

硕士学位论文

基于非线性优化的单目视觉与 IMU 融合的
SLAM 算法研究

RESEARCH ON MONOCULAR
VISUAL-INERTIAL SLAM BASED ON
NONLINEAR OPTIMIZATION

李庆峰



哈尔滨工业大学

2017 年 12 月

国内图书分类号：TP242.3 学校代码：10213

国际图书分类号：681.5 密级：公开

工学硕士学位论文

基于非线性优化的单目视觉与 IMU 融合的 SLAM 算法研究

硕 士 研 究 生：李庆峰

导 师：孟庆虎教授

申 请 学 位：工学硕士

学 科：控制科学与工程

所 在 单 位：深圳研究生院

答 辩 日 期：2017 年 12 月

授予学位单位：哈尔滨工业大学

Classified Index: TP242.3

U.D.C: 681.5

A dissertation submitted in partial fulfillment of
the requirements for the professional degree of
Master of Engineering

**RESEARCH ON MONOCULAR
VISUAL-INERTIAL SLAM BASED ON
NONLINEAR OPTIMIZATION**

Candidate:	LIQingfeng
Supervisor:	Prof. MENG Qinghu
Academic Degree Applied for:	Master of Engineering
Speciality:	Control Science and Engineering
Affiliation:	Shenzhen Graduate School
Date of Defence:	December, 2017
Degree-Conferring-Institution:	Harbin Institute of Technology

摘要

移动机器人已经成为了当前研究的焦点，尤其最近增强现实、虚拟现实与无人驾驶的火热进一步推动了移动机器人技术的发展。其中即时定位与建图（SLAM）是移动机器人技术的核心，而单一的传感器的 SLAM 算法无论是在精度还是鲁棒性上都不能满足一些应用的要求。

本文提出一种基于特征点法的实时紧耦合的单目视觉与 IMU 融合 SLAM 算法，该算法可以应用于室内外和大尺度环境。整个 SLAM 系统可以实现全自动初始化，不需要预先知道任何先验信息。该系统由三大部分组成：初始化、追踪、局部优化和闭环检测。在初始化部分本文采用单目视觉与 IMU 的松耦合来估计初始化参数：每一帧图像所对应的 IMU 相对世界坐标的变换关系 T_{WB} ，当地的重力矢量 g_w ，真实尺度 s ，每一帧图像所对应的 IMU 的速度 V_B ，加速度偏差 b_a 和陀螺仪偏差 b_g 。在追踪过程中，本文分为初始化成功前的仅视觉追踪和初始化成功后的视觉与 IMU 融合的追踪，在后者追踪的过程中本文又根据当前帧是否为关键帧而采用两种不同的目标方程。为了减少误差的累计和增强系统的鲁棒性，本文引入了局部地图。为了让局部地图在不丢失信息的同时计算量有界，本文引入了边缘化和稀疏化。为了避免引入错误信息，在边缘化中，本文采用固定求导线性点的方式求优化过程中雅可比矩阵。最后，为了减少 SLAM 过程中的累计误差，本文引入了闭环检测。由于每帧图像姿态的仰角和滚动角在 IMU 与相机的融合系统是可观的，因此，在闭环检测中本文优化的是 4 自由度的变量。

本文提出的 SLAM 系统在 EuRoC 数据集测试，初始化精度十分高，可以得到尺度 s 的误差小于 5%，重力矢量 g_w 的大小收敛在 9.8 m/s^2 左右，加速度偏差收敛在 0 左右，陀螺仪偏差收敛在 $10^{-3} \text{ }^\circ/\text{s}$ 左右。在整个 SLAM 过程中，定位精度也是十分具有竞争力，其中平均误差在 8cm 左右，最大误差也在 10 cm 以内。在实物测试中，本文的算法表现也十分出色，在小范围内，定位精度可以达到 10cm 左右。

关键词：非线性优化；传感器融合；追踪；边缘化与稀疏化；闭环检测

Abstract

Mobile robot has become the focus of current research. Recent augmented reality, virtual reality and unmanned vehicle promote development of the technology. Simultaneous localization and mapping (SLAM) is core of it. However, the SLAM of single sensor can't meet the demand of some applications either in accuracy or robustness.

The paper presents real-time monocular visual-inertial system based on features method. The system can be applied to indoor, outdoor, and large scale environments. The system don't require any prior information to initialization. it consists of three section,including initialization , tracking , local map and loop-closure. The paper adopts a loosely-coupled sensor fusion method to get initial parameters,which including transformation from body frame to world frame, gravity the real scale, velocity of IMU, accelerometer biases and gyroscope biases. The paper propose two ways of tracking.If the initialization is successful, we use visual inertial to tracking, otherwise only visual tracking is used. In visual inertial tracking, we also adopt two different cost function according to whether the map is updated. Our system constructs the local map to reduce the cumulative error and enhance robustness.We also use marginalization and sparsification to ensure that information is not lost and computationally bounded. We fixed the linearization point to compute Jacobian matrix in marginalization, which avoid misinformation. Finally, we also take advantage of loop-closure to eliminate long time cumulative error. We advocate the 4-DOF pose graph optimization in loop-closure owe to the pitch and roll are observable in system.

The system is tested in a recent public EuRoC dataset, and it get very high precision of initialization. It achieves error of less than 5% in scale factor, the gravity magnitude converge to about 9.8 m/s^2 , accelerometer biases converge to about 0 and gyroscope biases converge to about $10^{-3} \text{ } ^\circ/\text{s}$. Our algorithm also is competitive in positioning accuracy. It achieves average error less than 8 cm and the biggest error less than 10 cm. The result is the state-of-the-art in monocular visual-inertial odometry. In the physical test, our algorithm is also excellent. In the small space range, the accuracy of location is about 10 cm.

Keywords: nonlinear optimization, sensor fusion, tracking, marginalization and

sparsification, loop-closure

目录

摘要	I
ABSTRACT	II
第 1 章绪论	1
1.1 课题背景	1
1.2 课题研究的目的和意义	4
1.3 国内外研究现状与分析	5
1.3.1 国外研究现状	5
1.3.2 国内研究现状	6
1.3.3 研究现状分析	7
1.4 本文研究内容	7
第 2 章单目视觉与 IMU 初始化	10
2.1 引言	10
2.2 相机的地图初始化	10
2.2.1 寻找最佳的初始两帧图像	10
2.2.2 并行计算两个模型的分数的分数	11
2.2.3 模型选择	11
2.3 IMU 的预积分模型	12
2.4 IMU 与单目视觉融合初始化	14
2.5 本章小结	16
第 3 章视觉和 IMU 的融合追踪	18
3.1 引言	18
3.2 特征点的提取和匹配	18
3.3 视觉与 IMU 融合追踪	21
3.3.1 视觉部分的追踪	21
3.3.2 视觉与 IMU 的融合追踪	27
3.4 本章小结	29
第 4 章局部地图优化和闭环检测	30
4.1 引言	30
4.2 局部地图的建立	30
4.3 边缘化与稀疏化	32

4.4 闭环检测.....	38
4.5 本章小结.....	41
第 5 章实验结果	43
5.1 引言	43
5.2 EuRoC 数据集测试	43
5.3 实物小车中的测试	48
5.4 本章小结.....	51
结论	52
参考文献	54
哈尔滨工业大学学位论文原创性声明和使用权限	58
攻读硕士学位期间发表的论文及其他成果	59
致谢.....	60

第 1 章 绪论

1.1 课题背景

随着人工智能技术的日趋成熟，许多科幻小说和电影里的机器人出现在了人们的视野中。从几年前的扫地机器人和服务应答机器人到最近比较火热的无人驾驶，增强现实（AR）和虚拟现实（VR）都已经成为了人们生活和娱乐的一部分。而且这些人工智能产品的需求在不断的增加，如图 1-1 所示。这是我国从 2011 年到 2016 年扫地机器人的市场规模。不仅扫地机器人受到了人们的喜爱。增强现实和虚拟现实的相关产品也受到了人们的青睐。比如 Facebook 在 2015 年 6 月 13 号发布了一款名叫 Oculus Rift 的 VR 头盔，售价为 599 美元，短时间内这款头盔的销售量就达到了 10 万台。

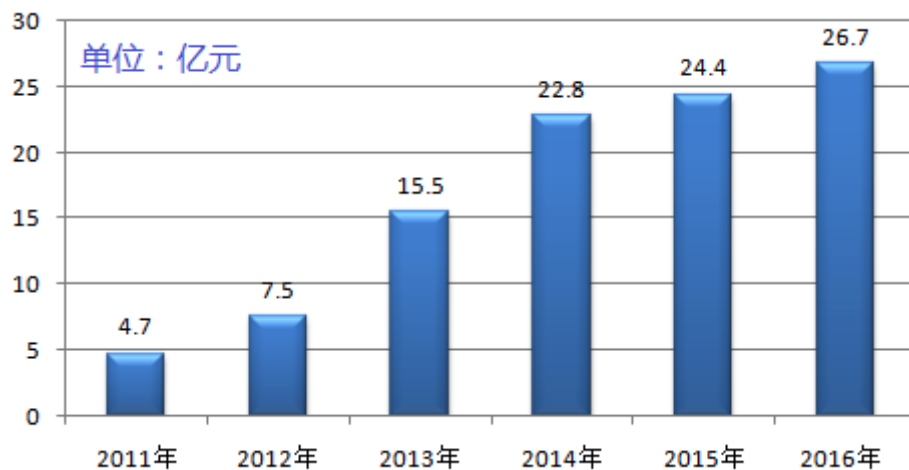
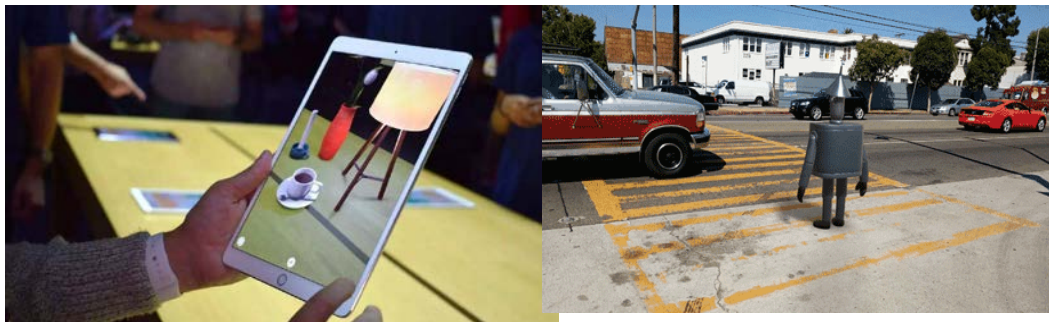


图 1-1 扫地机器人在我国的市场规模

在今年 6 月 6 号，在苹果的举行的 2017WWDC 大会上，苹果宣布了基于 IOS11 系统的增强现实组件 ARKit，这款增强现实组件不需要任何硬件外设，只需要一台带有 IOS11 的苹果手机或是平板电脑就可以操作。同年 8 月，谷歌也宣布了名为 ARCore 的增强现实组件，相比苹果的 ARKit，它不是基于苹果的 IOS 系统的，而是基于自家的 Android 系统。无论是苹果的 ARKit 还是谷歌的 ARCore 表现的效果都是令人惊叹的，如图 1-2 所示 a)图中的杯子和 b) 图的机器人原本并没有在现实场景中存在，但是却能在屏幕中显示的如此逼真。



a) ARKit b) ARCore

图 1-2 苹果和谷歌增强现实组件的效果

最近两年无人驾驶已经成为了所有的关注的焦点,早在 2012 年谷歌就开始启动了无人驾驶的项目,并且获得了内达华州的上路测试资格。2014 年谷歌宣布了名为“萤火虫”的无人驾驶汽车,如图 1-3 所示。从车身的设计到全自动的无人驾驶技术都是谷歌自己完成的,而且这是一款没有方向盘的 L5 级的无人驾驶车。自从 2015 年来,无论是传统的车厂还是互联网公司都加入了研发无人驾驶技术的行列。



图 1-3 谷歌的无人驾驶汽车

即时定位和建图 (SLAM) 不仅是移动机器人的核心,也是无人驾驶、增强现实和虚拟现实最关键的技术。SLAM 最早是由 Smith^[1-2]提出的,在他的论文中,机器人的运动路径视为定位,地图中的地图点视为路标点。Smith 将 SLAM 的运动方程和观测方程作为扩展卡尔曼滤波器 (Extended Kalman Filter),估计 SLAM 过程中机器人的位姿和路标点。在早期的 SLAM 领域中,滤波算法一直占有主导地位,例如在扩展卡尔曼滤波算法中,Lowe 使用 SIFT

特征点实现了视觉 SLAM^[3],那时的视觉 SLAM 算法主要是在 EKF^[4]基础下实现的。

在 2006 年以前由于计算机计算资源的有限,SLAM 主要是以滤波方法为主的。后来,研究者们认识到 SLAM 在优化问题中的稀疏性,并用图优化问题^[5]来表示 SLAM,如图 1-4 所示。在 SLAM 应用中,往往只保留关键帧之间的约束,故节点只有机器人的位姿,没有路标信息,这张图又称为位姿图 (Pose Graph)。相比于滤波器理论,位姿图不仅考虑当前帧和上一帧的关系,还考虑过去所有帧的信息。文献[6]表明,在相同计算量的前提下,优化算法的定位精度要高于滤波的方法,从而在总体上优于滤波器方法。

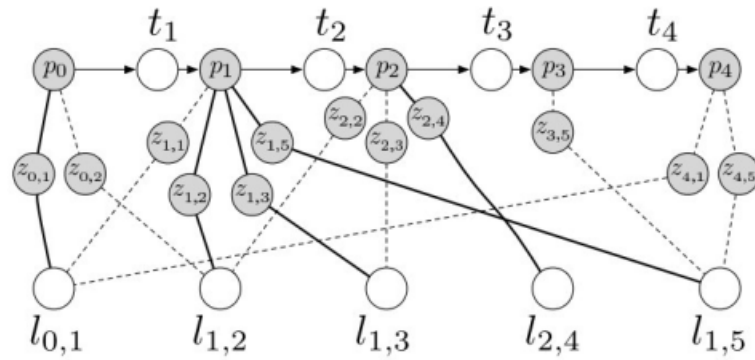


图 1-4 图优化示意图

由于单目相机体积小、价格廉价和不受视觉上距离范围的限制,可以满足多场景的 SLAM 要求。因此许多研究者开始研究单目视觉 SLAM,尤其是帝国理工的 Andrew Davison^[7]在 2007 年提出的单目视觉 SLAM,引起了单目 SLAM 在学术界的热潮。但是单目视觉由于无法估计出相邻两帧之间的位置矢量,因此,它无法满足一些需要实际尺度的应用要求例如 AR, VR 和无人驾驶的定位。因此,研究者们开始利用 IMU (Inertial Measurement Unit) 融合单目相机来解决这一问题。2007 年明尼苏达大学利用扩展卡尔曼滤波融合视觉和 IMU^[8],成为后来许多研究的基石。2013 年帝国理工大学 S Leutenegger^[9]等人发表的 OKVISLAM 算法,该算法是基于非线性优化做的,其定位精度在一些数据集上取得了最佳的结果,这引起的研究者们对基于非线性优化的视觉与 IMU 融合的 SLAM 算法的关注。因此,最近几年国外一些做 3D 视觉的实验室兴起了视觉与 IMU 融合算法研究的热潮。

虽然基于非线性优化的视觉与 IMU 融合的 SLAM 算法理论已经接近成熟,但是仍然有许多地方改进和完善,比如,目前的算法在初始化的时候缺

少对 IMU 偏差的估计,这容易导致 IMU 的误差项与视觉的误差项相差较大,最终导致系统跟踪失败。目前的算法在局部地图做的也不够完善,边缘化策略并不能适合所有的场景。在闭环检测中,大多数算法还是优化 6 自由度或是 7 自由度变量,而在整个系统中仰角和滚动角是可观的。

1.2 课题研究的目的是和意义

SLAM 在机器人和人工智能领域都是关键技术,例如扫地机器人要知道哪里已经打扫过了,哪里还没打扫,自己在哪里,又要去哪里等一系列问题,而这些问题的核心就是定位和导航。在增强现实和虚拟现实定位是核心技术,如果定位误差大会让消费者的体验极速下降。在无人车中定位也是核心技术,如果没有高精度的定位数据,这对车的控制是极其危险的。本文的目的是提高 SLAM 的精度和鲁棒性,使整个 SLAM 系统能够在多环境下满足要求。

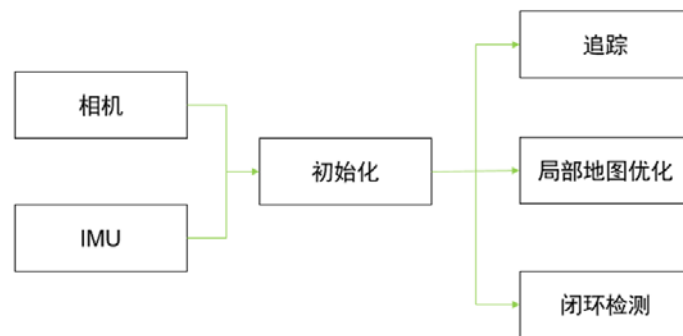


图 1-5 视觉与 IMU 融合的 SLAM 算法简单框架示意图

多传感融合的 SLAM 算法对提高整个系统的定位精度和鲁棒性有质的提高。而简单低价的相机与 IMU 传感器套件无论在理论上还是实际应用中都是多传感器融合的最佳选择。而基于非线性优化的视觉与 IMU 融合的算法是目前在精度和鲁棒性上最好的算法框架,整体系统算法框架如图 1-5 所示。单目视觉与 IMU 融合的 SLAM 算法不仅可以弥补仅单目视觉的无真实尺度的缺陷,而且还能提高跟踪的鲁棒性和快速性,就精度而言也远优于仅单目视觉。因此,该算法的研究对整个移动机器人技术有巨大的推动作用,是解决当前主流科技的关键所在。

1.3 国内外研究现状与分析

1.3.1 国外研究现状

1986-1989 Smith 系列论文《空间位置的表达与估计》提出了机器人定位和建图的问题。到目前为止 SLAM 主要有两种理论研究方法,一种是基于滤波的方法,主要包括卡尔曼滤波,扩展卡尔曼滤波,粒子滤波等。还有一种基于非线性的优化方法。

在地图的构建问题上。Cheesman^[10]和 Durrant-Whyte^[11]把路标(landmark)问题当作统计问题,利用统计学用于解决数学上的不确定性。这项工作证明了路标之间是有关联的。Ayache 和 Faugeras^[12]在早期做了视觉 SLAM 的相关工作。而 Chatila、Crowley^[13]和 Laumond^[14]早期利用声呐做过 SLAM 的工作。而后,Smith^[15]等人提出里程碑式的文章,他认为机器人的位姿和路标点是有一定关系的。随后 Matthies L^[16]提出了比较简单的传感器位姿与路标点的关系,但是它没考虑地图路标点的收敛性问题。于是在 1995 年的综述文章[1]提出了即时定位建图的概念和收敛性概念。

自从 2000 以后随着计算机计算能力的大幅度提升,单目视觉 SLAM 开始受到了研究者的关注。2007 年 Andrew Davio^[7]的单目 SLAM 是视觉 SLAM 的里程碑。随后 David Murray^[17]发表了基于关键帧的 PTAM,是现在大多数 SLAM 系统的基石,它主要的贡献在于把 SLAM 过程分为基于关键帧(keyframe)的追踪和建图两个并行线程。2014 年 J. Engel, T. Schops^[18]提出基于直接法的 LSD-SLAM,2015 年同组的人又提出了使用广角相机来做直接法^[43],这些工作的效果引起了国内外研究者们对直接法的关注,他的主要思想是基于相同空间点的光度不变性,在这个假设下利用优化最小二乘光度差来恢复相机的位姿,并且建立了半稠密地图。同年 R. Mur-Artal^[19]提出基于单目视觉的 ORBSLAM,其核心思想与 PTAM 很像,在关键帧(keyframe)的基础上把 SLAM 分为追踪,局部地图,闭环检测三个线程,不同的是 R. Mur-Artal 采用鲁棒性和精确度更高的 ORB 特征点,并且相比 PTAM 有了闭环检测,而且一致性观测地图的建立使得它可以在大尺度环境中运行。也在同年苏黎世大学机器人感知组提出了基于单目视觉的半直法 SVO^[20],主要思想是利用直接法匹配,利用特征点选取关键帧,使用滤波的方法估计地图点的深度。文献[18-20]是目前在单目 SLAM 中比较具有代表性的算法,其效果如图 1-6 所示,它们有共同的缺陷就是无法得到真实的尺度。



