally, The experimental and precision analysis of the constructed visual inertial SLAM system is carried out. The validity of the proposed visual-inertial SLAM system is verified, and the relative pose error, absolute pose error and the time required for the system operation are analyzed. The results show that he proposed system is superior to the OKVIS method in accuracy. The least RMSE of absolute pose error is up to 0.073m, and least RMSE of the relative pose error is 0.0026m. And the system has good real-time performance and robustness.

Keywords: SLAM, Visual Inertial Odometry, IMU, Pose Estimation, Loop Closure

目　录

第1章 绪论

1.1 课题背景及研究意义

随着人类探索领域的扩大与人工智能和模式识别等技术的快速发展，移动机器人技术逐渐从工业领域被应用在越来越多的领域，包括军事、服务、医疗、救援、宇宙探索等各个领域。这些环境通常具有不确定性和非结构化等特点。需要机器人具备感知，运动和自主规划等能力。移动机器人技术涉及运动学、传感器技术、计算机技术、图像处理、人工智能等多种学科技术。为了进一步提高移动机器人智能化属性以使其应用在更复杂的环境与任务中更好地为人类服务，对于移动机器人的自主性和适应环境的能力也提出了更高的要求。在众多移动机器人技术中，移动机器人的自主导航被认为是实现移动机器人完全自主控制和真正智能化的核心和基础。这个问题可以总结为以下四个子问题[1]：

1）定位：确定机器人在环境中的位置与姿态。

2）地图创建：感知机器人周围环境的三维信息。

3）任务规划：根据要执行的任务，确定短期目标位置。

4）路径规划：在当前位置与目标位置之间找到一条最优路径。

其中，定位和地图创建是实现路径规划和导航这两个问题的前提条件。在实际过程中，环境地图和移动机器人自身的位置信息都是未知的，且定位和建图二者相互耦合、互为依赖，使得问题求解非常困难[2]。因此，解决机器人的定位与建图问题是至关重要的。

移动机器人的定位技术可以分为两大类：绝对定位技术和相对定位技术。绝对定位技术的实现方式主要是全球定位系统（Global Position System，GPS）等。GPS 定位有许多优点如：定位方法比较成熟容易集成，在室外信号较好的情况下定位精度较高。但其有一个最主要的缺点：依赖外部信号。在 GPS 信号被遮挡，干扰或缺失的情况下定位会失效。在星际探测中如月球车的定位，同样无法使用 GPS 进行定位。此外，GPS 在室内定位中精度可能达不到移动机器人的需求。而相对定位技术则是利用机器人自身携带的传感器，根据机器人起始位置以及每个时刻的位置与运动状态进行推算来实现定位，具有不受外部信号影响的特点。传统的相对定位方法主要分为里程计法和惯性导航法[3]。里程计（odometry）一般采用车轮上的光电编码器记录车轮转数或旋转角速度来确定机器人速度从而确定机器人的行进轨迹。这种方法的主要

缺点是无法克服车轮打滑时引起的测量错误，并且只适用于水平方向上的运动估计，仅靠轮式里程计没有办法估计垂直方向上的运动状态。惯性导航系统（Inertial Navigation System，INS）一般利用惯性测量元件（Inertial Measurement Unit，IMU）即三轴加速度计和三轴陀螺仪来测量机器人的线加速度与角速度进行积分来推算机器人的运算轨迹。由于惯性导航法不需要借助其他外部传感器与信号，仅凭自身测量数据就能位置和姿态的解算，因此广泛应用在航空航天等技术中。但是由于位置估计是由加速度二次积分得到的，使得误差极易被累计从而导致轨迹随时间的漂移问题[4]。

随着传感器的发展以及集成电路的运算能力的提高，移动机器人使用自身携带的传感器感知周围环境信息，同时利用感知到的环境信息来进行自身定位的方法被提出，这种方法被称为同时定位与建图[5](Simultaneous Localization and Mapping，SLAM)。从 SLAM 的概念提出到现在的 30 年的时间里，移动机器人的同时定位与建图方法取得了长足的进步[2]，但是这些研究主要是基于激光雷达或超声波传感器等传感设备实现的。近些年来，基于视觉传感器的同时定位和建图引起了研究人员的广泛关注，成为机器人领域与计算机视觉领域一个非常活跃的研究方向。

对基于视觉的同时定位与建图方法的研究已经取得了一定的进展，但是也面临着一系列的挑战：首先，随着移动机器人应用领域的不断扩大，机器人所要面对的环境也越来越复杂；其次，基于视觉的同时定位与建图方法已不局限应用于移动机器人导航，它也被应用在自动驾驶，无人机定位，三维重建以及增强现实等领域。这就导致了基于视觉的同时定位与建图方法有许多难以处理的场景，如相机运动过快导致视觉跟踪失效，或场景本身动态变换难以判断自身的位置变化。IMU 能够测量传感器本体的角速度和加速度，被认为与视觉传感器具有明显的互补性，IMU 融合后可以处理视觉失效的情况，例如光照变化、遮挡、模糊、快速运动和动态场景；同时视觉也可以对 IMU 的本质误差零偏进行很好的估计，而且十分有潜力在融合之后得到更完善的 SLAM 系统[6]。因此，对基于单目视觉与惯性传感器结合的同时定位和建图方法开展进一步的研究工作具有重要的理论意义和实用价值。

1.2 单目视觉与 IMU 结合 SLAM 技术的发展与研究现状

1.2.1 SLAM 技术发展与研究现状

移动机器人 SLAM 问题首次被提出是在 1986 年的 ICRA 会议上。Peter Cheesman 提出采用贝叶斯计算来估计机器人状态,用于实现移动机器人

SLAM 问题。在 1986 年的 ICRA 会议上，讨论得出结论,认为一致性概率地图 SLAM 问题是极有研究价值的科学问题，并关系到移动机器人未来发展的趋势。随之，在接下来几年里,一些重要科研成果相继发表。相隔几年，一种建立环境路标几何不确定性的统计方法被 Cheesman Simith 和 Durrant-Whyte 提出。这种建立环境路标的方法揭示了路标位置信息与后续对该路标的观测值之间的数据高度相关性。相继 Leonard 和 Durrant-Whyte 提出新的观点,新观点指出只有移动机器人具备自主估计自身位姿状态并采集环境路标点信息的能力，才能得到一致性概率地图，完成地图构建。这也正是目前关于移动机器人 SLAM 研究的主导思想。随后，研究 SLAM 地图构建的收敛性成为了最热门的研究问题,学者通过实验，验证 SLAM 构建的地图误差会随时间增加不断积累增大。20 世纪末期，Durrant-Whyte 提出了一种建立系统性框架解决移动机器人 SLAM 的方法[7]，并验证了该方法可以使得 SLAM 构建的地图误差会收敛。Durrant-Whyte 的新方法思想是将位姿状态放入全状态估计器中，得到的状态信息数收敛。新方法未考虑到将庞大的位姿状态量放入全状态估计器中，会导致被估计状态维数增加,造成计算量以及复杂度加大,会影响 SLAM 系统的计算速度及实时性。至此，SLAM 问题已基本被确定，构建一致收敛的轨迹和地图并且控制计算量大小是此时 SLAM 系统研究的主要问题。

1.2.2 单目视觉 SLAM 技术发展与研究现状

早期的 SLAM 系统大多以声呐，激光测距仪以及红外测距仪作为传感器的。进入 21 世纪以来，随着三维计算机视觉的发展以及集成电路运算能力的提升，基于视觉的 SLAM 方法开始涌现。单目 SLAM 采用滤波器的方法解决[8-11]。在基于滤波器的方法中，每个帧都被滤波器处理，同时估计地图特征点位置和相机的位姿。它把计算时间浪费在处理只有少量新增信息的连续帧中，并且有线性化误差的累积。在另一类基于关键帧的方法中[12-13]，仅采用部分选定的帧（关键帧）来进行地图估计，使用计算量更大但精确的捆集调整(Bundle Adjustment)优化，这使得地图构建速率与帧速率不同步。SalasdAT 等人发表的论文展示了[14]基于关键帧的方案在计算成本相同的情况下比基于滤波器的方案更精确。

另一项对基于关键帧单目视觉 SLAM 方案具有创造性贡献的工作是 Klein 等人的 PTAM[12]。在该项工作中，首次引入了在并行线程中分离相机跟踪和建图的思想，并证明成功应用在小规模场景下的实时增强现实应用。在后来更新的版本中，增加了边缘特征，在跟踪过程中的旋转估计步骤和更

好的重定位方法[15]。PTAM 的地图点对应通过块匹配的 FAST 角点，该特征点仅用于跟踪，并没有用于位置识别。Strasdat 等人[16]提出了一套大场景单目视觉 SLAM 系统。该系统带有在 GPU 上实现的基于光流法的前端，跟着 FAST 特征匹配和仅有位姿的 BA 和一个后端基于滑动窗口的。闭环检测是通过基于相似约束（7 个自由度）的位姿图优化解决的，它可以纠正在单目视觉 SLAM 中出现的尺度漂移。

Stasdat 等人[17]使用 PTAM 的前端，但仅在从共视图中得到的局部地图上进行跟踪。他们提出了一种双窗口的后端优方法，在内部窗口中连续运行 BA，并在有限大小的外部窗口中进行位姿图优化。然而，回环检测仅有当外部窗口的大小是大到足以包括整个循环才是有效的

Pirker 等人[18]提出了 CD-SLAM，它是一个非常全面系统，包括回环检测，重定位，适应大尺度场景和一些在动态环境上的工作。然而该方法并没有提及如何地图初始化。

除特征点外，恩格尔等人提出了 LSD—SLAM 方法[19]，该方法能够建立大规模的半稠密地图，采用直接法（即直接在图像像素强度上进行的优化）代替基于特征的捆集调整。他们得到了非常惊艳的效果，该系统能在没有 GPU 加速的情况下建立半稠密地图。相比于输出稀疏地图的基于特征的 SLAM 方案，该方案更具有潜在的机器人应用前景。尽管如此，它们仍然需要闭环检测的功能且相机的定位精度明显低于 PTAM。

在直接法和基于特征的方法之间的折中是福斯特等人[20]的半直接视觉里程计 SVO(Semi-direct Visual Odometry，SVO)。它不需要在每一帧中提取特征就能够在高帧率下工作，在四旋翼飞行器上获得令人印象深刻的结果。然而，没有执行回环检测。当前的实现主要考虑的是向下朝向的相机。

2016 年 Mur-Artal 等人提出了 ORB-SLAM[21]。它是集成了前人的工作，是现代 SLAM 系统中做的比较完善的系统之一。它支持单目、双目、RGB-D 三种模式，泛用性较好。并且它创新性的使用了三个线程完成 SLAM：实时跟踪特征点的追踪线程，局部 BA 的优化线程和优化全局位姿图的回环检测以及优化线程。ORB-SLAM 的三线程结构取得了非常好的的跟踪和建图效果，本文的惯性 SLAM 系统也采用类似的架构。然而，该系统有基于纯视觉的 SLAM 系统故有的缺点，即对相机快速运动，场景特征点缺失以及动态场景会失效，这时就需要有惯性传感器对视觉补充。也正因为此，本文着力于研究视觉与惯性传感相结合的 SLAM 系统。

1.2.3 视觉与惯性结合的 SLAM 技术发展与研究现状

相机能够捕捉含有丰富细节场景信息，但无法应对快速运动与动态场景，而 IMU 虽然有较严重的长时间累计误差与漂移问题，但其有较高的测量速率且能够获得较为准确的短时间估计，这两个传感器具有明显的互补特征，从而在一起使用能够得到更好的结果。因此目前 SLAM 的研究热点的之一就是利用视觉传感器与 IMU 进行结合[22-25],实现视觉惯性里程计(Visual Inertial Odoinetry，VIO)，或视觉惯性 SLAM(Visual Inertial SLAM，VI-SLAM)。下面将介绍目前目前 VIO/VI-SLAM 的发展现状和主流的几种方法。

视觉惯性 SLAM 主要有两种分类方法，一种是基于后端状态估计方法分类，可分为基于滤波和基于优化的两大类方法。另一种是根据视觉与 IMU 的融合方式进行分类，可分为松耦合(loosely-coupled)和紧耦合(tightly-coupled)两大类。

由于最早的视觉与 IMU 结合的位姿估计方法采用的是滤波方法，首先介绍基于滤波的紧耦合方法。该类方法需要将图像的特征点放入到滤波的状态之中，因此整个系统需要估计的状态向量的维数与特征点数量正相关，显然随着相机的连续运动，特征点数量会不断增加，此时状态向量的维数会随着相机运动时间累积而大大增加，因此计算负担会越来越大，难以达到实时性的要求。

目前比较知名的紧耦合方法有鲁棒视觉惯性里程计 (Robust Visual Inertial Odometry, ROVIO)[26]和多状态限制卡尔曼滤波器(Multi-State Constraint Kalman Filter, MSCKF)[27]。传统的基于扩展卡尔曼滤波(Extend Kalman Filter, EKF)的 SLAM 方法中，特征点的信息会加入特征向量，多个特征点同时约束一个状态向量，这种方法计算协方差矩阵时计算量非常大，而且还要给定特征点深度和协方差的初始值，如初始条件给定不准确，极易导致之后进行的迭代不收敛，最终无法得到准确位姿估计。而 MSCKF 则用一个特征点约束多个状态向量，给定一个固定时间序列的滑动窗口，每当一个特征点在滑动窗口中被几个位姿被观测到，就会对这几个被观测到的位姿进行约束，从而实现了卡尔曼滤波的更新。在另一类基于松耦合的滤波方法中，与紧耦合不同，松耦合并不把图像的特征点当做状态变量，与之相反，将视觉算法即通过图像特征匹配解算位姿单独运行，仅当视觉里程计得到位姿估计后，再将得到的位姿估计作为状态向量加入滤波框架与 IMU 实现融合[28-29]。

与基于滤波的方法相对的是基于优化的方法。Mourikis 等人[30]将融合问题分为两个独立线程进行处理，第一个线程处理局部连续图像间的惯性测量

和特征跟踪，提供高频率的位姿估计，第二个线程包含一个间断工作的捆集调整的迭代估计，能够减少线性误差带来的影响。Leutenegger 等人[31]将 IMU 的误差以全概率形式融合到三维路标点的重投影误差里，构成将要被优化的非线性误差函数，在优化的过程中，通过持续边缘化较远的关键帧来维持一个大小固定的优化窗口，以保证优化进程的实时性。考虑到基于优化方法的视觉惯性 SLAM 问题的计算复杂度，Forster 等人[32]提出了预积分理论，预积分能够将两个相邻视觉关键帧的惯性测量到集中到一个独立的相对运动约束，这个 IMU 的预积分模型可以完美地融合到视觉惯性的因子图的框架下。

目前实现较好的视觉惯性 SLAM 算法都是使用较高精度的 IMU。但是由于其成本非常昂贵并限制了使用范围，因此实现一个实用的鲁棒性和高精度的惯性视觉 SLAM 系统依然是目前机器人和 SLAM 相关领域研究的热点和难点之一。

1.3 论文的主要研究内容

本文主要研究基于单目视觉与 IMU 结合的 SLAM 系统。对视觉 SLAM 相关理论进行分析和相关公式推导，重点研究视觉 SLAM 的数学建模，基于连续图像帧的位姿估计，IMU 预积分理论以及紧耦合视觉惯性融合方法。基于以上理论和方法，在现有惯性视觉系统框架进行扩展，提出了一种基于紧耦合非线性优化的在线单目视觉惯性 SLAM 系统，主要有以下两点特性：(1)通过基于关键帧滑动窗口紧耦合非线性优化提供稳定、准确、高效的位姿估计与地图构建。(2)系统带有回环检测功能，能利用检测到的回环进行重定位及全局位姿图优化。

本文结构安排如下：

第一章介绍了惯性视觉 SLAM 的背景及研究意义，梳理了 SLAM 技术的发展历程，分析了视觉 SLAM 的发展现状以及纯视觉方法的缺点，引出了基于惯性的 SLAM 的方法并介绍了其发展现状，最后论述了本文的主要研究内容与结构安排。

第二章分析阐述了惯性视觉 SLAM 的数学理论和模型，介绍了针孔相机模型以及世界，相机，IMU 之间的坐标系变换。介绍了基于四元数和李群李代数的位姿表示方法，并介绍了相关的数学运算。最后给出了 SLAM 问题的数学表述并推导了基于最大后验概率的后端优化方法。

第三章介绍了分别基于视觉信息和基于 IMU 信息的位姿恢复方法。在基于视觉的位姿恢复方法中，介绍了 Shi-Tomasi 特征提取方法与 KLT 光流法

追踪。在得到追踪到的特征点的基础上，分别介绍了基于对极几何，透视 N 点定位和光束法平差的位姿估计方式。在基于 IMU 的位姿恢复方法中，介绍了 IMU 的误差模型与相关运动学方程，推到了 IMU 预积分公式，最后得出基于 IMU 信息与视觉关键帧对齐的位姿估计结果。

第四章在前两章的基础上，提出了一种视觉惯性融合的在线 SLAM 系统。首先介绍了该系统的整体框架，随后介绍了基于滑动窗口视觉惯性紧耦合的非线性优化方法，给出了整体视觉惯性优化项与视觉，IMU 误差的求解方法。最后介绍了系统的回环检测和全局位姿图优化方法。

第五章进行了实验和精度分析。基于 Euroc 无人机惯性视觉数据集对视觉惯性状态估计器和整体 SLAM 系统进行了实验分析，给出了系统输出结果，并分析了系统的相对位姿误差与绝对位姿误差以及系统的运行时间。并与主流视觉惯性 SLAM 方案 OKVIS 进行对比，证明了本系统的用时短和精度高的特性。

第2章 惯性视觉 SLAM 的数学理论与模型

2.1 引言

SLAM 技术被认为的真正实现是机器人自主导航的前提之一，是机器人研究领域最关键的问题，也是过去二十多年来研究的重点课题。智能移动机器人要想实现真正的自主导航，需要具备定位，地图创建，任务规划和路径规划的四项能力。其中定位和地图创建是这四项能力的基础，这也是 SLAM 需要解决的问题。

目前，SLAM 技术已经取得了一定发展。研究者们把 SLAM 问题建模为状态估计问题，主要采用基于最大后验概率估计（Maximum a posteriori estimation, MAP）或扩展卡尔曼滤波器（Extend kalman filter, EKF）问题。其中，状态估计的变量为相机的位姿，大多数是由四元数和李代数所表示的，这又涉及了位姿的表达方式及其转换。而相机成像的过程也需要用几何模型描述。正是这些抽象建模，才使得求解 SLAM 问题变得可能。因此，本章在介绍如何解决惯性视觉 SLAM 问题之前，首先介绍了惯性视觉 SLAM 的建模和背后的数学原理。

2.2 相机模型与坐标系变换

2.2.1 针孔相机模型

基于视觉的 SLAM 方法利用相机获取场景中的信息进而后续的处理。相机成像的过程是将三维世界的坐标点映射到二维平面的过程，这个过程可以用一个几何模型描述。相机模型有很多种，本文采用了最常用的针孔相机模型，它描述了一束光线通过针孔之后，在针孔背面投影成像的关系。本节先介绍相机的针孔模型，再对由之带来的坐标系变换进行介绍。

如图 2-1 所示，设 $O-x-y-z$ 为相机坐标系，z 轴指向相机的前方，x 轴向右，y 轴向下，O 为相机的光心，也是针孔模型中的针孔。现实世界的空间点 P，经过小孔 O 投影之后，落在物理成像平面 $O'-x'-y'$ 上，成像点为 P'。设 P 的坐标为 $[X,Y,Z]^T$，P' 为 $[X',Y',Z']^T$，并且设物理成像平面到小孔的距离为 f。那么根据三角形相似关系，有：

$$\frac{Z}{f}=-\frac{X}{X'}=-\frac{Y}{Y'} \tag{2-1}$$

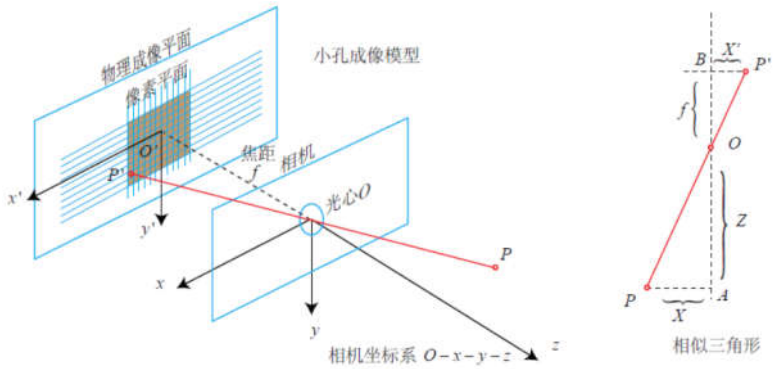


图 2-1 针孔相机模型

其中负号表示成的像是倒立的，为了简化模型，在不改变几何关系的前提下，将相平面对称到相机前方。在成像平面建立图像坐标系 O'–x'–y'，以主光轴与成像平面交点为原点，x’轴水平向右，y’轴竖直向下。如图 2-2 所示，

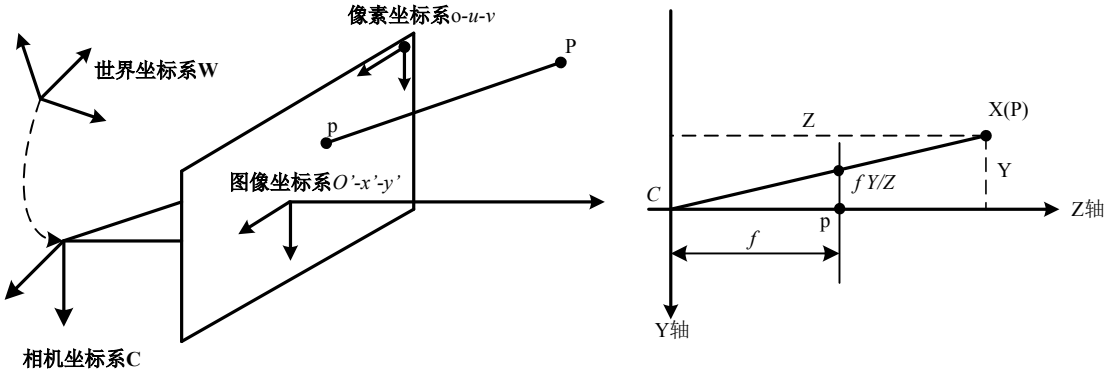


图 2-2 对称后的小孔成像模型

将 2-1 式中负号去掉并整理得：

$$X' = f\frac{x}{z}$$
$$Y' = f\frac{y}{z} \tag{2-2}$$

式 2-2 描述了点 P 和它的像空间之间的空间关系，也是相机坐标系到图像坐标系的变换。由于通过相机成像，我们最终获得的是像素坐标，这就需要对图像坐标系上的点进行采样和量化。这时我们设在物理成像平面上固定着一个像素平面 o-u-v。其中原点 o 为图像的左上角顶点，u 轴向右与 x 轴平行，v 轴向下与 y 轴平行。像素坐标系与图像坐标系坐标之间相差了一个缩放和一个原点平移。可得到相机坐标系与图像坐标系的转换关系：

$$\begin{aligned} \mathrm{u} &= f_x \frac{X}{Z} + c_x \\ \mathrm{v} &= f_y \frac{Y}{Z} + c_y \end{aligned} \tag{2-3}$$

其中 f_x、f_y 为以像素为单位的焦距，等于焦距程序乘以相应方向的尺度因子，$[c_x, c_y]^T$ 为原点在图像坐标系的坐标。为了表示方便，利用齐次坐标表示，将式(2-3)以矩阵的形式重写：

$$\mathrm{s}\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_v & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2-4}$$

在该式中，左侧的 s 为尺度因子，它与三维点在相机坐标系下的 Z 轴方向的坐标即点的深度相等。第一个 3×3 的矩阵称为内参数矩阵 K，内参数矩阵如未知，可较方便的由张正友标定法获得。

以上推导都是建立在基准坐标系为相机坐标系的前提下的。为了更一般的情况，比如以 IMU 的坐标系为基准或以世界坐标系为基准的情况，我们需要了解在任意一个三维坐标系下的三维坐标到像素坐标的过程。这就需要在之前的过程之前加上一个三维坐标系的变换。

两个三维坐标系的关系可由一个 3×3 的旋转矩阵 R 和一个 3×1 的平移向量 t 确定。以相机坐标系与世界坐标系为例，设某一 3D 点在世界坐标系下的坐标为 X_w，对应的像素坐标为 x。由式（2-4）可得：

$$sx = K(RX_w + t) = KTX_w \tag{2-5}$$

其中的 R，t 可以被称为相机外参，代表了相机在世界坐标系下的姿态。

2.2.2 相机畸变模型

为了获得更好的成像效果，通常会在在相机的前方加入透镜。透镜的加入对成像过程中光线的传播会产生新的影响：一是透镜自身的形状对光线传播的影响，二是在机械组装过程中，透镜和成像平面不可能完全平行，这也会使得光线穿过透镜投影到成像面时的位置发生变化。

