

硕士学位论文

融合 IMU 信息的双目视觉 SLAM 研究

**RESEARCH ON BINOCULAR VISION SLAM
WITH IMU INFORMATION FUSION**

徐宽

哈尔滨工业大学

2018 年 6 月

国内图书分类号: U469.72

学校代码: 10213

国际图书分类号: 656

密级: 公开

工学硕士学位论文

融合 IMU 信息的双目视觉 SLAM 研究

硕 士 研 究 生:	徐宽
导 师:	刘志远教授
申 请 学 位:	工学硕士
学 科:	控制科学与工程
所 在 单 位:	航天学院
答 辩 日 期:	2018 年 6 月
授 予 学 位 单 位:	哈尔滨工业大学

Classified Index: U469.72

U.D.C: 656

Dissertation for the Master Degree in Engineering

**RESEARCH ON BINOCULAR VISION SLAM
WITH IMU INFORMATION FUSION**

Candidate:	Xu Kuan
Supervisor:	Prof. Liu Zhiyuan
Academic Degree Applied for:	Master of Engineering
Speciality:	Control Science and Engineering
Affiliation:	School of Astronautics
Date of Defence:	June, 2018
Degree-Conferring-Institution:	Harbin Institute of Technology

摘 要

随着人工智能和机器人技术不断发展，机器人从原来的主要面向工业应用不断的走入人们的生活，为大众做熟知。其中，服务机器人和物流机器人很多已经商用，相信在不久的将来，机器人将在人们的生活中扮演越来越重要的角色。对于移动机器人，定位一直是其的一个核心能力，而本文的目的便是借助 IMU（Inertial measurement unit，惯性测量单元）的信息提升视觉 SLAM（simultaneous localization and mapping，即时定位与地图构建）的定位性能。

首先，对于视觉信息的处理部分，本文分别综述了相机的针孔模型，双目视觉模型等双目视觉的基础知识。然后又介绍了 FAST 特征点、BRIEF 描述符和 ORB 特征点。之后又介绍了词袋模型如何用到回环检测，以及如何如何进行回环的校正。

其次，对于 IMU 信息的处理部分，为了使 IMU 的频率和图像信息的频率一致，本文建立了 IMU 的噪声模型和运动学模型，对 IMU 的测量量进行了预积分，并推导了 IMU 测噪声传播方程，分析了在 IMU 的随机游走更新的时候 IMU 预积分量更新的方式。

然后，为了利用两个传感器的信息估计位姿，本文利用非线性优化的方法融合了 IMU 信息和视觉信息。之后还给出了一种对地图进行保存的方法，以及如何利用地图进行定位。

最后，针对提出的算法设计并进行了实验。其中，标定实验标定出了双目的内参和外参，相机 IMU 联合标定实验标定出了相机和 IMU 的外参。利用 Euroc 数据集的实验对比了本文算法的定位结果和真实的 ground truth 之间的误差。之后还利用了小觅双目相机对作者周围的环境进行了建图和定位的实验，都取得了良好的结果。

关键词：定位；视觉 SLAM；惯性测量单元；信息融合；地图保存和复用

Abstract

With the development of artificial intelligence and robotics technology, robots play a more and more important rule in people's lives. For mobile robots, localization has always been a core capability. The purpose of This article is to use the IMU (Inertial Measurement Unit) to improve the performance of the visual SLAM (simultaneous localization and mapping).

First of all, this article introduces the camera's pinhole model, binocular vision model, binocular distance measurement principle and other basic knowledge of binocular vision. Then FAST feature points, BRIFF descriptors, and ORB feature points is introduced. Afterwards, the method of loop detection is introduced.

Secondly, this article establishes a noise model and a kinematics model for the IMU, and pre-integrates the measured quantities of the IMU. And the noise propagation equation is derived and the influence of random walk is analyzed.

Then, in this paper, IMU information and visual information are combined using nonlinear optimization methods A method of saving the map and how to use the map for more precise and fast localizatio is also given later .

Finally, experiments are designed and conducted based on the proposed algorithm. Among them, the calibration experiment calibrated the parameters of camera and the external parameters of the camera and the IMU. Experiments using the Euroc dataset compare the error between the localization results of this algorithm and the ground truth. Finally, the map of some real scene around us is build using the mynt-eye binocular camera.

Keywords: localization, visual SLAM, IMU, information fusion, map saving and reuse

目录

摘 要.....	I
Abstract.....	II
目录.....	III
第 1 章 绪论.....	1
1.1. 课题背景与意义.....	1
1.2. 国内外研究现状.....	2
1.2.1. 视觉 SLAM 研究现状.....	2
1.2.2. 视觉 SLAM 和 IMU 信息融合研究现状.....	4
1.3. 本文主要研究内容与组织架构.....	6
第 2 章 视觉信息处理.....	9
2.1. 引言.....	9
2.2. 相机模型.....	9
2.2.1. 相机投影模型.....	9
2.2.2. 相机畸变模型.....	11
2.3. 特征点检测和描述.....	12
2.3.1. FAST 特征点.....	12
2.3.2. BRIEF 特征描述子.....	13
2.3.3. ORB 特征.....	14
2.4. 立体视觉模型.....	16
2.4.1. 双目测距原理.....	16
2.4.2. 迭代最近点法.....	17
2.5. 回环检测回环校正.....	19
2.5.1. 词袋模型和视觉字典.....	19
2.5.2. 回环检测.....	21
2.5.3. 回环校正.....	22
2.6. 本章小结.....	25