图 1-6 LSD,ORB 和 SVO 算法的里程示意图

为了解决单目尺度问题,2007 年明尼苏达大学提出了基于扩展卡尔曼滤波的单目视觉和 IMU 融合算法^[8]。2013 年苏黎世大学机器人与感知实验室首次提出在基于非线性优化下利用 IMU 预积分模型融合视觉与 IMU 的算法^[9],这项工作为基于非线性优化的多传感器融合的 SLAM 算法奠定了基础,但是它的初始化时 IMU 的偏差设置为 0,这增加了估计误差。2015 年苏黎世大学机器人感知实验室提出一种前端采用 SVO,后端采用非线性优化的 SLAM 算法,并在 2017 年整理该算法发表在 TRO 期刊上^[21],这个算法对整个 IMU 预积分的模型建立做了详细的推导,但是仍然忽略了初始化时 IMU 的偏差估计。2016 年西班牙的 R. Mur-Artal^[22]利用 ORB 算法在非线性优化的框架下融合了 IMU,其重点放在了初始化上,弥补了文献[21]初始化的缺陷,但是局部地图关键帧的数量不是一个恒值,这导致局部优化的计算量比较大,而且初始化需要一定的时间内完成。

1.3.2 国内研究现状

自从 2015 年以来,研究无人驾驶,增强现实和虚拟现实的实验室和公司越来越多。兴起了许多以这些技术为核心的创业公司,尤其是 VR 头盔,扫地机器人等公司。同时的一些国内实验室也开展了许多关于 SLAM 和人工智能的课题。

虽然国内研究 SLAM 方向相对国外比较晚,但是一些实验室也取得了显著的成果。浙江大学计算机辅助设计与图形学国家重点实验室(CAD&CG)章国锋老师组在 2014 年发表了基于关键帧的单目视觉追踪^[25],2016 年发表了单目视觉 SLAM 在大尺度环境中的应用^[26]。章老师组最近几年在 SLAM 的顶级会议上发表了许多论文,但是章老师主要研究仅视觉的 SLAM 问题。对多传感器的融合的 SLAM 算法缺少研究。而香港科技大学沈劭颀老师组在视觉与 IMU 融合的 SLAM 算法研究上有较大的成果。在 2015 沈老师年提出基于非线性优化的 IMU 与单目视觉的融合算法^[23],该算法创新在于在线标定了相机与 IMU 的位姿,但是在初始化的时候忽略的 IMU 的偏差,在 2017 年

沈劭颀老师组又完善该算法并取名为 VINS-Mono^[24]，在初始化过程中考虑了陀螺仪的偏差，并且在 GIT-HUB 上开源了他们的算法，效果如图 1-7 所示，但是该算法在初始化上仍然忽略了加速度偏差的影响。目前沈老师组对视觉与 IMU 融合的 SLAM 算法研究比较深入，在国际上有较大的影响力。

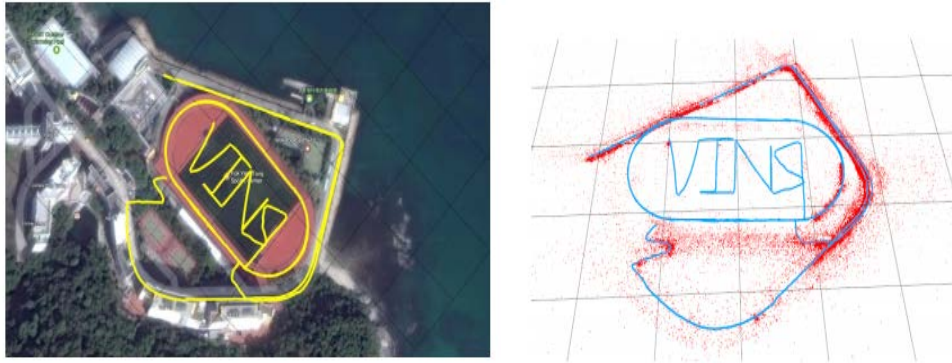


图 1-7 VINS-Mono 算法在香港科技大学运动场的实验效果

1.3.3 研究现状分析

对上述现状综合分析可以看出，近几年国内外对 SLAM 算法的研究已经取得了十分显著的成就，理论基础已经成熟。但是目前仍然有些问题解决的不够完善：

（1）在视觉与 IMU 初始化上大部分算法都缺少对 IMU 偏差的估计，这增加的对状态估计的误差，在不充分的运动中这很容易导致系统发散。

（2）在追踪过程中，目前的 SLAM 算法不够鲁棒，在纯旋转和激烈运动容易跟丢，而且追踪的地图点的数量不够稳定，这容易导致较大的误差。

（3）在局部地图的优化中，目前的算法的关键帧策略对不同的场景适应能力比较弱，有些算法直接丢掉过去帧的信息，并没有采用边缘化的思想保留丢掉的关键帧信息，部分算法虽然也采用了边缘化策略，但是对如何丢掉关键帧没有一个合适的策略。

（4）在闭环检测中，大部分算法优化的是 6 自由度或是 7 自由度变量，忽略了系统状态中仰角和滚动角的可观性。

1.4 本文研究内容

本文提出了一种基于非线性优化的视觉与 IMU 融合的 SLAM 算法，整体框图如图 1-8 所示，根据 SLAM 的理论基础和存在的问题把本文的研究分为初始化、追踪、局部地图和闭环检测三个部分。

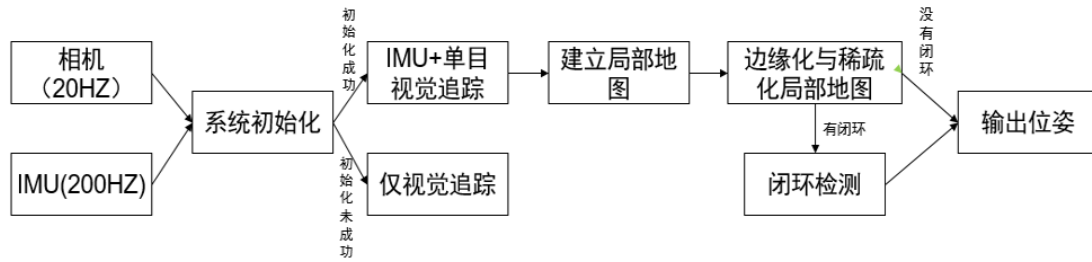


图 1-8 整个 SLAM 系统的框架图

（1）目前初始化多数需要人为的干涉，而不需要人为干涉的初始化精度较差，这不仅影响整个系统的鲁棒性，而且也严重影响后续的追踪的过程的定位估计和 IMU 的偏差估计。本文提出一种松耦合的融合方式，首先利用单目视觉找到初始两帧，然后使用单应矩阵和基础矩阵并行计算它们之间的变换关系，通过统计单应矩阵和基础矩阵的分数来决定它们之间的真实变换关系，最后通过三角化求出初始地图点，其中初始第一帧作为世界坐标原点。在单目视觉成功的得到初始两帧和初始地图点后，利用单目视觉的重投影误差计算相邻两帧之间的变换关系，同时也利用 IMU 预积分计算相邻两帧之间的变换关系。最后通过相应的融合方式求出初始化的参数：每帧图像对应的旋转矩阵、位置矢量、重力矢量、地图的尺度、IMU 的速度和偏差。

（2）本文提出了一种视觉与 IMU 的融合追踪的方法，追踪过程分为初始化前的仅视觉追踪和初始化成功后的视觉与 IMU 融合追踪两部分。在视觉部分本文并没有采用传统追踪方式：追踪上一帧的地图点，而是采用比较复杂的追踪局部地图点的方式，这使得关键帧之间的关系更加紧密。在视觉和 IMU 融合部分本文根据当前帧是否是关键帧而采用两种融合方式。除此之外，我们也在追踪的过程中选取关键帧，为了局部地图优化和闭环检测做准备。

（3）在局部地图中本文构建固定窗口（fixed window）和局部窗口（local window）两种窗口。其中固定窗口只提供视觉上的约束，不作为优化变量。而局部窗口不仅受到视觉上的约束，而且还受到 IMU 预积分的约束。为了减少在滑动窗口（sliding window）中的计算量和保证之前的信息不丢失，本文利用舍尔补（schur complement）根据当前帧的上一帧是否为关键帧来边缘化（marginalization）掉多余关键帧。为了减少里程过程的累积误差，本文还利用词袋（bag of words）引入了闭环检测。在闭环检测中本文只优化 4 个自由

度：位置矢量和偏航角。

第 2 章 单目视觉与 IMU 初始化

2.1 引言

在视觉和 IMU 融合的 SLAM 算法中，初始化过程是极其重要的，如果初始化得到的参数不收敛或是不足够精确，后面的跟踪系统就会发散，最终导致定位失败。本文在初始化过程中采用的是松耦合的方式，首先利用单目视觉找到初始两帧，然后使用单应矩阵和基础矩阵并行计算它们之间的变换关系，通过统计单应矩阵和基础矩阵的分数来决定它们之间的真实变换关系，最后通过三角化求出初始地图点，其中初始第一帧作为世界坐标原点。在单目视觉成功的得到初始两帧和初始地图点后，利用单目视觉的重投影误差计算相邻两帧之间的变换关系，同时也利用 IMU 预积分计算相邻两帧之间的变换关系。最后通过相应的融合方式求出初始化的参数：每帧图像对应的旋转矩阵、位置矢量、重力矢量、地图的尺度、IMU 的速度和偏差。

在本文的算法中不需要提前知道 IMU 的初始速度和当地的重力矢量。所有的 IMU 初始参数都可以在线估计，除此之外，本文在初始化的时候还考虑了 IMU 的加速度偏差和陀螺仪偏差，这是被很多研究者忽略的问题。本文初始化方式相比文献[27]可以得到更精确的初始化参数。

2.2 相机的地图初始化

地图初始化的目的是计算相邻两帧图像的变换关系，然后通过三角法计算初始地图点。这个过程应该不依赖场景和不要求人的干预。受文献[19]的启发，本文在地图初始化的时候设计了两种几何模型的初始化：一种是用来计算平面场景的单应矩阵几何模型，另一种的用来计算非平面场景的本质矩阵几何模型。本文采用启发式的方式来选择采用哪种模型进行初始化。无论采用哪种几何模型初始化都必须满足有足够大视差，这避免了得到糟糕的初始地图点。具体的初始化步骤可分为寻找最佳初始两帧图像、并行计算两个模型的分数和选择最佳模型三步。

2.2.1 寻找最佳的初始两帧图像

首先，提取当前帧的图像的 ORB 特征点，本文之所以采用 ORB 特征点

是因为它不仅具有 FAST 不具有的旋转不变性和尺度不变性外，计算速度比传统的 SIFT 和 SUFR 特征点快很多，满足系统的实时要求。在权衡计算时间和匹配鲁棒性的前提下本文在初始化中每幅图像提取 2000 个 ORB 特征点。除此之外，为了提高计算两帧之间变换关系的精度，本文也利用四叉树的数据结构保证每一个叶子节点下的网格至少有 5 个特征点，这保证了特征点在图像上尽可能的分布均匀。

当前帧提取 ORB 特征点之后，需要提取当前参考帧的 ORB 特征点，并且和当前帧做 ORB 匹配。本文采用快速最近邻（FLANN）的方式进行特征的匹配，当最佳匹配的欧式距离与次最佳匹配的欧式距离的比小于某一个阈值才满足匹配要求。同时统计所有特征匹配方向的直方图，找到最多的三个特征方向直方图，如果匹配的方向不满足以上三个直方图的匹配要丢弃。最后计算满足匹配要求的匹配个数和两帧图像的视差，如果满足阈值要求则作为初始两帧三角化求初始地图点，否则重新选择当前的参考帧。

2.2.2 并行计算两个模型的分數

成功选择初始两帧图像之后，并行利用单应矩阵 \mathbf{H} 和基础矩阵 \mathbf{F} 求解两帧之间的关系。

$$\begin{cases} \mathbf{x}_c = \mathbf{H}_{cr} \mathbf{x}_r, \\ \mathbf{x}_c^T \mathbf{F}_{cr} \mathbf{x}_r = 0 \end{cases} \quad (2-1)$$

其中 \mathbf{x}_c 是当前帧特征点位置集合， \mathbf{x}_r 是参考帧特征点的位置， \mathbf{H}_{cr} 和 \mathbf{F}_{cr} 分别代表单应矩阵和基础矩阵。根据文献[28]，单应矩阵可以通过归一化 DLT 算法求取，基础矩阵通过 8 点法求取，为了合理的计算单应矩阵和基础矩阵的分數，本文预先为两种模型设定好了迭代的步长。本文通过 S_{mod} 代表每种模型计算的分數：

$$\begin{aligned} S_{mod} &= L(e_{cr}^2(\mathbf{x}_c, \mathbf{x}_f, \mathbf{M})) + L(e_{rc}^2(\mathbf{x}_c, \mathbf{x}_f, \mathbf{M})), \\ L(e^2) &= \begin{cases} T_{mod} - e^2, & \text{if } e^2 < T_{mod} \\ 0, & \text{else } e^2 \geq T_{mod} \end{cases} \end{aligned} \quad (2-2)$$

e_{cr} 和 e_{rc} 是对称误差项， $L(e^2)$ 是排除外点的阈值， T_{mod} 是 χ 检测在 95% 的值。在这个过程中如果没有足够的匹配内点被发现将重新选取参考帧 \mathbf{x}_r 。

2.2.3 模型选择

如果初始化相机所看到的场景是平面，近视平面，或者当前帧 \mathbf{x}_c 和参考

帧 x_r 之间的视差比较小，本文都会选择单应矩阵求解。如果场景是非平面且视差较大本文会选择基础矩阵来恢复两帧之间的变换关系。受文献[19]的启发，本文采取了更加鲁棒的计算方式。

$$R_H = \frac{S_H}{S_F} \quad (2-3)$$

如果 R_H 小于 0.45 我们选择单应矩阵模型恢复两帧之间的变化关系，否则将选择基础矩阵模型。

模型被成功选出来之后，将对单应矩阵 H 和基础矩阵 F 求解恢复两帧之间的运动。在比较小的视差的情况下文献[29]求解方式很容易出现错误，本文提出一种更加鲁棒的求解方式。在恢复两帧之间的变换关系即：旋转矩阵 R 和位置矢量 t 中，或出现 $R(\pm\pi)$ 和 $\pm t$ 各两种解，一共是四种组合方式。本文采取分别按照四种解来求取重投影误差，如果最小的那组重投影误差比上次最小那组重投影误差的阈值小于某一范围，则满足恢复变换关系要求，否则重新选取参考帧 x_r 。在成功的获取初始两帧图像的变换关系之后，本文最后执行一次全局优化，其优化变量包括所有初始地图点和初始两帧的位姿。

2.3 IMU 的预积分模型

IMU 的模型是惯性导航里面最重要的一部分，本文采用了 IMU 的随机游走模型。在本文里 IMU 的参考坐标设置为 B ，而相机的参考坐标被设置为 C 。加速度计测量的加速度设置为 a_B 和陀螺仪的角速度测量设置为 ω_B 。其中加速度 a_B 和角速度 ω_B 的测量受到加速度偏差 b_a ，角速度偏差 b_g 和相应的噪声 η_a 和 η_g 的影响：

$$\begin{cases} \hat{\omega}_B = \omega_B + b_g + \eta_g \\ \hat{a}_B = a_B + b_a + \eta_a \end{cases} \quad (2-4)$$

这里的噪声本文视为服从高斯分布的白噪声即： $\eta_g \sim (0, \delta_g^2)$ ， $\eta_a \sim (0, \delta_a^2)$ ，在本文中设置它们为 0。为了与视觉融合求解，本文受文献[21]的启发，提出了一种新的预积分模型：

$$\begin{cases} \mathbf{R}_{WB}^{k+1} = \mathbf{R}_{WB}^k \text{Exp}\left((\mathbf{w}_B^k - \mathbf{b}_g^k) \Delta t\right) \\ {}_w\mathbf{V}_B^{k+1} = {}_w\mathbf{V}_B^k + \mathbf{g}_w \Delta t + \mathbf{R}_{WB}^k (\mathbf{a}_B^k - \mathbf{b}_a^k) \Delta t \\ {}_w\mathbf{P}_B^{k+1} = {}_w\mathbf{P}_B^k + {}_w\mathbf{V}_B^k \Delta t + \frac{1}{2} \mathbf{g}_w \Delta t^2 + \mathbf{R}_{WB}^k (\mathbf{a}_B^k - \mathbf{b}_a^k) \Delta t^2 \end{cases} \quad (2-5)$$

其中 $\mathbf{R}_{WB}^k \in \text{SO}(3)$ 是相机采集第 K 帧 IMU 相对世界坐标系的旋转关系。 $\exp(\cdot)$ 是李代数到李群的指数映射, Δt 是相邻两帧图像的时间间隔, ${}_w\mathbf{V}_B^k$ 是 IMU 在世界坐标系的速度, ${}_w\mathbf{P}_B^k$ 是 IMU 在世界坐标系下的位置, \mathbf{g}_w 是当地的重力矢量。一般 IMU 数据的发布频率要远大于相机发布图像的频率, 如图 2-1 所示。

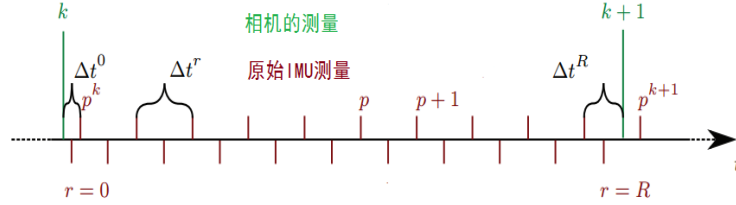


图 2-1 相机和 IMU 在时间上的关系图

因此, 本文还需要在第 k 帧图像与第 $k+1$ 帧图像之间预积分 IMU 的测量量。

$$\begin{cases} \mathbf{R}_{WB}^{i+1} = \mathbf{R}_{WB}^i \mathbf{R}_{i,i+1} \text{Exp}\left(\mathbf{J}_R^g \mathbf{b}_g^i\right) \\ {}_w\mathbf{V}_B^{i+1} = {}_w\mathbf{V}_B^i + \mathbf{g}_w t + \mathbf{R}_{WB}^i \left(\mathbf{V}_{i,i+1} + \mathbf{J}_V^g \mathbf{b}_g^i + \mathbf{J}_g^a \mathbf{b}_a^i\right) \\ {}_w\mathbf{P}_B^{i+1} = {}_w\mathbf{P}_B^i + {}_w\mathbf{V}_B^i t + \frac{1}{2} \mathbf{g}_w t^2 + \mathbf{R}_{WB}^i \left(\mathbf{P}_{i,i+1} + \mathbf{J}_V^g \mathbf{b}_g^i + \mathbf{J}_p^a \mathbf{b}_a^i\right) \end{cases} \quad (2-6)$$

其中雅可比 \mathbf{J}_R^g 和 \mathbf{J}_g^a 是陀螺仪偏差和加速度偏差的一阶近似。其中 IMU 和相机之间的变换关系为 \mathbf{T}_{CB} , 如图 2-2 所示。

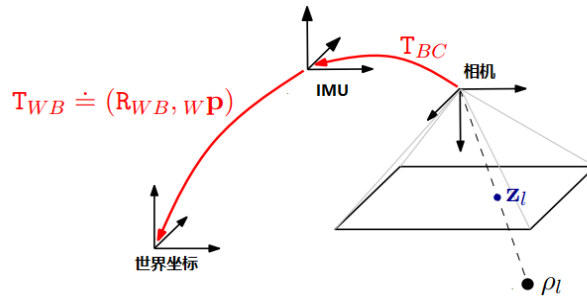


图 2-2 相机与 IMU 的坐标变换关系

2.4 IMU 与单目视觉融合初始化

本文提出一种能同时估计出地图尺度，重力矢量，IMU 速度和 IMU 偏差的初始化方法。通过单目视觉里程得到一系列时间上连续的关键帧，同时 IMU 通过预积分计算这些连续关键帧之间的变换关系，最后把它们计算的变换关系放到一个目标函数里优化。

(1) 陀螺仪的偏差估计

首先，通过单目视觉里程得到多组两个连续的关键帧之间的旋转关系，同时 IMU 通过预积分计算每组相邻关键帧之间的变换关系：

$$\min_{b_g} \sum_{i=1}^{N-1} \left\| \text{Log} \left(\left(\Delta R_{i,i+1} \text{Exp} \left(J_{\text{DR}}^g b_g^i \right) \right) R_{\text{BW}}^{i+1} R_{\text{BW}}^i \right) \right\| \quad (2-7)$$