由透镜形状引起的畸变称之为径向畸变。在实际拍摄的图像中，由于相机透镜的存在，使得原本真实环境中的一条直线在图像中编程了曲线，产生了畸变，而且越靠近图像的边缘，畸变越明显。由于透镜往往是中心对称的，因此畸变通常也是径向对称的。径向畸变主要分为两大类，桶形畸变和枕形畸变，如图 2-3 所示：

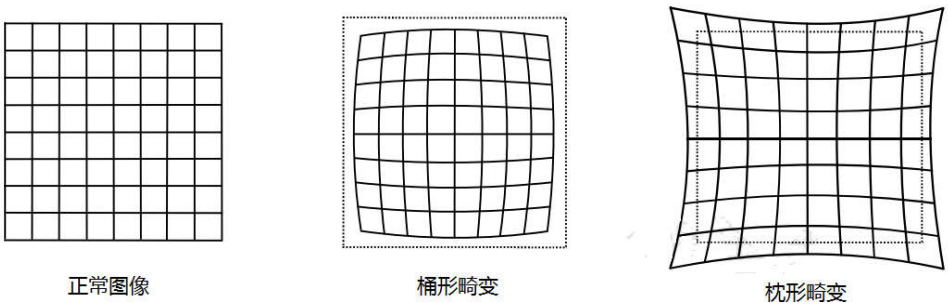


图 2-3 相机径向畸变模型

桶形畸变是由于图像放大率随着离光轴的距离增加而减小，而枕形畸变却恰好相反。在这两种畸变中，穿过图像中心和光轴有交点的直线还能保持形状不变。

除了透镜的形状会引入径向畸变外，在相机的组装过程中由于不能使得透镜和成像面严格平行也会引入切向畸变。

为了消除畸变对系统造成的影响，我们需要对畸变进行矫正。对于径向畸变，无论是桶形畸变还是枕形畸变，由于它们都是随着离中心的距离增加而增加。我们可以用一个多项式函数来描述畸变前后的坐标变化:这类畸变可以用和与中心距离有关的二次及高次多项式函数进行纠正:

$$\begin{aligned} x_c &= x(1+k_1r^2+k_2r^4) \\ y_c &= y(1+k_1r^2+k_2r^4) \end{aligned} \tag{2-6}$$

其中$[x,y]^T$是未校正的点的坐标，$[x_c,y_c]^T$是校正后的点的坐标，他们均为归一化平面上的点。

对于切向畸变，可以采用令两个参数p_1,p_2进行校正：

$$\begin{aligned} x_c &= x+2p_1xy+p_2(r^2+2x^2) \\ y_c &= y+p_1(r^2+2y^2)+2p_2xy \end{aligned} \tag{2-7}$$

联合式(2-6)和式(2-7)，对相机坐标系中的任意一点P（X,Y,Z），可以通过四个畸变参数找到该点在像素平面上正确的位置：

首先将图像坐标系的点投影到归一化平面上，设它的归一化坐标为$[x,y]^T$。

再对其进行畸变校正：

$$\begin{aligned} x_c &= x(1+k_1r^2+k_2r^4)+2p_1xy+p_2(r^2+2x^2) \\ y_c &= y(1+k_1r^2+k_2r^4)+p_1(r^2+2y^2)+2p_2xy \end{aligned} \tag{2-8}$$

最后将校正后的点通过内参数矩阵投影到像素平面，得到该点在图像上的正确位置：

$$\begin{aligned} u &= f_x x_c + c_x \\ v &= f_y y_c + c_y \end{aligned} \tag{2-9}$$

至此，我们即可将通过带有畸变的相机图像通过四个畸变参数 k_1, k_2, p_1, p_2 校正为正确的图像。

2.2.3 坐标系变换

惯性视觉系统比纯视觉系统多了一个惯性元件 IMU，在介绍其他原理之前，首先介绍一下惯性视觉系统中的坐标系变换，以及在本文中的字母表示。本系统中重要的坐标系共有三个：

1）世界坐标系 W(World)，它是一个绝对坐标系，在系统中恒定不动。

2）相机坐标系 C(Camera)，它的定义与 2.1 节中介绍的相机坐标系相同，会随着相机的移动而移动。

3）IMU 坐标系 B(Body)，如图 2-4 所示，它是在 IMU 芯片上建立的坐标系，为右手坐标系，Z 轴指向芯片外侧，X 轴水平向右。

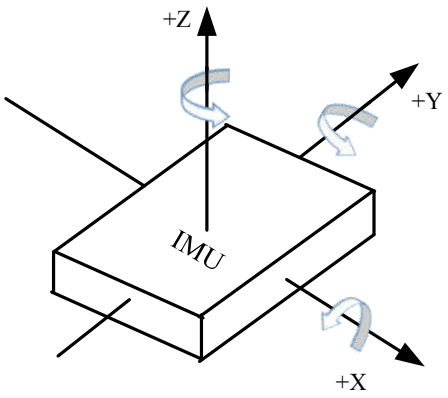


图 2-4 IMU 坐标系

本文对带有单目相机和 IMU 的机器人进行定位，当描述机器人的位姿状态时，以 IMU 的位姿代指实际的机器人的位姿，故机器人的坐标系与 IMU 坐标系一致。在某一时刻，可以用世界坐标系到 IMU 坐标系的变换，表示机器人当前时刻的位姿状态。

W, C, B 三个坐标系的变换如图 2-5 所示。世界坐标系 W 到 IMU 坐标系 B 的转换可用 T_{WB} 表示，下标 WB 表示坐标系 W 到坐标系 B 的转换。转换 T 中包含一个旋转矩阵 R 和一个平移向量 p，如 $T_{WB}=(R_{WB}, p_{WB})$。旋转也可

以用四元数的方式表示此时 $T_{WB}=(p_{WB}, q_{WB})$。

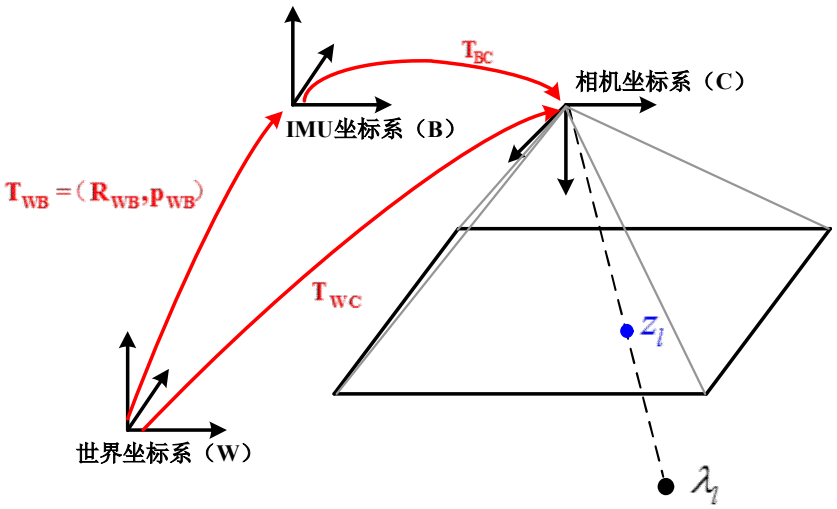


图 2-5 世界、IMU、相机坐标系变换

在三个坐标变换中，T_{BC} 为 IMU 坐标系到相机坐标系的变换，被称为惯性相机的外参，一般惯性元件和相机安装完位置是固定的，T_{BC} 的值不会变化。它的值一般由惯性相机生产厂家或利用视觉 IMU 标定算法进行标定[33]。

三个坐标系的关系可由以下公式给出：

$$T_{WB}=T_{BC}T_{WC} \tag{2-10}$$

2.3 四元数的运算及其旋转表示

四元数是一种旋转的表示方式，相比于旋转向量来讲，它的表示方式更紧凑。相比于欧拉角，它没有万象锁节效应，不具有奇异性，具有较好的特性。本文部分的位姿优化采用四元数的方法表示。

四元数是由汉密尔顿在 1983 年提出，它可以看做是复数的扩展。一个四元数 q 有一个实部和三个虚部，其定义为：

$$q=q_w+q_xi+q_yj+q_zk \tag{2-11}$$

其中 i，j，k 为四元数的三个虚部，它们满足以下关系式：

$$\begin{cases} i^2=j^2=k^2=-1 \\ ik=k, ji=-k \\ jk=i, kj=-i \\ ki=j, ik=-j \end{cases} \tag{2-12}$$

用四元数表达对一个点的旋转。假设一个空间三维点 $\boldsymbol{p}=[x,y,z]\in\mathbb{R}^3$，以

及一个由轴角 n，θ 指定的旋转。三维点 $\boldsymbol{p}$ 经过旋转之后变成为到 $\boldsymbol{p}$'如果使用矩阵描述，那么有 $\boldsymbol{p}'=\boldsymbol{R}\boldsymbol{p}$。用四元数描述旋转的方式如下：

首先，把三维空间点用一个虚四元数来描述：

$$\boldsymbol{p}=[0,x,y,z]=[0,\boldsymbol{v}] \tag{2-13}$$

这相当于把四元数的三个虚部与空间中的三个轴相对应。然后，用四元数 $\boldsymbol{q}$ 表示这个旋转：

$$\boldsymbol{q}=\left[\cos\frac{\theta}{2},n\sin\frac{\theta}{2}\right] \tag{2-14}$$

那么，旋转后的点 $\boldsymbol{p}$'即可表示为这样的乘积：

$$\boldsymbol{p}'=\boldsymbol{q}\boldsymbol{p}\boldsymbol{q}^{-1} \tag{2-15}$$

可以验证，计算结果的实部为 0，故为纯虚四元数。其虚部的三个分量表示旋转后 3D 点的坐标。

任意单位四元数描述了一个旋转，该旋转亦可用旋转矩阵或旋转向量描述。设四元数 $\boldsymbol{q}=\boldsymbol{q}_0+\boldsymbol{q}_1 i+\boldsymbol{q}_2 j+\boldsymbol{q}_3 k$，对应的旋转矩阵 R 为：

$$\boldsymbol{R}=\begin{bmatrix}1-2q_2^2-2q_3^2 & 2q_1q_2+2q_0q_3 & 2q_1q_3-2q_0q_2\\ 2q_1q_2-2q_0q_3 & 1-2q_1^2-2q_3^2 & 2q_2q_3+2q_0q_1\\ 2q_1q_3+2q_0q_2 & 2q_2q_3-2q_0q_1 & 1-2q_1^2-2q_2^2\end{bmatrix} \tag{2-16}$$

反之，由旋转矩阵到四元数的转换如下。假设矩阵为 $\boldsymbol{R}=\{m_{ij}\},i,j\in[1,2,3]$，其对应的四元数 q 由下式给出：

$$q_0=\frac{\sqrt{\mathrm{tr}(R)+1}}{2},\ q_1=\frac{m_{23}-m_{32}}{4q_0},\ q_2=\frac{m_{31}-m_{13}}{4q_0},\ q_3=\frac{m_{12}-m_{21}}{4q_0} \tag{2-17}$$

值得一提的是，由于 $\boldsymbol{q}$ 和-$\boldsymbol{q}$ 表示同一个旋转，事实上一个 $\boldsymbol{R}$ 对应的四元数表示并不是惟一的。

2.4李群与李代数

群（Group）是一种集合加一种运算组成的代数结构。把几何集合记为 G，运算记为 $\cdot$，满足以下四个条件的集合和运算记为群：

（1）封闭性：$\forall a,b\in G,\ a\cdot b\in G$.

（2）结合律：$\forall a,b,c\in G,\ (a\cdot b)\cdot c=a\cdot(b\cdot c)$.

（3）单位元：$\exists e\in A,\ s.t.\ \forall a\in G,\ e\cdot a=a\cdot e=a$.

（4）逆元：$\forall a\in A,\ \exists a^{-1}\in G,\ s.t.\ a\cdot a^{-1}=e$.

群结构保证了在群上运算具有良好的性质。矩阵中常见的群有：

一般线性群 GL(*n*)指 $n\times n$ 的可逆矩阵，它们对矩阵乘法成群。

特殊正交群 SO*(n)*为 *n* 维旋转矩阵构成的群；

特殊正交群 SE*(n)*为 *n* 维欧式变换矩阵构成的群。

由于 SO(3)和 SE(3)可以很好地描述刚体运动，对于相机位姿状态估计非常重要。由于 SO(3)和 SE(3)为李群（具有连续光滑性质的群），每一个李群都有一个李代数与之对应，李代数是一种位于向量空间的代数结构。本节主要介绍这两个群以及与之对应的李代数。

2.4.1 特殊正交群 SO(3)

特殊正交群 SO(3)描述了三维旋转矩阵构成的群，它的定义为：

$$SO(3)=\{\mathbf{R}\in\mathbb{R}^{3\times3}\mid \mathbf{R}\mathbf{R}^{T}=\mathbf{I},\det(\mathbf{R})=1\} \tag{2-18}$$

它对应的运算为矩阵乘法，逆运算为矩阵的转置。SO(3)也形成了一个光滑的流形。这个流形的切向空间称为 $\mathfrak{so}(3)$，它是与 SO(3)对应的李代数，它与 3×3 反对称矩阵构成的空间相一致。我们可以引用 ^ 符号，将任意一个在 $\mathbb{R}^3$ 上的向量映射到一个 3×3 反对称矩阵上：

$$\omega^{\wedge}=\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\end{bmatrix}^{\wedge}=\begin{bmatrix}0 & -\omega_3 & \omega_2\\ \omega_3 & 0 & -\omega_1\\ -\omega_2 & \omega_1 & 0\end{bmatrix}\in\mathfrak{so}(3) \tag{2-19}$$

与之相似的，对于任意一个反对称矩阵，我们可以用 ∨ 符号将之映射到 $\mathbb{R}^3$ 的向量上。对于反对称矩阵，后面要用到的一个重要的性质是：

$$a^{\wedge}b=-b^{\wedge}a,\ \forall a,b\in\mathbb{R}^3 \tag{2-20}$$

李群与李代数可以由指数映射和对数映射相互转换。$\mathfrak{so}(3)\to\mathrm{SO}(3)$ 的指数映射与矩阵的指数映射相一致，并且可以用罗德里格斯公式表示：

$$\exp(\phi^{\wedge})=\sum_{n=0}^{\infty}\frac{1}{n!}(\phi^{\wedge})^{n}=I+\frac{\sin(\|\phi\|)}{\|\phi\|}\phi^{\wedge}+\frac{1-\cos(\|\phi\|)}{\|\phi\|^{2}}(\phi^{\wedge})^{2} \tag{2-21}$$

指数映射的一阶近似为：

$$\exp(\phi^{\wedge})\approx I+\phi^{\wedge} \tag{2-22}$$

对数映射将一个 $R\neq I$ 的旋转矩阵映射到反对称矩阵，即从 $\mathrm{SO}(3)\to\mathfrak{so}(3)$ 的映射：

$$\log(\mathrm{R})=\frac{\varphi\cdot(\mathrm{R}-\mathrm{R}^{T})}{2\sin(\varphi)},\ \text{其中}\varphi=\cos^{-1}\left(\frac{\mathrm{tr}(\mathrm{R})-1}{2}\right) \tag{2-23}$$

实际上 $\log(\mathrm{R})^{\vee}=a\varphi$，其中 *a* 和 φ 分别是旋转 R 的旋转轴与旋转角度。这也说明了式（2-12）和式（2-14）的形式与旋转向量和旋转矩阵的公式相同的

原因。即 $\mathfrak{so}(3)$对应的三维向量的实际物理含义为旋转向量。需要注意的是，由于 ϕ 角取值具有周期性，在指数映射中任何 $\phi=(\varphi+2k\pi),k\in\mathbb{Z}$ 都会映射到相同的旋转矩阵 R 上。

为了表示上的方便，我们通常用向量的形式来表示对数和指数映射，SO3 与向量的映射关系可以用式(2-15)表示：

$$\begin{aligned}&\mathrm{Exp}:\quad \mathbb{R}^3 \rightarrow \mathrm{SO(3)}\ ;\quad \phi \rightarrow \exp(\phi^{\wedge})\\&\mathrm{Log}:\quad \mathrm{SO(3)} \rightarrow \mathbb{R}^3\ ;\quad \mathrm{R} \rightarrow \log(\mathrm{R})^{\vee}\end{aligned} \tag{2-24}$$

该式直接在三维向量上进行映射，而不是反对称矩阵 $\mathfrak{so}(3)$。

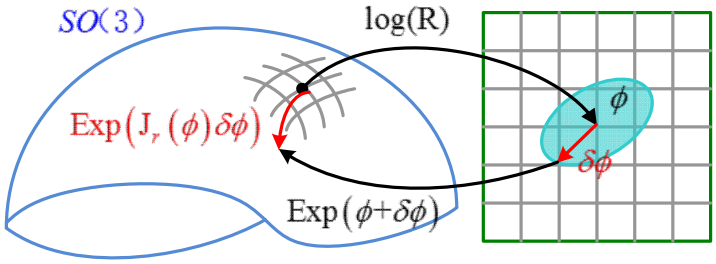


图 2-6 右雅各比 Jr 与切向空间的相加扰动 $\delta\phi$ 与流形 SO（3）上的相乘扰动的关系

指数映射的一阶近似扰动模型为：

$$\mathrm{Exp}(\phi+\delta\phi)\approx\mathrm{Exp}(\phi)\mathrm{Exp}(\mathrm{J}_r(\phi)\delta\phi) \tag{2-25}$$

其中 $\mathrm{J}_r(\phi)$ 的计算方式为：

$$\mathrm{J}_r(\phi)=\mathrm{I}-\frac{1-\cos(\|\phi\|)}{\|\phi\|^2}\phi^{\wedge}+\frac{\|\phi\|-\sin(\|\phi\|)}{\|\phi^3\|}(\phi^{\wedge})^2 \tag{2-26}$$

$\mathrm{J}_r(\phi)$ 称为 SO(3)的右雅各比矩阵，如图 2-6 所示，它与切向空间的微小相加量相关，即在李代数上加一个微小量可近似为在李群上带右雅各比的乘法。

与指数映射相似，对数映射的一阶近似扰动模型为：

$$\mathrm{Log}(\mathrm{Exp}(\phi)\mathrm{Exp}(\delta\phi))\approx\phi+\mathrm{J}_r^{-1}(\phi)\delta\phi \tag{2-27}$$

其中右雅各比的逆 $\mathrm{J}_r^{-1}(\phi)$ 的计算方式为：

$$\mathrm{J}_r^{-1}(\phi)=\mathrm{I}+\frac{1}{2}\phi^{\wedge}+\left(\frac{1}{\|\phi\|^2}+\frac{1+\cos(\|\phi\|)}{2\|\phi\|\sin(\|\phi\|)}\right)(\phi^{\wedge})^2 \tag{2-28}$$

右雅各比 $\mathrm{J}_r(\phi)$ 和它的逆 $\mathrm{J}_r^{-1}(\phi)$ 在 $\|\phi\|=0$ 时会退化为单位阵。

2.4.2 特殊欧式群 SE(3)

特殊欧式群 SE(3)描述了三维刚体运动构成的群，它的定义为：

$$SE(3)=\left\{\boldsymbol{T}=\begin{bmatrix}\boldsymbol{R} & \boldsymbol{t}\\ 0^T & 1\end{bmatrix}\in\mathbb{R}^{4\times4}\,\middle|\,\boldsymbol{R}\in SO(3),\boldsymbol{t}\in\mathbb{R}^3\right\} \tag{2-29}$$

对于特殊欧式群 SE(3)，也有对应李代数 $\mathfrak{se}(3)$，与 SO(3)相似，$\mathfrak{se}(3)$ 位于 $\mathbb{R}^6$ 空间中：

$$\mathfrak{se}(3)=\left\{\boldsymbol{\xi}=\begin{bmatrix}\boldsymbol{\rho}\\ \boldsymbol{\phi}\end{bmatrix}\in\mathbb{R}^6,\boldsymbol{\rho}\in\mathbb{R}^3,\boldsymbol{\phi}\in\mathfrak{so}(3),\boldsymbol{\xi}^{\wedge}=\begin{bmatrix}\boldsymbol{\phi}^{\wedge} & \boldsymbol{\rho}\\ 0^T & 0\end{bmatrix}\in\mathbb{R}^{4\times4}\right\} \tag{2-30}$$

把每个 $\mathfrak{se}(3)$ 元素记作 $\boldsymbol{\xi}$，它是一个六维向量。前三维为平移，记作 $\boldsymbol{\rho}$;后三维为旋转，记作 $\boldsymbol{\phi}$，实质上是 $\mathfrak{so}(3)$ 元素。同时，在这里拓展了 $^{\wedge}$ 符号的含义。$\mathfrak{se}(3)$ 中，同样使用 $^{\wedge}$ 符号，将一个六维向量转换成四维矩阵，但这里不再表示反对称：

$$\boldsymbol{\xi}^{\wedge}=\begin{bmatrix}\boldsymbol{\phi}^{\wedge} & \rho\\ 0^T & 0\end{bmatrix}\in\mathbb{R}^{4\times4} \tag{2-31}$$

这里仍使用 $^{\wedge}$ 和 $^{\vee}$ 符号来指代“从向量到矩阵”和“从矩阵到向量”的关系，以保持和 $\mathfrak{so}(3)$ 上的一致性。

与 $\mathfrak{so}(3)$ 类似，$\mathfrak{se}(3)$ 上也有指数映射，$\mathfrak{se}(3)$ 上的指数映射[34]的形式如下：

$$\begin{aligned}\exp\left(\boldsymbol{\xi}^{\wedge}\right)&=\begin{bmatrix}\sum_{n=0}^{\infty}\frac{1}{n!}\left(\boldsymbol{\phi}^{\wedge}\right)^n & \sum_{n=0}^{\infty}\frac{1}{(n+1)!}\left(\boldsymbol{\phi}^{\wedge}\right)^n\boldsymbol{\rho}\\ 0^T & 1\end{bmatrix}\\ &\triangleq\begin{bmatrix}\boldsymbol{R} & \boldsymbol{J}\boldsymbol{\rho}\\ 0^T & 1\end{bmatrix}=\boldsymbol{T}\end{aligned} \tag{2-32}$$

将 exp 进行泰勒展开推导此，从从结果上看，$\mathfrak{se}(3)$ 的指数映射左上角的 R 是我们熟知的 SO(3)中的元素，与 $\mathfrak{se}(3)$ 当中的旋转部分 $\boldsymbol{\phi}$ 对应。而右上角的 J 则可整理为(设 $\boldsymbol{\phi}=\theta\boldsymbol{a}$):

$$\boldsymbol{J}=\frac{\sin\theta}{\theta}\boldsymbol{I}+\left(1-\frac{\sin\theta}{\theta}\right)\boldsymbol{a}\boldsymbol{a}^T+\frac{1-\cos\theta}{\theta}\boldsymbol{a}^{\wedge} \tag{2-33}$$

同样的，虽然我们也可以类比推得对数映射，不过根据变换矩阵 T 求 $\mathfrak{so}(3)$ 上的对应向量有更方便的方式：从左上的 R 计算旋转向量，而右上的 t 满足：

$$\boldsymbol{t} = \boldsymbol{J}\rho \tag{2-34}$$

2.5 SLAM 问题模型框架

2.5.1 SLAM 问题的数学表述

SLAM 是要解决机器人在未知环境中定位并同时建图的问题，也就是给定一系列传感器输入，计算机器人所在的位置以及周围环境信息的问题。

由于 SLAM 要同时确定周围环境的信息与自身的定位，机器人身上一定有观测外部环境的信息的传感器，一般为相机或激光雷达。假设机器人在 k 时刻位于 x_k 处探测到一个路标 y_k，我们可以用式（2-35）来描述这个过程。

$$z_{k,j} = h(y_j, x_k, v_{k,j}) \tag{2-35}$$

其中 $z_{k,j}$ 代表机器人在 x_k 对路标 y_k 的一个观测数据，$v_{k,j}$ 代表观测的噪声，h 为观测函数，这个方程称为观测方程。需要注意的是，我们用函数 h 来描述这个过程，而不具体指明 h 的作用方式。这使得它可以指代任意的观测传感器，成为一个通用的方程，而不必局限于某个特定的传感器上。实际问题中该方程根据传感器的类型具体化，如传感器为相机，则该观测方程即为 2.2 节中介绍的成像模型。

仅用一种观测传感器可以完成 SLAM 的求解，如纯视觉或纯激光 SLAM。但实际机器人可能携带更多种传感器，比较典型的就是运动传感器如 IMU 或码盘等。因此 SLAM 也可以融合这些传感器提供的信息，使整个 SLAM 系统能够应对如场景动态变化，特征缺失，或自身运动速度过快等更难处理的情况。假设运动传感器在 $k-1$ 到 k 的之间输出为 $u_{k\text{-}1:k}$，可以用式（2-36）对描述机器人的运动。

$$x_k = f(x_{k-1}, u_{k-1:k}, w) \tag{2-36}$$

其中 x_{k-1}、x_k 代表机器人在 k -1 和 k 时刻的状态，w 代表运动传感器与运动方程的噪声，f 为描述运动的函数，这个方程称为运动方程。与观测方程一样，也是可以指代任意运动传感器。

更一般的 SLAM 过程可以总结成以上两个方程：

$$\begin{cases} x_k = f(x_{k-1}, u_{k-1:k}, w) \\ z_{k,j} = h(y_j, x_k, v_{k,j}) \end{cases} \tag{2-37}$$