第 3 章 IMU 信息预处理.....	26
3.1. 引言.....	26
3.2. IMU 建模.....	26
3.2.1. IMU 噪声模型.....	26
3.2.2. IMU 运动学模型.....	27
3.3. IMU 预积分.....	28
3.3.1. 预积分 IMU 测量值.....	29
3.3.2. 噪声的传播.....	32
3.3.3. 随机游走 bias 的更新.....	34
3.4. 本章小结.....	37
第 4 章 视觉和 IMU 信息融合.....	38
4.1. 引言.....	38
4.2. 系统初始化.....	38
4.2.1. 估计初始的陀螺仪 bias.....	38
4.2.2. 估计初始速度和重力向量.....	39
4.3. 非线性优化.....	41
4.3.1. 状态量和误差项.....	41
4.3.2. IMU 残差项的雅克比矩阵.....	43
4.3.3. LM 算法.....	46
4.4. 地图保存和复用.....	48
4.4.1. 地图保存.....	49
4.4.2. 地图复用.....	50
4.5. 本章小结.....	52
第 5 章 实验结果与分析.....	53
5.1. 引言.....	53
5.2. 标定实验.....	53
5.2.1. 实验平台.....	53
5.2.2. 双目相机标定.....	54
5.2.3. 相机和 IMU 联合标定.....	56

5.3. 使用 EUROC 数据集定位和建图.....	58
5.4. 使用小觅双目相机定位和建图.....	62
5.5. 本章小结.....	63
结论.....	64
参考文献.....	65
哈尔滨工业大学学位论文原创性声明和使用权限.....	69
致 谢.....	70

第 1 章 绪论

1.1. 课题背景与意义

近年来，人工智能技术不断发展，也带动了机器人产业的发展，使得机器人从原来的主要面向工业应用逐渐向物流、服务等贴近人们生活的方向发展。对于人们生活中的机器人，主要以轮式的小型移动机器人为主。这些机器人有行驶在室外的：如物流末端配送机器人、自动驾驶汽车等，也有行驶在室内的：如酒店的服务机器人、家庭的扫地机器人等。相信随着技术的不断发展，这些机器人会在我们的生活中扮演越来越重要的角色。

对于轮式的移动机器人，定位一直是其的一个核心能力。在室外，可用于定位的方法比较多，比如惯导、GPS（Global Positioning System，全球定位系统）、高精度地图等。而对于室内高精度地图和 GPS 无法工作，低精度的惯导设备难以满足精度的要求，高精度的惯导又价格昂贵，并且还有累计误差。UWB（Ultra Wideband）虽然可以全局定位，但是价格昂贵，难以普及，而利用 Wi-Fi、蓝牙等技术定位虽然价格低廉，但是精度过于粗糙，难以用于机器人。所以目前的室内移动机器人，使用的主流定位方法大多都是 SLAM 技术，而 SLAM 依据传感器的不同，又分为视觉 SLAM 技术和激光雷达 SLAM 技术。

相比于视觉 SLAM，激光雷达 SLAM 的发展起步较早，目前也更成熟一些，精度也相对较高，但是成本比较高，并且缺少语义信息。而视觉 SLAM 利用单目或者双目相机进行地图的构建和定位，成本低，信息量大，虽然目前发展的还不是很完善，但是拥有广阔的发展前景。

视觉 SLAM 一个明显的缺点就是鲁棒性能不太好，当遇到特征不够丰富或者光照情况不好的情况或者当机器人运动速度过快时都会出现跟丢的情况。而本次工作的目的便是结合 IMU 的信息去提升视觉 SLAM 的鲁棒性。IMU 则输出频率高，在快速移动和特征不丰富情况下依然能保持良好的测量效果，并且对光照不敏感，在一定程度上弥补了视觉传感器的缺点，而其自身的漂移问题则能够由视觉传感器去弥补。所以近年来视觉+惯导 slam 的方案受到越来越多的重视和研究。

1.2. 国内外研究现状

1.2.1. 视觉 SLAM 研究现状

在 2003 年的时候, A.J.Davision 提出的 MonoSLAM 是第一个实时的单目视觉 SLAM 系统^[1], MonoSLAM 前端跟踪稀疏的特征点, 后端使用扩展卡尔曼滤波, 以相机当前状态和所有路标点的世界坐标为状态量, 更新均值和协方差矩阵。虽然现在看来 MonoSLAM 有诸多缺点, 也难以直接运用到实际的场景中, 但是它是一个里程碑式的工作, 为视觉 SLAM 的实时化做出了巨大的贡献。之后, 在 2007 年, PTAM^[2] (Parallel Tracking and Mapping) 被 Kleun 等人提出。PTAM 主要的贡献有两点: 一是它创造性的将跟踪和建图过程放在两个线程里面并行的进行, 这大大提升了程序的运行效率; 二是它第一个使用了非线性优化作为后端, 现在已经有工作证明, 使用非线性优化作为后端的视觉 SLAM 系统在相同计算量的前提下精度要优于使用卡尔曼滤波作为后端的视觉 SLAM 系统。另外, PTAM 还引入了关键帧的机制, 使得算法不用对每帧图像都进行精细的处理, 这也大大提升了算法的效率。自 PTAM 之后, 以滤波为后端的视觉 SLAM 研究逐渐以非线性优化的后端为主导。但是, 在现在看来, PTAM 也存在一些比较明显的缺点, 比如它更适合小场景, 容易跟丢等, 不过以历史角度来看, PTAM 仍然是一个伟大的系统。