其中 N 是关键帧的个数， $R_{\text{WB}}^{(\bullet)} = R_{\text{WC}}^{(\bullet)} R_{\text{CB}}$ 是由单目视觉重投影误差计算的旋转矩阵。然后通过高斯牛顿法优化该方程求出 b_g 。

(2) 尺度和重力矢量的近似估计

单目视觉在初始化的过程中无法估计出相邻两关键帧的位置矢量 \mathbf{t} ，因此需要融合 IMU 计算出位置矢量，本文假设单目视觉通过重投影误差估计出来的位置矢量的尺度为 s ：

$${}_w \mathbf{P}_B = s \mathbf{P}_C + \mathbf{R}_{\text{WC}} \mathbf{P}_B \quad (2-8)$$

忽略加速度偏差的情况下，把公式 (2-8) 带入 (2-6) 得：

$$s {}_w \mathbf{P}_C^{i+1} = s {}_w \mathbf{P}_C^i + {}_w \mathbf{V}_B^i \Delta t_{i,i+1} + \frac{1}{2} \mathbf{g}_w \Delta t_{i,i+1}^2 + \mathbf{R}_{\text{WB}}^i \Delta \mathbf{P}_{i,i+1} + (\mathbf{R}_{\text{WC}}^i - \mathbf{R}_{\text{WC}}^{i+1}) {}_C \mathbf{P}_B \quad (2-9)$$

本文通过求解一个有关变量 s 和 \mathbf{g}_w 的线性方程来估计尺度和重力矢量，利用时间上连续的两个关键帧可获取两个类似公式 (2-9) 的线性方程，把两个线性方程做差可得：

$$\begin{bmatrix} \lambda_{(i)} & \beta_{(i)} \end{bmatrix} = \begin{bmatrix} s \\ \mathbf{g}_w \end{bmatrix} = \gamma(i) \quad (2-10)$$

假设三个连续的关键帧编号为 i ， $i+1$ ， $i+2$ ，把相邻的两个关键帧带入公式 (2-9) 并作差得：

$$\begin{cases} \lambda_{(i)} = (\mathbf{w} \mathbf{P}_C^{i+1} - \mathbf{w} \mathbf{P}_C^i) \Delta t_{i+1,i+2} - (\mathbf{w} \mathbf{P}_C^{i+2} - \mathbf{w} \mathbf{P}_C^{i+1}) \Delta t_{i,i+1} \\ \beta_{(i)} = \frac{1}{2} \mathbf{I}_{3 \times 3} (\Delta t_{i,i+1}^2 \Delta t_{i+1,i+2} + \Delta t_{i+1,i+2}^2 \Delta t_{i,i+1}) \\ \gamma_{(i)} = (\mathbf{R}_{WC}^{i+1} - \mathbf{R}_{WC}^i)_C \mathbf{P}_B \Delta t_{i+1,i+2} - (\mathbf{R}_{WC}^{i+2} - \mathbf{R}_{WC}^{i+1})_C \mathbf{P}_B \Delta t_{i,i+1} \\ \quad + \mathbf{R}_{WB}^{i+1} \Delta \mathbf{p}_{i+1,i+2} \Delta t_{i,i+1} + \mathbf{R}_{WB}^i \Delta \mathbf{V}_{i,j} \Delta t_{i,i+1} \Delta t_{i+1,i+2} - \mathbf{R}_{WB}^i \Delta \mathbf{P}_{i,i+1} \Delta t_{i+1,i+2} \end{cases} \quad (2-11)$$

把所有的连续 3 个关键帧都带入公式 (2-11) 可以得到类似于 $\mathbf{A}_{3(N-2) \times 4} \mathbf{X}_{4 \times 1} = \mathbf{B}_{3(N-2) \times 1}$ 的线性方程, 然后通过奇异值分解 (SVD) 求解尺度 s^* 和重力方向 \mathbf{g}_w^* 。此线性方程有 $3(N-2)$ 个方程和 4 个未知数, 因此至少需要 4 个关键帧。

(3) 加速度偏差的估计以及尺度和重力矢量的优化

公式 (2-9) 在计算尺度和重力矢量时没有考虑加速度偏差的影响, 因此现在要考虑加速度偏差的情况下重新计算尺度和重力方向。由于把加速度直接加入到公式 (2-9) 会使得重力和加速度偏差很难区分开, 所以本文对重力方向做了处理, 首先求出重力的大小 G , 假设一个重力参考方向 $\hat{\mathbf{g}}_I = \{0 \ 0 \ -1\}$, 然后计算出实际的重力方向 $\hat{\mathbf{g}}_w = \mathbf{g}_w^* / \|\mathbf{g}_w^*\|$, 最后计算出重力参考方向到实际重力方向的旋转关系 \mathbf{R}_{WI} :

$$\begin{cases} \mathbf{R}_{WI} = \text{Exp}(\hat{\mathbf{v}} \mathbf{q}) \\ \hat{\mathbf{v}} = \frac{\hat{\mathbf{g}}_I \times \hat{\mathbf{g}}_w}{\|\hat{\mathbf{g}}_I \times \hat{\mathbf{g}}_w\|}, \mathbf{q} = \text{atan2}(\|\hat{\mathbf{g}}_I \times \hat{\mathbf{g}}_w\|, \hat{\mathbf{g}}_I \cdot \hat{\mathbf{g}}_w) \end{cases} \quad (2-12)$$

可得到重力矢量为:

$$\mathbf{g}_w = \mathbf{R}_{WI} \hat{\mathbf{g}}_w G \quad (2-13)$$

重力参考方向到实际重力方向的旋转关系 \mathbf{R}_{WI} 可以使用一个扰动 $\sigma\theta$ 来优化:

$$\begin{cases} \mathbf{g}_w = \mathbf{R}_{WI} \text{Exp}(\sigma\theta) \hat{\mathbf{g}}_w G \\ \sigma\theta = \begin{bmatrix} \sigma\theta_{xy}^T & 0 \end{bmatrix}, \sigma\theta_{xy} = \begin{bmatrix} \sigma\theta_x & \sigma\theta_y \end{bmatrix} \end{cases} \quad (2-14)$$

一阶近似:

$$\mathbf{g}_w \approx \mathbf{R}_{WI} \hat{\mathbf{g}}_w G - \mathbf{R}_{WI} (\hat{\mathbf{g}}_w) G \sigma\theta \quad (2-15)$$

把公式 (2-15) 代入公式 (2-9) 得:

$$\begin{aligned}
 s_w \mathbf{P}_C^{i+1} = & s_w \mathbf{P}_C^i + {}_w\mathbf{V}_B^i \Delta t_{i,i+1} - \frac{1}{2} \mathbf{R}_{WI} \left(\hat{\mathbf{g}}_I \right)_{\times} \mathbf{G} \Delta t_{i,i+1}^2 \delta \theta \\
 & + \mathbf{R}_{WB}^i \left(\Delta \mathbf{P}_{i,i+1} + \mathbf{J}_{\Delta P}^a \mathbf{b}_a \right) + \left(\mathbf{R}_{WC}^i - \mathbf{R}_{WC}^{i+1} \right)_C \mathbf{P}_B + \frac{1}{2} \mathbf{R}_{WI} \hat{\mathbf{g}}_I \mathbf{G} \Delta t_{i,i+1}^2
 \end{aligned} \quad (2-16)$$

类似公式 (2-10) 利用三个连续的关键帧我们可以得到：

$$\begin{bmatrix} \lambda_{(i)} & \phi_{(i)} & \varsigma(i) \end{bmatrix} = \begin{bmatrix} s \\ s \mathbf{q}_{xy} \\ \mathbf{b}_a \end{bmatrix} = \boldsymbol{\psi}(i) \quad (2-17)$$

其中 $\lambda_{(i)}$ 和公式 (2-11) 相同， $\phi_{(i)}$ ， $\varsigma(i)$ 和 $\boldsymbol{\psi}(i)$ 如下：

$$\begin{cases} \phi_{(i)} = \frac{1}{2} \mathbf{R}_{WI} \left(\hat{\mathbf{g}}_I \right)_{\times} \mathbf{G} \left(\Delta t_{i,i+1}^2 \Delta t_{i+1,i+2} + \Delta t_{i+1,i+2}^2 \Delta t_{i,i+1} \right)_{(:,1:2)} \\ \varsigma_{(i)} = \mathbf{R}_{WB}^{i+1} \mathbf{J}_{\Delta P_{i+1,i+2}}^a \Delta t_{i,i+1} + \mathbf{R}_{WB}^i \mathbf{J}_{\Delta V_{i+1,i+2}}^a \Delta t_{i,i+1} \Delta t_{i+1,i+2} - \mathbf{R}_{WB}^i \mathbf{J}_{\Delta P_{i,i+1}}^a \Delta t_{i+1,i+2} \\ \boldsymbol{\psi}_{(i)} = \left(\mathbf{R}_{WC}^{i+1} - \mathbf{R}_{WC}^i \right)_C \mathbf{P}_B \Delta t_{i+1,i+2} - \left(\mathbf{R}_{WC}^{i+2} - \mathbf{R}_{WC}^{i+1} \right)_C \mathbf{P}_B \Delta t_{i,i+1} \\ \quad + \mathbf{R}_{WB}^{i+1} \Delta \mathbf{P}_{i+1,i+2} \Delta t_{i,i+1} + \mathbf{R}_{WB}^i \Delta \mathbf{V}_{i+1,i+2} \Delta t_{i,i+1} \Delta t_{i+1,i+2} \\ \quad - \mathbf{R}_{WB}^i \Delta \mathbf{P}_{i,i+1} \Delta t_{i+1,i+2} + \frac{1}{2} \mathbf{R}_{WI} \hat{\mathbf{g}}_I \mathbf{G} \Delta t_{i,i+1}^2 \end{cases} \quad (2-18)$$

其中 $(:,1:2)$ 表示矩阵的头两列，根据 (2-18) 可以得到线性方程 $\mathbf{A}_{3(N-2) \times 6} \mathbf{X}_{6 \times 1} = \mathbf{B}_{3(N-2) \times 1}$ ，通过奇异值分解 (SVD) 可以求解尺度 s^* ，重力方向的矫正量 $\sigma \theta_{xy}^*$ 和加速度偏差 \mathbf{b}_a^* 。公式 (2-18) 中有 $3(N-2)$ 个方程和 6 个未知数，因此至少需要 4 个关键帧。

(4) 速度的估计

根据公式 (2-18) 可以求出尺度，重力矢量和加速度偏差。把这些参数带入到公式 (2-16) 可求出所有关键帧的速度。而且通过公式 (2-6) 可以求得最近关键帧的速度。

2.5 本章小结

初始化是 SLAM 中至关重要的一步，本文使用单应矩阵和基础矩阵两种模型挑选初始两帧，并计算两种模型的分值，通过分值的大小来决定初始两帧的变换关系使用哪种模型，然后通过三角化计算初始地图点。在单目视觉初始化成功后，松耦合单目视觉和 IMU 计算 IMU 的初始参数。首先通过单目视觉与 IMU 的旋转关系估计陀螺仪的偏差，然后通过单目视觉与 IMU 的

位置矢量关系估计当地的重力矢量和尺度因子，最后估算加速度计的偏差和 IMU 的速度。

第 3 章 视觉和 IMU 的融合追踪

3.1 引言

当初始化成功后，整个系统进入三个并行线程：追踪、局部地图优化和闭环检测。其中追踪是整个系统的主线，这个过程也叫里程。其核心就是估计相邻两帧图像的变换关系。整个追踪过程都是以相机的频率为追踪的基准。因此图像上的处理更加重要，这部分主要分为两大部分，一部分是视觉追踪前的特征点的提取和匹配。另一部分就是初始化前的仅视觉追踪和初始化后的视觉与 IMU 融合的追踪。

在本文中，为了提高追踪的鲁棒性和精度，在特征点的提取和匹配上做了很多工作。比如采用具有尺度不变性和旋转不变性的 ORB 特征点；提取特定的数量的特征点并且采用 4 叉树的方法使所有特征点在整张图片上分布均匀；采用快速最近邻（FLANN）的方法加速匹配；采用随机采样一致性（RANSAC）剔除误匹配等。本文的追踪分为初始化前的仅视觉追踪和初始化成功后的视觉与 IMU 融合的追踪两部分。在视觉部分本文没有采用传统追踪方式：追踪上一帧的地图点。而是采用比较复杂的追踪局部地图点的方式，这使得关键帧之间的关系更加紧密构成共视图，这保证了该算法可以在大尺度下运行。同时也能有效的减少了追踪过程中的累积误差和在相机运动较激烈的情况下也可以保证追踪的鲁棒性。在视觉和 IMU 融合部分根据当前帧是否是关键帧而采用两种融合方式。除此之外，我们也在追踪的过程中选取关键帧，为了局部地图优化和闭环检测做准备。

3.2 特征点的提取和匹配

对于视觉里程，目前主要有两种方式，一种是特征点法，一种是直接法。本文采用的是特征点法，选取图像中具有代表性的像素点而不是整张图像的像素点，也不像文献[18-20]采用亮度梯度大的像素作为跟踪点。特征点法相对直接法具有匹配的基线大，定位精度高的优点。而直接法还需要对地图点进行多次的融合滤波来估计其深度。

具有代表的像素可以是角点、边缘和区块如图 3-1 所示。经过实验发现两张图像同一个角点比较容易，而边缘则稍微困难一，发现同一区块则是最困难的。所以一直以来都是以角点作为图像的特征。但是在大多数应用中单

纯的角点不能满足要求。比如，从远处看是角点，但是当相机走进的时候，可能就不是角点了。或当相机旋转的时候，角点的外观也会发生变化，也不容易辨认出是一个角点。因此在特征点的选取上，要有尺度不变性，旋转不变性等特点。经过长期的研究，研究者们设计出了许多稳定的局部图像特征。比如，SIFT^[30]，SURF^[31]，ORB^[32]。历史上，研究者们提出过许多图像特征，他们大部分都很精确，在相机运动和光照的情况下仍具有相似的表达，但都需要大量的计算，比如 SIFT 和 SUFT。视觉里程是一个实时的过程，所以计算量是一个重要的指标。因此本文选择 ORB 特征点作为视觉里程的特征点。

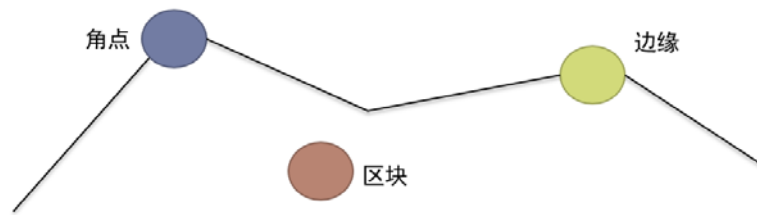


图 3-1 可以作为图像特征点的部分

ORB (Oriented Fast and Rotated BRIEF) 具有尺度不变性和旋转不变性，而且速度快。根据文献[32]的测试在提取 1000 个特征点的情况下，ORB 花费 15.3ms，SUFR 花费 217.3ms，SIFT 花费 5228.7ms。

ORB 特征由关键点和描述子两部分组成，即：FAST 角点和 BRIEF 描述子。FAST 是一种角点，主要检测局部图像灰度变化明显的地方。它检测的过程如图 3-2 所示。首先在图像中选取像素 p ，假设它的亮度为 I_p ；然后设定一个阈值 T ；以像素 P 为中心，选取半径为 3 的圆上的 16 个像素点；假如选取的圆上，有连续的 N 点的亮度大于 $I_p + T$ 或小于 $I_p - T$ ，那么像素 P 可以被认为是特征点。在本文中，为了高效对每个像素直接检测邻域圆上第 1,5,9,13 个像素的亮度。如果这 4 个像素中的 3 个像素同时大于 $I_p + T$ 或是小于 $I_p - T$ 时，则该像素被认为是特征点。为了避免 FAST 角点过于集中在图像某一个区域，本文使用非极大值抑制的方法，即：在一定区域内只保留响应极大值的角点。由于 FAST 角点数量很大接具有不确定性。由于 FAST 角点不具有尺度和旋转不变性，所以 ORB 通过构建金字塔来解决尺度不变形问题，通过灰度质心法来解决旋转不变性问题。

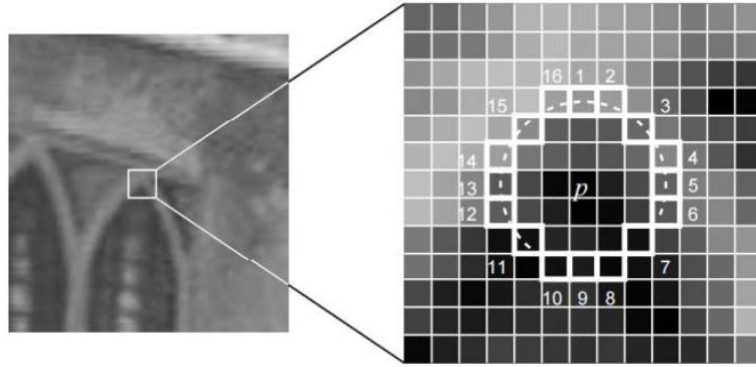


图 3-2 FAST 特征点

所谓的灰度质心法是以图像块的灰度值作为权重的中心。在一个小的图像块 B 中，定义图像块的矩位：

$$m_{pq} = \sum_{x,y \in B} x^p y^q I(x,y) \mid p,q = \{0,1\} \quad (3-1)$$

通过矩能够找到图像块质心：

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \quad (3-2)$$

通过连接图像块的几何中心 O 和质心 C ，得到一个向量 \overline{OC} ，于是特征点的方向可定义为：

$$\theta = \arctan\left(\frac{m_{01}}{m_{10}}\right) \quad (3-3)$$

通过以上方法 **ORB** 特征点就具有了尺度不变性和旋转不变性。本文为了在恢复 3D 地图点的时候尽量保证尺度的多样性，因此采用四叉树的方法来保证图像中的没一个网格都至少有 5 个 **ORB** 特征点。**ORB** 特征点被提取之后，需要对 **ORB** 特征点进行匹配，本文采用的是 **BRIEF** 描述子。**BRIEF** 是一种许多个 0 和 1 组成二进制描述子。这里的 0 和 1 代表了特征点附近两个像素值的大小关系：比如 P 比 Q 大，则取 1，否则取 0。本文取了 128 个这样的像素对，因此会有 128 维 0 和 1 组成的向量。其中 P 和 Q 的位置是按照概率随机选取的。按照汉明距离最小来找两帧的匹配点，但是由于 **BRIEF** 这种描述子的特性会存在大量的误匹配如图 3-3 所示，这里本文采用随机采样一致性（**RANSAC**）来剔除误匹配。



图 3-3 ORB 特征点的匹配

由图 3-3 可以看出未采用随机采样一致性方法之前，有许多误匹配。随机采样一致性的目的就是剔除这些误匹配，。经过随机采样一致性后的匹配如图 3-4 所示。



图 3-4 随机采样一致性后的匹配图

获取正确的匹配点对之后，就可以做上一章的基本矩阵或单应矩阵的初始化工作了，同时也可以构建本章的重投影误差项，用来追踪地图点和优化位姿。

3.3 视觉与 IMU 融合追踪

在运动模型的追踪过程，本文分为初始化成功前和初始化成功后。在初始化成功之前本文只采用仅视觉追踪。在初始化成功之后，采用视觉和 IMU 融合的追踪。

3.3.1 视觉部分的追踪

初始化成功前，本文利用恒速模型来获取当前帧的初始位姿。求出当前帧的初始位姿之后，本文利用重投影误差的方式来寻找匹配点

$$\begin{bmatrix} u_{\text{proj}} \\ v_{\text{proj}} \end{bmatrix} = \begin{bmatrix} f_u \frac{x_c}{z_c} + c_u \\ f_v \frac{y_c}{z_c} + c_v \end{bmatrix} \quad (3-4)$$

其中 x_c , y_c , z_c 是地图点在当前相机坐标系下的坐标。 c_u , c_v 是光心投影到图片上点的像素坐标, f_u , f_v 是以像素为单位的焦距。 u_{proj} , v_{proj} 是在当前相机坐标系下的地图点投影到 2D 平面的坐标点。