这两个方程描述了最基本的 SLAM 问题：当知道运动测量的读数 u，以及传感器读数 z 时，如何求解定位问题（估计 x）和建图问题（估计 y）。这样我们可以把 SLAM 问题建模成了一个状态估计问题。这样我们可以把

SLAM 问题建模成了一个状态估计问题。由于估计具有不确定性，在数学上一般用概率分布来表示估计，所以，SLAM 问题用概率表达就是求解条件概率分布 $P(y,x|z,u)$。

2.5.2 经典 SLAM 框架

视觉 SLAM 经过十几年的发展，已经形成了一个基本的框架[35]，如图 2-7 所示：

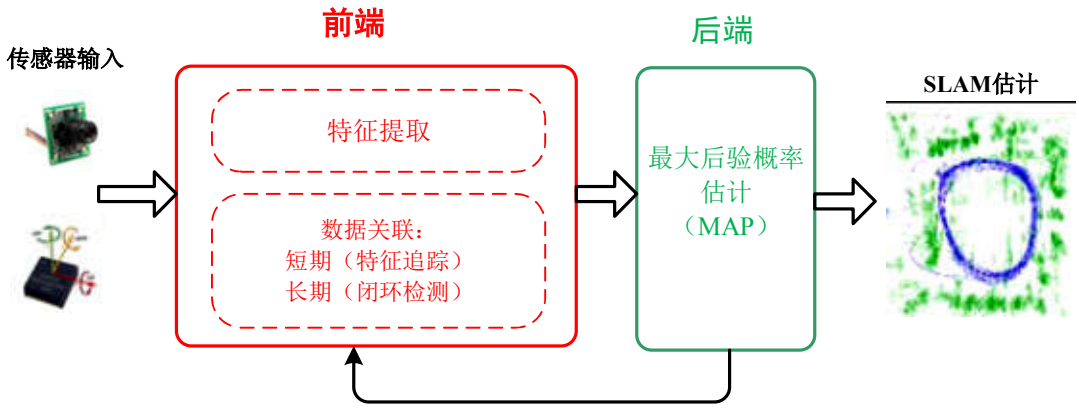


图 2-7 经典 SLAM 框架中的前端和后端

通常可以将 SLAM 问题的求解分为两部分：前端和后端。前端负责直接对传感器采集到的数据进行处理，后端负责利用前端处理后的数据生成全局一致的轨迹和地图。

前端的数据关联方式可分为短期和长期两类。其中短期的数据关联是指相邻帧之间的特征跟踪和匹配，再根据特征的匹配关系进行对运动的估计和地图的构建;长期的数据关联是指闭环检测，系统通过传感器的观测来判断是否返回了之前曾访问过的场景。

后端要解决的问题，就是在已给定的带有噪声的观测下，对整个系统进行状态估计，并求解状态估计的不确定性。这里的状态既包含相机的位姿，又包含地图中的路标。由于本文主要采用基于非线性优化的后端方法，因此下面将主要对其进行介绍：

在非线性优化中，所有待估计的变量放在一个状态变量中：

$$x=\{x_1,\cdots,x_N,\mathrm{y}_1,\cdots,\mathrm{y}_M\} \tag{2-38}$$

对机器人状态的估计，就是求已知输入数据 u 和观测数据 z 的条件下，计算状态 x 的条件概率分布：

$$P(x|z,u) \tag{2-39}$$

类似于 x，这里 u 和 z 也是对所有数据的统称。特别地，当没有测量运动的传感器，只有一张张的图像时，即只考虑观测方程带来的数据时，相当于估计 $P(x|z)$的条件概率分布。如果忽略图像在时间上的联系，把它们看作一堆彼此没有关系的图片，该问题也称为从运动中恢复结构（SFM，Structure from Motion），即如何从许多图像中重建三维空间结构。在这种情况下，SLAM 可以看作是图像具有时间先后顺序的，需要实时求解一个 SFM 问题。为了估计状态变量的条件分布，利用贝叶斯法则，有：

$$P(x|z)=\frac{P(z|x)P(x)}{P(z)}\propto P(z|x)P(x) \tag{2-40}$$

贝叶斯法则左侧称为后验概率。它右侧的 $P(z|x)$称为似然，另一部分 $P(x)$称为先验。直接求后验分布是困难的，但是求一个状态最优估计，使得在该状态下，后验概率最大化是可行的：

$$x^*_{MAP}=\arg\max P(x|z)=\arg\max P(z|x)P(x) \tag{2-41}$$

贝叶斯法则的分母部分与待估计的状态 x 无关，因而可以忽略。求解最大后验概率，相当于最大化似然和先验的乘积。由于对机器人位姿的先验分布未知，可以求解 x 的最大似然估计(Maximize Likelihood Estimation, MLE)：

$$x^*_{MLE}=\arg\max P(z|x) \tag{2-42}$$

似然是指在现在的位姿下，可能产生怎样的观测数据。由于观测数据是已知的，最大似然估计的直观意义为：在什么样的状态下，最可能产生现在观测到的数据。在高斯分布的假设下，最大似然能够有较简单的形式。根据观测模型，对于某一次观测：

$$z_{k,j}=h(y_j,x_k)+v_{k,j} \tag{2-43}$$

由于假设了噪声项 $v_k\sim N(0,Q_{k,j})$，所以观测数据的条件概率为：

$$P(z_{k.j}|x_k,y_j)=N(h(y_j,x_k),Q_{k,j}) \tag{2-44}$$

它依然是一个高斯分布。为了计算使它最大化的 x_k，y_j 式，来求一个高斯分布的最大似然。

高斯分布在负对数下有较好的数学形式。考虑一个任意的高维高斯分布 $x\sim N(\mu,\Sigma)$，它的概率密度函数展开形式为：

$$P(x)=\frac{1}{\sqrt{(2\pi)^N\det(\Sigma)}}\exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right) \tag{2-45}$$

取它的负对数，则变为：

$$-\ln\left(P\left(x\right)\right)=\frac{1}{2}\ln\left(\left(2\pi\right)^{N}\det\left(\Sigma\right)\right)+\frac{1}{2}\left(x-\mu\right)^{T}\Sigma^{-1}\left(x-\mu\right) \tag{2-46}$$

对原分布求最大化相当于对负对数求最小化。在最小化上式的 x 时，第一项与 x 无关，可以略去。于是，只要最小化右侧的二次型项，就得到了对状态的最大似然估计。代入 SLAM 的观测模型，相当于在求：

$$x^{*}=\arg\min\left(\left(z_{k,j}-h\left(x_{k},y_{j}\right)\right)^{T}Q_{k,j}^{-1}\left(z_{k,j}-h\left(x_{k},y_{j}\right)\right)\right) \tag{2-47}$$

该式等价于最小化噪声项（即误差）的平方（Σ 范数意义下）。因此，对于所有的运动和任意的观测，可以定义数据与估计值之间的误差：

$$\begin{aligned}e_{v,k}&=x_{k}-f\left(x_{k-1},u_{k}\right)\\e_{y,j,k}&=z_{k,j}-h\left(x_{k},y_{j}\right)\end{aligned} \tag{2-48}$$

2.6 本章小结

本章节主要介绍了惯性视觉 SLAM 中的数学理论与模型，首先阐述了针孔相机模型及世界，相机，IMU 之间坐标系之间的转换关系；介绍表示惯性元件运动姿态所使用的四元数，包括其基本的定义及其如何表征旋转运动等；然后引入了李群与李代数来表征刚体运动，简要介绍了其基本定义和计算；最后对 SLAM 问题进行数学建模，给出了基于图优化方法的 SLAM 的框架，推导了基于最大后验概率的后端优化方法。

第3章 基于单目视觉与 IMU 的位姿估计

3.1 引言

本文的目的是对单目视觉与 IMU 结合的 SLAM 技术进行研究。但其实两种传感器分别可以独立进行位姿运动估计。故在研究两种传感器融合算法之前，首先分别介绍基于纯单目视觉和基于纯 IMU 的位姿估计方法。这些位姿估计的方法通常在局部是有效的，属于 2.5.2 节所述的 SLAM 框架的前端部分。

基于视觉的 SLAM 方法是通过视觉传感器在机器人运动过程中拍摄不同的图像，通过检测这些图像的变化，提取并对特征点进行匹配，通过判断特征点的运动变化情况来估计机器人的运动情况。基于特征提取的方法能通过跟踪图像中的特征点，并计算其在连续图像间的位置变化，以求解机器人的位姿变化。本章将介绍系统所用的特征提取与跟踪方法，并介绍三种基于多视图特征匹配的位姿估计方法，这些方法在本系统中都会用到。

与视觉不同，IMU 并不依赖外界信息而只关注自身的运动状态，测量机器人自身的三轴角速度和三轴加速度。由于 IMU 的测量频率比视觉传感器高很多，而在 SLAM 问题中，我们相比于获取每个 IMU 测量值时的机器人状态更关注拍摄每一帧图像时的状态，甚至在运算能力不足时还要提取关键帧进一步减少待估计的变量。因此，如何将更高频率的 IMU 数据与帧率较低的相机图像对齐成为首先要解决的问题。本章将介绍如何利用 IMU 预积分的方式，将 IMU 的测量值积分为视觉帧间的位姿变换。

3.2 视觉特征提取与跟踪

在惯性视觉里程计中，视觉的部分根据相机在机器人运动过程中，需要对相同场景在相机不同位置时拍摄的图像帧的像素点进行跟踪，这就需要对前后帧的提取的特征点进行匹配和跟踪。然而连续的图像帧之间往往只有一部分重叠，不适宜将所有的像素点进行匹配，这将花费大量的计算资源。并且根据特征点匹配计算相机的位姿关系理论上并不需要太多的特征点。因此，在视觉里程计中，通常只提取部分图像的特征点，即图像中特征最明显的点，包括角点，亮度显著变化的点或轮廓边缘的点，之后对提取到的特征点进行跟踪。本文采用的特征点是 Shi-Tomas 角点，特征跟踪方法是 KLT 光流法。

3.2.1 Shi-Tomas 角点提取

Shi-Tomas 角点检测算法由 J.shi 和 C.Tomasi[36]在 1994 年提出，它是对 Harris 算法的一个改进算法，并取得了很好的效果。由于该算法主体框架与 Harris 算法相同，因此首先介绍 Harris 算法。

经典 Harris 算法是上世纪 80 年代末，C.Harris 等人[37]在 Alvey 视觉会议上提出的角点特征提取算法。其基本原理是将目标像素点作为中心，计算其窗口内的灰度曲率的变化情况，选取曲率变化值最大点作为特征点。其计算方法如下：

设 $G_{x,y}$ 为像素点 (x,y) 的邻域窗口 W 移动 (u,v) 时灰度变化量，当移动 (u,v) 为局部极小量时，根据 Talyor 级数展开进行对角化处理和二次型计算后可以得到:

$$\begin{aligned} G_{x,y} &= \sum_{x,y} w_{x,y}(I_{x+u,y+v} - I_{x,y}) \\ &= \sum_{x,y} w_{x,y}(u\frac{\partial I}{\partial x} + v\frac{\partial I}{\partial y} + o(\sqrt{u^2+v^2}))^2 \\ &\approx \sum_{x,y} w_{x,y}[u^2(I_x)^2 + v^2(I_y)^2 + 2uvI_xI_y] \\ &= [u \quad v]M\begin{bmatrix} u \\ v \end{bmatrix} \\ &= R^{-1}\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}R \end{aligned} \tag{3-1}$$

其中，I 为图像灰度，I_x、I_y 分别为 x、y 方向上的一阶偏导数，$M = \sum_{x,y} w_{x,y}\begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix}$ 为图像梯度的协方差矩阵，$w_{x,y} = e^{(x^2+y^2)/\sigma^2}$ 为窗口 W 的窗口函数，作用是降噪平滑，λ_1、λ_2 为矩阵 M 的特征值，R 为二维旋转矩阵。

如图3-1所示，特征值 λ_1、λ_2 表明了图像像素灰度值的分布情况。当 λ_1、λ_2 都比较小时，图像窗口在任意方向上移动都无明显灰度变化，说明图像窗口内对应的像素点处于平滑区域，当 λ_1、λ_2 中有一个较大，另一个较小时，图像窗口在一个方向上灰度变化较明显，另一方向上无明显灰度变化，说明图像窗口内对应像素点位于物体的边界处，当 λ_1、λ_2 都比较大时，沿着任意方向移动图像窗口内的像素点的曲率或灰度梯度变化都较大，该点即为所求的特征点。

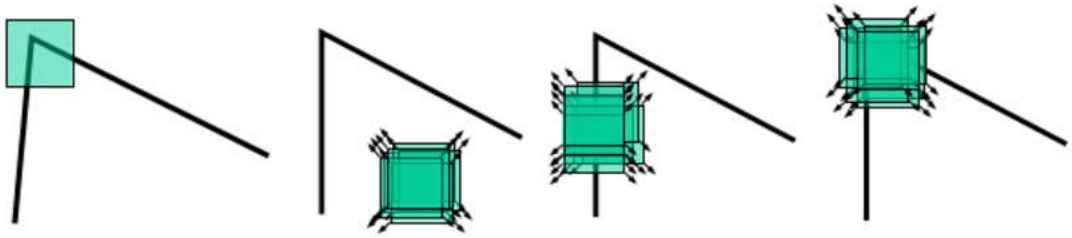


图 3-1 窗口滑动示意图

当对一整幅图像进行特征点提取时，需要对图像内所有像素点进行特征值的计算，计算量比较大。由于矩阵 M 为实对称矩阵，设

$$M=\sum_{x,y}w_{x,y}\begin{bmatrix}I_x^2 & I_xI_y\\ I_xI_y & I_y^2\end{bmatrix}=\begin{bmatrix}A & C\\ C & B\end{bmatrix} \tag{3-2}$$

则

$$tr(M)=\lambda_1+\lambda_2=A+B\ \ det(M)=\lambda_1\lambda_2=AB-C^2 \tag{3-3}$$

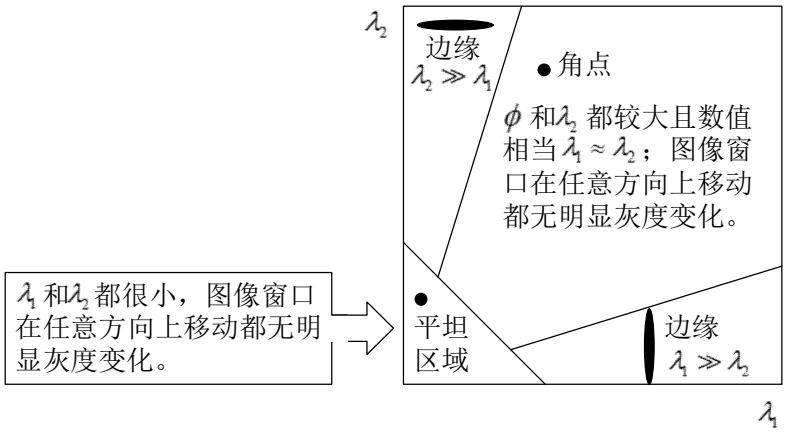


图 3-2 λ_1、λ_2 与像素点的分布情况

其中，$tr(M)$ 为矩阵 M 的迹，$det(M)$ 为其行列式。为了提高计算效率，通常利用 $tr(M)$ 和 $det(M)$ 避免 λ_1、λ_2 的求取。由此，得到角点特征检测函数

$$R(x,y)=det(M)-k(tr(M))^2=(AB-C^2)-k(A+B)^2 \tag{3-4}$$

这里，$k=0.04\sim0.06$，为经验常数。

给定阈值 T，对像素点 (x,y)，当 $R(x,y)>T$，且为局部邻域窗口内的最大值时，像素点 (x,y) 即为所求角点。

综上所述，可以得到经典 Harris 算法的框图，如图 3-3 所示。

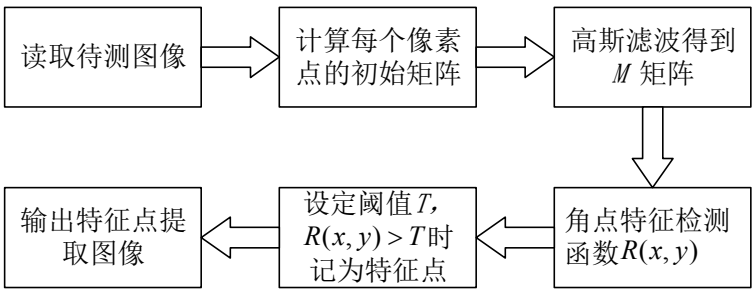


图 3-3 经典 Harris 算法框图

Shi 和 Tomasi 后来对经典 Harris 进行了改进，算法原始的定义是将矩阵 M 的行列式值与 M 的迹相减，再将差值与预先给定的阈值 T 进行比较。Shi 和 Tomasi 提出改进的方法，如果两个特征值中比较小的一个大于最小阈值，那么就会得到强角点。即：

$$R(x,y)=\min(\lambda_1,\lambda_2) \tag{3-5}$$

由于较大的不确定度取决于较小的特征值，故可以通过找小特征值的最大值来寻找好的特征点。

Shi-Tomas 算法仅修改了 $R(x,y)$ 的计算方式，其余与 Harris 算子无差异，故其算法框图也为图 3-2。Shi-Tomas 算法在大多数情况下都会得到比 Harris 算法更好的结果。该算子可利用 Opencv 中的 goodFeaturesToTrack 函数实现。

3.2.2 KLT 光流法跟踪

KLT 算法全称为 Kanade-Lucas-Tomasi 方法。Kanade 和 Lucas 在 1981 年首先提出一种通过求解偏移量 d 来进行图像匹配的方法[38]，1994 年 Jianbo Shi 和 Carlo Tomasi 对其进行了补充，说明了哪些情况下一定能找到 d 的解。在过去的这二十多年中，KLT 算法被广泛的应用于各种运动跟踪。

KLT 算法是基于特征点的跟踪方法，对连续两帧图像的相同特征点的进行跟踪。KLT 算法有三个假设前提：1) 图像要保持亮度恒定；2) 运动时间连续或者运动是小运动；3) 空间一致，临近点的运动变化相似。这三点在通常情况下可以得到满足。

假设 I 和 J 为图像序列中连续的两幅图，点 (x,y) 的灰度值分别记为 $I(x,y)$ 和 $J(x,y)$。根据 KLT 算法的前提条件，图像 I 和 J 在某个小邻域内运动变化相似，设前一帧图像 I 的某个特征点坐标为 u，在下一帧 J 与之对应的特征点坐标为 v ，点 v 由点 u 移动一段距离 d 得到，可得 $v=u+d$，即 $t+1$ 时刻的点 v 可由 t 时刻的 u 移动 d 得到，KLT 算法的目的就是求位移的变化量 d。

KLT 求解时用到的是特征点何其周围邻域对应的一个小窗口图像块，设 ω_x 和 ω_y 分别是点 u 左右扩展的窗口范围，窗口大小为(2ω_x+1)×(2ω_y+1)，求解 d 就是求解函数 $\varepsilon(d)$ 的最小值，见公式(3-6)：

$$\varepsilon(d)=\sum_{x=u_x-\omega_x}^{u_x+\omega_x}\sum_{y=u_y-\omega_y}^{u_y+\omega_y}\left(I(x,y)-J\left(x+d_x,y+d_y\right)\right)^2 \tag{3-6}$$

用积分表示(3-6)，可等效为：

$$\varepsilon(d)=\iint\left(J\left(x+\frac{d}{2}\right)-I\left(x-\frac{d}{2}\right)\right)^2 w(x)dx \tag{3-7}$$

可通过求解向量 d 的偏导使其等于 0，来求解 $\varepsilon(d)$ 的最小值，得：

$$\frac{\partial_\varepsilon}{\partial_d}=2\iint\left(J\left(x+\frac{d}{2}\right)-I\left(x-\frac{d}{2}\right)\right)\left(\frac{\partial J\left(x+\frac{d}{2}\right)}{\partial d}-\frac{\partial I\left(x-\frac{d}{2}\right)}{\partial d}\right)w(x)dx \tag{3-8}$$

利用泰勒级数展开得：

$$J(\xi)\approx J(a)+(\xi_x-a_x)\frac{\partial J}{\partial x}(a)+(\xi_y-a_y)\frac{\partial J}{\partial y}(a) \tag{3-9}$$

根据式(3-9)对 $J\left(x+\frac{d}{2}\right)$ 和 $I\left(x-\frac{d}{2}\right)$ 进行泰勒展开可以得：

$$\begin{aligned}J\left(x+\frac{d}{2}\right)&\approx J(x)+\frac{d_x}{2}\frac{\partial J}{\partial x}(x)+\frac{d_y}{2}\frac{\partial J}{\partial y}(x)\\ I\left(x-\frac{d}{2}\right)&\approx I(x)-\frac{d_x}{2}\frac{\partial I}{\partial x}(x)+\frac{d_y}{2}\frac{\partial I}{\partial y}(x)\end{aligned} \tag{3-10}$$

代入到(3-8)可得：

$$\frac{\partial_\varepsilon}{\partial_d}\approx\iint\left(J(x)-I(x)+g^Td\right)g(x)w(x)dx \tag{3-11}$$

其中，g 的表达式为：

$$g=\left[\frac{\partial}{\partial x}\left(\frac{I+J}{2}\right)\ \frac{\partial}{\partial y}\left(\frac{I+J}{2}\right)\right]^T \tag{3-12}$$

最终需要求解的是：

$$\frac{\partial_\varepsilon}{\partial_d}=\iint\left(J(x)-I(x)+g^T(x)d\right)g(x)w(x)dx=0 \tag{3-13}$$

打开后得：

$$\iint\left[J(x)-I(x)\right]g(x)w(x)dx=-\iint g^T(x)dg(x)w(x)dx \tag{3-14}$$

单独提出 d 后得：

$$-\iint g^T(x)dg(x)w(x)dx=-\left[\iint g(x)g^T(x)w(x)dx\right]d \tag{3-15}$$

可简写为：

$$Zd=e \tag{3-16}$$

其中 Z 和 e 分别为：

$$\begin{aligned} Z&=\iint g(x)g^T(x)w(x)dx \\ e&=\iint\left[I(x)-J(x)\right]g(x)w(x)dx \end{aligned} \tag{3-17}$$

如果要 d 有解必须保证 Z 是可逆的，Jianbo Shi 和 Carlo Tomasi 证明了角点可保证 Z 可逆，本文即采用了他们提出的 Shi-Tomasi 角点。

公式(3-16)求解偏移量 d 的时候，通过迭代的方式进行计算，可以得到一个较为准确的位移矢量。对于第 $k(k\geq 1)$ 次迭代，设第 k -1次迭代得到的位移为 $d^{k-1}=[d_x^{k-1}d_y^{k-1}]$，则第 k 次迭代时，$J(x,y)$ 可用第 k -1次迭代表示：

$$J(x,y)=J\left(x+d_x^{k-1},y+d_y^{k-1}\right) \tag{3-18}$$

式(3-6)变为：

$$\varepsilon(d)=\sum_{x=u_x-w_x}^{u_x+w_x}\sum_{y=u_y-w_y}^{u_y+w_y}\left(I(x,y)-J\left(x+d_x^{k-1},y+d_y^{k-1}\right)\right)^2 \tag{3-19}$$

经过几次标准的算法，得到第 k 次的位移 d^k：

$$d^k=Z^{-1}e_k \tag{3-20}$$

假设经过 K 次迭代后收敛，最终位移 d 可表示为：

$$d=\sum_{K=1}^{K}d^k \tag{3-21}$$

通过位移 d 我们便可以得到两帧图像特征点间的运动关系，来对特征点进行跟踪，也就可以将特征点匹配起来。

3.3 基于多视图几何的位姿估计

上一节介绍了视觉特征点的提取和追踪方法，本节将根据不同视图中追踪到的特征点恢复来获得位姿估计的方法。根据图像间特征点匹配从而计算拍摄图像的相机间的位姿变化关系方法可分为两大类:已知特征点的三维信息和特征点三维信息未知。其中特征点信息未知的一类可叫做 2D-2D 方法恢复位姿，已知的一类称为 2D-3D 的方法，也可以叫做 N 点透视定位（Perspective n Points，PnP）。

3.3.1 对极几何恢复位姿

3.3.1.1 基于本质矩阵 E 的位姿恢复

利用对极几何研究图像间的约束关系，是利用几何不变量解决透视投影问题的关键。如图 3-4 所示，O_1和O_2分别表示两个视图中相机的光学中心。P为空间中一点，p_1为其在第一幅图像I_1中的像，p_2为其在第二幅图像I_2中的像。在这种情况下，p_2被约束在p_1位于图像I_2的极线l_2上，相对应地，p_1被约束图像I_1中的极线l_1上。图像I_1中的像点位于图像I_2中的极线都通过光学中心连线O_1O_2和像平面I_1的交点e_2，相对应地，图像I_2中的像点位于图像I_1中的极线也都通过光学中心连线O_1O_2和像平面I_1的交点e_1。在不同视图同一场景的图像匹配过程中，对于第一张图像中的任意一点，该点在第二张图像中所对应的点一定位于该点在第二张图像中所对应的极线上。而该约束关系，又与不同视图的相机旋转平移变化有关。因此，通过多组已知二维点对匹配可反求解出对极几何约束条件，进而恢复出不同视图间相机的旋转平移变化。