在 2014 年, J.Engle 提出了 LSD-SLAM^[3] (Large Scale Direct monocular SLAM), 它使用直接法进行半稠密地图的重建。在 LSD-SLAM 中, 作者提出了像素梯度和直接法的关系, 也提出了像素梯度和极线方向的角度关系。另外, LSD-SLAM 做出的另一个成就是它仅仅依靠 CPU 就实现了半稠密场景的重建, 在它之前很少有方案能够只用 CPU 就重建出稠密的地图。并且, 在进行半稠密跟踪的时候, LSD-SLAM 没用利用单个像素或者图像块, 而是在极线上等距离的取 5 个点, 度量它们的 SSD; 在估计深度的时候, LSD-SLAM 先用随机数初始化深度, 之后又把深度均值归一化, 调整尺度。在进行深度不确定性度量的时候, LSD-SLAM 不仅考虑了三角化的几何关系, 还考虑了极线和深度的夹角, 将其归纳成了一个深度不确定项。另外, LSD-SLAM 的关键帧之间的约束使用了相似变换群和李代数表达出尺度, 在后端优化的时候可以将不同尺度的场景考虑进来, 减小了尺度的漂移现象。上面的一些优点使得 LSD-SLAM 成为了视觉 SLAM 历史上又一里程碑式的

工作。

同样是在 2014 年, Forster 等人提出了 SVO^[4] (Semi-direct Visual Odoemtry), 它是一种基于稀疏直接法的视觉里程计。SVO 将直接法和特征点混合使用: SVO 跟踪了一些没有描述子的角点, 然后利用直接法的思想, 根据这些角点周围的信息估计相机的运动。在具体实现中, SVO 使用角点周围边长为 4 的正方形小块进行块匹配, 求解相机的运动。由于 SVO 不用计算角点的描述子, 也不用向稠密和半稠密的方案那样处理那么多信息, 甚至它还舍弃了后端优化和回环检测, 所以 SVO 的速度极快, 非常适合向无人机、AR 设备这种计算能力不高的场合。另外, SVO 推导出了基于高斯混合分布的深度滤波器, 这是首次提出深度滤波器的概念, 利用深度滤波器, SVO 能够更好的计算特征点位置。

对于 RGB-D 的方案, Mathieu Labbe 等人于 2014 年提出了 RTAB-MAP^[5] (Real Time Apperance-Based Mapping)。RTAB-MAP 是一个非常完整的 RGB-D SLAM 方案, 它实现了基于特征点的前端、后端的位姿图优化、基于词袋模型的回环检测以及点云和三角网格地图, 在 ROS 上可以直接获得其二进制程序。但是由于 RTAB-MAP 的集成度比较高, 使得在其基础上进行二次开发比较困难。

在 2015 年, Raul Mur-Artal 等人提出了 ORB SLAM^[6], 之后又提出了 ORB SLAM2^[7], ORB SLAM 是一个比较现代一些的 SLAM 系统, 由于它非常完善, 精度很不错, 并且代码清晰易读, 简单易用, 便于二次开发, 使得 ORB SLAM 成为了一个非常有名的视觉 SLAM 系统。相比其它视觉 SLAM 系统, ORB SLAM 有如下优点: 首先它既支持单目相机, 同时也支持双目和 RGBD 相机, 具有良好的泛用性。其次它前端的跟踪, 后端的局部地图优化和回环检测使用的都是 ORB 特征点^[8], 提升了特征点的利用率。另外, ORB SLAM 的回环检测做的很好, 它使用词袋模型和 ORB 视觉字典, 在跟丢了之后能够迅速的找回。类似但不同于 PTAM, ORB SLAM 采用三线程的模式, 其中, Tracking 线程进行特征点的检测、求当前帧的初始位姿、插入关键帧和插入地图点等工作; LocalMapping 线程进行局部地图优化, 其优化的是共视图 (Covisibility Graph) 的相机位姿和地图点在世界坐标系下的坐标等; LoopClosing 线程进行回环检测和全局的优化, 这个线程优化的是本质图 (Essential Graph), 会对全局的地图进行优化, 消除累计误差, 但是这个线程的优化只优化相机的位姿, 不优化地图点的坐标。另外, ORB SLAM 围绕

特征点也做了很多优化，它在特征点提取之后利用四叉树对特征点进行分散处理，使得特征点能够更均匀的分布在整张图像上。上面的这些优势使得 ORB SLAM 被广泛的应用和二次开发。但是，ORB SLAM 也存在一些不足，比如 ORB 特征点的检测比较耗费计算量，非常容易跟丢等。