把 (3-4) 投影得到的 u_{proj} , v_{proj} 作为在当前帧搜索的中心点。搜索的窗口大小依赖当前帧的金字塔尺度的大小。所以上一帧中的每一个特征点都会在当前帧所对应的一个搜索区域中寻找匹配点。通过 BRIEF 描述子来计算最佳匹配点, 注意并不是上一帧的每一个特征点在当前帧都能找到匹配点, 本文会判断该最佳匹配点的汉明距离是否小于阈值的要求。如果大于阈值, 则丢弃。统计所有满足阈值要求的匹配点对的旋转方向, 建立以度数为单位的直方图, 找到最大的三个直方图作为标准。然后不在这三个直方图上的匹配点将被剔除。满足这三个直方图的所有匹配对应的地图点被视为跟踪成功。跟踪成功的地图点的个数将被统计, 如果个数大于 20 被认为跟踪成功, 否则扩大匹配中最小汉明距离的阈值, 重新计算跟踪成功的地图点的个数, 如果跟踪成功的个数仍小于 20, 则跟踪失败。

如果跟踪成功需要优化当前相机的位姿。其实我们优化的是一个观测方程:

$$z_{k,j} = h(y_j, x_k) + v_{k,j} \quad (3-5)$$

其中 y_j 代表地图点, x_k 代表当前帧的位姿。 $v_{k,j}$ 代表白噪声。由于我们假设了噪声项 $v_{k,j} \sim N(0, Q_{k,j})$, 所以观测数据的条件概率为:

$$P(z_{k,j} / x_k, y_j) = N(h(y_j, x_k), Q_{k,j}) \quad (3-6)$$

公式 (3-6) 依然是一个高斯分布, 为了计算使它最大化的 x_k, y_j , 本文使用最小化负对数的方法来求一个高斯分布的最大似然。考虑任意一个高维的高斯分布 $x \sim N(\mu, \Sigma)$, 它的概率密度函数的展开形式为:

$$P(x) = \frac{1}{\sqrt{(2\pi)^N \det(\Sigma)}} \exp\left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu)\right) \quad (3-7)$$

取 (3-7) 的负对数, 则变为:

$$-\ln(P(x)) = \frac{1}{2} \ln((2\pi)^N \det(\Sigma)) + \frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \quad (3-8)$$

对 (3-7) 的最大化相当于对 (3-8) 的最小化。最小化 (3-8) 时第一项与 x 无关，可以忽略。于是最大似然方程变成了：

$$x^* = \min \left(\left(z_{k,j} - h(y_j, x_k) \right)^T Q_{k,j}^{-1} \left(z_{k,j} - h(y_j, x_k) \right) \right) \quad (3-9)$$

由于视觉观测方程是重投影误差项如图 3-5 所示。因此，本文定义需要优化的误差为：

$$E_{\text{proj}}(j) = \rho \left(\left(x - \pi(X_c) \right)^T Q_{k,j}^{-1} \left(x - \pi(X_c) \right) \right) \quad (3-10)$$

其中向量 x 代表图像坐标 u, v 。 X_c 代表在相机坐标系下的地图点的坐标。 $\pi(\cdot)$ 代表相机坐标系下地图点的坐标 X_c 到图像坐标的映射关系， j 代表跟踪成功的第 j 个地图点。因此，在视觉跟踪成功之后，我们要优化的方程为：

$$f(x) = \min_x \sum_{j \in C} E_{\text{proj}}(j) \quad (3-11)$$

其中 $j \in C$ 代表当前帧跟踪成功的所有地图点。

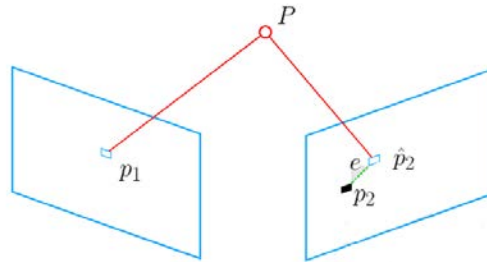


图 3-5 重投影误差示意图

式 (3-11) 是一个非线性函数，自变量 x 是当前帧的位姿，它是一个 6 自由度的未知数。如果 $f(x)$ 是一个数学形式上简单的函数，那就可以用解析的形式来求。令目标函数的导数为 0，然后求解 x 的最优解即：

$$\frac{df(x)}{dx} = 0 \quad (3-12)$$

但实际上 $f(x)$ 是一个非常复杂的非线性方程，因此本文采取迭代的方式，即从一个初始值出发，不断的迭代更新当前的优化变量，使整个目标函数下降。

对于优化方法，本文并没有采取最速下降法和牛顿法，因为最速下降法过于贪心，容易走锯齿路线，反而增加迭代的次数，而牛顿法的 H 矩阵在 SLAM 中规模较大，因此计算量很大。本文也尝试使用高斯牛顿法，但是发

现在实际的数据中 $\mathbf{J}^T \mathbf{J}$ 是半正定的，因此 $\mathbf{J}^T \mathbf{J}$ 是一个奇异矩阵，此时求出的变量的增量稳定性较差，导致算法不收敛。因此本文采用列文伯格-马夸尔特方法。

列文伯格-马夸尔特方法实际上是一种信赖域方法，在信赖域里面我们认为近似是有效的。除了这个信赖域近似可能无效。信赖域的确定是靠近似的差异大小来决定的。如果差异小，我们就让范围尽可能的大；反之，则要缩小范围：

$$\rho = \frac{f(\mathbf{x} + \Delta \mathbf{x}) - f(\mathbf{x})}{f(\mathbf{x}) \Delta \mathbf{x}} \quad (3-13)$$

其中 ρ 的分母是近似模型的下降值， ρ 的分子是实际函数下降的值。如果 ρ 接近 1 说明近似的效果很好，如果 ρ 太小则说明近似减小的值远小于实际减小的值，则认为近似效果比较差，需要缩小近视范围。反之，如果 ρ 比较大。则说明实际下降的比近视的小，我们可以放大近似范围。

由恒速模型可以得到初始的位姿：

$$\begin{cases} \mathbf{R}_{WB}^{i+1} = \mathbf{R}_{WB}^i \Delta \mathbf{R}_{i-1,i} \\ {}_w \mathbf{P}_B^{i+1} = {}_w \mathbf{P}_B^i + \Delta \mathbf{P}_{i-1,i} \end{cases} \quad (3-14)$$

其中 \mathbf{R}_{WB}^{i+1} 和 ${}_w \mathbf{P}_B^{i+1}$ 分别代表当前帧的初始姿态和位置矢量。 $\Delta \mathbf{R}_{i-1,i}$ 和 $\Delta \mathbf{P}_{i-1,i}$ 是上上帧到上一帧之间的旋转矩阵和位置矢量。假设初始优化的半径为 μ 。则对于第 K 姿迭代，求解：

$$\min_{\Delta \mathbf{x}} = \frac{1}{2} \|f(\mathbf{x}_k) + \mathbf{J}(\mathbf{x}_k) \Delta \mathbf{x}_k\|^2, s.t. \|D \Delta \mathbf{x}_k\|^2 \leq \mu \quad (3-15)$$

其中 \mathbf{D} 是单位矩阵，相当于把 $\Delta \mathbf{x}$ 约束在一个半径为 μ 的球里。然后再计算 ρ ，如果 $\rho > 3/4$ ，则 $\mu = 2\mu$ ，若 $\rho < 1/4$ ，则 $\mu = 0.5\mu$ 。 $\mathbf{J}(\mathbf{x}_k)$ 是目标方程对变量的导数。它在整个 SLAM 中有着重要的意义。(3-15) 误差是像素误差，对于每一个地图点的投影来讲该误差是二维的，而优化变量 \mathbf{x}_k 是 6 维的。所以 $\mathbf{J}(\mathbf{x}_k)$ 是一个 2×6 的矩阵。这里我们引入了李群李代数，因为我们的优化的变量是一个 4×4 的矩阵，是一个 $SE(3)$ 群，它没有良好的定义加法，而且如果把它当成一个普通矩阵来处理的话，还需要对它进行约束。但是从李代数的角度来看， $se(3)$ 是一个向量，有良好的加法运算。因此，本文采用李代数的扰动模型，即投影误差对李代数扰动的导数：

$$\mathbf{J}(\mathbf{x}) = \frac{\partial f(\xi)}{\partial \xi} = \lim_{\delta \xi \rightarrow 0} \frac{f(\delta \xi \oplus \xi) - f(\xi)}{\delta \xi} = \frac{\partial f(\xi)}{\partial \mathbf{P}} \frac{\partial \mathbf{P}}{\partial \xi} \quad (3-16)$$

其中 ξ 是优化变量 \mathbf{x} 的李代数。指的是李代数上的左乘扰动。 $\partial f(\xi)/\partial \mathbf{P}$ 是误差项关于地图点的导数：

$$\frac{\partial f(\xi)}{\partial \mathbf{P}} = - \begin{bmatrix} \frac{\partial u}{\partial X} & \frac{\partial u}{\partial Y} & \frac{\partial u}{\partial Z} \\ \frac{\partial v}{\partial X} & \frac{\partial v}{\partial Y} & \frac{\partial v}{\partial Z} \end{bmatrix} = - \begin{bmatrix} \frac{f_x}{Z} & 0 & -\frac{f_x X}{Z^2} \\ 0 & \frac{f_y}{Z} & -\frac{f_y Y}{Z^2} \end{bmatrix} \quad (3-17)$$

而第二项为变换后的地图点关于李代数的导数：

$$\frac{\partial(\mathbf{TP})}{\partial \delta \xi} = \begin{bmatrix} \mathbf{I} & -\mathbf{P}^{\hat{v}} \\ \mathbf{0}^T & \mathbf{0}^T \end{bmatrix} \quad (3-18)$$

本文取出前三维，于是得：

$$\frac{\partial(\mathbf{TP})}{\partial \delta \xi} = \begin{bmatrix} \mathbf{I} & -\mathbf{P}^{\wedge} \end{bmatrix} \quad (3-19)$$

将 (3-17) 与 (3-19) 相乘，就得到了 2×6 的雅克比矩阵：

$$\mathbf{J}(\mathbf{x}) = - \begin{bmatrix} \frac{f_x}{Z} & 0 & -\frac{f_x X}{Z^2} & -\frac{f_x XY}{Z^2} & f_x + \frac{f_x X^2}{Z^2} & -\frac{f_x Y}{Z} \\ 0 & \frac{f_y}{Z} & -\frac{f_y X}{Z^2} & -f_y - \frac{f_y Y^2}{Z^2} & \frac{f_y XY}{Z^2} & \frac{f_y X}{Z} \end{bmatrix} \quad (3-20)$$

在列文伯格-马夸尔特优化中，需要解 (3-15) 来获得梯度。而 (3-15) 是一个带不等式约束的优化问题，本文用拉格朗日乘子将它转化为一个无约束的问题。

$$\min_{\Delta \mathbf{x}} \frac{1}{2} \|f(\mathbf{x}_k) + \mathbf{J}(\mathbf{x}_k) \Delta \mathbf{x}_k\|^2 + \frac{\lambda}{2} \|D \Delta \mathbf{x}\|^2 \quad (3-21)$$

这里的 λ 是拉格朗日乘子，类似于高斯牛顿的做法，把它展开后，发现其核心与高斯牛顿一样计算增量的线性方程：

$$(\mathbf{H} + \lambda \mathbf{D}^T \mathbf{D}) \Delta \mathbf{x} = \mathbf{g} \quad (3-22)$$

可以看到，当参数 λ 较小时 \mathbf{H} 占主导地位，说明二次近视模型在该范围内是比较好的，列文伯格-马夸尔特法更近似于高斯牛顿法。而当 λ 较小时， $\lambda \mathbf{D}^T \mathbf{D}$ 占主导地位，此时，列文伯格-马夸尔特法更接近一阶梯度下降法，说明附近的二次近似不够好，列文伯格-马夸尔特法可以在一定程度上的避免线性方程组的系数矩阵的非奇异的问题，提供相对准确的增量 $\Delta \mathbf{x}$ 。

按照列文伯格-马夸尔特优化方法可以找到当前帧的位姿的最优解。在本文中设当前帧追踪上一帧得到的优化姿态和位置矢量为： $\mathbf{R}_{\text{clast}}$ 和 $\mathbf{P}_{\text{clast}}$ 。本文在优化位姿后，会对重投影的误差项进行判别：

$$\begin{cases} 0, & \text{如果 } chi > th \\ 1, & \text{否则} \end{cases} \quad (3-23)$$

其中 chi 是重投影误差值， th 是设定的阈值，如果重投影的误差值大于阈值，则剔除该重投影匹配对，同时跟踪成功的地图点个数也相应的减 1。判别所有误差项之后，利用所有小于阈值的误差项对当前相机的位姿再做一次优化。最后利用最终优化的位姿剔除当前帧地图点的外点。

通过追踪上一帧，并优化其重投影误差项可以估计出当前帧的位姿和地图中被跟踪成功的地图点。但是如果仅仅依靠追踪上一帧地图点来优化当前帧的位姿，不仅会导致位姿估计精度不够好，而且还会使整个系统鲁棒性下降。因此，我们在追踪上一帧成功之后，还会让当前帧追踪局部地图中的地图点，即凡是局部地图中能被当前帧观测到的地图点，就会被重投影到当前帧的图像坐标，形成重投影误差项。然后利用这些重投影误差项优化位姿。当前帧成功的追踪上一帧的之后，首先要做的就是更新局部地图中的关键帧，即统计当前帧的地图点能被哪些其他关键观测到，然后把这些关键帧放到局部地图中。本文为了保证局部地图中的关键帧数量足够，采取的以下策略：

(1) 找到和当前局部地图中每一个关键帧相邻的一些关键帧，然后把它们放到局部地图中。

(2) 找到当前局部地图中每一个关键帧的“子帧”，然后把它们放到局部地图中。

(3) 找到当前局部地图中每一个关键帧的“父帧”，然后把它们放到局部地图中。

(4) 找到当前局部地图中每一个关键帧的“之前帧”，然后把它们放到局部地图中。

(5) 找到当前局部地图中每一个关键帧的“下一帧”，然后把它们放到局部地图中。

更新局部关键帧之后，更新局部地图点。遍历局部地图所有关键帧的地图点，对每一个地图点进行判别：

(1) 在相机坐标系下该地图点的深度是否大于 0，如果小于 0 则舍弃。

(2) 在尺度一致性区域内，该地图点和相机的距离是否在尺度的变化

范围内。

(3) 计算该地图点的视角和所有地图点的平均视角的夹角是否小于 60 度。若大于，则丢弃。

(4) 预测该地图点对应当前帧图像的特征点所在的金字塔的层数。

通过以上 4 步可以得到满足要求的局部地图点。下面就要统计哪些地图点能被当前帧观测到，然后把当前帧观测到的地图点重投影到当前帧形成重投影误差项。最后视觉部分的优化目标方程为：

$$f(\mathbf{x}) = \min_{\mathbf{x}} \sum_{j \in L} E_{\text{proj}}(\mathbf{j}) \quad (3-24)$$

其中 $E_{\text{proj}}(\mathbf{j})$ 是重投影误差项，即式 (3-10)。(3-19) 与 (3-10) 类似，只是多了个 k 。其中 k 代表局部地图中的第 k 帧。优化过程和追踪上一帧的一样，采用列文伯格-马夸尔特的方法优化。

当前帧的位姿被优化后，需要判断当前帧是否是关键帧，如果当前帧是关键帧就必需满足以下要求：

- (1) 自从上一帧插入局部地图，已经有连续 20 帧是非关键帧了。
- (2) 当前帧的地图点与参考帧的地图点的重合率小于 90%
- (3) 当前帧追踪成功的地图点个数至少 50 个。

条件 (1) 可以保证当追踪失败的时候，系统可以得到一个比较好的重定位效果。而条件 (3) 可以保证比较好的追踪效果。

3.3.2 视觉与 IMU 的融合追踪

在初始化成功之后，本文采用单目视觉和 IMU 融合的方式追踪。在整个追踪过程将以接受相机图像的频率为基准，每接受到新的图片将更新 IMU 的位姿，速度和偏差。由于相邻两帧图像的变换关系可以由 IMU 预积分计算从而代替了运动模型，这可以更加精确的确定了优化的初始位置，有效的较少了优化的迭代次数。一旦估计出 IMU 的位姿，就可以通过 IMU 和相机的固定的变换关系 T_{CB} 得到相机的位姿，然后把局部地图中的地图点投影到当前帧并且进行匹配，最后通过优化局部地图点的重投影误差和 IMU 误差项来求解当前帧的位姿。优化的目标函数依赖与局部地图是否更新，如图 3-6 所示。

(1) 地图更新

每当有新的关键帧插入局部地图时，则局部地图更新。假设当前关键帧为 j 和上一关键帧为 i ，则优化的变量和优化方程为：

$$\min_{\theta} \left(\sum_{j \in L} E_{\text{proj}}(j) + E_{\text{IMU}}(i, j) \right) \quad (3-25)$$

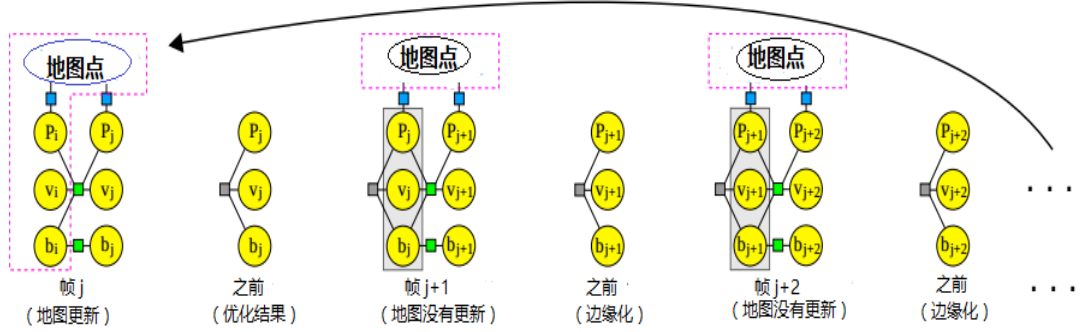


图 3-6 根据地图更新与否确定相应的优化方程

其中 $\theta = \{ \mathbf{R}_{WB}^j, {}_W\mathbf{P}_B^j, {}_W\mathbf{V}_B^j, \mathbf{b}_g^j, \mathbf{b}_a^j \}$ 是优化变量，比仅视觉的追踪的优化变量多了 IMU 的速度 ${}_W\mathbf{V}_B^j$ ，陀螺仪偏差 \mathbf{b}_g^j 和加速度偏差 \mathbf{b}_a^j 。 $\sum_{j \in L} E_{\text{proj}}(j)$ 是当前帧所对应的局部地图点的重投影误差：

$$\begin{cases} E_{\text{proj}}(j) = \rho \left((\mathbf{x} - \mathbf{p}(\mathbf{X}_C))^T \Sigma_X (\mathbf{x} - \mathbf{p}(\mathbf{X}_C)) \right) \\ \mathbf{X}_C = \mathbf{R}_{CB} \mathbf{R}_{BW}^j (\mathbf{X}_W - {}_W\mathbf{P}_B^j) + {}_C\mathbf{P}_B \end{cases} \quad (3-26)$$

其中 \mathbf{x} 是和局部地图点重投影到当前帧图像上点所匹配的特征点。 \mathbf{X}_W 是在相机坐标下的地图点。 Σ_X 是跟局部地图点尺度相关的信息矩阵。 $E_{\text{IMU}}(i, j)$ 是 IMU 误差项：

$$\begin{cases} E_{\text{IMU}}(i, j) = \rho \left(\begin{bmatrix} \mathbf{e}_R^T & \mathbf{e}_v^T & \mathbf{e}_p^T \end{bmatrix} \Sigma_I \begin{bmatrix} \mathbf{e}_R^T & \mathbf{e}_v^T & \mathbf{e}_p^T \end{bmatrix}^T \right) + \rho(\mathbf{e}_b^T \Sigma_R \mathbf{e}_b) \\ \mathbf{e}_R = \log \left(\left({}_{\Delta}\mathbf{R}_{ij} \text{Exp} \left(\mathbf{J}_{\Delta R}^g \mathbf{b}_g^j \right) \right)^T \mathbf{R}_{BW}^i \mathbf{R}_{WB}^j \right) \\ \mathbf{e}_v = \mathbf{R}_{BW}^i \left({}_W\mathbf{V}_B^j - {}_W\mathbf{V}_B^i - \mathbf{g}_w \mathbf{V} t_{ij} \right) - \left({}_{\Delta}\mathbf{V}_{ij} + \mathbf{J}_{\Delta V}^g \mathbf{b}_g^i + \mathbf{J}_{\Delta V}^a \mathbf{b}_a^j \right) \\ \mathbf{e}_p = \mathbf{R}_{BW}^i \left({}_W\mathbf{P}_B^j - {}_W\mathbf{P}_B^i - {}_W\mathbf{V}_B^i \Delta t_{ij} - \frac{1}{2} \mathbf{g}_w \Delta t_{ij}^2 \right) - \left(\Delta \mathbf{p}_{ij} + \mathbf{J}_{\Delta P}^g \mathbf{b}_g^i + \mathbf{J}_{\Delta P}^a \mathbf{b}_a^j \right) \\ \mathbf{e}_b = \mathbf{b}^j - \mathbf{b}^i \end{cases} \quad (3-27)$$