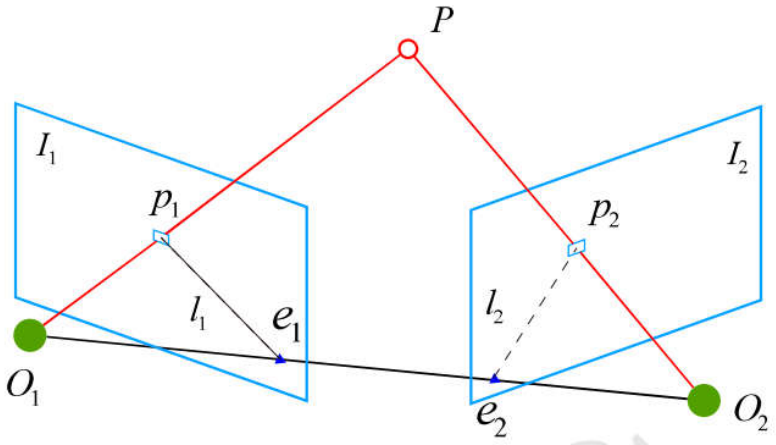


图 3-4 对极几何约束

在这种对极几何约束下，上述两个相机系统的位姿关系可由旋转矩阵 R 和转移向量 t 来描述，用公式表达为：

$$x_2^T t^\wedge R x_1 = 0 \tag{3-22}$$

其中^为 2.4 节中介绍的反对称矩阵转化符号。x_1, x_2分别为两个像素点p_1, p_2在归一化平面下的坐标。它的几何意义为像素点点p_1, p_2的归一化平面的坐标在O_1PO_2三点构成的平面上。对极几何约束中同时包含了平移和旋转。中间的部分称作本质矩阵 **E**（Essential Matrix），可简化对极约束：

$$E = t^\wedge R, \quad x_2^T E x_1 = 0 \tag{3-23}$$

对极约束简洁地给出了两个匹配点的空间位置关系。于是，相机位姿估计问题可分解为：首先根据像素点对的匹配求出本质矩阵 E，再根据 E 求出旋转平移变化 R, t。

本质矩阵可由八对点来进行估计估计，这就是经典的八点法(Eight-point-algorithm)[40-41]八点法只利用了 E 的线性性质，因此可以在线性代数框架下求解。

考虑一对匹配点，它们的归一化坐标为：$x_1=[u_1,v_1,1]^T$，$x_2=[u_2,v_2,1]^T$。根据对极约束，有：

$$(u_1,v_1,1)\begin{pmatrix}e_1 & e_2 & e_3\\ e_4 & e_5 & e_6\\ e_7 & e_8 & e_9\end{pmatrix}\begin{pmatrix}u_2\\ v_2\\ 1\end{pmatrix}=0 \tag{3-24}$$

把矩阵 $\boldsymbol{E}$ 展开，写成向量的形式：

$$e=[e_1,e_2,e_3,e_4,e_5,e_6,e_7,e_8,e_9]^T \tag{3-25}$$

那么对极约束可以写成与 e 有关的线性形式：

$$[u_1u_2,u_1v_2,u_1,v_1u_2,v_1,u_2,v_2,1]\cdot e=0 \tag{3-26}$$

同理，对于其它点对也有相同的表示。我们把所有点都放到一个方程中，变成线性方程组(u^i, v^i 表示第 i 个特征点，以此类推)：

$$\begin{pmatrix}u_1^1u_2^1 & u_1^1v_2^1 & u_1^1 & v_1^1u_2^1 & v_1^1v_2^1 & v_1^1 & u_2^1 & v_2^1 & 1\\ u_1^2u_2^2 & u_1^2v_2^2 & u_1^2 & v_1^8u_2^8 & v_1^2v_2^2 & v_1^2 & u_2^2 & v_2^2 & 1\\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\ u_1^8u_2^8 & u_1^8v_2^8 & u_1^8 & v_1^8u_2^1 & v_1^8v_2^8 & v_1^8 & u_2^8 & v_2^8 & 1\end{pmatrix}\begin{pmatrix}e_1\\ e_2\\ e_3\\ e_4\\ e_5\\ e_6\\ e_7\\ e_8\\ e_9\end{pmatrix}=0 \tag{3-27}$$

这八个方程构成了一个线性方程组。它的系数矩阵由特征点位置构成，大小为8×9。e 位于该矩阵的零空间中。如果系数矩阵是满秩的（即秩为 8），那么它的零空间维数为 1，也就是 e 构成一条线。这与 e 的尺度等价性是一致的。如果八对匹配点组成的矩阵满足秩为 8 的条件，那么 E 的各元素就可由上述方程解得。

接下来的工作是根据估得的本质矩阵 $\boldsymbol{E}$，恢复出相机的运动 $\boldsymbol{R}$，$\boldsymbol{t}$。这个过程是由奇异值分解（SVD）得到的。设 $\boldsymbol{E}$ 的 SVD 分解为：

$$t_1^{\wedge} = UR_Z\left(\frac{\pi}{2}\right)\Sigma U^T, R_1 = UR_Z^T\left(\frac{\pi}{2}\right)V^T$$
$$t_2^{\wedge} = UR_Z\left(-\frac{\pi}{2}\right)\Sigma U^T, R_1 = UR_Z^T\left(-\frac{\pi}{2}\right)V^T \tag{3-28}$$

其中 $R_Z\left(\frac{\pi}{2}\right)$ 表示沿 Z 轴旋转 90 度得到的旋转矩阵。同时，由于-E 和 E 等价，所以对任意一个 t 取负号，也会得到同样的结果。因此，从 E 分解到 t, R 时，一共存在四个可能的解。

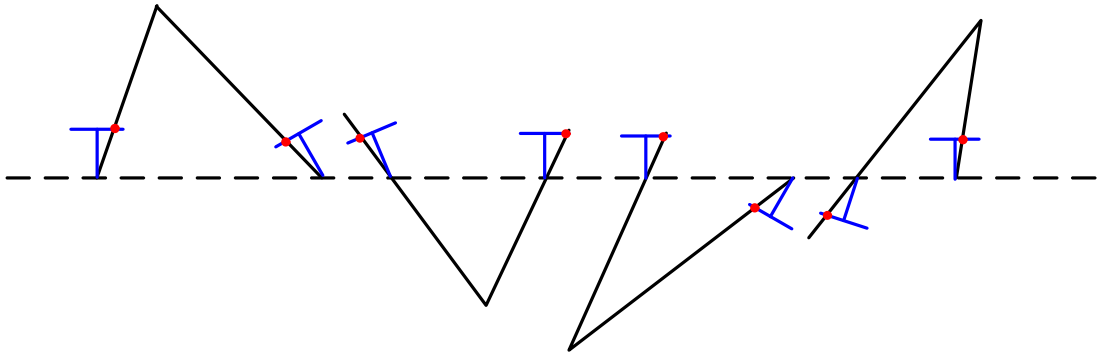


图 3-5 分解本质矩阵得到的四个解

图 3-5 表示在保持投影点(红点)不变的情况下，两个相机以及空间点一共有四种可能的情况。图形象地显示了分解本质矩阵得到的四个解。我们已知空间点在相机（蓝色线）上的投影（红点），想要求解相机的运动。在保持红点不变的情况下，可以画出四种可能的情况，只有第一种解中，P 在两个相机中都具有正的深度。因此，只要把任意一点代入四种解中，检测该点在两个相机下的深度，即可确定正确的解。

需要注意的是，根据线性方程解出的 E，可能不满足 E 的内在性质，即它的奇异值不一定为 $\sigma,\sigma,0$ 的形式。这时，在进行 SVD 时，我们需要把 Σ 矩阵调整成上面的样子。通常的做法是，对八点法求得的 E 进行 SVD 分解后，会得到奇异值矩阵 $\Sigma = diag(\sigma_1,\sigma_2,\sigma_3)$，不妨设 $\sigma_1 \geq \sigma_2 \geq \sigma_3$。取：

$$E = U\mathrm{diag}\left(\frac{\sigma_1+\sigma_2}{2}, \frac{\sigma_1+\sigma_2}{2}, 0\right)V^T \tag{3-29}$$

这相当于是把求出来的矩阵投影到了 E 所在的流形上。

3.3.1.2 基于单应矩阵 H 的位姿恢复

实际上，并不是所有的场景都适合用本质矩阵 E 去估计，若场景中的特

征点都落在同一平面上(如墙面，地面等)，本质矩阵的自由度下降，则会发生退化的情况。现实中的数据总包含一些噪声，这时候如果继续使用八点法求解基础矩阵，基础矩阵多余出来的自由度将会主要由噪声决定。

这种情况可以用单应矩阵 H(Homography Matrix)估计两帧之间的运动关系。单应矩阵描述了两个平面之间的映射关系，也可从中恢复出相机的位姿变化，适用于场景特征点都在一个平面上的情况。

记单应矩阵为 H，两帧图像齐次坐标分别为 p_2, p_1，单应矩阵关系可用下式描述：

$$p_2 = Hp_1 \tag{3-30}$$

单应矩阵描述了两帧图像之间像素点的映射关系，它的定义与旋转、平移以及平面的参数有关。与本质矩阵 E 类似，单应矩阵 H 也是一个 3x3 的矩阵，求解时的思路与 E 相类似，同样地可以先根据匹配点计算 H，然后将它分解以计算旋转和平移。把上式展开，得:

$$\begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} \tag{3-31}$$

需要注意的是这里的等号是在非零因子下成立的。在实际处理中，通常乘以一个非零因子使得 $h_9 = 1$(在它取非零值时)。然后根据第三行，去掉这个非零因子，可以整理得：

$$\begin{aligned} h_1u_1 + h_2v_1 + h_3 - h_7u_1u_2 - h_8u_1u_2 = u_2 \\ h_4u_1 + h_5v_1 + h_6 - h_7v_1v_2 - h_8v_1v_2 = v_2 \end{aligned} \tag{3-32}$$

这样一组匹配点对就可以构造出两项约束，于是在特征点没有三点贡献的情况下，自由度为 8 的单应矩阵可以通过 4 对匹配特征点算出，即求解以下的线性方程组 $h_9 = 0$时，右侧为零，通过解该向量的线性方程来恢复 H：

$$\begin{pmatrix} u_1^1 & v_1^1 & 1 & 0 & 0 & 0 & -u_1^1u_2^1 & -v_1^1u_2^1 \\ 0 & 0 & 0 & u_1^1 & v_1^1 & 1 & -u_1^1v_2^1 & -v_1^1v_2^1 \\ u_1^2 & v_1^2 & 1 & 0 & 0 & 0 & -u_1^2u_2^2 & -v_1^2u_2^2 \\ 0 & 0 & 0 & u_1^2 & v_1^2 & 1 & -u_1^2v_2^2 & -v_1^2v_2^2 \\ u_1^3 & v_1^3 & 1 & 0 & 0 & 0 & -u_1^3u_2^3 & -v_1^3u_2^3 \\ 0 & 0 & 0 & u_1^3 & v_1^3 & 1 & -u_1^3v_2^3 & -v_1^3v_2^3 \\ u_1^4 & v_1^4 & 1 & 0 & 0 & 0 & -u_1^4u_2^4 & -v_1^4u_2^4 \\ 0 & 0 & 0 & u_1^4 & v_1^4 & 1 & -u_1^4v_2^4 & -v_1^4v_2^4 \end{pmatrix} \begin{pmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \end{pmatrix} = \begin{pmatrix} u_2^1 \\ v_2^1 \\ u_2^2 \\ v_2^2 \\ u_2^3 \\ v_2^3 \\ u_2^4 \\ v_2^4 \end{pmatrix} \tag{3-33}$$

与本质矩阵相似，求出单应矩阵以后需要对其进行分解，才可以得到相应的旋转矩阵 R 和平移向量 t。分解的方法包括数值法[42-43]与解析法[44]。与

本质矩阵的分解类似，单应矩阵的分解同样会返回四组旋转矩阵与平移向量，并且同时可以计算出它们分别对应的场景点所在平面的法向量。如果已知成像的地图点的深度全为正值(即在相机前方)，则又可以排除两组解。最后仅剩两组解，这时需要通过更多的先验信息进行判断。通常我们可以通过假设已知场景平面的法向量来解决，如场景平面与相机平面平行，那么法向量**n**的理论值为$\mathbf{1}^{\mathrm{T}}$。

单应性在 SLAM 中具重要意义。当特征点共面，或者相机发生纯旋转的时候，基础矩阵的自由度下降，这就出现了所谓的退化(degenerate）现实中的数据总包含一些噪声，这时候如果我们继续使用八点法求解基础矩阵，基础矩阵多余出来的自由度将会主要由噪声决定。为了能够避免退化现象造成的影响，在本文中会同时估计基础矩阵 F 和单应矩阵 H，选择重投影误差比较小的那个作为最终的运动估计矩阵。

3.3.2 透视 N 点定位

透视 N 点定位是求解 3D 到 2D 点对运动的方法。它描述了当我们知道 n 个 3D 空间点以及它们的投影位置时，如何估计相机所在的位姿。上一节 2D-2D 的对极几何方法需要八个或八个以上的点对(以八点法为例)，且存在着初始化、纯旋转和尺度的问题。然而，如果两张图像中，其中一张特征点的 3D 位置已知，那么最少只需三个点对(需要至少一个额外点验证结果)就可以估计相机运动。要想使用 PnP 方法，必须提前获得特征点的 3D 位置，特征点的 3D 位置可以由三角化或由 RGB-D 相机的深度图确定。但由于本系统中仅有单目相机，因此在本系统首先进行初始化，即通过对极几何获得相机位姿变化而后通过三角过获取特征点的 3D 位置，然后才使用 PnP。3D-2D 方法不需要使用对极约束，又可以在很少的匹配点中获得较好的运动估计，是最重要的一种姿态估计方法。它的解法如下所示:

考虑某个空间点 P，它的齐次坐标为 $P=(X,Y,Z,1)^T$。在图像 I_1 中，投影到特征点 $x_1=(u_1,v_1,1)^T$（以归一化平面齐次坐标表示）。此时相机的位姿 R,t 是未知的。与单应矩阵的求解类似，我们定义增广矩阵$[R|t]$为一个3×4的矩阵，包含了旋转与平移信息。我们把它的展开形式列写如下：

$$s\begin{pmatrix}u_1\\v_1\\1\end{pmatrix}=\begin{pmatrix}t_1&t_2&t_3&t_4\\t_5&t_6&t_7&t_8\\t_9&t_{10}&t_{11}&t_{12}\end{pmatrix}\begin{pmatrix}X\\Y\\Z\\1\end{pmatrix} \tag{3-34}$$

用最后一行把 s 消去，得到两个约束：

$$u_1 = \frac{t_1X + t_2Y + t_3Z + t_4}{t_9X + t_{10}Y + t_{11}Z + t_{12}} \quad v_1 = \frac{t_5X + t_6Y + t_7Z + t_8}{t_9X + t_{10}Y + t_{11}Z + t_{12}} \tag{3-35}$$

为了简化表示，定义 T 的行向量：

$$\boldsymbol{t}_1 = (t_1, t_2, t_3, t_4)^T, \boldsymbol{t}_2 = (t_5, t_6, t_7, t_8)^T, \boldsymbol{t}_3 = (t_9, t_{10}, t_{11}, t_{12})^T \tag{3-36}$$

于是有：

$$\begin{aligned} \boldsymbol{t}_1^T P - \boldsymbol{t}_3^T P u_1 = 0 \\ \boldsymbol{t}_2^T P - \boldsymbol{t}_3^T P v_1 = 0 \end{aligned} \tag{3-37}$$

请注意 t 是待求的变量，可以看到每个特征点提供了两个关于 t 的线性约束。假设一共有 N 个特征点，可以列出线性方程组：

$$\begin{pmatrix} P_1^T & 0 & -u_1 P_1^T \\ 0 & P_1^T & -v_1 P_1^T \\ \vdots & \vdots & \vdots \\ P_N^T & 0 & -u_N P_N^T \\ 0 & P_N^T & -v_N P_N^T \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} = 0 \tag{3-38}$$

由于 t 一共有 12 维，因此最少通过六对匹配点即可实现矩阵 T 的线性求解，这种方法称为直接线性变换(DLT, Direct Linear Transform 就)。当匹配点大于六对时，可以使用 SVD 等方法对超定方程求最小二乘解。

在 DLT 求解中，我们直接将 T 矩阵看成了 12 个未知数，忽略了它们之间的联系。由于旋转矩阵用 $R \in S0(3)$，DLT 求出的解不一定满足该约束，它是一个一般矩阵。对于旋转矩阵 R，我们必须针对 DLT 估计的 T 的左边 3×3 的矩阵块，寻找一个最好的旋转矩阵对它进行近似。这可以由 QR 分解完成，相当于把结果从矩阵空间重新投影到 $SE(3)$流形上，转换成旋转和平移两部分。

3.3.3 光束法平差（Bundle Adjustment, BA）

除了使用线性方法之外，我们可以把 PnP 问题构建成一个定义于李代数上的非线性最小二乘问题。上两节的线性方法，往往是先求相机位姿，再求空间点位置，而非线性优化则是把它们都看成优化变量，放在一起优化。这是一种非常通用的求解方式，可以用它对 PnP 或 ICP 给出的结果进行优化。在 PnP 中，这个 Bundle Adjustment 问题，是一个最小化重投影误差（Reprojection Error）的问题，如图 3-6 所示：

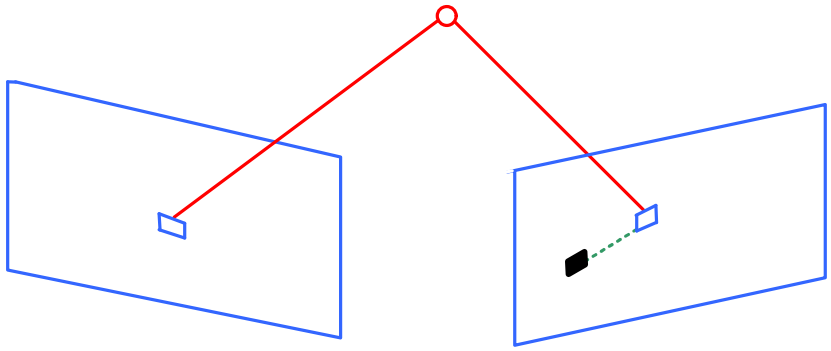


图 3-6 重投影误差示意图

考虑 n 个三维空间点 P 和它们的投影 p，我们希望计算相机的位姿 R，t，它的李代数表示为 ξ。假设某空间点坐标为 $P_i=[X_i,Y_i,Z_i]^T$，其投影的像素坐标为 $\boldsymbol{u}_i=[u_i,v_i]^T$。

根据第二章的内容，像素位置与空间点位置的关系如下：

$$s_i\begin{bmatrix}u_i\\v_i\\1\end{bmatrix}=K\exp(\xi^\wedge)\begin{bmatrix}X_i\\Y_i\\Z_i\\1\end{bmatrix} \tag{3-39}$$

除了用 ξ 为李代数表示的相机姿态之外，别的都和前面的定义保持一致。写成矩阵形式就是：

$$s_i\boldsymbol{u}_i=K\exp(\xi^\wedge)P_i \tag{3-40}$$

由于相机位姿未知以及观测点的噪声，该等式存在一个误差。因此，我们把误差求和，构建最小二乘问题，然后寻找最好的相机位姿，使它最小化：

$$\xi^*=\arg\min_\xi\frac{1}{2}\sum_{i=1}^{n}\left\|\boldsymbol{u}_i-\frac{1}{s_i}K\exp(\xi^\wedge)P_i\right\|_2^2 \tag{3-41}$$

该问题的误差项，是将像素坐标（观测到的投影位置）与 3D 点按照当前估计的位姿进行投影得到的位置相比较得到的误差，所以称之为重投影误差。使用齐次坐标时，这个误差有 3 维。不过由于 u 最后一维为 1，该维度的误差一直为零，因而更多时候使用非齐次坐标，于是误差就只有 2 维了。通过特征匹配，知道了 p1 和 p2 是同一个空间点 P 的投影，但是我们不知道相机的位姿。在初始值中，P 的投影 p1 与实际的 p2 之间有一定的距离。于是我们调整相机的位姿，使得这个距离变小。不过，由于这个调整需要考虑很多个点，所以最后每个点的误差通常都不会精确为零。

3.4 基于 IMU 数据的视觉帧间位姿估计

由于 IMU 的测量频率比视觉相机快很多，如图 3-7 所示。而且实际视觉优化项是以关键帧为单位的。因此要想在同一框架下同时优化视觉和 IMU 的约束，需要将两个相邻视觉关键帧的众多 IMU 的测量整合成一个约束。本文所采用的基于 SO3 流形的预积分公式理论由 Forster 等人在 2016 年提出[32]，就是用 IMU 预积分的方法将 IMU 测量数据转化成视觉关键帧约束。本节将从 IMU 误差模型与运动学方程出发，最终推导出 IMU 预计分的表达式。

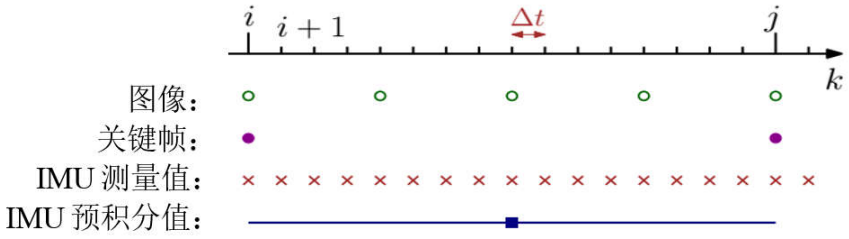


图 3-7 图像与 IMU 测量频率

3.4.1 IMU 误差模型与运动学模型

IMU 包括一个三轴加速度计和一个三轴陀螺仪，可以在惯性系内测量三个方向的加速度和角速度。IMU 测量值受高斯白噪声与零偏影响，设定 IMU 的测量值为 $\tilde{\omega}_B(t)$ 和 $\tilde{a}_B(t)$，那么 IMU 的噪声模型可用以下公式描述：

$$\tilde{\boldsymbol{\omega}}_{\mathrm{B}}(t)=\boldsymbol{\omega}_{\mathrm{B}}(t)+\mathbf{b}_g(t)+\boldsymbol{\eta}_g(t) \tag{3-42}$$

$$\tilde{\mathbf{a}}_B(t)=\mathbf{R}_{\mathrm{WB}}^{\mathrm{T}}(\mathbf{a}_{\mathrm{W}}(t)-\mathbf{g}_w)+\mathbf{b}_a(t)+\boldsymbol{\eta}_a(t) \tag{3-43}$$

在式(3-42)和式(3-43)中，右下标 B 表示在 IMU 坐标系下的对应量。$\omega_B(t)\in\mathbb{R}^3$ 表示陀螺仪测量的角速度，$\mathbf{a}_{\mathrm{B}}(t)\in\mathbb{R}^3$ 表示加速度计测量的角速度。$\mathbf{b}$ 和 η 分别代表相应的零偏和高斯白噪声，

为了从 IMU 的测量值中计算出 IMU 的运动，需要引入下列的运动学模型：

$$\dot{R}_{WB}=R_{WB}\omega_B^{\wedge},\quad \dot{\mathrm{v}}_{\mathrm{w}}=a_w,\quad \dot{p}_w=v_w \tag{3-44}$$

式(3-44)以微分的形式描述了 IMU 的位姿与速度的变化。为了获取 IMU 在每个时刻状态的值，需要对式(3-44)进行积分，令 Δt 为两次相邻 IMU 测量值的时间间隔，可得：

$$\begin{aligned}\mathbf{R}_{\mathrm{WB}}(t+\Delta t)&=\mathbf{R}_{\mathrm{WB}}(t)\mathrm{Exp}(\int_t^{t+\Delta t}\omega_B(\tau)d\tau)\\ \mathbf{v}_{\mathrm{W}}(t+\Delta t)&=\mathbf{v}_{\mathrm{W}}(t)+\int_t^{t+\Delta t}\mathbf{a}_{\mathrm{W}}(\tau)d\tau\\ \mathbf{p}_{\mathrm{W}}(t+\Delta t)&=\mathbf{v}_{\mathrm{W}}(t)+\int_t^{t+\Delta t}\mathbf{v}_{\mathrm{W}}(\tau)d\tau+\iint_t^{t+\Delta t}\mathbf{a}_{\mathrm{W}}(\tau)d\tau^2\end{aligned}\tag{3-45}$$

在实际应用中，IMU 测量时间间隔 Δt 非常短，故假设 $\mathbf{a}_{\mathrm{W}}$ 和 ω_{W} 在间隔 $[t,t+\Delta t]$ 中保持不变，可将式(3-45)中积分变为乘法得：

$$\begin{aligned}\mathbf{R}_{\mathrm{WB}}(t+\Delta t)&=\mathbf{R}_{\mathrm{WB}}(t)\mathrm{Exp}(\omega_B(t)\Delta t)\\ \mathbf{v}_{\mathrm{W}}(t+\Delta t)&=\mathbf{v}_{\mathrm{W}}(t)+\mathbf{a}_{\mathrm{W}}(t)\Delta t\\ \mathbf{p}_{\mathrm{W}}(t+\Delta t)&=\mathbf{v}_{\mathrm{W}}(t)+\frac{1}{2}\mathbf{a}_{\mathrm{W}}(t)\Delta t^2\end{aligned}\tag{3-46}$$