除此之外，还有一些工作诸如 DVO-SLAM^[9]，DSO^[10]，DTAM^[11]等也都取得了不错的结果，为视觉 SLAM 的发展做出了巨大的贡献。

1.2.2. 视觉 SLAM 和 IMU 信息融合研究现状

对于视觉 SLAM 和 IMU 信息融合的方案研究，目前主要分为两种：松耦合和紧耦合^[12]。松耦合是指相机和 IMU 先分别估计自身的运动，然后对其位姿估计的结果进行融合，而紧耦合是指利用 IMU 和相机的状态共同构建运动方程和观测方程，然后进行位姿的估计。而如果按融合算法去分的话，主要分为以扩展卡尔曼滤波^[13]为后端的系统和以非线性优化为后端的系统。

在 2007 年，Mourikis 等人提出了 MSCKF^[14]算法，MSCKF 维持了一个滑动窗口，以扩展卡尔曼滤波为后端，取得不错的效果。传统的以扩展卡尔曼滤波为后端的 SLAM 和 IMU 进行信息融合时，在每个时刻的状态向量都会保存当前帧的位姿、速度、地图点的三维坐标以及 IMU 的随机游走等，然后用 IMU 做预测，再优化图像的重投影误差。而 MSCKF 则利用多帧的信息，优化最近多帧图像的重投影误差，所以取得更好的效果。在 2013 年，Mingyang Li 提出了 MSCKF2.0^[15]的算法，相比于 MSCKF，MSCKF2.0 指出了在 VIO（Visual-Inertial Odometry）系统中 yaw 轴角的不可观性，并修改了线性系统模型中雅克比矩阵的计算方式。另外 MSCKF2.0 还将 IMU 坐标系和相机坐标系的外参进行在线的标定，取得了比 MSCKF 更好的效果。

在 2015 年 Michael Bloesch 等人提出了 ROVIO^[16]（Robust Visual Inertial Odometry Using a Direct EKF-Based Approach）。ROVIO 是一个单目 VIO 系统，它前端使用直接法，通过最小化图像块的光度误差，取得了不错的跟踪精度和鲁棒性。对于后端，ROVIO 使用扩展卡尔曼滤波。ROVIO 将 3D 地图点表示在当前帧的相机坐标系下，并且其将地图点的三维坐标用一个单位向量和一个距离参数去表示，这样做改善了计算性能。值得一提的是，ROVIO 不需要向其他 VIO 方案那样需要一个初始化的过程，这得益于其地图点的表示形式。但是，ROVIO 只是一个里程计，其并没有回环检测和回环校正的过

程，使得其存在累计误差。

同样是在 2015 年，Stefan Leutenegger 等人提出了 OKVIS^[17]。OKVIS 是一个支持双目 SLAM 融合 IMU 信息的方案，它使用 SSE 优化的角点为特征点，使用 BRISK^[18]描述符作为描述子，然后通过特征点匹配的方法匹配这些特征点。OKVIS 匹配特征点的方法有两种：3D-2D 匹配和 2D-2D 匹配。所谓 3D-2D 匹配就是通过 IMU 的数据和上一帧的状态对当前帧的状态进行预测，利用预测的当前帧的位姿可以获得当前帧可以观测到的地图点，将这些地图点和当前帧检测到的特征点做暴力匹配，即计算当前帧的特征点和可见地图点特征描述子之间的距离，通过这个距离来确定匹配关系。之后通过计算 Mahalanobis 距离和 RANSAC^[19]（Random Sample Consensus，随机抽样一致）算法来消除误匹配。而 2D-2D 匹配则是通过暴力匹配的方法得到初始的匹配关系，然后三角化特征点，通过计算其三维坐标消除一些误匹配。另外，OKVIS 使用了关键帧的机制，当检测到特征点的区域占图像总面积不到 50% 或者匹配到的特征点占检测到的特征点总数不到某一阈值的时候，会将当前帧选作关键帧。对于视觉和 IMU 信息融合，OKVIS 对 IMU 进行了预积分^[20]，并使用非线性优化的方法优化视觉的重投影误差和 IMU 残差。另外 OKVIS 还有在线标定的功能，可以在线标定 IMU 坐标系和相机坐标系的旋转矩阵和平移矩阵。上面的这些特点使得 OKVIS 取得了非常不错的性能。但是 OKVIS 没有回环检测和回环校正的功能，使得其累计误差会随着里程的增加而不断增加。

在 2017 年，ORB SLAM 的作者 Raul Mur-Artal 等人提出了融合 IMU 信息的 VI ORB SLAM^[21]。其在单目的 ORB SLAM 的基础上利用 IMU 预积分和非线性优化融合 IMU 信息，并在局部地图优化的时候也会将 IMU 项加入优化。从作者给出的实验结果来看，融合 IMU 信息的单目 ORB SLAM 比原来精度提高了很多，但是不同于 ORB SLAM 和 ORB SLAM2，作者并没有将代码开源。

同样是在 2017 年，沈劭劼等人提出了 VINS-Mono^[22]。VINS-Mono 是一个单目 SLAM 融合 IMU 的方案，其前端使用最简单的角点检测，并使用 KLT 光流法对这些角点进行跟踪。在选取关键帧上，VINS-Mono 有两个标准：一是通过平均视差，如果某一帧中跟踪到的特征点的平均视差大于某一阈值，则将该帧作为关键帧。不过需要说明的是，相机的平移和旋转都会造成视差，但是纯旋转不能三角化特征点，所以为了避免这种情况，在计算视差的时候