Σ_I 是 IMU 预积分的信息矩阵， Σ_R 是偏差的信息矩阵。

(2) 地图没有更新

如果没有新的关键帧插入，则地图没有更新。那么下一帧 $j+1$ 被优化的

时候将与当前帧 j 有联系：

$$\min_{\theta} \left(\sum_{j \in L} E_{\text{prior}}(j) + \sum_{j \in L} E_{\text{proj}}(j+1) + E_{\text{IMU}}(j, j+1) + E_{\text{prior}}(j) \right) \quad (3-28)$$

其中 $E_{\text{prior}}(j)$ 为：

$$\begin{cases} E_{\text{IMU}}(i, j) = \rho \left(\begin{bmatrix} \mathbf{e}_R^T & \mathbf{e}_v^T & \mathbf{e}_p^T & \mathbf{e}_b^T \end{bmatrix} \Sigma_p \begin{bmatrix} \mathbf{e}_R^T & \mathbf{e}_v^T & \mathbf{e}_p^T & \mathbf{e}_b^T \end{bmatrix}^T \right) + \rho(\mathbf{e}_b^T \Sigma_R \mathbf{e}_b) \\ \mathbf{e}_R = \log \left(\bar{\mathbf{R}}_{\text{BW}}^j \mathbf{R}_{\text{WB}}^j \right), \mathbf{e}_v = {}_W \bar{\mathbf{V}}_{\text{B}}^j - {}_W \mathbf{V}_{\text{B}}^j \\ \mathbf{e}_p = {}_W \bar{\mathbf{P}}_{\text{B}}^j - {}_W \mathbf{P}_{\text{B}}^j, \mathbf{e}_b = \bar{\mathbf{b}}^j - \mathbf{b}^j \end{cases} \quad (3-29)$$

其是 $(\bar{\cdot})$ 是之前优化估计的状态， Σ_p 是之前的信息矩阵。到下一帧 $j+1$ 被优化之后，一帧将会被边缘化掉，这个优化目标函数将一直到有新的关键插入的时候才会被改变。

3.4 本章小结

本文的跟踪是以相机的帧频为基准。在视觉部分我们采用了许多策略来保证跟踪的准确性和鲁棒性。比如固定特征点数量，采用 4 叉树保证特征点在整张图像分布均匀，采用随机采样一致性剔除外点等。在整个追踪过程中我们分初始化成功前的仅视觉跟踪和初始化成功后的视觉和 IMU 融合跟踪两种方式。在视觉和 IMU 融合跟踪过程中本文又根据当前帧是否为关键帧而采用两种不同的目标方程。在优化目标方程中本文的初始位姿是通过 IMU 预积分来预测的，这大大的减少了优化迭代的次数。在追踪成功后，本文会判断当前帧是否为关键帧为后续局部地图和闭环检测做准备。

第 4 章 局部地图优化和闭环检测

4.1 引言

局部地图的优化是减少误差累积的核心步骤，同时也确保整个系统能在大尺度环境中运行。本文的局部地图是由一个滑动窗口（sliding window）构成，在初始化成功之前，滑动窗口中的关键帧是由与当前帧公视的关键帧以及与这些关键帧公视的关键帧组成的，正如第三章仅视觉追踪部分所说的一样。在初始化成功后，局部地图的滑动窗口（sliding window）是由固定窗口（fixed window）和时间序列上局部窗口（local window）组成。其中固定窗口只提供视觉上的约束，不作为优化变量。而局部窗口不仅受到视觉上的约束，而且还受到 IMU 预积分的约束。在本文中，为了减少计算量在滑动窗口中设定固定数量的关键帧，但是又不能直接丢掉之前的关键帧，以此，本文利用舍尔补（schur complement）来边缘化（marginalization）掉之前的关键帧。

为了减少里程过程的累积误差，本文还利用词袋（bag of words）引入了闭环检测。在闭环检测中并没有像传统那样优化 6 个自由度。而是优化 4 个自由度：位置矢量(x, y, z)和偏航角(yaw)。因为有 IMU 信息，因此仰角(pitch)和滚动角(roll)是可观的。

4.2 局部地图的建立

初始化前的局部地图是由当前帧公视的关键帧以及与这些关键帧公视的关键帧组成的，而每一个关键帧都包含该关键帧能够看到的地图点，如图 4-1 所示。

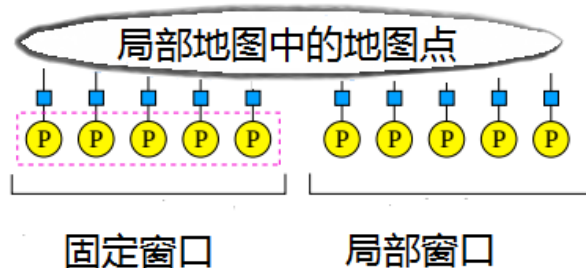


图 4-1 初始化成功前的局部窗口

由图 4-1 图所示整个局部地图分两个窗口，一个是固定窗口，一个是局部窗口。在固定窗口中的关键帧和地图点只参与方程上的约束，而不作为被优化的优化变量。而局部窗口中的关键帧和地图点即作为方程上的约束有作为待优化的变量。其优化的目标方程为：

$$\begin{cases} f(\mathbf{x}) = \min_{\chi} \sum_{j \in L} E_{\text{proj}}(j) \\ E_{\text{proj}}(j) = \rho\left(\left(\mathbf{x} - \pi(\mathbf{X}_C)\right)^T \mathbf{Q}_{k,j}^{-1} \left(\mathbf{x} - \pi(\mathbf{X}_C)\right)\right) \end{cases} \quad (4-1)$$

这个公式和（3-24）极其相似，但不一样的是这里的优化变量：

$$\begin{cases} \chi = \{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{j}_0, \mathbf{j}_1, \dots, \mathbf{j}_m\} \\ \mathbf{x}_k = [\mathbf{p}_{b_k}^w, \mathbf{q}_{b_k}^w], k \in [0, n] \end{cases} \quad (4-2)$$

其中 \mathbf{x}_k 是第 k 个关键帧所对应 IMU 的位姿，即位置矢量 $\mathbf{p}_{b_k}^w$ 和旋转矩阵 $\mathbf{q}_{b_k}^w$ ， \mathbf{j} 是局部地图中的地图点。而（3-24）的公式中优化变量只有当前帧的位姿。

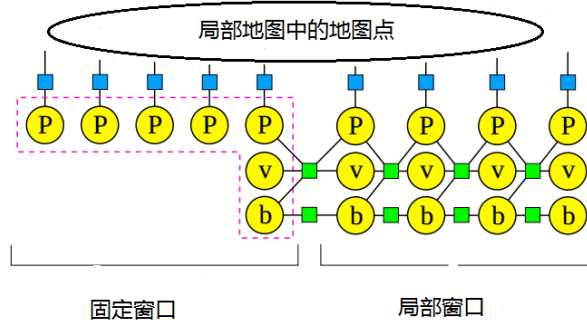


图 4-2 初始化成功后的局部地图

在初始化成功之后，本文固定了局部窗口中参与优化的关键帧数量，并且这些关键帧是时间上连续的，在相邻两个关键帧之间又多加了 IMU 预积分的约束，如图 4-2 所示，可以看出在局部窗口中，跟初始化成功前的局部地图相比多了 IMU 的约束，同时在固定窗口和局部窗口之间也多了 IMU 的约束。其优化的目标方程为：

$$\begin{cases} f(\mathbf{x}) = \min_{\chi} \left(\sum_{j \in L} E_{\text{proj}}(j) + \sum_{i \in L} E_{\text{IMU}}(i, i+1) \right) \\ E_{\text{proj}}(j) = \rho\left(\left(\mathbf{x} - \pi(\mathbf{X}_C)\right)^T \mathbf{Q}_{k,j}^{-1} \left(\mathbf{x} - \pi(\mathbf{X}_C)\right)\right) \\ E_{\text{IMU}}(i, j) = \rho\left(\left[\mathbf{e}_R^T \mathbf{e}_v^T \mathbf{e}_p^T\right] \Sigma_l \left[\mathbf{e}_R^T \mathbf{e}_v^T \mathbf{e}_p^T\right]^T\right) + \rho\left(\mathbf{e}_b^T \Sigma_R \mathbf{e}_b\right) \end{cases} \quad (4-3)$$

其中优化变量：

$$\begin{cases} \chi = \{x_0, x_1, \dots, x_n, j_0, j_1, \dots, j_m\} \\ x_k = [p_{b_k}^w, q_{b_k}^w, v_{b_k}^w, b_a, b_g], k \in [0, n] \end{cases} \quad (4-4)$$

这里的优化变量比式 (4-3) 中多了许多，每一帧图像所对应的优化变量除了位置矢量 $p_{b_k}^w$ 和旋转矩阵 $q_{b_k}^w$ 外，还有 IMU 的速度，加速度偏差 b_a 和陀螺仪偏差 b_g 。

4.3 边缘化与稀疏化

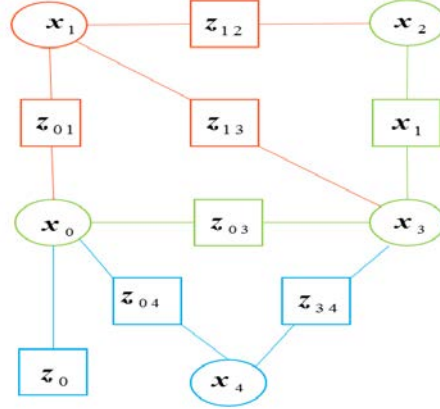
如果局部地图中的关键帧和地图点不做删除操作的话，那么随着时间的积累，局部地图中会有越来越多的关键帧和地图点，整个海森矩阵会变得特别庞大，导致计算量特别大，无法实现实时 SLAM。因此，要采取一定的措施剔除一些关键帧和地图点，保证局部地图中的关键帧和地图点的数量是有界的。本文采取的策略是每当一个关键帧插入到局部地图，都会剔除当前地图中最老的那帧或是时间上第二近的那帧，并做一次局部地图优化。这样做的目的是保证算法的计算量在一定范围内。可是剔除的那帧图像和局部地图中的某些关键帧和地图点都有关系，这样直接剔除是会造成信息的丢失。因此，本文采用舍尔补 (schur) 的方法剔除旧的帧和地图点，即保证了计算量的有界性，又保证了信息不会丢失。

无论是初始化前的局部地图的 Bundle Adjustment 优化，还是初始化后的局部地图的 Bundle Adjustment 优化。需要优化的目标方程都是非线性的，最后需要解的是一个最小二乘问题，而这个最小二乘问题是通过列文伯格-马夸尔特方法不断迭代求解的：

$$H\delta x = -b \quad (4-5)$$

其中 H 是海森矩阵， δx 是优化变量的增量。求解这个方程可以通过 QR 分解来解。

但是本文关注想要求解的变量，不计算那些被边缘化的变量，从而达到减少计算量的目的。假设在整个优化变量 x 中边缘化掉的变量是 x_m 和这些边缘化掉的变量有约束关系的变量为 x_b ，剩余的变量为 x_r 。即 $x = [x_m, x_b, x_r]$ ，相应的测量值 $z = \{z_b, z_r\} = \{z_m, z_c, z_r\}$ ，其中 $z_b = \{z_m, z_c\}$ 如图 4-3 所示，假设局部地图中有 5 个优化的变量即 $x = \{x_1, x_2, x_3, x_4, x_5\}$ ，假设变量 x_1 需要被边缘化掉，其中 x_0, x_2, x_3 和边缘化掉的 x_1 变量有约束关系。则对应图 4-3 有： $x_m = x_1$ ， $x_b = [x_0, x_2, x_3]^T$ ， $x_r = x_4$ 。它们之间的约束为： $z_m = \{z_{01}, z_{12}, z_{13}\}$ ，



若要边缘化掉变量 x_m 而不丢失信息，正确的方法就是把丢失掉的信息封装成未边缘化变量的先验信息：

公式 (4-6) 可以通过下面公式来求解未知数 $\hat{\mathbf{x}}_{\mathbf{b}}, \mathbf{H}_{\mathbf{t}}^{-1}$:

求解公式 (4-7) 这个最小二乘问题可以得到下面的海森矩阵:

需要边缘化掉 x_m ，对 (4-8) 消元得到：

其中 $\mathbf{H}_{bb} - \mathbf{H}_{mb}^T \mathbf{H}_{mm}^{-1} \mathbf{H}_{mb}$ 称为 \mathbf{H}_m 在 \mathbf{H}_b 中的舍尔补，有了公式（4-9）我们就可以计算 $\delta \mathbf{x}_b$ 了：

这样我们就能够迭代的更新部分变量，从而维持计算量不增加。公式 (4-7)

$$\min \left(\frac{1}{2} \left\| \hat{\mathbf{x}}_b - \mathbf{x}_b \right\|_{H_t}^2 + \sum_{i,j \in (z_c, z_r)} \frac{1}{2} \left\| z_{ij} - h_{ij}(\mathbf{x}_i, \mathbf{x}_j) \right\|_{H_{ij}}^2 \right) \quad (4-11)$$

从上面的过程中，我们要注意，构建出来的 $\mathbf{H}\delta\mathbf{x} = -\mathbf{b}$ 是利用了边缘化变量的信息。也就是说我们没有丢掉边缘化掉的信息，但是在计算的时候本文却只计算了希望优化的变量。在整个局部地图优化的过程中，不断地由新的变量加入，也有旧的变量被边缘化掉。因此，在本文中局部地图的优化在没有丢掉边缘化信息的同时，保证了计算量的有界性。

在边缘化的过程中我们需要不断的迭代海森矩阵 \mathbf{H} 和残差 \mathbf{b} ，而迭代过程中，状态变量也会随之更新。本文采用固定求导变量的方式（First Estimate Jacobians）计算雅克比。假设之前的目标函数可以写成：

$$c(\mathbf{x}) = c_m(\mathbf{x}_m, \mathbf{x}_b) + c_r[\mathbf{x}_b, \mathbf{x}_r] \quad (4-12)$$

因此，我们有：

$$\min_{\mathbf{x}} c(\mathbf{x}) = \min_{\mathbf{x}_b, \mathbf{x}_r} \left(\min_{\mathbf{x}_m} c(\mathbf{x}_m, \mathbf{x}_b, \mathbf{x}_r) \right) = \min_{\mathbf{x}_b, \mathbf{x}_r} \left(\min_{\mathbf{x}_m} c_m(\mathbf{x}_m, \mathbf{x}_b) + c_r(\mathbf{x}_b, \mathbf{x}_r) \right) \quad (4-13)$$

这里先求 $\min_{\mathbf{x}_m} c_m(\mathbf{x}_m, \mathbf{x}_b)$ ，通过把 $c_m(\mathbf{x}_m, \mathbf{x}_b)$ 二阶泰勒展开：

$$c_m(\mathbf{x}_m, \mathbf{x}_b) \approx c_m(\hat{\mathbf{x}}_m, \hat{\mathbf{x}}_b) + \mathbf{G}^T \begin{bmatrix} \mathbf{x}_m - \hat{\mathbf{x}}_m \\ \mathbf{x}_b - \hat{\mathbf{x}}_b \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \mathbf{x}_m - \hat{\mathbf{x}}_m \\ \mathbf{x}_b - \hat{\mathbf{x}}_b \end{bmatrix}^T \mathbf{H} \begin{bmatrix} \mathbf{x}_m - \hat{\mathbf{x}}_m \\ \mathbf{x}_b - \hat{\mathbf{x}}_b \end{bmatrix} \quad (4-14)$$

其中：

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_{mm} \\ \mathbf{G}_{mb} \end{bmatrix}, \mathbf{H} = \begin{bmatrix} \mathbf{H}_{mm} & \mathbf{H}_{mb} \\ \mathbf{H}_{mb}^T & \mathbf{H}_{bb} \end{bmatrix} \quad (4-15)$$

该论文求导的线性化点是 $(\hat{\mathbf{x}}_m, \hat{\mathbf{x}}_b)$ ，使用舍尔补边缘化掉 \mathbf{x}_m ，其中：

$$\mathbf{x}_m = \hat{\mathbf{x}}_m - \mathbf{H}_{mm}^{-1} \left(\mathbf{G}_{mm} + \mathbf{H}_{mb}^T (\mathbf{x}_b - \hat{\mathbf{x}}_b) \right) \quad (4-16)$$

把上公式中的 \mathbf{x}_m 带入公式（4-14）得：

$$\min_{\mathbf{x}} c_m(\mathbf{x}_m, \mathbf{x}_b) = \min_{\mathbf{x}} \varsigma + \mathbf{G}_t^T (\mathbf{x}_b - \hat{\mathbf{x}}_b) + \frac{1}{2} (\mathbf{x}_b - \hat{\mathbf{x}}_b)^T \mathbf{H}_t (\mathbf{x}_b - \hat{\mathbf{x}}_b) \quad (4-17)$$

其中：

$$\begin{cases} \mathbf{G}_t^T = \mathbf{G}_{mb} - \mathbf{H}_{mb}^T \mathbf{H}_{mm}^{-1} \mathbf{G}_{mm} \\ \mathbf{H}_t = \mathbf{H}_{bb} - \mathbf{H}_{mb}^T \mathbf{H}_{mm}^{-1} \mathbf{H}_{mb} \end{cases} \quad (4-18)$$

由以上结果的得出公式（4-13）的等价方程：

$$\begin{aligned} \dot{c}_r(\mathbf{x}_b, \mathbf{x}_r) = & \mathbf{G}_t^T \left(\mathbf{x}_b - \hat{\mathbf{x}}_b \right) + \frac{1}{2} \left(\mathbf{x}_b - \hat{\mathbf{x}}_b \right)^T \mathbf{H}_t \left(\mathbf{x}_b - \hat{\mathbf{x}}_b \right) \\ & + \sum_{(i,j \in Z_c, Z_r)} \frac{1}{2} \left\| z_{ij} - h_{ij}(\mathbf{x}_i, \mathbf{x}_j) \right\|_{\mathbf{H}_{i,j}}^2 \end{aligned} \quad (4-19)$$

本文固定求雅克比时的线性化点 $(\hat{\mathbf{x}}_m, \hat{\mathbf{x}}_b)^{[40]}$ ，如图 4-4 所示，因此 $\mathbf{G}_t^T = 0$ 那公式 (4-11) 就等价于公式 (4-19)。如图 4-4 所示， ζ_0 是目标方程迭代的初始值，即初始的线性化点。 x 是迭代过程的累积值， δ 是每一步迭代的值。虽然采用固定的线性化点去求雅克比，得到的不是最优解，但是这样做可以保证 $\mathbf{G}_t^T = 0$ ，保证了优化的目标方程没有加入伪造信息。

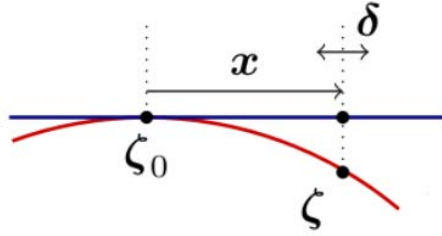


图 4-4 固定求雅克比时的线性化点的示意图

在本文中并没有像传统的边缘化那样只是边缘化掉最老的那帧图像帧，而是根据当前帧的上一帧是否为关键帧而选择边缘化掉哪一帧图像，如图 4-5 所示，如果当前帧的上一帧是关键帧，则边缘化掉局部地图中最老的那一帧，如果当前帧的上一帧不是关键帧则边缘化掉该帧。