根据式(3-42)和式(3-43)，可以将式(3-46)中的 $\omega_B(t)$ 和 $\mathbf{a}_{\mathrm{W}}(t)$ 改写为与测量项相关的表达式：

$$\begin{aligned}\mathbf{R}(t+\Delta t)&=\mathbf{R}(t)\mathrm{Exp}\left(\left(\tilde{\omega}(t)-\mathbf{b}_g(t)-\boldsymbol{\eta}_g(t)\right)\Delta t\right)\\ \mathbf{v}(t+\Delta t)&=\mathbf{v}(t)+\mathbf{g}\Delta t+\mathbf{R}(t)\left(\tilde{\mathbf{a}}(t)-\mathbf{b}_a(t)-\boldsymbol{\eta}_a(t)\right)\Delta t\\ \mathbf{p}(t+\Delta t)&=\mathbf{p}(t)+\mathbf{v}(t)\Delta t+\frac{1}{2}\mathbf{g}\Delta t^2+\frac{1}{2}\mathbf{R}(t)\left(\tilde{\mathbf{a}}(t)-\mathbf{b}_a(t)-\boldsymbol{\eta}_a(t)\right)\Delta t^2\end{aligned}\tag{3-47}$$

在式(3-47)中，为了公式的易读性将参考坐标系的下标隐去了（此时应该没有歧义了）。在式(3-6)对速度和位置的积分中，在 IMU 相邻两次测量之间假定 IMU 的姿态 $\mathbf{R}(t)$ 是恒定的。当 IMU 的角速度不为零时，它不是微分方程（3-43）的精确的解。在实际应用过程中，由于使用的 IMU 测量频率，可以很大程度上减轻由此带来的误差影响。

3.4.2 基于 IMU 预积分的视觉帧间位姿估计

由上一节中式(3-47)可得到惯性元件在相邻 Δt 时间间隔的位姿关系，Δt 是 IMU 的采样频率，如果想对其进行优化，则需要在每次采集到 IMU 数据时添加一个需要估计的状态，这会导致计算量过大无法实施对 IMU 的数据进行处理。

通过 IMU 预积分方法，可将相邻两个视觉关键帧 i 和 j 之间的 IMU 测量合并成为一个复合的项，构成两个相邻视觉关键帧的运动约束。假设 IMU 与视觉帧检测时间同步，并且在离散时间 k 采集测量数据，那么可以通过 IMU 测量量获得在 k=i 与 k=j 两个关键帧之间的位姿关系：

$$\begin{aligned}\mathbf{R}_j &= \mathbf{R}_i \prod_{k=i}^{j-1} \mathrm{Exp}\left(\left(\tilde{\boldsymbol{\omega}}_k - \mathbf{b}_k^g - \boldsymbol{\eta}_k^{gd}\right)\Delta t\right) \\ \mathbf{v}_j &= \mathbf{v}_i + \mathbf{g}\Delta t_{ij} + \sum_{k=i}^{j-1} \mathbf{R}_k \left(\tilde{\mathbf{a}}_k - \mathbf{b}_k^a - \boldsymbol{\eta}_k^{ad}\right)\Delta t \\ \mathbf{p}_j &= \mathbf{p}_i + \sum_{k=i}^{j-1}\left[\mathbf{v}_k \Delta t + \frac{1}{2}\mathbf{g}\Delta t^2 + \frac{1}{2}\mathbf{R}_k \left(\tilde{\mathbf{a}}_k - \mathbf{b}_k^a - \boldsymbol{\eta}_k^{ad}\right)\Delta t^2\right]\end{aligned} \tag{3-48}$$

式(3-48)已经提供了在 t_i 到 t_j 时刻运动估计，但其有一个缺点，即每当 t_i 时刻的运动估计改变时，需要对式(3-48)重新计算，这会导致大量的重复计算。为了避免位姿估计起始帧的估计变化导致积分重新计算的情况，本文采用增量式的表达方式，将 IMU 测量获得的运动约束与位姿估计的起点独立：

$$\begin{aligned}\Delta\mathbf{R}_{ij} &\doteq \mathbf{R}_i^{\mathrm{T}}\mathbf{R}_j = \prod_{k=i}^{j-1} \mathrm{Exp}\left(\left(\tilde{\boldsymbol{\omega}}_k - \mathbf{b}_k^g - \boldsymbol{\eta}_k^{gd}\right)\Delta t\right) \\ \Delta\mathbf{v}_{ij} &\doteq \mathbf{R}_i^{\mathrm{T}}\left(\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t_{ij}\right) = \sum_{k=i}^{j-1} \mathbf{R}_{ik}\left(\tilde{\mathbf{a}}_k - \mathbf{b}_k^a - \boldsymbol{\eta}_k^{ad}\right)\Delta t \\ \Delta\mathbf{p}_{ij} &\doteq \mathbf{R}_i^{\mathrm{T}}\left(\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i\Delta t_{ij} - \frac{1}{2}\mathbf{g}\Delta t_{ij}^2\right) \\ &= \sum_{k=i}^{j-1}\left[\mathbf{v}_{ik}\Delta t + \frac{1}{2}\Delta\mathbf{R}_{ik}\left(\tilde{\mathbf{a}}_k - \mathbf{b}_k^a - \boldsymbol{\eta}_k^{ad}\right)\Delta t^2\right]\end{aligned} \tag{3-49}$$

需要注意的是，式中的 $\Delta\mathbf{R}_{ij}$ 代表第 t_i 到 t_j 时刻的姿态变化，但式中 $\Delta\mathbf{v}_{ij}$ 与 $\Delta\mathbf{p}_{ij}$ 并不代表实际在物理上的速度和位置变化，而是为了右侧 IMU 测量项与 t_i 时刻的状态和重力效应想独立而构造的变化量。所以，我们可以直接通过两个关键帧之间惯性测量量计算出式(3-48)的右侧表达式即视觉关键帧之间的运动约束。

3.5 本章小结

本章节给出了惯性视觉 SLAM 系统中的提供局部位姿初值估计的方法，包括纯单目视觉和基于纯 IMU 的位姿估计方法，他们通常在局部是准确有效的。即就是通过相邻图像的信息以及 IMU 的测量数据粗略估计相机运动，给后端提供更好的初始值。在视觉的部分，本章首先给出了 Shi-Tomas 角点提取方法和 KLT 光流法跟踪的原理，实现了从原始的图像帧到含有特征匹配（跟踪）的图像帧。随后介绍了基于连续图像特征匹配恢复图像间位姿的方法，它们可分为 2D-2D 方法和 2D-3D 方法。在 2D-2D 方法中，主要根据对极几何原理恢复位姿信息，在通常情况下恢复本质矩阵 E，退化情况下恢复单应矩阵 H。在 2D-3D 方法中，主要介绍了透视 N 点定位恢复位姿的方法，它比

2D-2D 的方法需要更少的特征点匹配。接着介绍了光束法平差，对相机和三维特征点位置进行优化。最后介绍了根据 IMU 误差模型和运动学模型，推导了以 IMU 测量值形式表达的视觉帧之间的位姿信息。

第4章　基于单目视觉与惯性传感器融合的在线 SLAM 系统

4.1 引言

前两章已经介绍了 SLAM 的基本原理，以及基于独立视觉与独立 IMU 的位姿估计方法，然而仅有局部的位姿信息时远远不够的，视觉惯性 SLAM 系统需要构建全局一致的轨迹和地图。 然而要想得到全局一致的还需要做更多的工作。本章致力于构建一套在线单目视觉惯性 SLAM 系统，能输出全局一致的姿态轨迹和三维路标点构成的地图，且具有实时性。

4.2 系统整体框架

根据基于单目视觉和惯性相机的定位和建图需求，本章提出一套在线单目视觉惯性 SLAM 系统方案，它是一套完整的惯性视觉 SLAM 框架。能根据输入的单目相机图像和 IMU 的测量值实时计算出全局一致的惯性相机（机器人）的位姿和三维路标点的坐标，并且系统具有回环检测功能，能够进一步减小累计误差。整个系统的框架如图 4-1 所示：

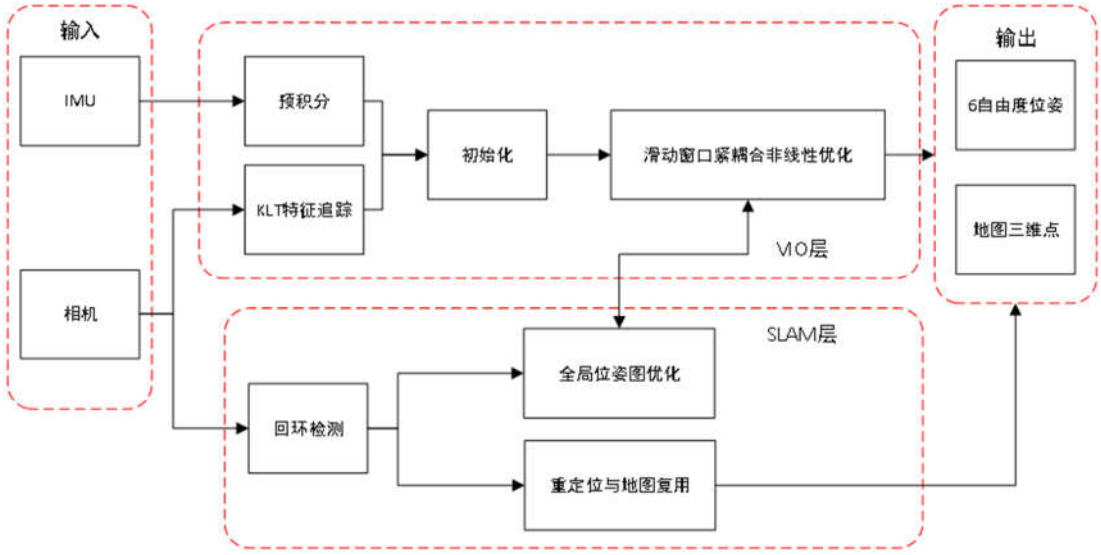


图 4-1 系统结构框图

系统的输入为单目相机采集的图像和 IMU 测量的加速度与角速度。输出

为与视觉关键帧对齐的含有真实尺度的六自由度位姿和路标点三维坐标，即惯性相机(机器人)运动的轨迹。整个框架可以分为两个大层：VIO 层和 SLAM 层，共有四个线程。

其中 VIO 层包括视觉特征追踪与后端优化两个线程，它主要解决前端视觉惯性里程计增量地实时估计位姿的问题。

SLAM 层包括回环检测与位姿图优化两个线程，它主要接受前端惯性视觉里程计得到的位姿估计，结合闭环检测对其进行优化，最后得到全局一致的轨迹和地图。

前端视觉特征部分已在 3.2 节中进行了介绍，本章主要介绍系统的优化后端，即紧耦合非线性视觉惯性状态估计器和回环检测的部分。

4.3 系统初始化

视觉惯性 SLAM 系统初始化的主要目的是获取系统进行优化所必要的参数以及状态的初值。由于单目惯性紧耦合系统是一个非线性程度很高的系统，对一些初值非常敏感，初始化的好坏会直接影响整个紧耦合系统的鲁棒性以及定位精度。因此用特定的方法对系统进行的初始化以提供精确的参数和初值是十分必要的。

在初始化的过程中，需要初始化或估计的信息可分为两大类：1）在系统运行过程中几乎不变或变化不大的参数，如绝对尺度以及重力加速度（默认惯性相机内外参已知）；2）系统起始状态量的初值，包括前几个关键帧的位姿与速度信息与三维路标点的位置，以及 IMU 加速度计和陀螺仪的零偏。

初始化分为两个过程，基于滑动窗口单目视觉初始化可以初始化出随尺度而变的初始关键帧位姿信息与三维路标点位置信息，和视觉惯性联合初始化可以初始化出绝对尺度，重力加速度，相机状态的速度信息以及加速度计和陀螺仪的零偏。

4.3.1 基于滑动窗口的单目视觉初始化

在基于滑动窗口的单目视觉初始化的过程中，会构建一个滑动窗口的随尺度变化的(Up-to-Scale)纯视觉结构以恢复初始关键帧位姿信息与三维路标点位置信息。

首先，在滑动窗口中选择包含足够特征视差的两个关键帧。接下来，分别使用 3.3.1 节中介绍的对极几何恢复位姿的八点法和直接线性变换法分别恢复出本质矩阵 E 与单应矩阵 H，通过计算两个模型的重投影误差来进行打分，采用重投影误差较小的模型。固定平移变换的尺度，利用选择的模型 E

或 H 恢复出运动的姿态，并三角化出三维地图点。在初始化一批三维点后，采用 3.3.2 节中介绍的 PnP 方法去求解滑动窗口内剩余帧的位姿信息。

然后应用 3.3.3 节介绍的 BA 方法，以最小化所有关键帧中所有特征观测的总重投影误差。至此可以得到所有关键帧的位姿信息 (p_i, q_i) 和特征点的三维信息。由于相机和 IMU 之间的外参 $T_{CB}=(p_{CB},\ q_{CB})$ 已知，所有的变量可以转化到 IMU 坐标系中表示：

$$sp_{Bi} = sR_{CB}p_{Ci} + p_{CB} \tag{4-1}$$

其中 s 是未知的尺度因子，将在下一节中进行估计。

4.3.2 视觉惯性联合初始化

通过视觉惯性联合初始化可以初始化出绝对尺度，重力加速度，相机状态的速度信息以及 IMU 的零偏。

首先是 IMU 偏置的初始化：假设 IMU 陀螺仪零偏 $\mathbf{b}_g$ 在当前窗口是不变的。考虑窗口中相邻的第 k 和第 k+1 关键帧，在上一步单目视觉初始化中，我们可以得到它们的相对世界坐标系的旋转 $\mathbf{R}_k$ 和 $\mathbf{R}_{k+1}$，在 IMU 预积分的结果中，由式（3-49）可以得到角速度的预积分结果 $\Delta\mathbf{R}_{k,k+1}$。可以通过最小化这两项的误差来估计陀螺仪的偏置：

$$\min_{b_g}\sum_{k}\left\|Log(\mathbf{R}_k^{\mathrm{T}}\mathbf{R}_{k+1}\Delta\mathbf{R}_{k,k+1})\right\| \tag{4-2}$$

通过求解这个最小二乘问题，即可得到对陀螺仪零偏 $\mathbf{b}_g$ 的估计。

对于加速度计偏置，在初始化过程中很难解决，由于在估计加速度计偏置时要同时估计重 $\mathbf{g}$，需要足够的旋转来区分加速度计偏置和重力。然而，加速度计偏置 $\mathbf{b}_a$ 对系统稳定性的影响并不大，可以给出粗略的初始猜测是，而且在后续的优化过程中将不断对偏置进行优化。因此，在初始化步骤中，我们将加速度计偏置 $\mathbf{b}_a$ 设置为零。

将剩余的待估计量：状态的速度，重力以及尺度因子定义为初始化待估计的变量：

$$\chi_I = [\mathbf{v}_0, \mathbf{v}_1, ..., \mathbf{v}_n, \mathbf{g}, s] \tag{4-3}$$

通过解决以下的最小二乘问题，可以得到初值的估计：

$$\min_{\chi_I}\sum\left\|\hat{\mathbf{z}}_{k+1}^{k} - \mathbf{H}_{k+1}^{k}\chi_I\right\|^2 \tag{4-4}$$

即可以得到每一个局部帧的速度，在视觉帧下的重力向量（包括方向和强度），以及尺度因子。从视觉结构中得到的平移分量 $\mathbf{P}_i$ 便可对齐到实际尺

度上。同时将 IMU 坐标系下的估计值与重力方向竖直向下的世界坐标系对齐。其示意图如图 4-2 所示：

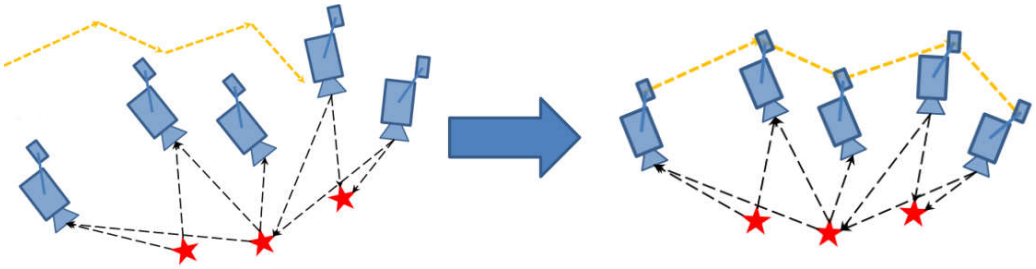


图 4-2 视觉与 IMU 数据对齐示意图

图 4-2 中左图为未对齐的状态，黄色线为 IMU 估计的结果，蓝色相机移动轨迹为相机图像估计的位姿变化。经过对齐后，最小化 IMU 轨迹与相机轨迹的误差。由于 IMU 的位姿估计数据是带有绝对尺度的，而相机的位姿估计是不带有漂移的。两者对齐之后可以很好的估计相机位姿的绝对尺度与 IMU 的偏置。至此，初始化过程完成，这些带有尺度的估计值将被送给紧耦合非线性视觉惯性状态估计器。

4.4 紧耦合非线性视觉惯性状态估计器

在状态初始化之后，采用滑动窗非线性估计器对高精度状态进行估计。本节主要介绍基于单目视觉与 IMU 结合的状态估计器，也可以理解为基于非线性图优化的后端。它对三维路标点的坐标与惯性相机的位姿，速度和 IMU 零偏进行同时优化。基于实时性的考虑，本方法对最近的一部分关键帧进行优化，形成一个滑动窗口。它将非关键帧以及离目前时间较远的帧边缘化来保持一个前后一致的先验项和估计的不确定度。尽管仅在局部窗口优化，本方法通过对边缘化项的重复线性化可以实现对惯性相机的叫高精度追踪。

在本节中，首先定义了系统估计的状态变量，接着给出视觉惯性优化项的表达式，之后给出了视觉误差项与惯性误差项详细的计算公式，最后介绍系统的边缘化方法。

4.4.1 系统状态变量

首先定义系统的状态变量，待估计的状态变量有惯性相机(B)在图像时间 k 时刻的状态变量 $\mathbf{x}_B^k$，以及三位路标点(L)坐标 $\mathbf{x}_{\mathrm{B}}$。状态变量 $\mathbf{x}_{\mathrm{B}}$，包括位置 p，姿态 q，速度 v，陀螺仪偏置 $\mathrm{b_g}$ 以及加速度计偏置 $\mathrm{b_a}$：

$$\mathbf{x}_{\mathrm{B}} = [\mathbf{p}, \mathbf{q}, \mathbf{v}, \mathbf{b}_{\mathrm{g}}, \mathbf{b}_{\mathrm{a}}] \in \mathbb{R}^3 \times S^3 \times \mathbb{R}^9 \tag{4-5}$$

路标点的位置$\mathbf{x}_{\mathrm{L}}$用齐次坐标表示，第 j 个路标点表示为$\mathbf{x}_{\mathrm{Lj}}=l_j\in\mathbb{R}^4$，均为归一化的坐标，第四个分量设置为 1。

总的来讲，系统的状态是存在于流形上的，因此本方法采用位于切向空间$\mathfrak{g}$的扰动量，且引入了群的运算$\bullet$，以及相应的指数和对数映射。定义在估计值$\bar{\mathbf{x}}$附近的扰动为$\delta\mathbf{x}:=\mathbf{x}\bullet\bar{\mathbf{x}}^{-1}$。采用极小坐标表示$\delta\chi\in\mathbb{R}^{\dim\mathfrak{g}}$。通过双射$\Phi:\mathbb{R}^{\dim\mathfrak{g}}\to\mathfrak{g}$将极小化坐标转化为切向空间。因此，可以得到极小化坐标与待估计状态变量之间的相互转换：

$$\delta\mathbf{x}=\exp\left(\Phi\left(\delta\boldsymbol{\chi}\right)\right) \tag{4-6}$$

$$\delta\boldsymbol{\chi}=\Phi^{-1}\left(\log\left(\delta\mathbf{x}\right)\right) \tag{4-7}$$

具体来讲，本文采用极小化的三轴角度扰动量$\delta\boldsymbol{\alpha}\in\mathbb{R}^3$，它可以通过指数映射转化为等价的四元数$\delta\mathbf{q}$：

$$\delta\mathbf{q}:=\exp\left(\begin{bmatrix}\frac{1}{2}\delta\boldsymbol{\alpha}\\0\end{bmatrix}\right)=\begin{bmatrix}\operatorname{sinc}\left\|\frac{\delta\boldsymbol{\alpha}}{2}\right\|\frac{\delta\boldsymbol{\alpha}}{2}\\\cos\left\|\frac{\delta\boldsymbol{\alpha}}{2}\right\|\end{bmatrix} \tag{4-8}$$

因此，采用群的运算$\otimes$可以将旋转表示为$\mathbf{q}_{\mathrm{WB}}=\delta\mathbf{q}\otimes\bar{\mathbf{q}}_{\mathrm{WB}}$。可以获得极小化的机器人误差状态变量

$$\delta\boldsymbol{\chi}_{\mathrm{R}}=\left[\delta\mathbf{p}^{\mathrm{T}},\delta\boldsymbol{\alpha}^{\mathrm{T}},\delta\mathbf{v}^{\mathrm{T}},\delta\mathbf{b}_{\mathrm{g}}^{\mathrm{T}},\delta\mathbf{b}_{\mathrm{a}}^{\mathrm{T}}\right]^{\mathrm{T}}\in\mathbb{R}^{15} \tag{4-9}$$

4.4.2 视觉惯性 SLAM 优化项

在传统的视觉 SLAM 或者视觉里里程计中，非线性优化是通过最小化相机帧中特征点的重投影误差以求得相机位姿与三维路标点的最佳估计。一旦惯性测量量被引入，它们不仅在相机连续运动位姿之间产生约束。而且在连续时间的加速度计和陀螺仪的速度和 IMU 偏差估计之间也有约束，从而增加了要估计的系统状态变量。

故在对惯性视觉 SLAM 优化时，需要同时考虑基于视觉的重投影误差，以及基于 IMU 的误差。令 J(x)为待优化的代价函数，视觉的重投影误差为$\mathbf{e}_{proj}$，IMU 的误差$\mathbf{e}_{imu}$。代价函数可由定义为：

$$J(\mathbf{x}):=\underbrace{\sum_{i\in I}\sum_{k\in K}\mathbf{e}_{\mathrm{proj}}^{i,k\ \mathrm{T}}\mathbf{W}_{\mathrm{proj}}^{i,k}\mathbf{e}_{\mathrm{proj}}^{i,k}}_{\text{视觉}}+\underbrace{\sum_{k\in K}\mathbf{e}_{\mathrm{imu}}^{k\ \mathrm{T}}\mathbf{W}_{\mathrm{imu}}^{k}\mathbf{e}_{\mathrm{imu}}^{k}}_{\text{IMU}} \tag{4-10}$$

式中第一项为视觉误差项，第二项为 IMU 误差项为它会优化惯性相机的状态$\mathbf{x}_{\mathrm{B}}$，以及三维路标点$\mathbf{x}_{\mathrm{L}}$。式中 k 代表关键帧的序号，i 代表路标点的序

号，I 为图像追踪到路标点的集合，K 为关键帧的集合。$\mathbf{W}_{\text{proj}}^{i,k}$ 表示相应路标点测量的信息矩阵（协方差矩阵的逆），$\mathbf{W}_{\text{imu}}^{k}$ 表示 IMU 测量误差的信息矩阵。

接下来，将会介绍视觉误差项与 IMU 误差项的计算方法。

4.4.3 视觉与惯性误差项

本文使用的重投影误差公式为：

$$\mathbf{e}_{proj}^{i,k} = \mathbf{z}^{i,k} - \mathrm{h}(T_{BW}^{k}, l^{i}) \tag{4-11}$$

其中 h(·)表示相机成像模型，本文所用的即是 2.2.1 节介绍的针孔相机模型。$\mathbf{z}^{i,k}$ 代表图像测量的像素坐标。除了误差项外，还需要计算该项的雅各比矩阵，它不仅会在优化求解时使用，而且在 4.4.5 节的边缘化中也起到重大作用，视觉误差项的雅各比[47]为：

$$\frac{\partial \mathbf{e}_{\mathrm{r}}^{i,j,k}}{\partial \delta \boldsymbol{\chi}_{\mathrm{T}}^{k}} = \mathbf{J}_{\mathrm{r},i} \overline{\boldsymbol{T}}_{C_iB}^{k} \begin{bmatrix} \overline{\mathbf{R}}_{BW}^{k} \bar{\boldsymbol{l}}_{4}^{j} & \mathbf{R}_{BW}^{k} \left[\bar{\boldsymbol{l}}_{1:3}^{j} - \mathbf{p}_{B}^{k} \bar{\boldsymbol{l}}_{4}^{j} \right] \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} \end{bmatrix} \tag{4-12}$$

$$\frac{\partial \mathbf{e}_{\mathrm{r}}^{i,j,k}}{\partial \delta \boldsymbol{\chi}_{\mathrm{L}}^{j}} = \mathbf{J}_{\mathrm{r},i} \overline{\boldsymbol{T}}_{C_iB}^{k} \begin{bmatrix} \overline{\mathbf{R}}_{BW}^{k} \\ \mathbf{0}_{1\times 3} \end{bmatrix} \tag{4-13}$$

$$\frac{\partial \mathbf{e}_{\mathrm{r}}^{i,j,k}}{\partial \delta \boldsymbol{\chi}_{Ci}^{k}} = \mathbf{J}_{\mathrm{r},i} \begin{bmatrix} \mathbf{R}_{CiB}^{k} \bar{\boldsymbol{l}}_{4}^{j} \mathbf{R}_{CiB}^{k} \left[\bar{\boldsymbol{l}}_{1:3}^{j} - \mathbf{p}_{Ci}^{k} \bar{\boldsymbol{l}}_{4}^{j} \right] & \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} \end{bmatrix} \tag{4-14}$$