VINS-Mono 会使用 IMU 的测量结果去补偿旋转带来的视差。第二个标准是跟踪特征点的数量，如果某一帧跟踪到的特征点数量低于某一阈值，则将该帧当做一个关键帧。在对 IMU 信息的处理上，VINS-Mono 也采用的是预积分的方法，然后利用非线性优化的方法将 IMU 信息和视觉信息进行融合。另外，VINS-Mono 还有在线标定和回环检测的功能，除了用于跟踪的角点，其还会检测出 500 个 FAST 角点^[23]，使用 BRIEF^[24]描述子进行描述，使用词袋模型的方法进行回环的检测，取得了非常不错的效果。

在 2017 年末，苏黎世联邦理工学院的 ASL 实验室开源了 maplab^[25-38]，这是一个非常完整且庞大的 SLAM 框架。它基本实现 SALM 所需要的所有功能：融合 IMU 信息的前端，后端优化，离线稠密地图重建，多设备协作建图，地图可视化等。除此之外，maplab 还能利用保存的地图进行进行更精准的定位，是目前视觉 SLAM 的集大成者。

1.3. 本文主要研究内容与组织架构

本文主要研究双目视觉 SLAM 和 IMU 信息融合的问题，为了更好能够将双目视觉的信息和 IMU 的信息融合，本文的主要研究了下面几部分内容：

(1) 双目视觉 SLMA 的基础理论。包括双目视觉的模型，标定方法，特征点检测和匹配的方法，以及利用匹配到的特征点如何求解两帧的图像的相对位姿。以及视觉 SLAM 的基本算法，包括关键帧和地图点的选取方法，回环检测和回环校正的方法等；

(2) 视觉 SLAM 和 IMU 信息的融合。本文运用了 IMU 预积分的技术，利用非线性优化的方法将 IMU 预积分的信息和视觉特征点的信息进行了融合。虽然本文是在 ORB SLAM 的基础上进行工作的，但是不同于 ORB SALM 作者提出的融合 IMU 的算法，本文使用的是双目融合的 IMU，并且考虑到 IMU 的引入可能导致数值优化的不稳定性，本文并没有过多使用 IMU 的信息，仅仅是利用 IMU 进行了预测和初步的跟踪，并没有行 ORB SLAM 作者那样在 Local Mapping 线程的优化中也使用 IMU 的信息。这样做的目的是降低对 IMU 精度的要求，使得使用廉价的 IMU 也能取得很好的效果，在保证 ORB SLAM 系统精度的前提下提升其跟踪的鲁棒性。

(3) 地图的保存和复用。本文提出了一种保存地图和复用地图的方法，通过地图的保存和复用，使得系统定位得更加准确，同时还节省了计算量。

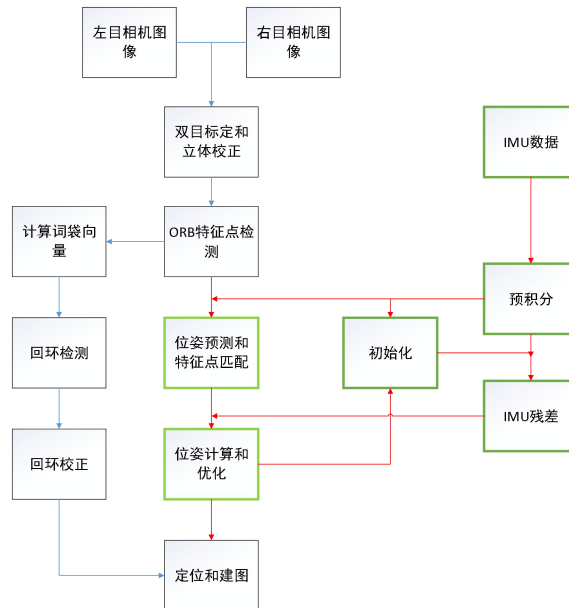


图 1-1 系统结构图

本次系统的结构如上图 1-1 所示。本文的具体章节安排如下：

第一章主要介绍了视觉 SLAM 以及视觉 SLAM 与 IMU 信息融合的意义，以及它们目前的研究现状，重点介绍几个比较著名的视觉 SLAM 方案和融合 IMU 信息的视觉 SLAM 方案，包括它们融合 IMU 的方式等。

第二章则介绍本次工作所用到的双目视觉部分的基础知识。包括相机的模型，双目视觉的基础知识，特征点检测和匹配的方法，利用匹配到的 3D 点进行位姿解算，还用回环检测和回环校正的算法，词袋模型等等。

第三章则主要介绍对 IMU 数据的处理方法。包括对 IMU 建立噪声模型和运动学模型，对 IMU 的测量数据进行预积分，推导 IMU 误差传播的方式，分析的 IMU 随机游走对预积分量的影响等等。

第四章主要介绍了如何利用非线性优化的方法融合视觉和 IMU 的信息。包括对状态量和误差项的定义，如何利用高斯牛顿法或者 LM 算法对误差项进行优化。除此之外还介绍了如何利用 boost 库序列化的方法对建立的地图进行保存，以及如何利用保存的地图进行二次定位等。