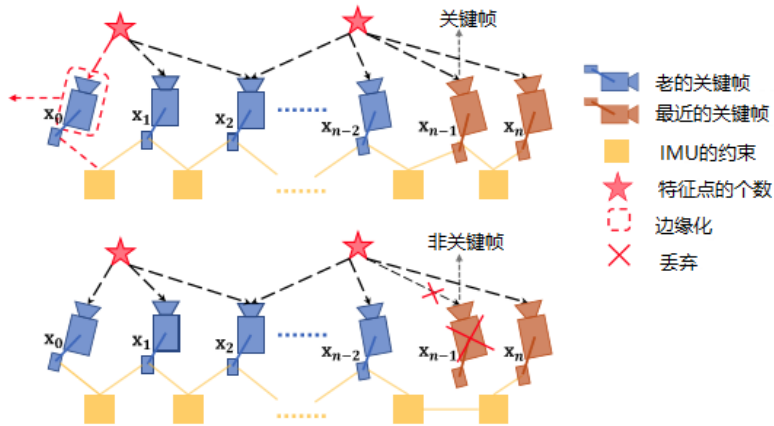


图 4-5 根据当前帧的上一帧是否为关键帧而边缘化掉不同的关键帧

边缘化掉过去的关键帧和地图点，虽然减少了优化变量的个数和海森矩阵的维度，但是与边缘化掉的变量 \mathbf{x}_m 有关系的变量 \mathbf{x}_b 所对应的海森矩阵块会变得稠密。视觉 SLAM 的 Bundle Adjustment 中的海森矩阵本身具有稀疏性，这是因为在整个局部地图中，每一帧的图像只能观测部分地图点，如图 4-6 所示。

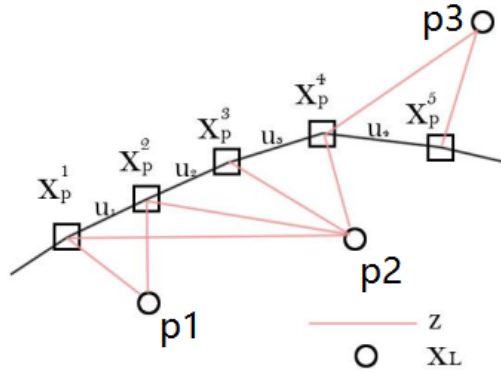


图 4-6 相机运动轨迹和路标示意图

所以，一般 SLAM 中的 Bundle Adjustment 的海森矩阵如图 4-7 所示，其中左上角局部地图的目标函数对相机位姿的导数相乘得到，右下角是局部地图的目标函数对地图点的导数相乘得到，绿色部分是局部地图的目标函数对相机位姿的偏导和地图点的偏导乘积部分。

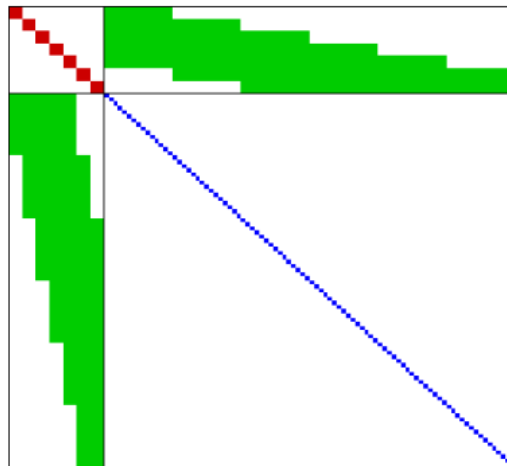


图 4-7 SLAM 中的 Bundle Adjustment 的海森矩阵简单示意图

假设局部地图中有四个关键帧，六个地图点，每个关键帧可以观测到三个地图点，如图 4-8 所示。

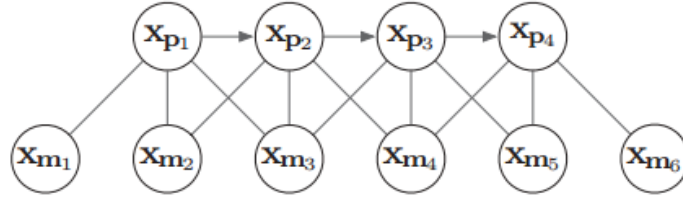
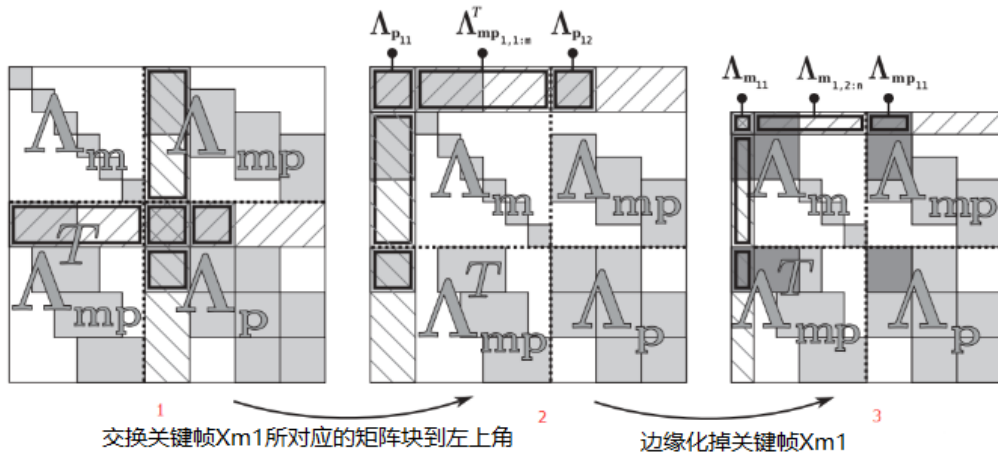
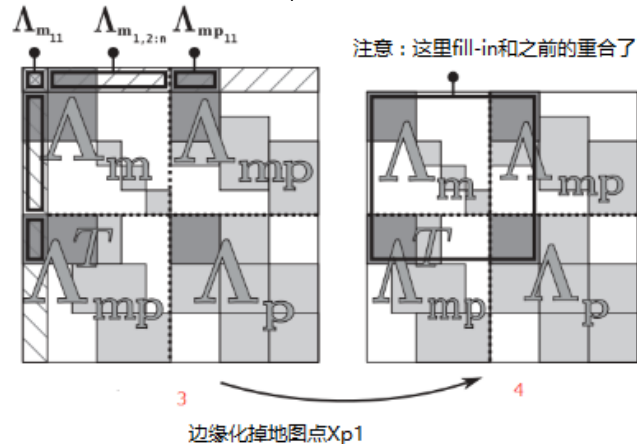


图 4-8 局部地图的关键帧与地图点的关系

假设在局部地图中要边缘化掉关键帧 x_{m1} 和地图点 x_{p1} ，则整个局部地图的海森矩阵会发生如图 4-8 的变化。


 图 4-9 边缘化掉关键帧 x_{m1} 海森矩阵的变化

如图 4-9 所示，边缘化掉地图点 x_{p1} 后，整个海森矩阵的维度变小了，但是与边缘化掉 x_{m1} 有约束的那些变量所对应的矩阵块变得稠密了。边缘化掉关键帧 x_{m1} 后，再边缘化掉地图点 x_{p1} 整个海森矩阵会发生如图 4-10 的变化。


 图 4-10 边缘化掉地图点 x_{p1} 海森矩阵的变化

由图 4-10 可以看出边缘化掉地图点 x_{p1} 后，整个海森矩阵的维度变小了，而且海森矩阵与之前相比并没有变得稠密，这是因为地图点 x_{p1} 与关键帧 x_{m1}

和其他的变量有共同的约束,也就是说边缘化地图点 \mathbf{x}_{p1} 使海森矩阵变稠密的区域与边缘化关键帧 \mathbf{x}_{m1} 使海森矩阵变稠密的区域重合了。

由边缘化的过程可知,虽然边缘化掉一些变量可以使得海森矩阵维度变小,但是却使得海森矩阵变得稠密,如果不对这稠密的海森矩阵做处理,那么实际的计算量往往会比没有边缘化之前的计算量还要大。因此,需要对边缘化后的海森矩阵进行稀疏化处理。相比 GLC^[33]全局线性化点,本文采用类似 NFR^[34]局部线性化点的方式。本文选择用一个稀疏海森矩阵 Ω 来近似局部地图中被边缘化后的海森矩阵 ${}^L\mathbf{A}$ 。通过正则 KLD 最小化^[35]可以获得近似的海森矩阵:

$$\min_{\Omega} \langle \Omega, \Sigma \rangle - \log \det(\Omega) + \lambda \|\Omega\|_1 \quad s.t. \Omega \geq 0 \quad (4-20)$$

其中 Σ 是海森矩阵 ${}^L\mathbf{A}$ 的逆, $\langle *, * \rangle$ 代表矩阵的内积。 λ 是用来控制矩阵 Ω 的稀疏性的。同时解上式要满足解是半正定的。为了解公式 (4-20) 本文使用了文献[36]的方法。

4.4 闭环检测

即使采取局部地图优化可以有效的减少估计误差,但是随着时间的累计,也会出现误差的累计。特别是在室内的环境中,视觉—IMU 系统经常会出现同一个地方多次,但是如果没有闭环检测,误差会逐渐变大,而且这个误差是无界的,如果有闭环检测,那么累计误差将会被消除,如图 4-11 所示。

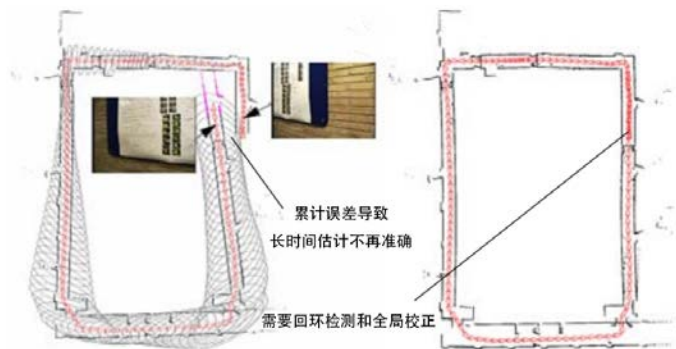


图 4-11 无闭环检测与有闭环检测的位姿估计对比

与文献[41-42]相似,在本文采用词袋模型 (bag of words) 做闭环检测。首先需要生成视觉词典,所谓的“视觉词典”就是每帧图像里的特征点。在本文中利用 K-means 算法把相似的特征点聚为一类,其中的相似性就是特征

点描述子之间的汉明距离。整个建立词典的过程如图 4-11 所示，首先提取 ORB 特征点和 BRIEF 描述子，这步在追踪部分已经完成；然后利用 K-means 算法把相似的描述子聚为一类作为一个节点；然后对产生的节点依次进行 K-means 聚类，直至产生根节点。

由于样本图片多达几万张，甚至几十万张所以视觉字典都是线下建立的，本文为了方便，采用的是别人已经建立好的 ORB 视觉字典。视觉字典建立好后，本文对 SLAM 过程中的每一个关键帧的图像都按照视觉字典的方式建立单词。也就是说每一个关键帧都有自己的视觉单词树以供后面的图像帧闭环检测。每一个视觉单词都有自己的序号和权重，即每个视觉单词是一个向量：

$$V = \langle index, weight \rangle \quad (4-21)$$

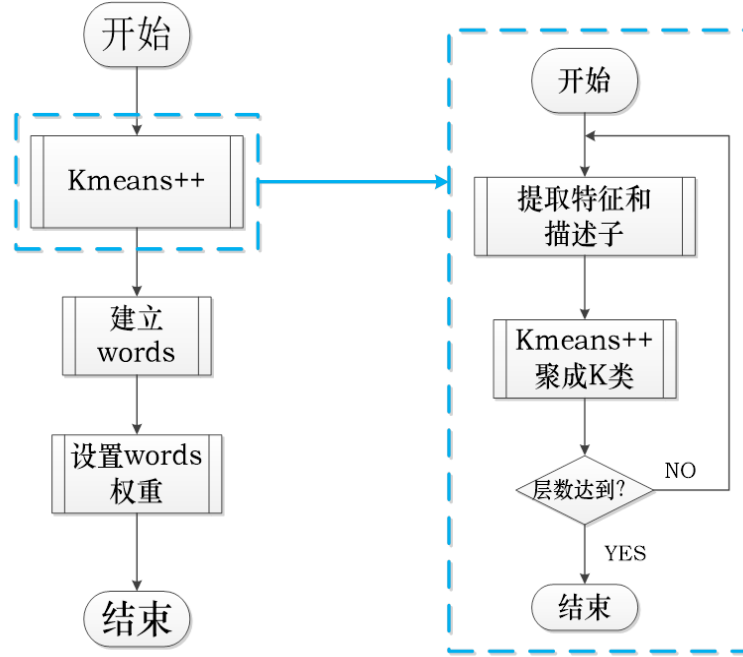


图 4-12 视觉单词树的建立过程

其中 $index$ 是视觉单词的 ID，而 $weight$ 是该单词的权重：

$$w_t^i = tf(i, I_t) \times idf(i), w_t^i = w_t^i / \sum w_t^i \quad (4-22)$$

$tf(i, I_t)$ 是生成单词向量时的权重分量：

$$tf(i, I_t) = \frac{n_{it}}{n_t} \quad (4-23)$$

(4-23) 的分母是当前图片出现单词的总个数，分子是当前图片出现 i 类特征

的个数。其中 $idf(i)$ 是生成字典时的权重分量：

$$idf(i) = \log \frac{N}{n_i} \quad (4-24)$$

$idf(i)$ 的值是固定不变的，因为在生成词典的时候就确定下来了，其中分母表示出现单词 i 的图片个数，分子是训练样本图片的总个数。

在本文中，采取逆序索引单词（inverseindexwords）的方式加快闭环检测的速度，即只检测与当前帧图像有共同单词的图片。其中逆序索引单词的结构如图 4-13 所示。

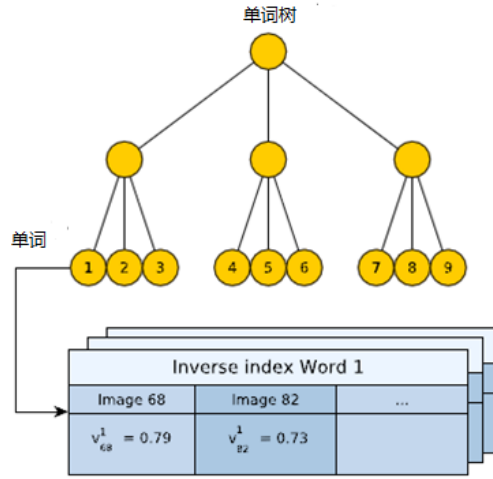


图 4-13 逆序索引单词结构

逆序索引单词涵盖了所有出现的单词，每个单词都指向包含它的所有图片，这样就可以采用投票机制加速了与候选帧的匹配。其中两帧图片之间的归一化得分如下：

$$\eta(v_t, v_{t_j}) = \frac{s(v_t, v_{t_j})}{s(v_t, v_{t-\Delta t})} \quad (4-25)$$

其中：

$$s(v_1, v_2) = 1 - \frac{1}{2} \left| \frac{v_1}{|v_1|} - \frac{v_2}{|v_2|} \right| \quad (4-26)$$

公式（4-26）代表两帧图片的相似性。在本文中我们对公式（4-25）的分数设定了阈值，大于阈值的候选帧被筛选出来进入组匹配校验。

当图像 I_{t_1} 与图像 I_{t_2} 之间有可能形成一个闭环的时候，则 I_{t_1} 同样和 $I_{t_2 \pm \Delta t}$ ，

$I_{t_2 \pm 2\Delta t}$ 也有着较高的相似性。因此，组匹配得分函数定义如下：

$$H(v_t, V_{T_j}) = \sum_{j=n_i}^{m_i} \eta(v_t, v_{t_j}) \quad (4-27)$$

其中 V_{T_j} 代表候选图像所有集合。然后选择组匹配最高分作为闭环检测的唯一候选帧。如果这个最高分满足闭环检测的阈值要求，则该帧图像与当前帧形成闭环。当成功的检测到闭环的时候，我们开始执行 4 自由度的优化：

$$\min_{p_i^w, \varphi_i, p_j^w, \varphi_j} r_{i,j}(p_i^w, \varphi_i, p_j^w, \varphi_j) \quad (4-28)$$

其中 i 代表当前帧， j 代表与当前帧形成闭环的那帧图像。 p_i^w 表示当前帧的位置， φ_i 代表当前帧的偏航角 (yaw)， p_j^w, φ_j 分别表示与当前帧形成闭环的那帧的位置和偏航角 (yaw)。其中：

$$r_{i,j}(p_i^w, \varphi_i, p_j^w, \varphi_j) = \begin{bmatrix} R(\hat{\varphi}_i, \hat{\theta}_i, \hat{f}_i)^{-1} (p_j^w - p_i^w) - \hat{p}_{ij}^i \\ \varphi_j - \varphi_i - \hat{\varphi}_{ij} \end{bmatrix} \quad (4-29)$$

其中 $\hat{\theta}_i, \hat{\varphi}$ 分别是当前帧的仰角 (pitch) 和滚动角 (roll)。因为它们是可观的，可以直接通过本系统的观测直接获得。在闭环检测中，除了闭环两帧图像以外，还有他们之间的关键帧，因此，整个闭环的目标函数为：

$$\min_{p, \varphi} \left\{ \sum_{(i,j) \in S} \|r_{i,j}\|^2 + \sum_{(i,j) \in L} \rho(\|r_{i,j}\|^2) \right\} \quad (4-30)$$

其中 L 是所有闭环帧序列， S 是闭环中除了闭环帧的其他关键帧。 $\rho(\bullet)$ 是损失函数，目的是为了减少错误闭环的影响。当整个系统走了很远之后才检测到闭环，那么公式 (4-30) 的计算量是巨大的，为了让闭环检测的计算量是有界的，且维持在一定范围。本文设定闭环最大关键帧数量，一旦超出这个数量，本文会剔除那些相似的关键帧。

4.5 本章小结

本章主要解决系统追踪过程误差和跟踪特征点较少的问题。本文通过关键帧之间的关系构建固定窗口和局部窗口两种窗口，其中固定窗口只负责构建关键帧之间的约束，不作为优化的变量，而局部窗口不仅参与关键帧之间的约束，而且也作为优化变量。除此之外，本文在局部地图时间连续的关键

帧之间添加了 IMU 积分的约束。在优化局部地图中，本文引入边缘化和稀疏化，这确保了在不丢失信息的前提下可以有效的减少局部地图的计算量。局部地图优化后，本文引入了闭环检测，这可以有效的减少累计误差，在闭环检测成功后，本文提出了一种 4 自由度的优化方式，相比 6 自由度的优化精度更高。

第 5 章 实验结果

5.1 引言

为了验证本文提出的算法。该算法分别在 EuRoC 数据集上和搭建的小车上做了测试。本文的算法在 EuRoC 测试，可以得到尺度 s 的误差小于 5%，重力矢量的大小收敛在 9.8 m/s^2 左右，加速度偏差收敛在 0 左右，陀螺仪偏差收敛在 $10^{-3} \text{ }^\circ/\text{s}$ 。这个初始化结果相对目前最好的几个算法是具有竞争力的。其中平均误差在 8 cm 左右，最大误差也在 10cm 以内,该算法的定位结果在相对目前最好的几个开源算法相比也是具有竞争力的，本文提出的算法和目前双目视觉最好的开源算法 ORB_SLAM 分别在 EuRoC 数据集上做了对比，本文的算法虽然在平均时间上比 ORB_SLAM 慢，但是在定位精度上比 ORB_SLAM 高，并且最大误差要比 ORB_SLAM 小很多。

本文提出的算法也在移动小车上做了测试，在这部分本文提出一种自动估计 IMU 与相机的旋转关系的方法，解决旋转对不齐的问题。由于硬件上相机与 IMU 的时间不对齐，而且移动小车中的相机与 IMU 相对 EuRoC 数据集都是十分廉价的，因此在初始化结果和定位精度比较差。

5.2 EuRoC 数据集测试

为了验证本文提出的算法，本算法在 EuRoC^[37]数据集上做了测试，该数据集来至 2016 年苏黎世联邦理工大学发表的顶级期刊 IJRR 的一篇论文。该数据集是被配置了一个频率为 20Hz 的双目摄像头和一个频率为 200Hz 的 IMU 的无人机所采集的，其中整体硬件设备如图 5-1 所示。

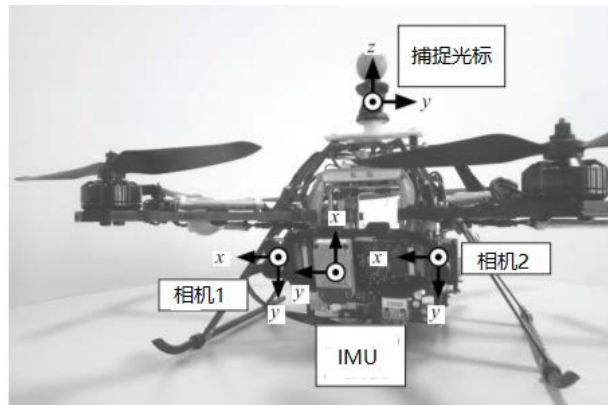


图 5-1 EuRoC 数据集采集硬件设备

两个其中两个相机和 IMU 采集的时间是同步的，并且该数据集给出了精确的 6 自由的真实轨迹，相机的内参和 IMU 与相机的相对位姿，其中 IMU，相机和光标之间的位姿关系如图 5-2 所示。