IMU 误差项的模型为：

$$\mathbf{e}_{\mathrm{s}}^{k}\left(\mathbf{x}_{\mathrm{R}}^{k}, \mathbf{x}_{\mathrm{R}}^{k+1}, \mathbf{z}_{\mathrm{s}}^{k}\right) = \begin{bmatrix} \hat{\mathbf{p}}_{S}^{k+1} - \mathbf{p}_{S}^{k+1} \\ 2\left[\hat{\mathbf{q}}_{BS}^{k+1} \otimes \mathbf{q}_{BS}^{k+1}\right]_{1:3} \\ \hat{\mathbf{x}}_{\mathrm{sb}}^{k+1} - \mathbf{x}_{\mathrm{sb}}^{k+1} \end{bmatrix} \in \mathbb{R}^{15} \tag{4-15}$$

根据出差传播定律，联合的信息矩阵可被表示为：

$$\frac{\partial \mathbf{e}_{\mathrm{s}}^{k}}{\partial \delta \hat{\boldsymbol{\chi}}_{\mathrm{R}}^{k+1}} = \begin{bmatrix} \mathbf{I}_{3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 9} \\ \mathbf{0}_{3\times 3} & \left[\hat{\mathbf{q}}_{BS}^{k+1} \otimes \mathbf{q}_{BS}^{k+1}\right]_{1:3,1:3}^{\oplus} & \mathbf{0}_{3\times 9} \\ \mathbf{0}_{9\times 3} & \mathbf{0}_{9\times 3} & \mathbf{I}_{9} \end{bmatrix} \tag{4-16}$$

雅各比矩阵：

$$\mathbf{e}_{\mathrm{s}}^{k}\left(\mathbf{x}_{\mathrm{R}}^{k}, \mathbf{x}_{\mathrm{R}}^{k+1}, \mathbf{z}_{\mathrm{s}}^{k}\right) = \begin{bmatrix} \hat{\mathbf{p}}_{S}^{k+1} - \mathbf{p}_{S}^{k+1} \\ 2\left[\hat{\mathbf{q}}_{BS}^{k+1} \otimes \mathbf{q}_{BS}^{k+1}\right]_{1:3} \\ \hat{\mathbf{x}}_{\mathrm{sb}}^{k+1} - \mathbf{x}_{\mathrm{sb}}^{k+1} \end{bmatrix} \in \mathbb{R}^{15} \tag{4-17}$$

4.4.4 边缘化（Marginalization）

随着时间的增长，特征点和相机位姿会积累的越来越多，随之而来的优化的计算量也会随之增加，由于视觉惯性系统的待优化变量维数要远比纯视觉系统待优化变量维数多，如不对待估计变量进行限制，计算量会随着时间增加的非常快。图像的边缘化的目的就是为了限制计算量的无限制增加，保持优化计算量维持在一个恒定的范围内。本文在进行非线性优化的同时对待估计系统状态进行边缘化处理，即在不改变估计一致性的前提下在去除掉时间局限在较久远的关键帧。

4.4.2 节中的视觉惯性优化项本质上是一个最小二乘问题，一般可以通过高斯牛顿迭代法来求解，表示如下:

$$\mathbf{H}\delta\chi=\mathbf{b} \tag{4-18}$$

设 x_μ 为将要边缘化的状态变量，x_λ 为要保留的状态变量，根据条件独立，我们可以简化边缘化的过程并将之应用到子问题中：

$$\begin{bmatrix}\mathbf{H}_{\mu\mu} & \mathbf{H}_{\mu\lambda}\\ \mathbf{H}_{\lambda\mu} & \mathbf{H}_{\lambda\lambda}\end{bmatrix}\begin{bmatrix}\delta\chi_\mu\\ \delta\chi_\lambda\end{bmatrix}=\begin{bmatrix}b_\mu\\ b_\lambda\end{bmatrix} \tag{4-19}$$

应用舒尔补(Schur complement)可得:

$$\begin{aligned}\mathbf{H}^*_{\lambda\lambda}&=\mathbf{H}_{\lambda\lambda}-\mathbf{H}_{\lambda\mu}\mathbf{H}^{-1}_{\mu\mu}\mathbf{H}_{\mu\lambda}\\ \mathbf{b}^*_{\lambda}&=\mathbf{b}_{\lambda}-\mathbf{H}_{\lambda\mu}\mathbf{H}^{-1}_{\mu\mu}\mathbf{b}_{\mu}\end{aligned} \tag{4-20}$$

其中，$\mathbf{H}^*_{\lambda\lambda}$ 和 $\mathbf{b}^*_{\lambda}$ 为被边缘化的 $\mathbf{H}$ 矩阵和误差量，状态变量 χ_μ 则被边缘化掉了。这是每次进行边缘化时进行的操作，通过重复进行边缘化，可在增加新状态变量时保持待优化的状态变量数量保持恒定，可大大减小随时间增加的计算量。

将该边缘化方法引入到视觉惯性的后端优化中，最初的边缘化误差项是由 N 个图像关键帧组成的项，每当一个新的关键帧加入到优化的滑动窗口中时，算法进行一次边缘化操作。这时，被边缘化掉的变量将将以先验信息的形式保留在系统中，以保证边缘化前后状态估计的一致性。由于要保证算法的实时性，本方法仅将关键帧加入到滑动窗口中，未加入非关键帧。

4.5回环检测与闭环

本章前面所述部分可组成视觉惯性里程计，虽然可以得到位姿状态地图三维特征点坐标的估计，但是由于进行的非线性优化仅在局部的滑动窗口中运行，并没有对全局的位姿和地图进行优化，系统会不可避免地出现误差累

积，即产生漂移现象。 因此，要想减小累计误差的影响，需要对系统进行闭环检测，对闭环的位姿进行优化。

回环检测也成闭环检测，是指通过特定的算法检测机器人是否到访过曾经到过的地方，它是 SLAM 中的一个重要组成部分，由于对机器人的位姿估计仅考虑相邻时间或相邻帧的约束，因此之前帧就已经产生的误差将不可避免地累积到下一帧，结果会使得 SLAM 系统估计的位姿和地图出现累积误差的情况，不能构建全局一致的轨迹和地图。而回环检测通过识别曾经到过的相似场景，增加远期回环间位姿的约束，通过这种方式可以很好地消除累积误差，以保持地图和位姿的误差一致性。需要注意的是，一定要保证检测到的回环的正确性，引入错误的回环将会使整个系统的误差增加。检测到回环后，需要使用一定的策略将检测到的回环约束加入到系统中，对系统的状态重新优化，以将累计误差平均分配到整个轨迹上。下面将分别介绍回环检测的方法，和检测到回环后的优化方法。

4.5.1 回环检测方法

回环检测一般采用视觉的方式进行，即通过图像检索的方式检测历史图像中与当前图像相似的图像。传统方法采用穷举的方法，将当前的图像与所有的历史图像进行相似度计算，然而对于运行时间较长的 SLAM 系统，这种方法效率非常低，不能满足实时性要求。

为了不影响系统的实时性，需要采用更高效的图像检索算法。本文采用了目前最先进的基于词袋(Bag of Words)模型的回环检测方法。词袋模型源于自然语言处理中的信息检索，后在图像检索中也展现了非常好的效果。在基于词袋模型的图像检索中，图像可以用由视觉词汇组成的向量进行描述。图像中不同的特征描述子对应着不同的视觉词汇向量，可将这些视觉词汇进行离散分类，离散后构成的的描述子空间叫做视觉词典，通过视觉词典可将任意一图像描述子转换为视觉词汇。

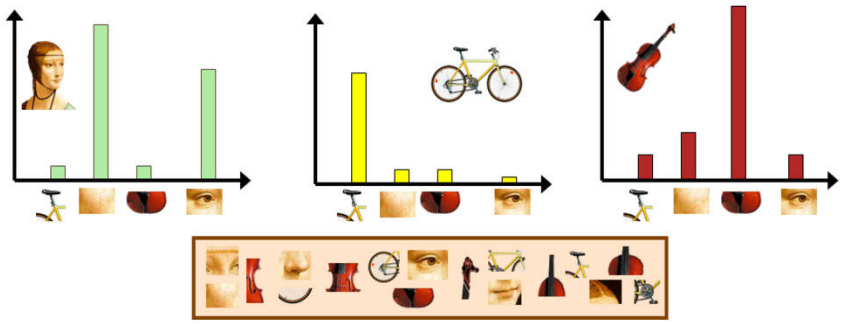


图 4-3 词袋模型示意图

如图 4-3 所示，图中下方的特征块的代表视觉词汇，他们一起构成了视觉词典。上方的三幅图像人脸，自行车和小提琴均可以用视觉词汇进行描述。故可通过描述不同图像视觉词汇的相似性来进行回环检测。

在本文的方法中，视觉词汇采用 ORB 描述子进行提取，视觉词典采用 ORB-SLAM 中预训练的词典。当关键帧滑动非线性优化窗口中被边缘化后作为查询关键帧进入回环检测数据库，当回环检测线程接受到新关键帧时，即会从回环检测数据库中检索与之匹配的关键帧。

4.5.2 回环闭合

检测到匹配后，会在全局地图中增加一条边的约束，然后在匹配的关键帧之间通过 3.3 节介绍的 2D-3D 的位姿估计方法算法来得到匹配关键帧之间的变换位姿，同时以此验证验证由 DBOW2 得到的匹配关键帧的正确性。

确认回环检测约束关系后，将进行全局位姿图优化，这里的位姿约束共有两种：一种是由非线性状态估计器得到的相邻关键帧之间的位姿变换，另一种是由闭环检测得到的全局地图中匹配的关键帧之间的位姿变换。可以通过优化下列代价函数：

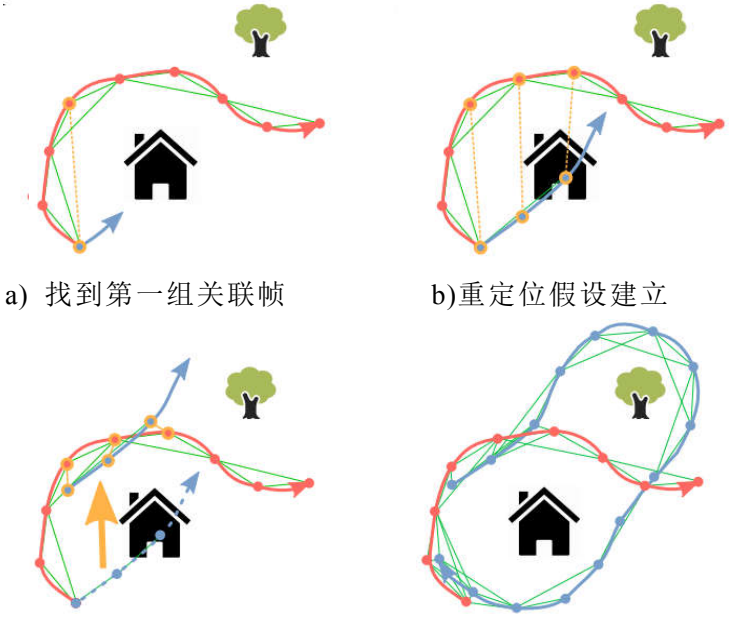
$$J(\mathbf{x}) \coloneqq \sum_{k=1}^{K-1} \mathbf{e}_{\text{vio}}^{k,k+1} \mathbf{W}_{\text{vio}}^{k,k+1} \mathbf{e}_{\text{vio}}^{k,k+1} + \sum_{(k,k')\in K} \mathbf{e}_{\text{loop}}^{k,k'} \mathbf{W}_{\text{loop}}^{k,k'} \mathbf{e}_{\text{loop}}^{k,k'} \tag{4-2}$$

式(4-19)中包括两个误差项，第一项为惯性视觉里程计误差项，第二项为回环误差项，$\mathbf{e}_{\text{vio}}^{k,k+1}$ 和 $\mathbf{e}_{\text{loop}}^{k,k'}$ 分别为视觉里程计的回环关键帧约束的残差，$\mathbf{W}_{\text{vio}}^{k,k+1}$ 和 $\mathbf{W}_{\text{loop}}^{k,k'}$ 分别为对应的信息矩阵，整个误差项衡量的是误差的马氏距离，可以用非线性优化求解器进行求解。

4.6 重定位与地图复用

重定位或地图复用，为机器人在已有先验地图信息且自身位置未知的情况下在已有地图上进行定位的方法。在视觉 SLAM 中，由于遮挡，纹理缺失或相机运动过快造成的跟踪失效非常常见，重定位可以在跟踪失效后进行位姿恢复。而在某些机器人或现实增强应用领域，可事先对地图场景进行构建，再在已知的场景中进行定位。这就需要 SLAM 系统支持地图复用功能，即系统要允许在之前构建的地图中进行定位，并且可以无缝地在新增加的环境部分中继续构建地图。重定位和地图复用的难点在于检测重定位并且进行地图的融合。

本文通过上一节介绍的回环检测方法来进行重定位的检测，通过新旧轨迹关键帧的连续匹配来判断新老地图的关系，重定位的过程如图 5-3 所示：



a) 找到第一组关联帧　　b)重定位假设建立

c) 地图对齐　　d)继续定位建图

图 5-3 重定位过程

图中红色的轨迹为前一次构建的轨迹，蓝色的轨迹为新轨迹，橙色的连线为回环检测匹配帧。图中 5-3 a)、5-3 b)、5-3 c)、5-3 d)即展示了重定位的过程：开始时，新旧轨迹地图相互独立，图 5-3 a)回环检测检测到第一个匹配的关键帧；图 5-3 b)随着新轨迹的构建发现连续的关键帧匹配，重定位假设成立；图 5-3 c)利用回环检测的匹配约束融合新旧地图和轨迹；图 5-3 d)在融合后的地图中继续进行定位建图。

4.7 本章小结

本章提出了一套基于单目视觉与惯性传感器融合的在线 SLAM 系统。首先介绍了系统的框架，它是一套完整的惯性视觉 SLAM 框架，具有目前最先进的三线程结构。接着介绍了整个系统初始化的过程，通过视觉惯性联合初始化将惯性位姿和视觉位姿联合起来，得到系统的初始估计值。接着介绍了系统最重要的后端的非线性优化方法，即紧耦合非线性视觉惯性状态估计器，给出了整个视觉惯性优化误差项，以及其计算方法和边缘化的过程。然后介绍了系统的回环检测与回环闭合方法，这使系统拥有识别回路并进行全局优化的能力，最后介绍了系统的重定位与地图复用的功能及其实现方式。

第5章　系统实现与实验分析

5.1 引言

本章旨在详细描述对本系统所进行的实验，给出实验的实验条件和结果，对所提出的系统进行实验测试以及进行精度分析，对所提出方法的性能进行评估，得出结论。

5.2实验条件与环境

为了验证本文提出算法的性能，采用了标准的惯性视觉数据集 Euroc 无人机数据集[45]。Euroc 无人机数据集包括机械室，Vicon 室等共三个场景共 11 个惯性视觉序列。数据集使用 Asctec Firefly 六旋翼直升机进行数据集采集，携带视觉惯性传感器单元，如图 5-1 所示。Vicon 6D 动作捕捉系统被用来捕捉六旋翼飞行器飞行时六自由度的位姿变换，并作为轨迹的真值。下面将以数据集中的一个序列为例，详细介绍实验的过程和结果。

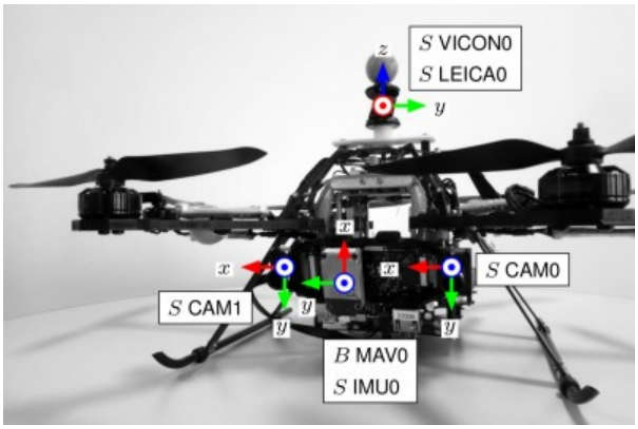


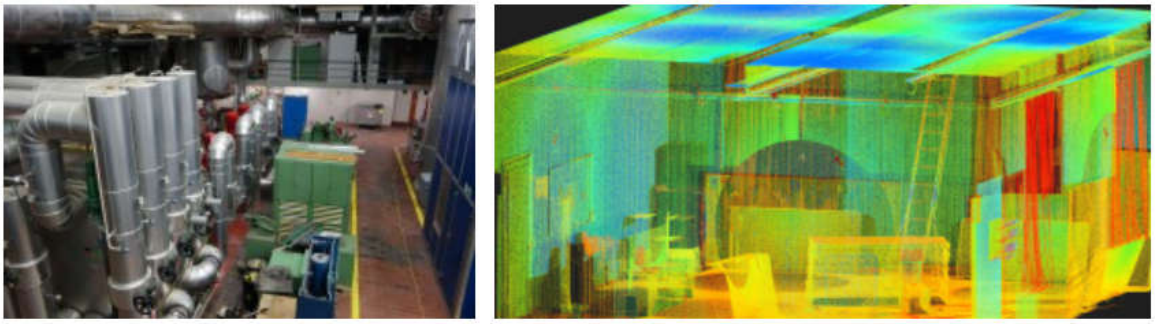
图 5-1 数据集采集数据使用的六旋翼飞行器

Euroc 数据集共记录了三个场景：一个机械室(MH, Machine Hall)和两个 Vicon 室(VR1，VR2)，共记录了 11 个序列。三个场景对应的序列分别为：MH01~MH05，VR11~VR13，VR21~VR23。序列的难度分为三个等级，如下表所示：

表 5-1Euroc 序列与难度表

序列	MH01	MH02	MH03	MH04	MH05	VR11	VR12	VR13	VR21	VR22	VR23
难度	易	易	中	难	难	易	中	难	易	中	难

下面以难度等级为难的 MH04 序列为例，介绍数据集的输入信息。该序列的场景为机械室，如图 5-2 所示，其中图 5-2 a)为机械室的场景图，图 5-2 b)为采用激光跟踪仪测对机械室得的真实三维数据。



a)机械室场景图　　b）激光跟踪仪测得的真实三维数据

图 5-2 MH04 序列所在的场景-机械室

数据集的图像与 IMU 均提供了统一的纳秒级别的 Unix 时间戳，将其转换为北京时间可知序列的采集时间为 2014/6/25 3:28:47 到 2014/6/25 3:30:28。这长度为 101 秒的视觉惯性序列，包含了 2033 对图像和 20320 组 IMU 加速度和角速度数据。

由于本系统输入是单目视觉与 IMU 信息，系统的输入仅采用了双摄像机即中左相机拍摄的图像，图像序列如图 5-3 所示，图像的分辨率为 752×480，帧率为 20Hz，为 8 位灰度图像。

图 5-3 输入图像序列示例

系统输入的 IMU 信息分别为三轴角速度和三轴加速度，如图 5-4 所示，

图像的横坐标为 IMU 的帧数，其中左侧的图像为输入的三轴角速度信息，单位为 rad/s。右侧的图像为输入的三轴加速度信息，加速度的单位为 m/s^2。

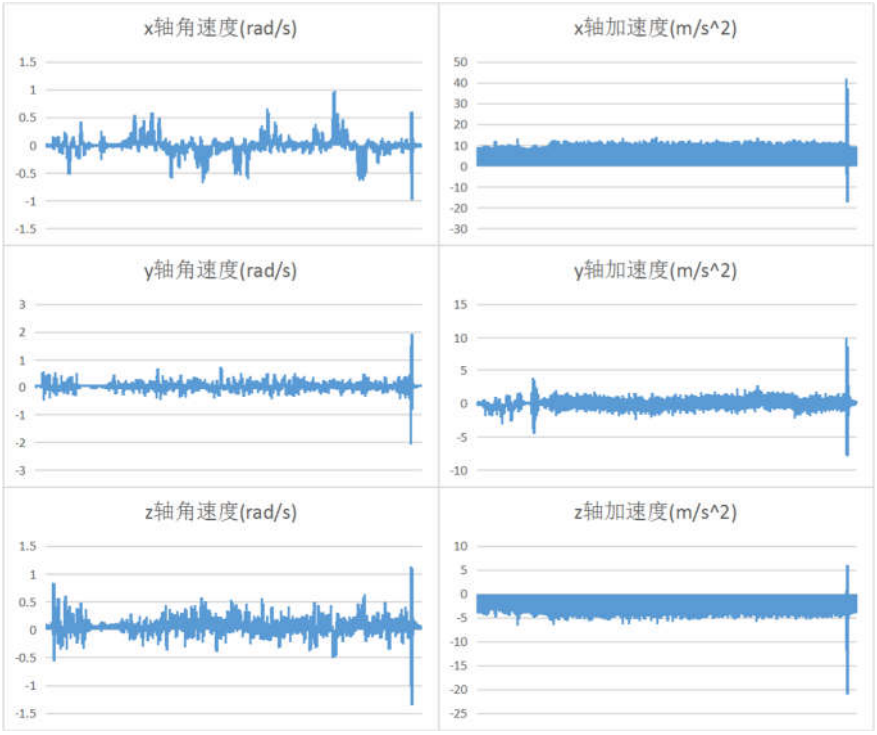


图 5-4 系统输入的 IMU 序列

相机和 IMU 的参数如表 5-2 所示：

表 5-2 相机与 IMU 参数

项目	参数
相机内参	$\begin{pmatrix} 458.654 & 0 & 367.215 \\ 0 & 457.296 & 248.375 \\ 0 & 0 & 1 \end{pmatrix}$
径向畸变	[-0.28340811, 0.07395907]
切向畸变	[0.00019359, 1.76187114e-05]
加速度计噪声	2.0000e-03
加速度计随机游走	3.0000e-03
陀螺仪噪声	1.6968e-04
陀螺仪随机游走	1.9393e-05

算法运行的硬件设备为 Inter NUC 迷你 PC，CPU 为 i7-6770HQ。实验所

用的系统为 Ubuntu16.04，整体软件架构搭载在机器人操作系统 ROS 上，ROS 版本文为 ROS Kinetic。系统主体在以 C++语言开发，程序主要用到四个开源库：计算机视觉库 Opencv 3.4.0，矩阵运算库 Eigen 3.3.3 与非线性优化求解器 Ceresolver，以及视觉惯性导航架构 VINS，可视化界面采用 ROS 自带的 Rviz 工具编写。

系统线程以 ROS 节点的形式存在，系统各个节点之间通过发布/订阅(publish/subscribe)消息模型进行通信。输入的数据以 ROS 数据包的格式存在，ROS 数据包可以记录传感器产生的数据，并以相同的频率发布所接收到的数据。在本系统中，ROS 数据包发布的是带有时间戳的图像信息和信息，以 MH04 序列为例，使用 rosbag info 命令可以查看 ROS 数据包的信息，输出信息如图 5-5 所示：

```
path:        MH_04_difficult.bag
version:     2.0
duration:    1:42s (102s)
start:       Jun 25 2014 03:28:47.33 (1403638127.33)
end:         Jun 25 2014 03:30:29.91 (1403638229.91)
size:        1.4 GB
messages:    25855
compression: none [1356/1356 chunks]
types:       geometry_msgs/PointStamped [c63aecb41bfdfd6b7e1fac37c7cbe7bf]
             sensor_msgs/Image          [060021388200f6f0f447d0fcd9c64743]
             sensor_msgs/Imu            [6a62c6daae103f4ff57a132d6f95cec2]
topics:      /cam0/image_raw    2033 msgs    : sensor_msgs/Image
             /cam1/image_raw    2032 msgs    : sensor_msgs/Image
             /imu0             20320 msgs    : sensor_msgs/Imu
             /leica/position    1470 msgs    : geometry_msgs/PointStamped
```

图 5-5 ROS 数据包信息

从图中显示的数据包中信息可知数据包包含的数据种类和数量，以及记录的时间。从图中可以看出共包含 4 个主体，本文系统只用到了其中两个，即/cam0/image_raw 和/imu0，前者为左相机图像数据，后者为 IMU 三轴加速度与角速度数据。

5.3 实验测试结果

5.3.1 视觉前端测试结果

前端已图像为输入，输出实时提取并跟踪的特征点，前端处理过程由 3.2 节给出，主要采用 Shi-Tomas 角点提取与 KLT 光流法进行跟踪，为了保证实时性和跟踪效果，保持特征点数量不超过 300 个，并且设定特征点提取的最小间隔为 15 个像素。实际前端节点订阅 ROS 数据包发布的图像信息，利用回调函数处理图像，最终发布提取和跟踪到的图像特征点。下图为 MH04 序列无人机起飞时，视觉前端输出的结果：