第五章主要设计并进行了实验。实验包括三个部分，第一个是标定实验，包括对双目相机的标定和相机与 IMU 的联合标定。第二个是利用 Euroc 数据集^[39]进行的实验，Euroc 数据集由 ground truth，所以这部分的实验会给出本

次工作算法的误差信息。第三部分实验则是利用廉价的双目相机和廉价的 IMU 进行实际室内外实验，这部分实验中会给出建图的结果。

第 2 章 视觉信息处理

2.1. 引言

因为本次工作是通过非线性优化的方法优化视觉的重投影误差和 IMU 残差来获得某一帧图像时刻的位姿的，并且在初始化的时候需要单独的视觉解算出位姿，所以视觉部分承担的任务主要包括特征点的检测，特征点的匹配以及对特征点匹配良好的两帧图像进行位姿解算。本章将从相机模型、双目视觉、特征点以及回环检测四个部分进行叙述，为后续的视觉与 IMU 融合、相机标定以及视觉与 IMU 联合标定等工作奠定基础。

2.2. 相机模型

对于相机模型的介绍包括两部分：相机的投影模型和相机的畸变模型。相机的投影模型是指相机如何将相机坐标系下的点投影到图像坐标上，即无相机畸变影响下的图像坐标系与相机坐标系之间的函数关系。但是受相机制作工艺的影响，真实相机的相机坐标系和图像坐标系的函数关系往往不是理想的相机模型所描述的那样的函数关系，这部分的差别主要是由于畸变引起的，下面将对相机的投影模型和畸变模型展开叙述。

2.2.1. 相机投影模型

在本次工作中，使用的相机模型为小孔成像模型：

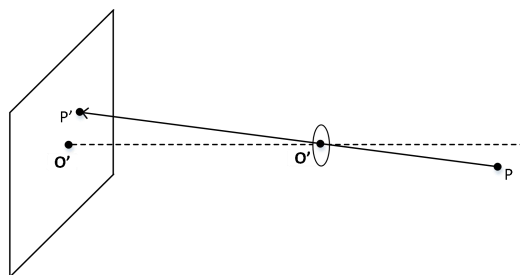


图 2-1 小孔成像示意图

如上图 2-1 所示， O 称为相机的光心， OO' 称为相机的光轴，空间上的一点通过相机的光心投影到成像平面上。在投影过程中，涉及几个坐标系的概念，下面对这些坐标系做一下介绍：

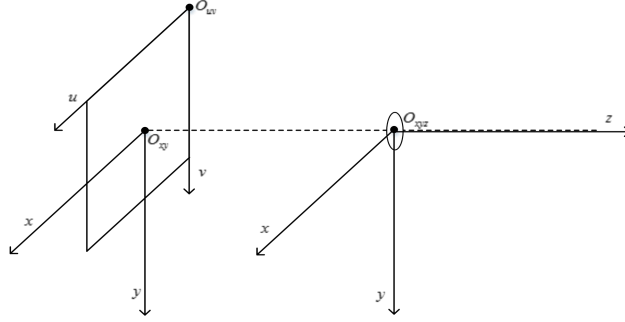


图 2-2 相机模型中使用的坐标系

(1) 像素坐标系：像素坐标系建立在获取的数字图像上，以左上角的像素点为坐标原点， x 轴正方向向右， y 轴正方向向下，一个分离的像素值为一个单位，通常用 (u,v) 表示像素坐标系下的像素坐标，图 2-2 中 $O_{uv} - uv$ 为像素坐标系。

(2) 图像坐标系：图像坐标系也建立在图像上，只不过图像坐标系的原点放在了图像的中心，即光轴与图像的交点上。 x 轴和 y 轴的正方向和像素坐标系相同，并且以实际的毫米(mm)为单位，通常以 (x,y) 表示图像坐标系下的坐标，图 2-2 中 $O_{xy} - xy$ 为图像坐标系。

(3) 相机坐标系：相机坐标系固连在相机上，以相机的光心为坐标原点， x 轴正方向向右， y 轴正方向向下， z 轴正方向向前，以实际的物理单位为单位，通常以 (X_C, Y_C, Z_C) 表示相机坐标系下的坐标，图 2-2 中 $O_{xyz} - xyz$ 为相机坐标系。

在相机的针孔模型下，像素坐标系和图像坐标系下的坐标转换关系用下面式子 (2-1) 描述：

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (2-1)$$

上式 (2-1) 中的 dx 和 dy 分别表示一行或者一列中一个像素代表的实际毫米数， (u_0, v_0) 表示光轴和图像平面交点的像素坐标。

图像坐标系和相机坐标系下的坐标转换关系用下面式子 (2-2) 描述：

$$Z_c \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} \quad (2-2)$$

上式中的 f 表示实际的焦距值。将上面两个式子结合起来可以得到像素坐标系下的像素坐标与相机坐标系下的坐标的相互转换关系，如式子 (2-3) 所示：

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} \frac{f}{dx} & 0 & u_0 \\ 0 & \frac{f}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2-3)$$