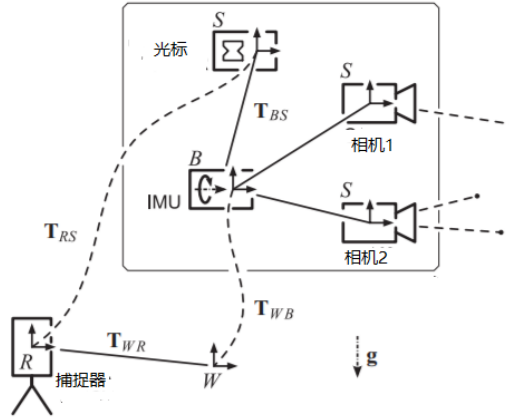
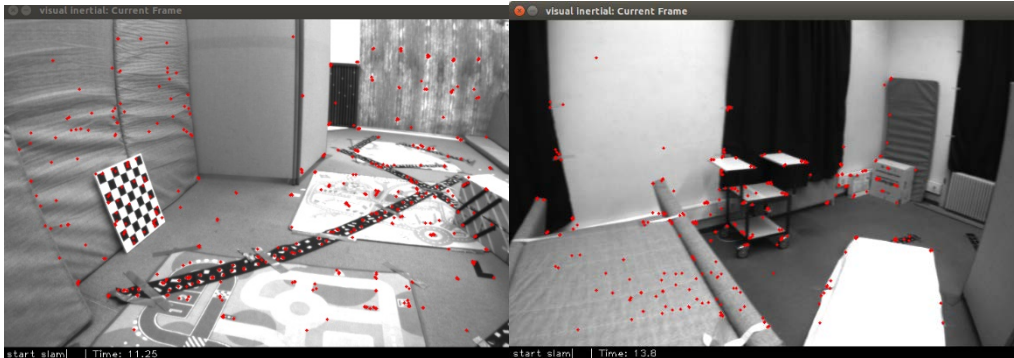


图 5-2 IMU，相机和光标之间的关系示意图

该无人机分别在两个不同的房间采集了 11 个数据集。根据房间的纹理，无人机的运动快慢这 11 个数据集分为简单，中等和困难三类。

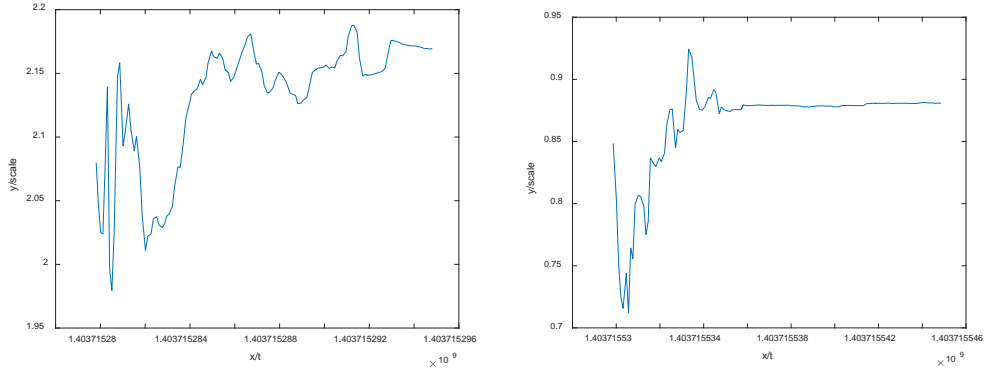
本实验在一个带有 8GB 内存的 i7-4790 CPU 台式电脑对其中两个数据集做了测试。其中两个数据集的环境如图 5-3 所示。



a) V1_01_easy 数据集场景 b) V2_01_easy 数据集场景

图 5-3 本文提出的算法测试的两个数据集的场景

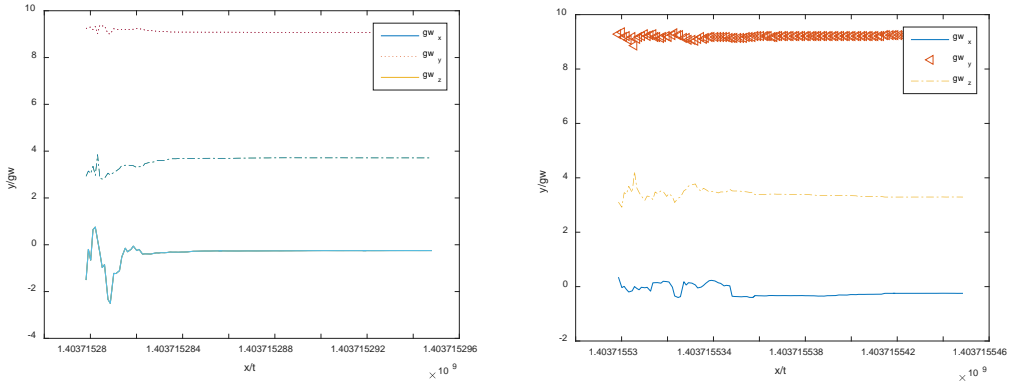
以下是分别使用该算法对两个数据集的尺度，重力矢量，加速度偏差和陀螺仪偏差估计的数据。其中分别在两个数据集上尺度的估计如图 5-4 所示。由图 5-4 的数据显示，在初始化 10 秒左右尺度开始收敛。经过和真实的尺度对比，该算法估计出的尺度误差小于 5%。



a) V1_01_easy 数据集 b) V2_01_easy 数据集

图 5-4 本文提出的算法测试的两个数据集的尺度估计结果

本文的初始化算法分别在两个数据集上估计的重力矢量如图 5-5 所示。

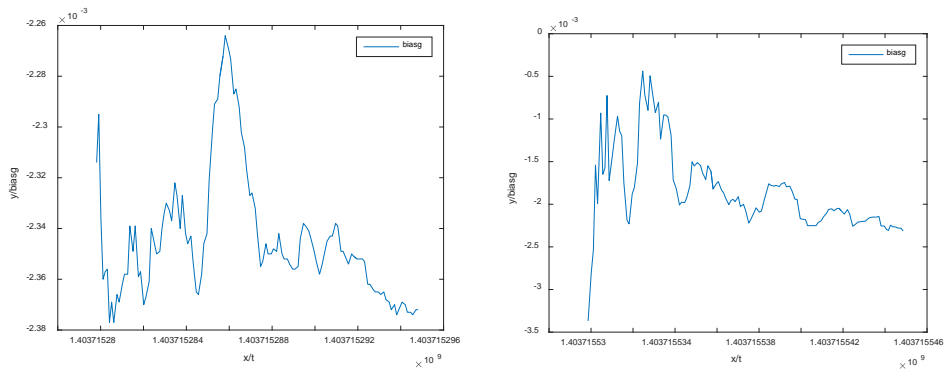


a) V1_01_easy 数据集 b) V2_01_easy 数据集

图 5-5 本文提出的算法测试的两个数据集的重力矢量估计结果

由图 5-5 可以看出加速度偏差在初始化 10 秒~15 秒左右开始收敛到 0。

本文算法分别在两个数据集上对陀螺仪偏差的估计如图 5-6 所示。



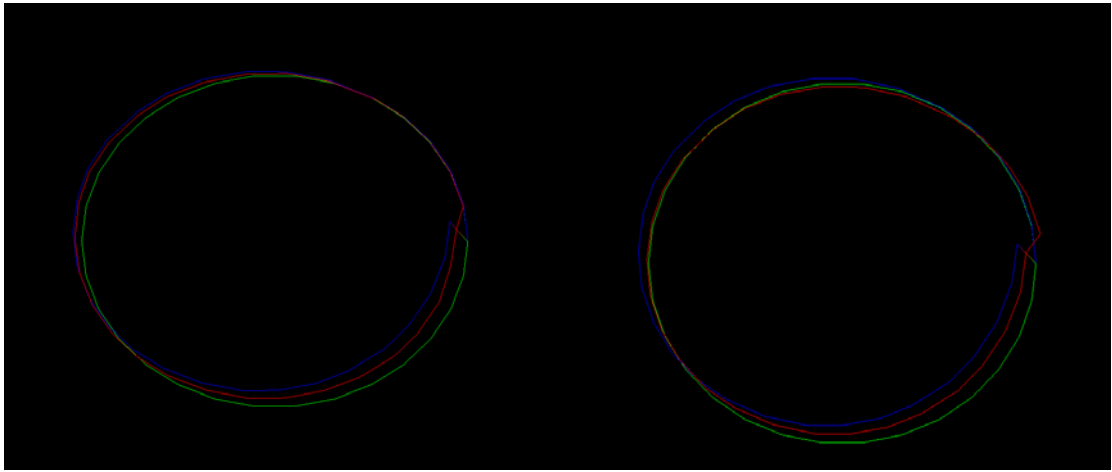
a) V1_01_easy 数据集 b) V2_01_easy 数据集

图 5-6 本文提出的算法测试的两个数据集的陀螺仪偏差估计结果

由图 5-6 所示陀螺仪偏差逐渐收敛在 10^{-3} 范围内。

本文的初始化算法和当前最好的几个开源算法相比具有明显的优势，首先，本文的算法不需要 IMU 的初始化速度；其次本算法对硬件和场景要求不高；最后也是最重要的本算法对尺度，重力方向，加速度偏差和陀螺仪偏差估计的精度要比其他算法高，但是本文提出的算法在初始化期间花费的时间要大于其他开源算法。

在闭环检测上，本文提出了优化 4 自由度状态变量，而不是传统的 6 自由度或是 7 自由度变量，本文提出的闭环检测算法在单位圆的数做了测试。如图 5-6 所示。



a) 优化 6 自由度闭环检测结果 b) 优化 4 自由度闭环检测结果

图 5-7 本文提出的 4 自由度闭环检测与 6 自由度闭环检测的对比

在图 5-7 中，其中黄色曲线为真实位置轨迹，蓝色曲线是未优化前的位置轨迹，黄色曲线是优化后的位置轨迹。从上图可以看出本文提出的优化 4 自由度的闭环检测相对于传统的优化 6 自由度闭环检测更接近真实的位置轨迹。因为在本文提出的视觉与 IMU 融合 SLAM 算法系统中，姿态中的仰角和滚动角是可观的。

本文提出的 SLAM 算法在 V1_01_easy 和 V2_01_easy 数据集上做了定位精度的测试。参考文献[38]，通过相似变换把本文提出的算法估计的位姿和运动捕捉系统得到的真实值做对比，如图 5-8 所示，可以看出估计出的轨迹和真实轨迹几乎完全重合。

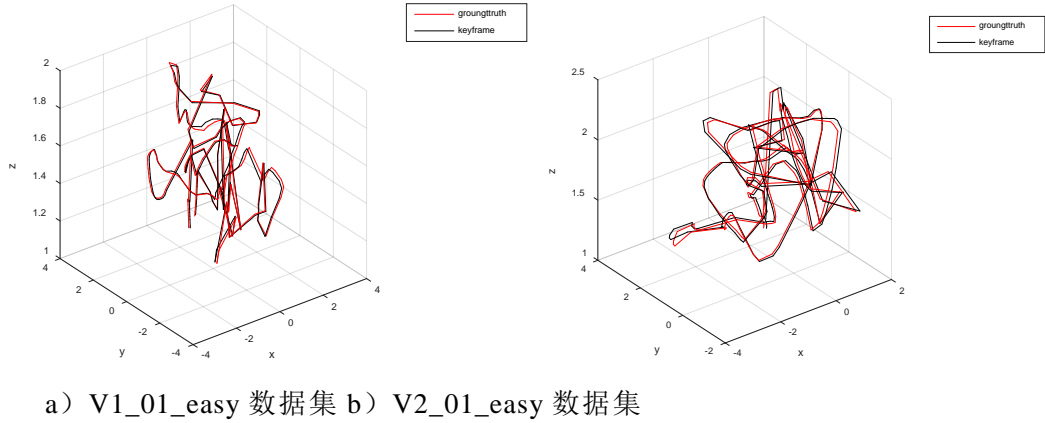


图 5-8 本文提出的算法估计的轨迹与真实轨迹对比结果

本文还对自己的算法估计出来的位置和视觉捕捉系统得到的真实位置做了误差对比，如图 5-9 所示。

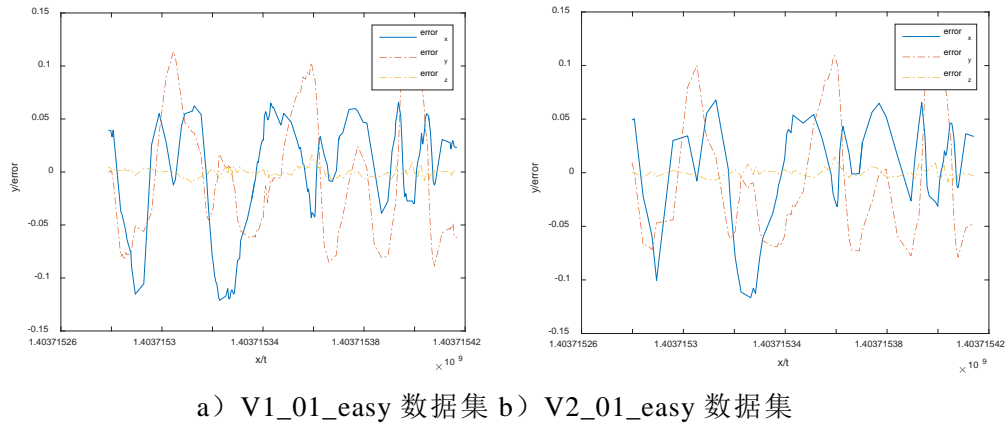


图 5-9 本文提出的算法估计的位置误差

由图 5-9 可以看出本文的算法位置估计误差在 0 上下波动，最大误差是 10cm，这个结果与目前最好的开源算法相比是十分具有竞争力的。

本文的算法在 EuRoC 的两个数据集上和目前最好的双目 SLAM 算法 ORB_SLAM 算法做了位置和姿态的误差对比，对比结果如图 5-10 所示，由图 5-10 所示本文提出的视觉与 IMU 融合的算法相比目前最好的开源双目 SLAM 算法 ORB_SLAM 具有绝对的优势。在里程的开始阶段本文提出的算法在位置和姿态上的误差都大于 ORB_SLAM，但是在后阶段本文提出的 SLAM 算法的误差要小于 ORB_SLAM，这说明本文提出的视觉与 IMU 融合的 SLAM 算法收敛性要比 ORB_SLAM 算法好。

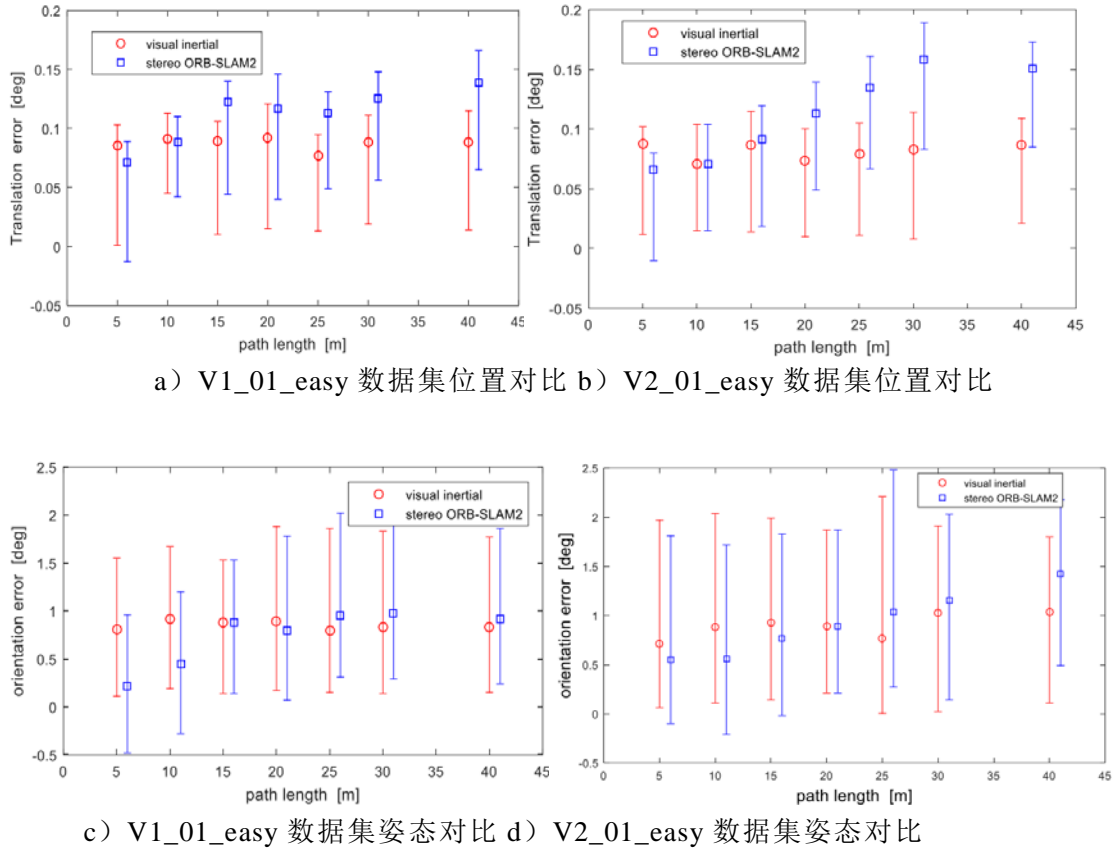


图 5-10 本文提出的算法双目 ORB_SLAM 算法在位置和姿态误差上对比

5.3 实物小车中的测试

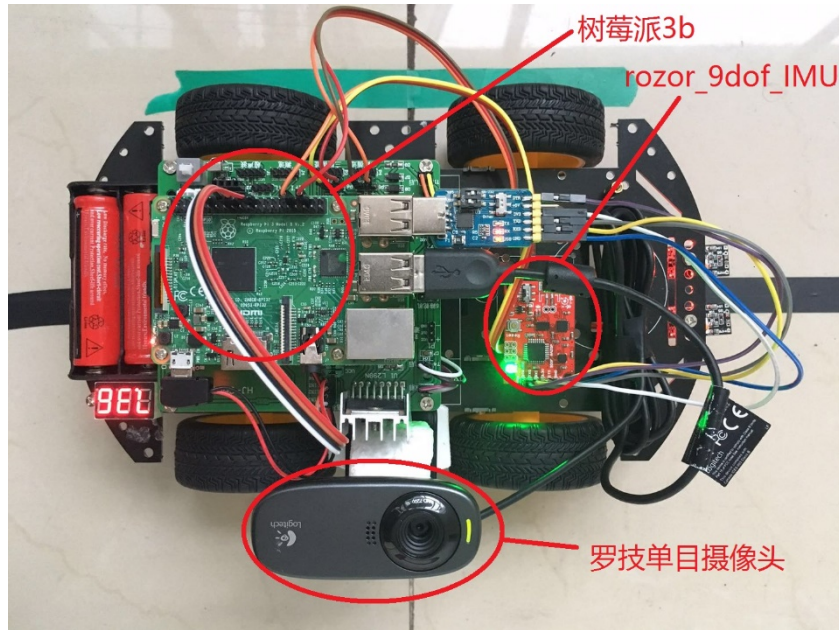


图 5-11 整体硬件示意图

为了验证本文提出的 SLAM 算法的鲁棒性和实用价值, 该算法被测试在一个装有树莓派 3B 和 9 轴 IMU 的小车上, 如图 5-10 所示。其中树莓派被装配 4 核 64 位的 RAM 芯片, 运行频率为 1.2GHZ, 1GB 的 LPDDR2 内存。相机采用的是廉价的罗技免驱动的 500 万摄像头。IMU 采用的也是廉价的 Spark Fun 九轴传感器。供电系统有两节 3.7V 的 18650 锂电池组成。由于我们的硬件的设备性能不好, 且没有外设标定相机与 IMU 之间的位姿, 因此本文受文献[39]启发, 在初始化的时候利用软件来标定相机和 IMU 之间的位姿:

$$\mathbf{R}_{b_{k+1}}^{b_k} \cdot \mathbf{R}_c^b = \mathbf{R}_c^b \cdot \mathbf{R}_{c_{k+1}}^{c_k} \quad (5-1)$$