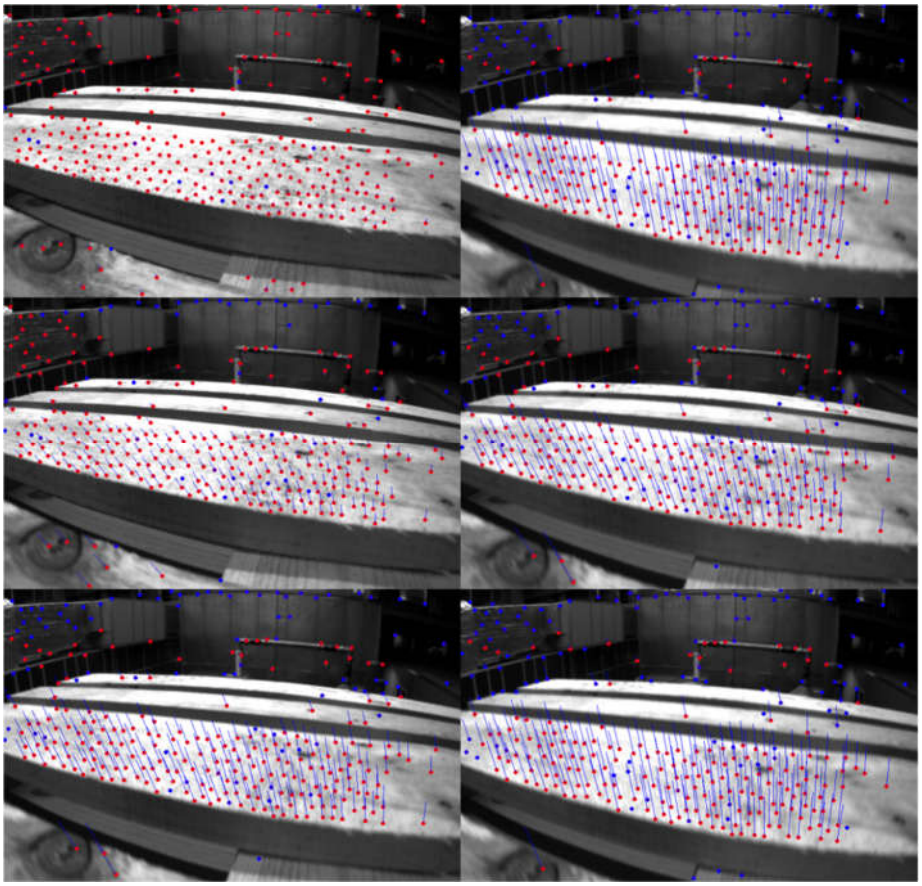


图 5-6 视觉前端输出结果

图中选取的是连续六帧视觉图像进行特征提取和跟踪的结果，图像顺序为逆时针顺序，左上角图像为第一帧，右上角图像第六帧，为无人机刚起飞时拍摄到的图像。图中红色和蓝色点即为提取到的 Shi-Tomas 角点，其中红色的点为跟踪次数较多的点，说明跟踪比较稳定，蓝色的点为新提取到的点。可以看出，提取到的特征点比较多，且分布较平均。这是由于设置了特征点的最小间隔，进行非极大值原因。图中蓝色的连线即为 KLT 光流法跟踪的特征点运动。可以看出图像中的大部分点都能准确跟踪，跟踪效果较好。且随着无人机的起飞，近处的场景特征点运动较快，远处场景特征点基本保持不变，这也在光流跟踪中体现了出来。

5.3.2 优化后端测试结果

优化后端包括两个过程，初始化和滑动窗口状态估计。其中初始化仅在系统刚运行时进行，主要为了计算系统运行必要的初值，以匹配的视觉帧和 IMU 信息为输入，输出估计的系统参数，包括重力 g，陀螺仪偏置 b_g，尺度

因子 s 以及初始的位姿估计，初始化的具体过程已在 4.3 节进行介绍。滑动窗口状态估计以匹配的视觉帧和 IMU 信息作为输入，输出每一个视觉帧对应的位姿估计和三维地图点，具体处理过程已在 4.4 节中进行介绍。

以 MH04 序列为例，系统初始化共耗时 5.56s ，计算得重力加速度为 $\mathbf{g}=[3.9968\times10^{-15},\ -3.1225\times10^{-17},\ 9.81007]$，尺度因子为 $s=0.007282$，陀螺仪偏置为 $\mathbf{b}_{\mathbf{g}}=[-0.00221163, 0.0219409, 0.0759545\ ]$。

初始化完毕后，系统进行正常的后端优化，输出无人机时实时三维位姿信息。系统输出信息如图 5-7 所示，该图为程序在命令行中的输出片段，实时输出带有时间戳的位姿，为了显示效果将图像进行反色显示。

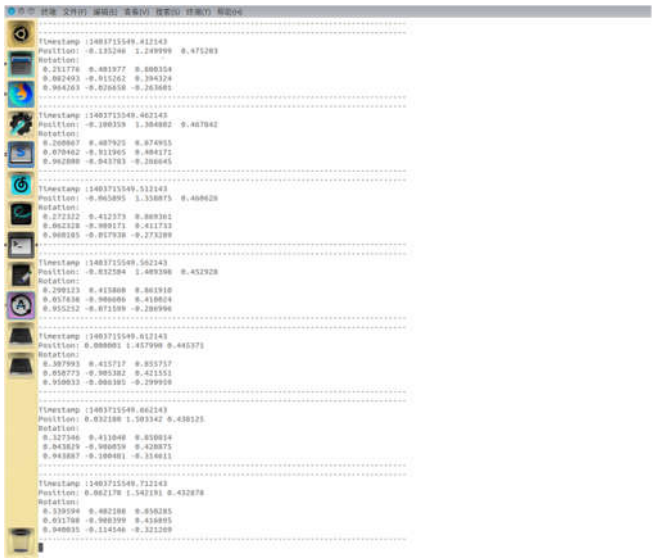


图 5-7 程序输出片段-无人机坐标

如图中所示，每个输出块内的第一行位该帧的时间戳，单位为秒。第二行为无人机该帧在世界坐标系（初始化成功时，第一个视觉帧与重力对齐的坐标系）下的位置信息，单位为米。最后的 3x3 矩阵为世界坐标系到该帧 IMU 坐标系的旋转矩阵，表示了无人机在该帧下的姿态。

除了直接输出数据，还可以通过 Rviz 可视化界面观察无人机运行轨迹。Rviz 为 ROS 系统中专门的可视化工具，可以较方便的显示系统的轨迹和三维点信息。系统输出的无人机轨迹在 Rviz 中显示的结果如图 5-8 所示：

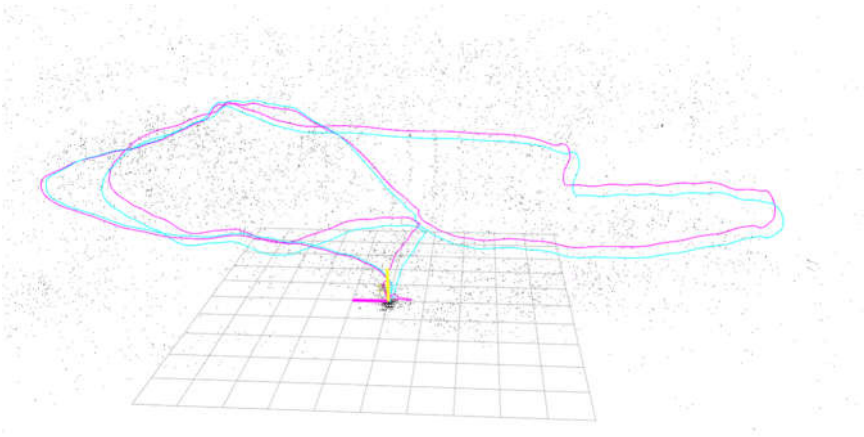


图 5-8 无人机轨迹与三维点

图中紫色的线为用系统输出的位姿信息画出的无人机轨迹，蓝色的线为无人机数据集提供的轨迹的真值，可以看出两条轨迹较为接近，说明计算出的位姿状态比较准确。周围稀疏的点为计算出来的三维路标点，由于本文方法要保持实时性，故对提取的特征点数量有所限制，只求解出稀疏的三维路标点。同时我们注意到计算估计值与真值的差距明显会随着距离与时间的增长而增长，说明系统还是存在一定的累积误差。

5.3.3 回环检测与全局位姿优化结果

该部分以后端优化出的带有位姿信息的视觉关键帧和三维特征点为输入，对视觉关键帧的相似性进行检测，并进行优化，最终输出全局误差一致的关键帧位姿信息。具体方法在 4.5 与 4.6 中进行介绍。

回环检测线程实时检索当前帧与关键帧数据库中的关键帧匹配情况，当发现匹配关键帧时，计算两帧图像的 ORB 特征点匹配关系，如图 5-9 所示：

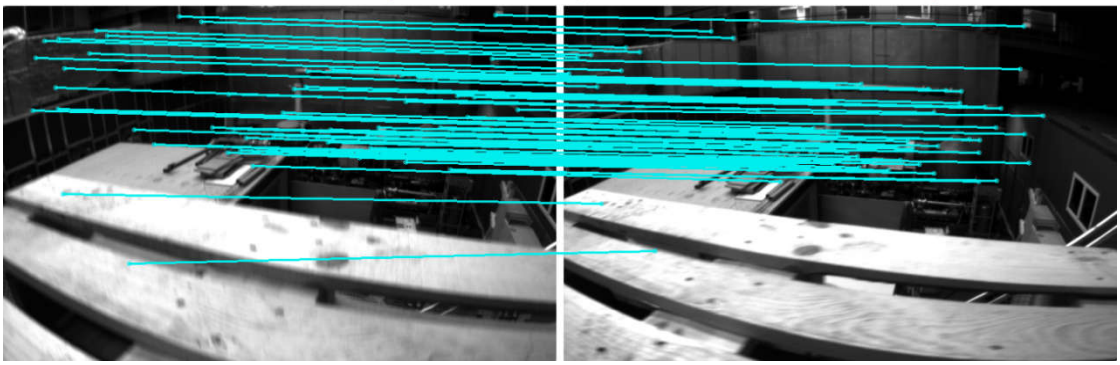


图 5-9 回环检测匹配

图中蓝色的连线即为 ORB 特征点的匹配关系，肉眼可看出匹配到的特征对均为正确的匹配点对。回环检测后的位姿图优化主要有两个功能：进行回环闭合或重定位和地图复用。

通过回环检测，可通过将检测到匹配的关键帧进行约束，从而将其误差平均分配到整个轨迹上，如图 5-10 所示：

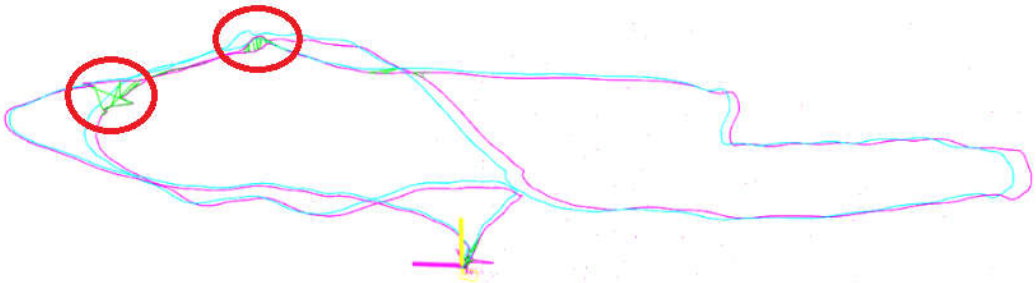


图 5-10 带有回环闭合的轨迹

图中轨迹与图 5-8 相同，为 MH04 序列计算出的无人机轨迹。其中画红圈的部分即为回环检测到匹配并且约束的帧，他们的约束在图中以绿色的短连线表示。从图中可以看出估计值与真值的误差要比不带回环检测的结果要小，尤其是在检测到回环处附近。通过回环检测，可以通过将误差平均分配到整个轨迹上以得到全局一致的轨迹和地图，从而大大改善累积误差的问题。

除回环闭合外，为了验证系统对重定位和地图复用功能的支持，对其进行了实验。首先运行 MH05 序列，计算相应的位姿轨迹和地图三维点信息并将将信息保存。接着运行在同一场景下采集的 MH04 序列，此时的视觉和惯性序列与 MH05 序列并不连续，测试其输出的轨迹信息。输出结果如图 5-11 所示。

图 5-11 中深蓝色的线为 MH05 序列已经计算出的轨迹，紫色的线为本次 MH04 序列计算出的轨迹，通过回环检测的 ORB 特征匹配，可将本次输出的轨迹对应到之前计算出的轨迹上，图中浅蓝色的线即为回环检测构成的约束。并在相同坐标系下继续构建轨迹和地图。

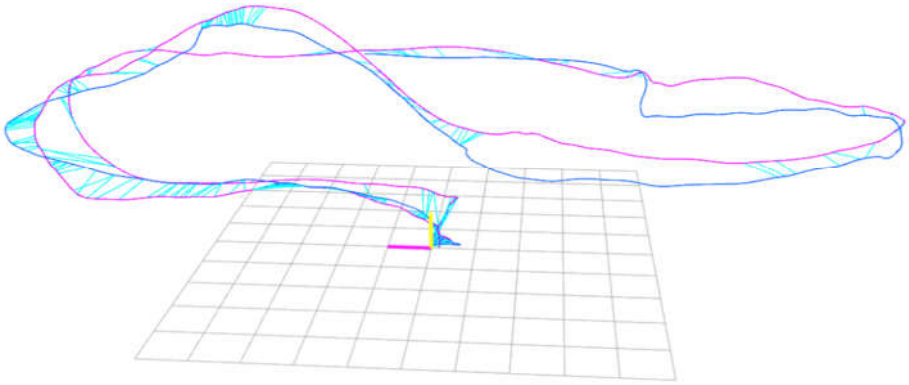


图 5-11 地图复用输出结果

5.4 实验精度分析

5.4.1 精度评价指标

上一节对实验的结果进行的是定性的分析，证明了本文提出算法的有效性和回环检测的有效性。本小节将从定性的指标上系统性的分析本文所提出算法的精度，并与目前主流视觉惯性 SLAM 算法进行对比。

在进行精度分析之前，首先介绍一下衡量精度的指标，在评价 SLAM 算法的精度时，主要采用的指标有两个：相对位姿误差(Relative Pose Error，RPE)[46]和绝对位姿误差（Absolute Pose Error，APE）。下面将分别介绍这两个指标：

1）相对位姿误差：RPE 计算在固定的时间 Δ 间隔内，它可以衡量路径的局部正确率。设估计的位姿为 $P_i \in SE(3), i=1...n$, 位姿的真值为 $Q_i \in SE(3), i=1...n$, 定义时刻 i 的相对位姿误差为:

$$\mathrm{E}_i = (\mathrm{Q}_i^{-1}\mathrm{Q}_{i+\Delta})^{-1}(\mathrm{P}_i^{-1}\mathrm{P}_{i+\Delta}) \tag{5-1}$$

对于有 n 个相机姿态的轨迹，共可以得到 $m=n-\Delta$ 段相对位姿误差，对所有的 RPE 可计算均方根误差(RMSE, Root Mean Squared Error):

$$\mathrm{RMSE}\left(\mathrm{E}_{i:n},\Delta\right) = \left(\frac{1}{m}\sum_{i=1}^{m}\left\|trans(\mathrm{E}_i)\right\|^2\right)^{\frac{1}{2}} \tag{5-2}$$

其中，$trans(\mathrm{E}_i)$ 表示相对位姿误差 E_i 中的平移部分，此时的均方根误差代表轨迹的均方根误差。实验过程中，一般比较相邻帧的相对位姿误差，故 Δ 一般取 1。

2）绝对位姿误差：通过比较估计的位姿与真值之间的距离来衡量轨迹的全局误差，此误差可反映全局路径的偏差程度。设估计的位姿为

$P_i \in SE(3), i=1...n$, 位姿的真值为 $Q_i \in SE(3), i=1...n$, 此时定义 i 时刻的误差为：

$$\mathrm{E}_i = \mathrm{Q}_i^{-1}\mathrm{P}_i \tag{5-3}$$

利用所有时刻的绝对误差可计算它的均方根误差：

$$\mathrm{RMSE}(\mathrm{E}_{i:n}) = \left(\frac{1}{m}\sum_{i=1}^{m}\left\|trans(\mathrm{E}_i)\right\|^2\right)^{\frac{1}{2}} \tag{5-4}$$

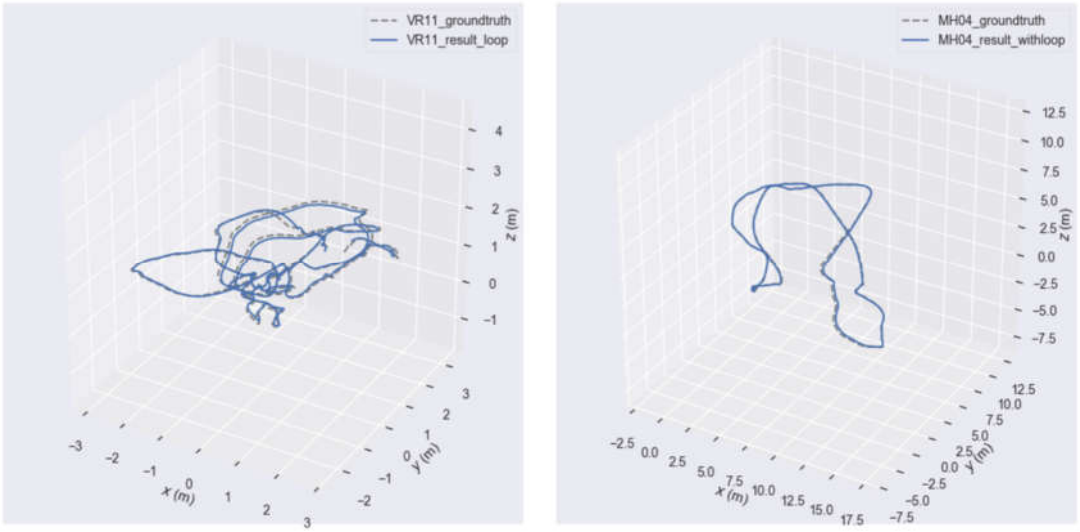
5.4.2 实验结果精度分析

本节将对系统计算出的轨迹信息进行定量评估，分别以较简单的 VR11 序列和较难的 MH04 序列为例。针对每一个序列输出序列，系统分别有两个输出序列：一个为后端优化线程实时输出的无回环位姿，另一个是位姿图优化线程输出的有回环约束的位姿估计，下面分析如不加特别说明均指有回环的位姿输出。系统输出的位姿帧数，时间与位姿总长度如表 5-3 所示，其中真值来自于 Vicon 6 自由度光学运动捕捉系统。

表 5-3 系统输出数据帧数，时间与长度

属性	VR11 真值	VR11 无回环	VR11 有回环	MH04 真值	MH04 无回环	MH04 有回环
总帧数(帧)	28712	2784	1461	19753	1961	904
总时间(s)	143.555	139.150	135.650	98.760	98.000	96.400
总长度(m)	58.592	57.922	56.942	91.747	90.991	91.150

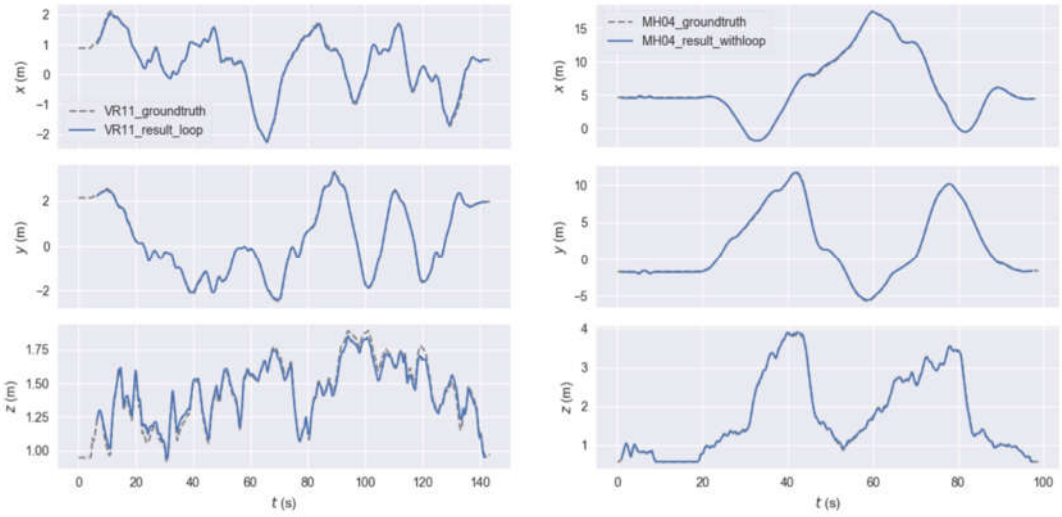
两个序列输出的轨迹在三维坐标系下如图 5-12 所示：



a)VR11 序列　　b)MH04 序列

图 5-12 两个序列输出的轨迹

两个序列输出的轨迹在 x，y，z 一维坐标系下如图 5-13 所示：

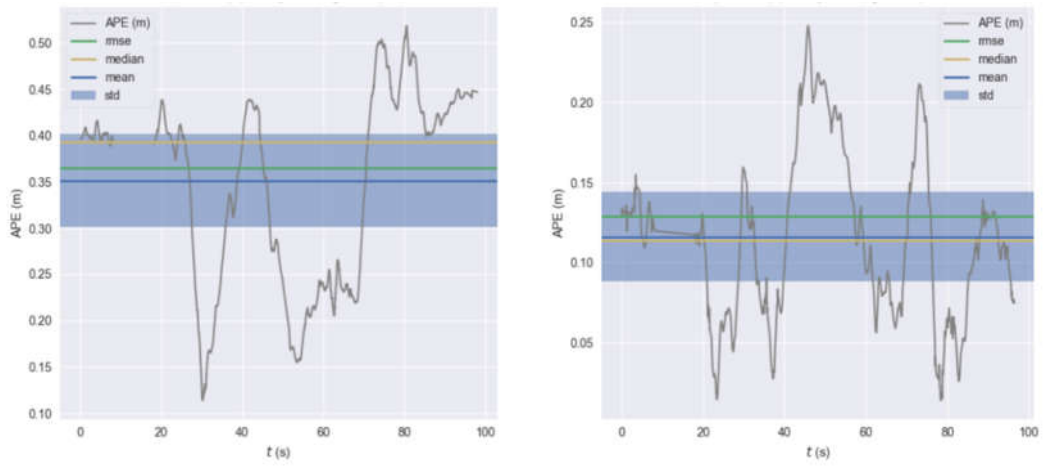


a)VR11 序列　　b)MH04 序列

图 5-13 两个序列分别在 x，y，z 坐标下输出的轨迹

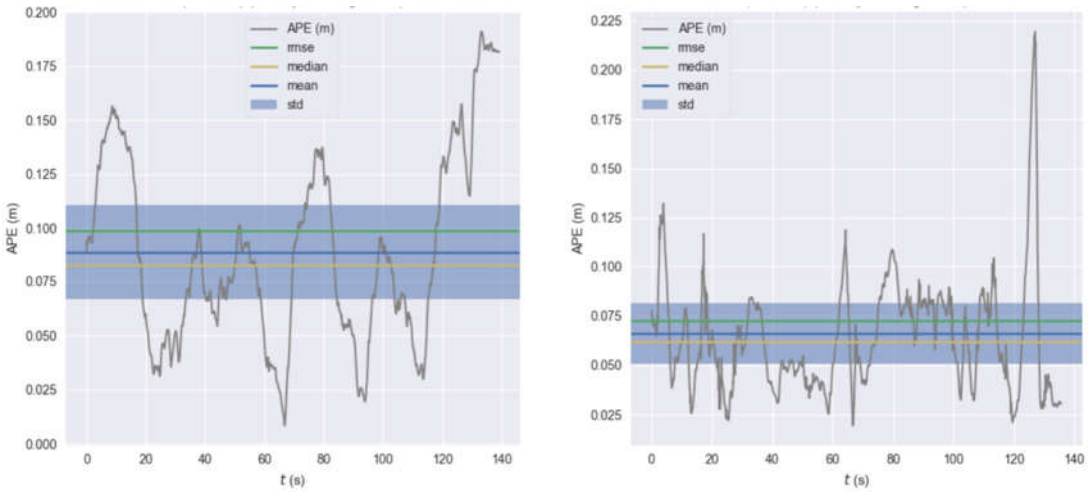
在 5-12 及图 5-13 中蓝色的线为系统输出的随时间变化的轨迹，虚线为轨迹的真值，两条线若重合真值会被覆盖显示为蓝色。可以看出在图中坐标系下大部分系统预测的轨迹均与真值重合，说明系统输出的轨迹信息较为准确。由于系统输出位姿误差相对于无人机运动的尺度来讲比较小，因此在上两幅图中并不能很好的误差的大小。