上式中 f_x 和 f_y 分别表示以行像素和列像素为单位的焦距值。

2.2.2. 相机畸变模型

如果没有畸变的影响，一般相机的投影关系将完全可以由 2.2.1 中介绍的小孔模型去描述。但是，由于相机的镜头在径向方向上的曲率是不规则变化的，投影的图像会有些扭曲，这种现象的畸变称为径向畸变。另外，由于用于成像的 CCD 或者 CMOS 所在的平面不可能完全安装的与相机光轴垂直，距离光心远近不同的区域成像的大小也会稍有差别，这种现象的畸变称为切向畸变，下面分别对这两种畸变进行介绍。

对于径向畸变，每个相机的畸变程度是不同。一般来讲，鱼眼相机的径向畸变比较明显，比一般相机要大。通常用三个参数去描述径向畸变：

$$\begin{aligned} x_0 &= x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \\ y_0 &= y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \end{aligned} \quad (2-4)$$

式子 (2-4) 的坐标是在图像坐标系下坐标， (x, y) 为无畸变情况下坐标， (x_0, y_0) 为有畸变情况下的坐标， r 为点 (x, y) 到原点的距离，即有： $r^2 = x^2 + y^2$ 。 k_1 ， k_2 ， k_3 为三个径向畸变参数，通常这三个参数可以由标定步骤得到。通常来讲，对于鱼眼或者广角相机，这三个参数都要使用，对于一般的非鱼眼、非广角的相机，只使用前两个参数即可。

对于切向畸变，每个相机的畸变程度也是不同的。通常用两个畸变参数去描述切向畸变：

$$\begin{aligned} x_0 &= x + 2p_1xy + p_2(r^2 + 2x^2) \\ y_0 &= y + 2p_2xy + p_1(r^2 + 2y^2) \end{aligned} \quad (2-5)$$

上式中的 p_1 和 p_2 是两个切向畸变参数，通常可以由标定步骤得到。

将上面的径向畸变和切向畸变综合考虑进来，便可以得到相机的畸变模型：

$$\begin{aligned} x_0 &= x(1 + k_1r^2 + k_2r^4 + k_3r^6) + 2p_1xy + p_2(r^2 + 2x^2) \\ y_0 &= y(1 + k_1r^2 + k_2r^4 + k_3r^6) + 2p_2xy + p_1(r^2 + 2y^2) \end{aligned} \quad (2-6)$$

描述畸变的参数一共有 5 个： $[k_1 \ k_2 \ p_1 \ p_2 \ k_3]$ 。

2.3. 特征点检测和描述

计算机视觉经过多年的发展，目前已经有了多种检测特征点的方法，如 Harris 角点^[40]、FAST 特征点、ORB 特征点、SURF 特征点^[41]、SIFT 特征点^[42]等等。通常来讲针对不同的用途，要选取不同的特征点方案。由于 SURF 特征点和 SIFT 特征点的检测过于耗时，难以达到实时检测，而 Harris 角点和 FAST 特征点缺少旋转不变性和尺度不变性，所以本次工作选取的是 ORB 特征点。ORB 特征点是在 FAST 特征点和 BRIEF 特征描述子的基础上添加了尺度和方向的描述，所以本节首先会介绍一下 FAST 特征点检测方法以及 BRIEF 特征描述子，然后再叙述 ORB 做的一些优化，最后给出怎么利用特征点描述子匹配特征点。

2.3.1. FAST 特征点

概括来讲，如果一个像素点与周围邻域内足够多的像素点相差足够大的话，该像素点便会被检测成一个 FAST 特征点。具体的检测步骤如下：

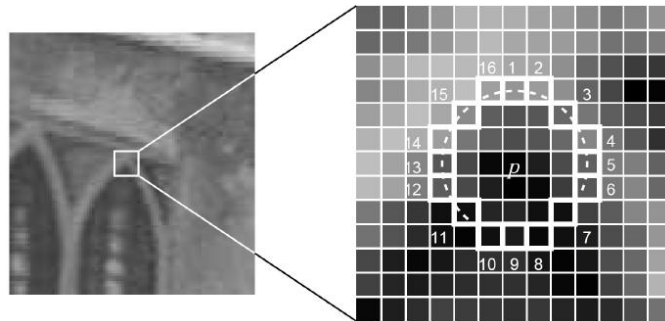


图 2-3 FAST 特征点检测

(1) 首先, 对于一帧图像上的某一个点 p , 以其为圆心, 以三个像素的长度为半径画出一个圆, 并按图 2-3 的方式把圆上的像素点编号 $p_1, p_2 \dots p_{16}$, 对某一个像素点 p_i , 把其像素值即为 I_{p_i} 。

(2) 比较编号为 1、5、9、13 的像素点和圆心像素点的像素值差。为了度量像素值的差, 定义一个阈值 ΔI , 如果这四个像素点与圆心像素点 p 差值的绝对值超过阈值的数量大于等于 3, 则认为点 p 是一个候选的 FAST 特征点, 继续进行下面的步骤, 否则点 p 不是一个 FAST 特征点。

(3) 继续比较点 p 与圆上 16 个像素点的像素差值, 如果有至少连续 9 个像素点与点 p 的像素差值绝对值超过阈值 ΔI , 则认为点 p 是一个 FAST 特征点, 继续进行后面的步骤, 否则点 p 不是一个 FAST 特征点。