公式 (5-1) 是一个线性的旋转标定法, 其中 $\mathbf{R}_{c_{k+1}}^{c_k}$ 代表第 k 帧图像到第 $k+1$ 帧通过优化重投影误差求得到的旋转矩阵。 $\mathbf{R}_{b_{k+1}}^{b_k}$ 代表第 k 帧图像到第 $k+1$ 帧通过 IMU 预积分求得的旋转矩阵, \mathbf{R}_c^b 代表 IMU 到相机的旋转矩阵。用四元数来代表公式 (5-1) 的旋转矩阵:

$$\mathbf{q}_{b_{k+1}}^{b_k} \otimes \mathbf{q}_c^b = \mathbf{q}_c^b \otimes \mathbf{q}_{c_{k+1}}^{c_k} \quad (5-2)$$

把等式的右侧移动到等式的左侧得:

$$\left[\varphi_1(\mathbf{q}_{b_{k+1}}^{b_k}) - \varphi_2(\mathbf{q}_{c_{k+1}}^{c_k}) \right] \cdot \mathbf{q}_c^b = \mathbf{Q}_{k+1}^k \cdot \mathbf{q}_c^b = 0 \quad (5-3)$$

其中:

$$\begin{aligned} \varphi_1(\mathbf{q}) &= \begin{bmatrix} \mathbf{q}_w \mathbf{I}_3 + [\mathbf{q}_{xyz} \times] & \mathbf{q}_{xyz} \\ -\mathbf{q}_{xyz} & \mathbf{q}_w \end{bmatrix} \\ \varphi_2(\mathbf{q}) &= \begin{bmatrix} \mathbf{q}_w \mathbf{I}_3 + [\mathbf{q}_{xyz} \times] & \mathbf{q}_{xyz} \\ -\mathbf{q}_{xyz} & \mathbf{q}_w \end{bmatrix} \end{aligned} \quad (5-4)$$

其中 $[\mathbf{q}_{xyz} \times]$ 是四元数变量中的 3 个元素的斜对称矩阵, \otimes 代表四元素的乘积操作。在初始化的时候, 会有 N 个连续的图像帧, 则 N 个连续的关键帧构建的线性约束方程为:

$$\begin{bmatrix} \mathbf{Q}_1^0 \\ \mathbf{Q}_2^1 \\ \mathbf{M} \\ \mathbf{Q}_N^{N-1} \end{bmatrix} \cdot \mathbf{q}_c^b = \mathbf{Q}_N \cdot \mathbf{q}_c^b = 0 \quad (5-5)$$

最后通过奇异值分解来求解 \mathbf{q}_c^b 。文献[39]证明了在整个 SLAM 过程中, IMU 与相机的位置矢量的精度对最终的定位结果影响十分小, 因此在这部分本文采用人工标定相机与 IMU 的位置矢量 \mathbf{p}_c^b 。

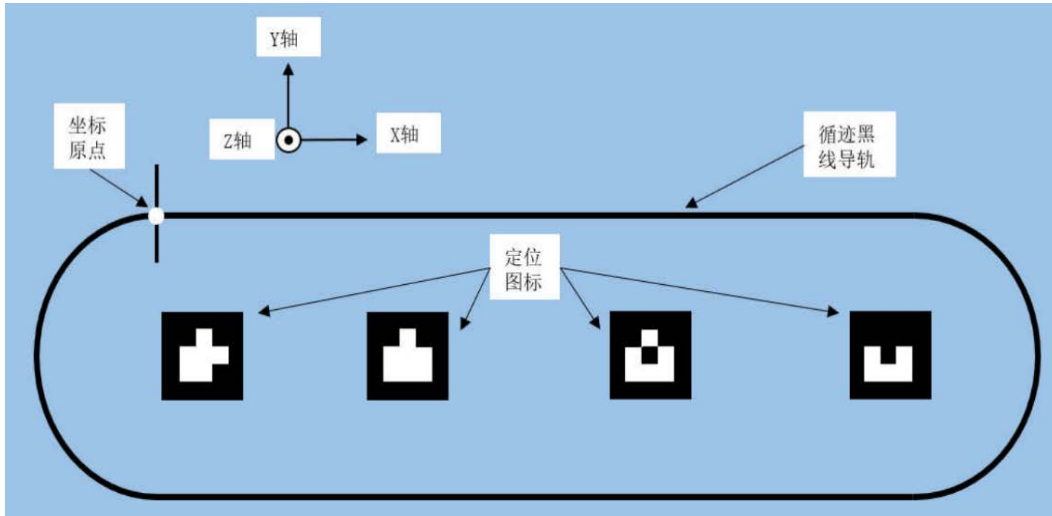


图 5-12 真实系统简单示意图

由于本实验室没有运动捕捉系统，因此本实验采用二维码定位作为真实数据。如图 5-12 所示，本实验采用 4 个二维码，每个二维码都采用 3×3 黑白相间的小正方形作为 ID，按照顺序贴在楼道的天花板上，每两个相邻的二维码之间的 x 轴上相距 1.5m ， y 轴相距为 0 。整个二维码定位系统的世界坐标放在了小车的初始位置上，即相机捕捉到二维码的第一帧，坐标系是按照右手坐标系，如图 5-12 所示。为了方便计算小车的位姿，本实验在地面上贴了一圈接近椭圆的黑线，小车可以按照黑线自动循迹移动。其中本实验的定位轨迹如图 5-13 所示。

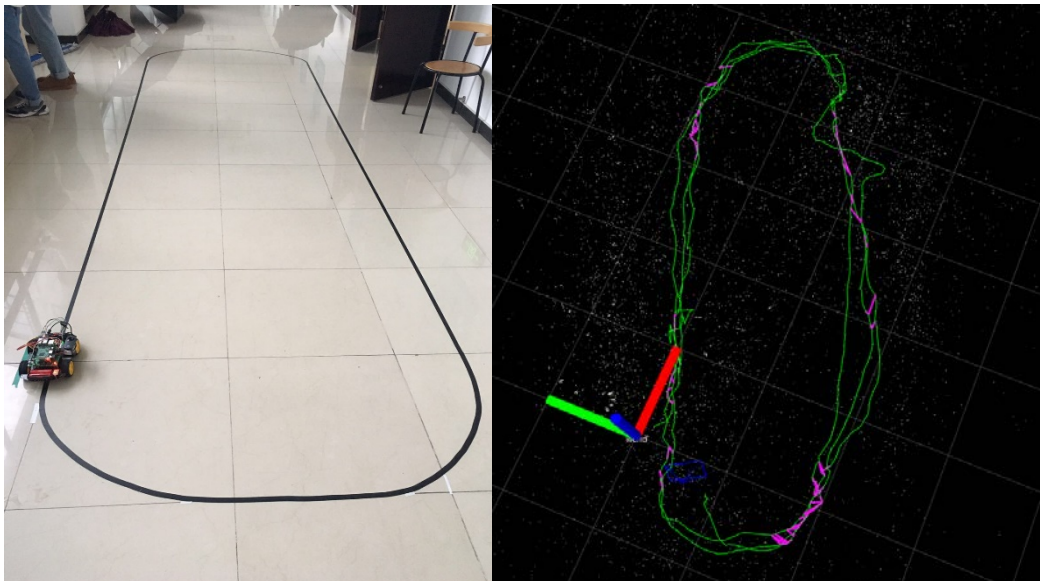


图 5-13 小车定位的轨迹图

表 5-1 本文提出的算法在实物小车上的误差分析

圈数	误差误差		误差平均误差	
	x (cm)	y (cm)	z (cm)	ave(cm)
1	14.36	5.87	2.11	7.45
2	23.95	5.34	1.65	10.31
3	18.79	11.95	2.22	11.37
4	24.27	8.94	3.36	12.19
5	19.26	8.18	3.02	10.15

在本实验中，为了验证本文提出算法在实际应用中的可靠性，我们让小车按照黑色的轨迹线跑了 5 圈，然后与二维码的定位信息进行对比。如表 5-1 所示，每一圈回到原点位置，会统计一次平均误差。从表 5-1 可以看出本文提出的单目视局与 IMU 融合的 SLAM 系统在实物中的里程是有一定的误差的，这主要是因为 IMU 与相机的时间没有对齐，除此之外低廉的 IMU 获取的加速度和角速度的精度不高，导致本文提出的算法在实物中有误差。

5.4 本章小结

本章主要是分别在 EuRoC 数据集和实物小车上验证了本文提出的视觉与 IMU 融合的 SLAM 算法。本文的算法在 EuRoC 测试，可以得到尺度的误差小于 5%，重力矢量的大小收敛在 9.8 m/s^2 左右，加速度偏差收敛在 0 左右，陀螺仪偏差收敛在 $10^{-3} \varphi/t$ 。这个初始化结果相对目前最好的几个算法是具有竞争力的。其中平均误差在 8 厘米左右，最大误差也在 10cm 以内，该算法的定位结果在相对目前最好的几个开源算法相比也是具有竞争力的，本文提出的算法和目前双目视觉最好的开源算法 ORB_SLAM 分别在 EuRoC 数据集上做了对比，本文的算法虽然在平均时间上比 ORB_SLAM 慢，但是在定位精度上比 ORB_SLAM 高，并且最大误差要比 ORB_SLAM 小很多。本文提出的算法也在移动小车上做了测试，由于硬件上相机与 IMU 的时间不对齐，而且移动小车中的相机与 IMU 相对 EuRoC 数据集都是十分廉价的，因此在初始化结果和定位精度比较差。

结论

本文基于开源的 ROS 系统平台，利用单目相机和 IMU 传感器，提出了一种鲁棒的高精度的基于非线性优化的视觉与 IMU 融合的 SLAM 算法。课题主要完成的工作如下：

(1) 本文提出了一种高精度的初始化方法。首先仅利用单目视觉寻找最佳的初始两帧图像，然后并行计算两个模型来构建初始化地图，得到地图之后利用松耦合的方法估计 IMU 与单目视觉的初始化参数：每帧图像对应的旋转矩阵、位置矢量、重力矢量、地图的尺度、IMU 的速度和偏差。在实物的实验中，本文在初始化估计的参数中，还多了 IMU 与单目相机之间的姿态。

(2) 本文提出了一种鲁棒的追踪方法。在初始化成功前仅使用视觉追踪和初始化成功后的视觉与 IMU 融合追踪两种追踪方式，在初始化成功后本文根据当前帧是否为关键帧而采用两种不同的优化目标函数的跟踪方式。如果当前帧是关键帧，跟踪的目标函数为当前帧的重投影误差项和当前帧与上一帧之间的 IMU 误差项；如果当前帧不是关键帧，那么跟踪的优化方程还多了之前的先验误差项。

(3) 本文提出了一种计算量更低局部地图优化方法和精度更高的闭环检测方法。局部地图由固定窗口和局部窗口组成，其中固定窗口仅提供局部地图中的约束，而局部窗口除了提供约束外，还作为优化变量。在局部优化中，本文根据时间上第二近的图像帧是否为关键帧而采用两种不同的边缘化方式。在局部地图优化迭代的过程中，本文采用了固定线性化点的方式求目标函数的雅克比。在闭环检测中，本文采用视觉词袋的方式来检测闭环，成功检测闭环后，本文采用的是 4 自由度的优化，而不是传统的 6 自由度或是 7 自由度优化。

(4) 本文提出的 SLAM 算法分别在 EuRoC 数据集和实物小车上做了测试。在 EuRoC 数据集上，本文提出的算法在初始化上表现十分突出，可以得到尺度 s 的误差小于 5%，重力矢量的大小收敛在 9.8 m/s^2 左右，加速度偏差收敛在 0 左右，陀螺仪偏差收敛在 $10^{-3} \text{ } ^\circ/\text{s}$ 。闭环检测上本文提出的 4 自由度优化要比传统的优化方式定位误差小。在平均值误差上本文提出的 SLAM 算法的平均误差在 8cm 左右，最大误差也在 10cm 以内，和目前最优秀的开源算法相比也是具有优势的。同时本文的 SLAM 算法也同双目 ORB_SLAM 算法进行对比，我们的算法在定位精度和收敛性上要占有绝对优势。本文提

出的 SLAM 算法在低价的实物小车上也可以得到较好的效果，但是由于硬件上的缺陷，相比数据集上的结果还是有一定的差距。

对后续研究工作的设想：

目前本文提出的单目视觉与 IMU 融合的 SLAM 算法计算量比较大，在未来的工作里还需要减少该算法的计算量。由于本算法为了得到高精度的初始化参数，因此在初始化上花费了大量的时间，未来工作会减少初始化时间。除此之外，该算法不支持双目与 IMU 的融合，在未来本作者会让该算法支持双目系统与 IMU 的融合。

参考文献

- [1] Smith R C, Cheeseman P. On the representation and estimation of spatial uncertainty[J]. International Journal of Robotics Research, 1987, 5:56-68.
- [2] Smith R, Self M, Cheeseman P. Estimating Uncertain Spatial Relationships in Robotics[M]// Autonomous robot vehicles. Springer-Verlag New York, Inc. 1990: 435-461.
- [3] Se S, Lowe D G, Little J J. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks[J]. International Journal of Robotics Research, 2002, 21(8): 735-760.
- [4] Paz L M, Jensfelt P, Tardos J D, et al. EKF SLAM updates in $O(n)$ with divide and conquer SLAM[J]. 2007: 1657-1663.
- [5] Sibley D, Mei C, Reid I, et al. Adaptive relative bundle adjustment[J]. Proceedings of Robotics Science & Systems, 2010: 976-982.
- [6] Hauke Strasdat, J.M.M. Montiel, Andrew J. Davison. Visual SLAM: Why filter? [J]. Image & Vision Computing, 2012, 30(2): 65-77.
- [7] 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, 29(6): 1052.
- [8] 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.
- [9] Leutenegger S, Furgale P, Rabaud V, et al. Keyframe-based visual-inertial SLAM using nonlinear optimization[C]// Robotics: Science and Systems. 2013: 789-795.
- [10] Smith R C, Cheeseman P. On the representation and estimation of spatial uncertainty[J]. International Journal of Robotics Research, 1987, 5:56-68.
- [11] Durrantwhyte H F. Uncertain geometry in robotics[J]. IEEE Journal on Robotics & Automation, 1988, 4(1): 23-31.
- [12] Ayache N, Faugeras O D. Building, registering, and fusing noisy visual maps[J]. International Journal of Robotics Research, 1988, 7(7): 45-65.
- [13] Crowley J L. World modeling and position estimation for a mobile robot using ultrasonic ranging[C]// IEEE International Conference on Robotics and Automation, 1989. Proceedings. IEEE, 1989: 674-680 vol.2.

- [14] Chatila R, Laumond J. Position referencing and consistent world modeling for mobile robots[C]// IEEE International Conference on Robotics and Automation. Proceedings. IEEE, 2014: 138-145.
- [15] Wang C M. Location estimation and uncertainty analysis for mobile robots[M]// Autonomous Robot Vehicles. Springer New York, 1990:1231-1235.
- [16] Matthies L, Kelly A, Litwin T, et al. Obstacle detection for unmanned ground vehicles: a progress report[C]// Intelligent Vehicles '95 Symposium. Proceedings of the. IEEE, 1995: 66-71.
- [17] 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.
- [18] Engel J, Schöps T, Cremers D. LSD-SLAM: Large-scale direct monocular SLAM[J]. 2014, 8690: 834-849.
- [19] Mur-Artal R, Tardós J D. ORB-SLAM: Tracking and mapping recognizable features[C]// The Workshop on Multi View Geometry in Robotics. 2014.
- [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] 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.
- [22] Mur-Artal R, Tardós J D. Visual-inertial monocular SLAM with map reuse[J]. IEEE Robotics & Automation Letters, 2016, 2(2): 796-803.
- [23] Yang Z, Shen S. Monocular visual-inertial fusion with online initialization and camera-IMU calibration[C]// IEEE International Symposium on Safety, Security, and Rescue Robotics. IEEE, 2015: 1-8.
- [24] Qin T, Li P, Shen S. VINS-Mono: A robust and versatile monocular visual-inertial state estimator[J]. 2017.
- [25] Dong Z, Zhang G, Jia J, et al. Efficient keyframe-based real-time camera tracking[J]. Computer Vision & Image Understanding, 2014, 118(2): 97-110.
- [26] 刘浩敏, 章国锋, 鲍虎军. 面向大尺度场景的单目同时定位与地图构建[J]. 中国科学: 信息科学, 2016(12).
- [27] 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.
- [28] Hartley R, Zisserman A. Multiple view geometry in computer vision. With foreword by Olivier Faugeras. 2nd edition[J]. 2003.
- [29] Bartoli A, Sturm P, Haraud R. Projective structure and motion from two views of a piecewise planar scene[C]// Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on. IEEE, 2001: 593-598 vol.1.
- [30] Lowe D G. Distinctive image features from scale-invariant keypoints[J]. International Journal of Computer Vision, 2004, 60(2): 91-110.
- [31] Bay H, Ess A, Tuytelaars T, et al. Speeded-up robust features[J]. Computer Vision & Image Understanding, 2008, 110(3): 404-417.
- [32] Rublee E, Rabaud V, Konolige K, et al. ORB: An efficient alternative to SIFT or SURF[C]// IEEE International Conference on Computer Vision. IEEE, 2012: 2564-2571.
- [33] Carlevaris-Bianco N, Kaess M, Eustice R M. Generic node removal for factor-graph SLAM[J]. IEEE Transactions on Robotics, 2017, 30(6): 1371-1385.
- [34] Mazuran M, Burgard W, Tipaldi G D. Nonlinear factor recovery for long-term SLAM[J]. International Journal of Robotics Research, 2016, 35.
- [35] Huang G, Kaess M, Leonard J J. Consistent sparsification for graph optimization[C]// European Conference on Mobile Robots. IEEE, 2013: 150-157.
- [36] Boyd S, Parikh N, Chu E, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers[J]. Foundations & Trends in Machine Learning, 2010, 3(1): 1-122.
- [37] Burri M, Nikolic J, Gohl P, et al. The EuRoC micro aerial vehicle datasets[J]. International Journal of Robotics Research, 2016, 35(10): 1157-1163.
- [38] Horn B K P. Closed-form solution of absolute orientation using unit quaternions[J]. Journal of the Optical Society of America A, 1987, 4(4): 629-642.
- [39] Lin Y, Gao F, Qin T, et al. Autonomous aerial navigation using monocular visual - inertial fusion[J]. Journal of Field Robotics, 2017(4).
- [40] Engel J, Koltun V, Cremers D. Direct sparse odometry[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2016, PP(99):1-1.

- [41] Galvez-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.
- [42] Mur-Artal R, Tardós J D. Fast relocalisation and loop closing in keyframe-based SLAM[C]// IEEE International Conference on Robotics and Automation. IEEE, 2014: 846-853.
- [43] Caruso D, Engel J, Cremers D. Large-scale direct SLAM for omnidirectional cameras[C]// IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2015: 141-148.

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《基于非线性优化的单目视觉与IMU融合的SLAM算法研究》，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：李庆峰

日期：2017年01月10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：李庆峰

日期：2017年1月10日

导师签名：孟庆尧

日期：2017年1月10日

攻读硕士学位期间发表的论文及其他成果

(一) 发表的学术论文

- [1] Qingfeng Li, Hao Gu, Cuihong Han, Weimeng Gong, Shuang Song, Max Q. -H. Meng. Robust visual inertial monocular using nonlinear optimization[C]. International Conference on Information and Automation(ICIA), 2017:483-488.
- [2] Weimeng Gong, Hao Xu, Qingfeng Li, Hao Gu, Cuihong Han, Shuang Song, Max Q. -H. Meng. Mobile robot manipulation system design in given environments[C]. International Conference on Information and Automation(ICIA), 2017:609-613.

致谢

衷心感谢导师孟庆虎教授对本人的悉心指导。他的前瞻远瞩给我坚定不移完成课题的信心，他的支持和鼓励是我在研究过程中克服困难的动力，他的教导将使我终生受益。

感谢宋霜老师，在我课题研究及论文撰写的过程中都给予了宝贵的建议和帮助，并对课题进度进行了必要的督促。感谢徐文福老师的学位论文撰写规范讲座，使我轻松地组织论文结构，调整逻辑，规范格式。感谢吴晓君老师在课题中给予的关心和鼓励。

感谢实验室同窗们的热情支持和帮助，我们是同门，更是一个温暖的集体。感谢舍友为我细心地检查格式错误。

感谢家人遥远的牵挂，对我生活上的关心，让我踏踏实实地投入到科研之中。