下面将利用 5.4.2 中介绍的相对位姿误差与绝对位姿误差分析输出的位姿信息。首先是绝对位姿误差 APE，图 5-14 和图 5-15 分别给出了两个序列 APE 随时间变化情况，表 5-4 给出了两个序列的 APE 的具体统计学指标数据。



a)无回环　　b)有回环

图 5-14 MH04 序列的绝对位姿误差



a)无回环　　b)有回环

图 5-15 VR11 序列的绝对位姿误差

表 5-4 绝对位姿误差统计学指标数据

评价指标	VR11 无回环	VR11 有回环	MH04 无回环	MH04 有回环
均方根误差(m)	0.0988	0.0725	0.3652	0.1289
最大值(m)	0.1909	0.2194	0.5182	0.2476
平均值(m)	0.0886	0.0659	0.3514	0.1162
中位数(m)	0.0824	0.0616	0.3929	0.1133
标准差(m)	0.0436	0.0301	0.0995	0.0558

从上面图 5-14，图 5-15 以及与表 5-4 中数据中可以得出以下几点结论：

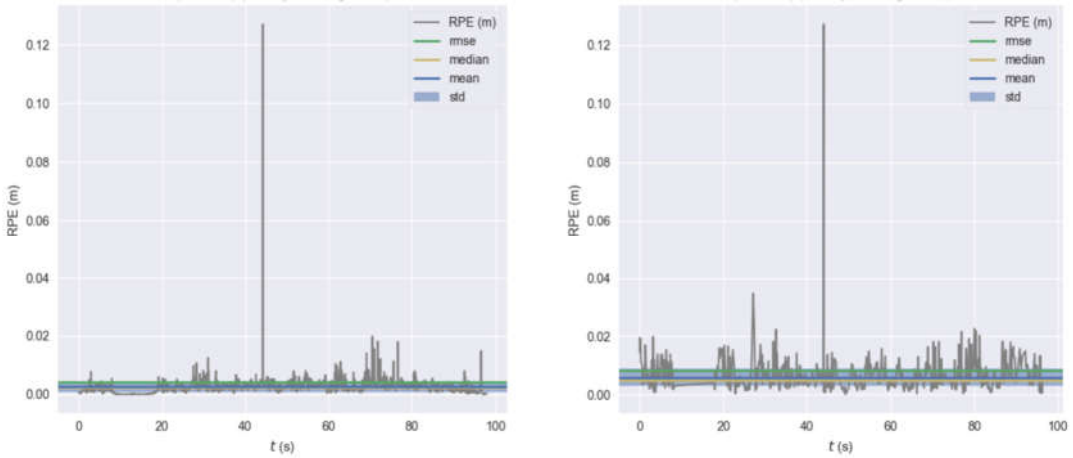
1. VR11 序列输出 APE 的均方根误差为 0.073m，MH04 序列输出的均方根误差为 0.129m。相相比于 58.592m 和 91.747m 的序列长度误差较小，系统精度较高。

2. 通过两个序列间对比，MH04 的 APE 明显要比 VR11 要大。这与 MH04 序列更难且轨迹更长的条件相符。

3. 从相同序列有无回环的对比中可以看出，通过加入回环检测，系统输出轨迹的 APE 误差得到了明显的改善，说明回环检测对系统绝对位姿误差的减小起到了较好的效果。

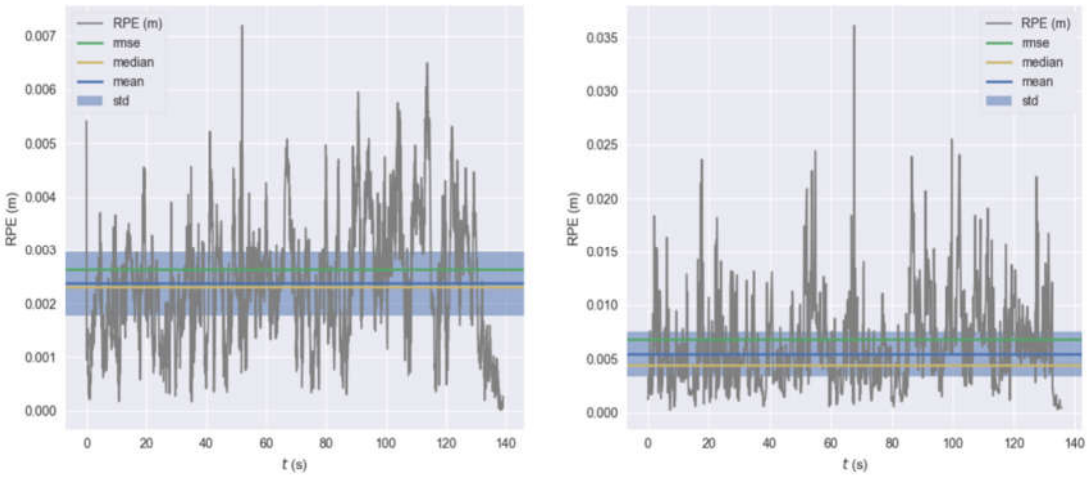
4. MH04 回环检测使其 APE 从 0.365m 下降到 0.129m，共下降 0.236m，而 VR11 序列仅使其下降 0.028m。说明回环检测的效果对整体误差较大，轨迹更长的输入效果更好。

下面分析相对位姿误差 RPE，图 5-16 和图 5-17 分别给出了 RPE 随时间变化情况，表 5-5 给出了两个序列的 RPE 的具体统计学指标数据。



a)无回环　　b)有回环

图 5-16 MH04 序列的相对位姿误差



a)无回环　　b)有回环

图 5-17 VR11 序列的相对位姿误差

表 5-5 相对位姿误差统计学指标数据

评价指标	VR11 无回环	VR11 有回环	MH04 无回环	MH04 有回环
均方根误差(m)	0.00264	0.00681	0.00423	0.00830
最大值(m)	0.00719	0.03608	0.12700	0.12710
平均值(m)	0.00237	0.00547	0.00254	0.00597
中位数(m)	0.00231	0.00447	0.00228	0.00485
标准差(m)	0.00116	0.03019	0.00338	0.00576

从上面图 5-16，图 5-17 以及与表 5-5 中数据中可以得出以下几点结论：

1. VR11 序列输出 RPE 的均方根误差为 0.0068m，MH04 序列输出的均方根误差为 0.0083m，误差级别为毫米级，比 APE 误差精度高 1-2 个数量级，说明系统局部的定位精度很高。

2. 通过两个序列间对比，MH04 的 RPE 的平均值与 VR11 相差不多。但 MH04 序列 RPE 的最大值 0.127m 远大于 VR11 的 0.036m，使得 MH04 序列 RPE 的均方根值更大。说明序列的难度的增加会使得局部 RPE 误差增大，从而使得 RPE 的均方根误差增大。

3. 从相同序列有无回环的对比中可以看出，通过加入回环检测，系统输出轨迹的 RPE 误差不降反增，结合 APE 的误差分析结果，可以得出回环检测会使全局误差 APE 减小，而使局部误差 RPE 增加的结论。分析原因可知，回环检测和全局优化的目的即是减小累计误差，使系统输出轨迹具有全局误差一致性，这与本文得出的使 APE 误差减小的实验结果相符，但 RPE 误差增大的实验结果表明，减小累计误差是以牺牲局部定位精度为代价的。

除了上述两个序列，本文对 Euroc 视觉惯性数据集 10 个序列均进行了实验测试，计算了 10 个序列的绝对位姿误差的均方根值，并将其与目前主流的视觉惯性 SLAM 方法 OKVIS[47]进行对比，其中 OKVIS 的输入的视觉与 IMU 信息与本文系统完全一样，评价指标同样为绝对位姿误差的均方根值，其方法的误差结果由论文[48]给出 。本文方法 VI-SLAM 与主流方法 OKVIS 针对不同 Euroc 序列的绝对位姿均方根误差结果对比如表 5-3 所示：

表 5-6 本文方法 OKVIS 方法对比结果

序列	VI-SLAM APE(m)	OKVIS APE(m)
MH1	0.106	0.342
MH2	0.121	0.363
MH3	0.113	0.301
MH4	0.129	0.483
MH5	0.264	0.473
VR11	0.073	0.125
VR12	0.077	0.160
VR13	0.089	0.246
VR21	0.073	0.125
VR22	0.093	0.224

通过对比可得，本文提出的 VI-SLAM 系统的精度全面优于目前主流 OKVIS 方法。

除了对系统的定位精度进行了测试分析，本文还对系统各个线程模块的运行时间进行了测试。系统各个模块的运行时间如下表所示：

表 5-7 系统模块运行时间

线程	模块	时间(ms)	频率(Hz)
特征提取与跟踪	Shi-Tomas 角点提取	15.5	25
	KLT 光流法跟踪	4.9	25
滑动窗口非线性优化	非线性优化	52.4	10
回环检测与全局优化	回环检测	98.2	
	全局位姿图优化	134.7	

其中特征提取与跟踪部分为实时跟踪，运行一次耗时 20.4ms，可稳健的在 25Hz 的频率下进行跟踪。非线性优化部分耗时 52.4ms，运行在 10Hz 的频率下，该频率也是系统输出状态的频率。回环检测线程在后台运行，不需要实时输出结果，而全局位姿图优化线程仅在检测到回环后创建。因此这两部分耗时较长并不影响系统的实时性。

5.5 本章小结

本章主要进行了对所构建的视觉惯性 SLAM 系统进行实验测试。首先介绍实验的条件与环境，包括所采用的实验数据，即 Euroc 无人机惯性视觉数据集，以及实验系统的软件环境。展示了 Shi-Tomas 提取以及光流法跟踪效果，接着给出了系统的运行结果，即实时位姿轨迹输出，并将带有回环检测系统运行结果不带的结果进行对比，说明了回环检测在视觉惯性系统中的有效性，并给出了带有地图复用的输出结果，成功地在有先验场景信息的基础上给出在坐标系下的无人机轨迹。

接着对系统进行了定量分析，分析了输出结果的绝对位姿误差与相对位姿误差，其中 VR11 序列的绝对位姿误差的精度可达 0.073m，并将结果与目前现有的基于优化方法的视觉惯性算法 OKVIS 进行对比，证明了本方法在精度上由于 OKVIS 方法。最后展示了系统运行时间，说明了本系统具有实时性。结果表明，所构建单目视觉与 IMU 结合的 SLAM 的系统具有精度高，实时性好，鲁棒性好等特点，并且具有回环检测与地图复用功能。

结论

本文点对基于单目视觉与 IMU 结合的 SLAM 技术进行了深入的研究。对 SLAM 基于尤其是基于视觉与 IMU 结合的 SLAM 技术做了全面的综述。深入分析了 SLAM 相关技术原理，提出并构建了一套完整的惯性视觉 SLAM 技术方案。最后，通过基于 Euroc 无人机惯性视觉数据集的实验，对系统进行测试，并对结果进行精度分析。实验结果表明，本文所提出的视觉惯性 SLAM 系统方案准确有效，具有良好的实时性和鲁棒性。系统的绝对位姿误差均方根值可达 0.073m，相对位姿误差均方根值最小可达 0.0026m，优于主流视觉惯性方法 OKVIS。

本文的完成的主要工作如下：

（1）对视觉 SLAM 相关理论进行分析和相关公式推导，主要包括：惯性视觉 SLAM 的数学建模，基于连续图像帧的位姿估计，IMU 预积分理论及公式以及惯性视觉紧耦合非线性优化的状态估计器。

（2）提出了一套完整的在线紧耦合惯性单目视觉的 SLAM 系统方案，将视觉误差与 IMU 误差融合在一个统一的后端优化框架中。使用了基于关键帧，滑动窗口等策略优化系统运行时间。尤其在原有视觉里程计的框架下加入 SLAM 层的回环检测和重定位功能。在现有开源单目视觉 SLAM 框架基础下，结合计算机视觉库 opencv 和非线性优化库 ceres 在机器人操作系统框架 ROS 上，完成了对本文所提出系统的构建。

（3）对所构建的视觉惯性 SLAM 系统在 Euroc 无人机视觉惯性数据集中进行了实验和测试，实验结果表明，系统可以 10Hz 的频率实时输出无人机的位姿信息，具有良好的实时性。系统的绝对位姿误差均方根值最小可达 0.073m，相对位姿误差均方根值最小可达 0.0026m。与目前主流视觉惯性 OKVIS 方法相比，本文所提系统计算的位姿定位精度更高。

尽管基于单目视觉与 IMU 结合的 SLAM 技术虽然已有一定的发展，本文所提出的系统也可以完成惯性相机（机器人）的实时定位与地图构建。但惯性视觉 SLAM 技术离真正全面应用还有一段距离，还有很多方面的问题可以继续去完善解决。待优化改进的部分包括：

（1）尽管惯性视觉融合的 SLAM 系统相比纯视觉对快速运动的鲁棒性有所提升，但当机器人变向剧烈时，还是不能完全保证定位成功，系统的鲁棒性需要进一步提高。

（2)虽然本文所提出的惯性视觉SLAM系统在PC上已能在线实时运行，现实生活中很多应用都运行在轻量级设备上，系统的计算复杂度需要进一步优化改进。

（3）后续可在本系统的基础上，在更多传感器融合，或多机器人协作方面继续进行研究和开发，增强系统的实用性与稳定性。

参考文献

[1] J.J Lenonard, H.F Durrant-Whyte. Mobile Robot Localization by Tracking Geometric Beacons[J]. IEEE Transactions on Robotics and Automation, 1991, 7:376-382.

[2] H.Durrant-Whyte, T.Baily.Simultaneous Localization and Mapping(SLAM)[J]. Part I The Essential Algorithms. IEEE Journal of Robotics and Automation,2006,13(2):99-110.

[3] Baird W H. An introduction to inertial navigation[J]. American Journal of Physics, 2009, 77(9):844-847.

[4] Ronnback S. Development of a INS/GPS navigation loop for an UAV[D]. Lulea Tekniska University of Technology,2000.

[5] Leonard J J, Durrant-Whyte H F. Simultaneous Map Building and Localization for an Autonomous Mobile Robot[C] Ieee/rsj Int. Workshop on Intelligent Robots and Systems. 1991:1442-1447 vol.3.

[6] Gui J, Gu D, Wang S, et al. A review of visual inertial odometry from filtering and optimisation perspectives[J]. Advanced Robotics, 2015, 29(20):1289-1301.

[7] Barshan B, Durrant-Whyte H F. Evaluation of a solid-state gyroscope for robotics applications[J]. IEEE Transactions on Instrumentation & Measurement, 1995, 44(1):61-67.

[8] Davison A J, Reid I D, Molton N D, et al. MonoSLAM: Real-Time Single Camera SLAM[J]. IEEE Trans Pattern Anal Mach Intell, 2007, 29(6):1052-1067.

[9] Civera J, Davison A J, Montiel J M M. Inverse Depth Parametrization for Monocular SLAM[J]. IEEE Transactions on Robotics, 2008, 24(5):932-945.

[10] Chiuso A, Favaro P, Jin H, et al. Structure from motion causally integrated over time[M] Computer Vision — ECCV 2000. Springer Berlin Heidelberg, 2000:523-535.

[11] Eade E, Drummond T. Scalable Monocular SLAM[C] Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. IEEE, 2006:469-476.

[12] Klein G, Murray D. Parallel Tracking and Mapping for Small AR Workspaces[C] IEEE and ACM International Symposium on Mixed and Augmented Reality. IEEE Computer Society, 2007:1-10.

[13] Mouragnon E, Lhuillier M, Dhome M, et al. Real Time Localization and 3D

Reconstruction[C] IEEE Computer Society Conference on Computer Vision & Pattern Recognition. IEEE Computer Society, 2006:363-370.
[14] Hauke Strasdat, J.M.M. Montiel, Andrew J. Davison. Visual SLAM: Why filter [J]. Image & Vision Computing, 2012, 30(2):65-77.
[15] Klein G, Murray D. Improving the Agility of Keyframe-Based SLAM[C] European Conference on Computer Vision. Springer-Verlag, 2008:802-815.
[16] Hauke Strasdat, J.M.M. Montiel, Andrew J. Davison. Scale drift-aware large scale monocular SLAM[J] Robotics:Science and System(RSS), Zaragoza, Spain, 2010
[17] Strasdat H, Davison A J, Montiel J M M, et al. Double window optimisation for constant time visual SLAM[C] International Conference on Computer Vision. IEEE Computer Society, 2011:2352-2359.
[18] Pirker K, Rüther M, Bischof H. CD SLAM - continuous localization and mapping in a dynamic world[C] Ieee/rsj International Conference on Intelligent Robots and Systems. IEEE, 2011:3990-3997.
[19] Engel J, Schöps T, Cremers D. LSD-SLAM: Large-Scale Direct Monocular SLAM[J]. 2014, 8690:834-849.
[20] Forster C, Pizzoli M, Scaramuzza D. SVO: Fast semi-direct monocular visual odometry[C] IEEE International Conference on Robotics and Automation. IEEE, 2014:15-22.
[21] Mur-Artal R, Montiel J M M, Tardós J D. ORB-SLAM: A Versatile and Accurate Monocular SLAM System[J]. IEEE Transactions on Robotics, 2017, 31(5):1147-1163.
[22] 夏凌楠, 张波, 王营冠,等. 基于惯性传感器和视觉里程计的机器人定位[J]. 仪器仪表学报, 2013, 34(1):166-172.
[23] 叶波. 基于四旋翼平台的融合单目视觉与惯性传感的里程计方法研究[D]. 浙江大学, 2017.
[24] 王亭亭. 无人机室内视觉/惯导组合导航方法[J]. 北京航空航天大学学报, 2018, 44(1):176-186.
[25] 王德智. 基于 ROS 的惯性导航和视觉信息融合的移动机器人定位研究[D]. 哈尔滨工业大学, 2017.
[26] Bloesch M, Omari S, Hutter M, et al. Robust visual inertial odometry using a direct EKF-based approach[C] Ieee/rsj International Conference on Intelligent Robots and Systems. IEEE, 2015:298-304.
[27] Mourikis A I, Roumeliotis S I. A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation[C] IEEE International Conference on Robotics and Automation. IEEE, 2007:3565-3572.
[28] Weiss S, Achtelik M W, Chli M, et al. Versatile distributed pose estimation

and sensor self-calibration for an autonomous MAV[C] IEEE International Conference on Robotics and Automation. IEEE, 2012:31-38.

[29] Weiss S, Siegwart R. Real-time metric state estimation for modular vision-inertial systems[C] IEEE International Conference on Robotics and Automation. IEEE, 2011:4531-4537.

[30] Mourikis A I, Roumeliotis S I. A dual-layer estimator architecture for long-term localization[C] Computer Vision and Pattern Recognition Workshops, 2008. CVPRW08. IEEE Computer Society Conference on. IEEE, 2008:1-8.

[31] Leutenegger S, Furgale P, Rabaud V, et al. Keyframe-Based Visual-Inertial SLAM using Nonlinear Optimization[C]// Robotics: Science and Systems. 2013:789–795.

[32] Forster C, Carlone L, Dellaert F, et al. On-Manifold Preintegration for Real-Time Visual--Inertial Odometry[J]. IEEE Transactions on Robotics, 2017, 33(1):1-21.

[33] 李立杭. 视觉SLAM中的多传感器标定及稠密地图创建的研究[D]. 中国科学院大学, 2015.

[34] Holmgren D E. An invitation to 3-d vision: from images to geometric models[J]. Photogrammetric Record, 2010, 19(108):415-416.

[35] Cadena C, Carlone L, Carrillo H, et al. Simultaneous Localization And Mapping: Present, Future, and the Robust-Perception Age[J]. IEEE Transactions on Robotics, 2016, 32(6).

[36] Shi J, Tomasi C. Good Features to Track[M]. Cornell University, 1993.

[37] Harris C J. A combined corner and edge detector[J]. Proc Alvey Vision Conf, 1988, 1988(3):147-151.

[38] Lucas B D, Kanade T. An iterative image registration technique with an application to stereo vision[C] International Joint Conference on Artificial Intelligence. Morgan Kaufmann Publishers Inc. 1981:674-679.

[39] Fischler M A, Bolles R C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography[M]. ACM, 1981.

[40] Longuet-Higgins H C. A computer algorithm for reconstructing a scene from two projections[J]. Readings in Computer Vision, 1987, 293(5828):61-62.

[41] Hartley R I. In defense of the eight-point algorithm[J]. IEEE Pami, 1997, 19(6):580-593.

[42] O.D. FAUGERAS, F. LUSTMAN. MOTION AND STRUCTURE FROM MOTION IN A PIECEWISE PLANAR ENVIRONMENT[J]. International Journal of Pattern Recognition & Artificial Intelligence, 1988, 02(03):-.

[43] ZHANG,Z. 3D Reconstruction based on homography mapping[J]. Proc Arpa, 1996:0249--6399.

[44] Malis E, Vargas M. Deeper understanding of the homography decomposition for vision-based control[J]. HAL - INRIA, 2007.

[45] Michael Burri, Janosch Nikolic, Pascal Gohl, et al. The EuRoC micro aerial vehicle datasets[J]. International Journal of Robotics Research, 2016, 35(10):1157-1163.

[46] Endres F, Hess J, Engelhard N, et al. An evaluation of the RGB-D SLAM system[C] IEEE International Conference on Robotics and Automation. IEEE, 2012:1691-1696.

[47] Stefan Leutenegger, Simon Lynen, Michael Bosse, et al. Keyframe-based visual-inertial odometry using nonlinear optimization[J]. International Journal of Robotics Research, 2015, 34(3):314-334.

[48] Kasyanov A, Engelmann F, Stückler J, et al. Keyframe-Based Visual-Inertial Online SLAM with Relocalization[J]. 2017:6662-6669.

哈尔滨工业大学学位论文原创性声明和使用权限

学位论文原创性声明

本人郑重声明：此处所提交的学位论文《基于单目视觉与 IMU 结合的 SLAM 技术研究》，是本人在导师指导下，在哈尔滨工业大学攻读学位期间独立进行研究工作所取得的成果，且学位论文中除已标注引用文献的部分外不包含他人完成或已发表的研究成果。对本学位论文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明。

作者签名：李建禹 日期：2018 年 6 月 29 日

学位论文使用权限

学位论文是研究生在哈尔滨工业大学攻读学位期间完成的成果，知识产权归属哈尔滨工业大学。学位论文的使用权限如下：

（1）学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文，并向国家图书馆报送学位论文；（2）学校可以将学位论文部分或全部内容编入有关数据库进行检索和提供相应阅览服务；（3）研究生毕业后发表与此学位论文研究成果相关的学术论文和其他成果时，应征得导师同意，且第一署名单位为哈尔滨工业大学。

保密论文在保密期内遵守有关保密规定，解密后适用于此使用权限规定。

本人知悉学位论文的使用权限，并将遵守有关规定。

作者签名：李建禹 日期：2018 年 6 月 29 日

导师签名：孙怡 日期：2018 年 6 月 29 日

致谢

光阴荏苒，日月如梭，两年的硕士生涯匆匆而过，即将接近尾声。此时此刻，我的心中充满浓浓不舍。回顾这两年的美好时光，我从懵懵懂懂的本科毕业生，到一名合格的硕士毕业生，这其中不仅学术知识上的成长，更有心智上的成熟。在这期间，太多人给过我指导和帮助，帮助我成长，指导我渡过难关。

首先要感谢我的导师刘国栋教授，他教授渊博的学识、严谨的治学态度及认真负责的工作责任感深深感染着我，而且在思想、生活上激励着我不断积极进取，严格要求自己。谆谆教诲，感恩在心。在此，谨向恩师刘国栋教授致以最高的敬意和诚挚的感谢！

其次我要感谢研究所的陈凤东副教授，在两年的硕士生涯期间，他无数次的帮助指导我，帮我纠正错误。他严谨的治学态度也深深感染了我，让我认识到了做学术不能马马虎虎，对每一个细节要清楚明白。再次向陈凤东副教授致以最诚挚的谢意。同时感谢实验室的刘炳国老师，庄志涛老师，甘雨老师，他们虽未亲身指导我，但耳濡目染，亦令我获益匪浅。

非常感谢我的师兄、师姐们，感谢我的同学和朋友，是你们陪伴我度过了这美好的两年。感谢单梦圆师姐，是你让我在轻松愉快的氛围中进入了研究生生活状态；感谢我的同窗好友郭佳皓、孙佳睿、黄卓、李雪莹、程书博、魏云清、霍智翔、时圣星、郭攀、洪凯程，我会永远记得我们一起学习、生活时的美好回忆；感谢我的室友沈飞跃、刘小勇、周子坤，谢谢你们从本科到硕士六年的陪伴。

我要特别感谢我的女朋友周晓柳婷，硕士两年期间，无论是学习，工作还是生活，她总是无条件的支持和陪伴着我，陪我渡过了一个又一个的难关。谢谢你一直以来的支持和鼓励。

感谢我亲爱的家人，谢谢你们一直以来给予我无私的爱和关怀，感谢你们在我伤心失落时的安慰与鼓励，在我开心时的分享与支持，你们是我前进的动力，我将会更加努力地走下去！

最后，感谢母校哈尔滨工业大学对我的培养，我将继续秉承规格严格，功夫到家的校训，不断努力进取，继续锤炼自己、充实自己、超越自己，为社会的发展贡献自己的一份力!