(4) 最后一步被称作非极大值抑制, 主要目的是为了防止多个 FAST 特征点“聚集”的现象, 主要方法是在一个小区域内只保留“最优”的特征点, 抑制其它次优的特征点。为了对特征点进行度量, 用如下式子去计算一个特征点的得分:

$$S = \max \begin{cases} \sum_{i=1}^{16} p_i - p & \text{if } (p_i - p) > \Delta I \\ \sum_{i=1}^{16} p - p_i & \text{if } (p - p_i) > \Delta I \end{cases} \quad (2-7)$$

计算得到点 p 的得分之后, 将其与一个邻域内其它 FAST 特征点进行比较, 如果点 p 的得分是最高的, 则保留, 否则抑制。

上面的四步即为检测 FAST 的步骤。不过需要注意的是, 在第 3 步中, 超过阈值的个数设的是 9 个, 所以上面也被称作为 FAST-9。类似的, 还有 FAST-10, FAST-11 和 FAST-12。

从上面步骤可以看出, FAST 特征点检测起来简单快速, 并且易于实现。但是上面算法仅仅是特征点检测, 并没有对特征点进行描述, 这样是无法用来匹配的, 所以下面介绍一下 BRIEF 特征描述子。

2.3.2. BRIEF 特征描述子

BRIEF 特征描述子是一个二值化的特征描述算法, 它采用一个长度为 n 的二进制码串作为描述子向量, 非常节省内存。下面介绍一下算法的具体步骤:

- (1) 首先对图像进行高斯平滑，去除噪声；
- (2) 以检测到的特征点 p 为中心，取一个 $d \times d$ 大小的 patch 邻域；
- (3) 在这个 patch 邻域内以某一规则选出 n 个点，比较这 n 个点的像素值，并做以下标记：

$$\tau(p; x, y) := \begin{cases} 1 & \text{if } I_x < I_y \\ 0 & \text{otherwise} \end{cases} \quad (2-8)$$

- (4) 把上面步骤得到的二进制串写成一个 N 维向量的形式：

$$f_{n_d}(p) := \sum_{1 \leq i \leq n_d} 2^{i-1} \tau(p; x_i, y_i) \quad (2-9)$$

以上步骤便是 BRIEF 描述子的计算方法，可以看出，BRIEF 特征描述子是没有对特征点的方向和尺度进行描述的，所以当尺度发生变化或者方向发生旋转时，使用 BRIEF 特征描述子匹配特征点是难以取得很好的效果。所以，接下来介绍一下 ORB 特征。

2.3.3. ORB 特征

ORB 特征是在上面介绍的 FAST 特征点和 BRIEF 特征描述子的基础上，添加了方向的检测和描述，本节重点介绍一下 ORB 特征是如何对特征点进行方向上进行优化的，并介绍一下如何解决尺度不变性的问题。

首先介绍一下 ORB 特征对 FAST 特征点如何添加方向上的检测。首先介绍一下图像矩的概念。定义下面式子：

$$m_{pq} = \sum_{x,y} x^p y^q I(x, y) \quad (2-10)$$

上面式子 (2-10) 的计算结果被称为图像的 $(p+q)$ 阶矩。比如图像的 0 阶矩 m_{00} ，表征了图像像素值的总和，类似的还有一阶矩 m_{01} 、 m_{10} 和图像的二阶矩 m_{02} 、 m_{11} 、 m_{20} ，以及图像的更高阶矩等等。

然后对图像的质心做出如下的定义：

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \quad (2-11)$$

有了上面式子定义的图像质心，便可以对特征点进行方向上的检测。首先在特征点的附近邻域定义一个以特征点 P 为中心的小 patch，求出这个小 patch 的质心 C 。但是需要注意的是，求这个小 patch 的时候，式子 (2-10) 中的 x 和 y 值并不是图像的像素坐标值，而是相对于 patch 中心亦即特征点 P 相对

坐标值。特征点的方向被定义为特征点 P 和质心 C 连线的方向，方向角的计算方式如下：

$$\theta = \text{atan2}(m_{01}, m_{10}) \quad (2-12)$$

用这个方向角去表征特征点的方向。

在特征点有了方向之后，对原来的 BRIEF 描述也需要做出相应的改进。为了使 BRIEF 特征描述子具备旋转不变性，最直接的一个方法就是将两个待匹配的特征点的邻域 patch 首先旋转到一个方向上，但是旋转整个邻域 patch 的计算量是比较大的，一个替代的方法便是只旋转 BRIEF 使用的 n 个点对。将这 n 个点对的坐标写成一个矩阵的形式：

$$S = \begin{pmatrix} x_1, & \dots, & x_n \\ y_1, & \dots, & y_n \end{pmatrix} \quad (2-13)$$

利用上面的方向角 θ ，可以得到如下的旋转矩阵：

$$R_\theta = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \quad (2-14)$$

利用旋转矩阵得到旋转后的 n 个点对的坐标：

$$S_\theta = R_\theta S \quad (2-15)$$

利用旋转后的点对生成 BRIEF 特征描述子。这种改进后的 BRIEF 算法被称为 Steered BRIEF 算法。

解决了 FAST 和 BRIEF 的旋转不变性后，下面介绍一下怎么解决尺度的问题。

首先介绍一下图像金字塔的概念。图像金字塔是由多层不同分辨率的图像组成，可以表达出图像的多尺度概念，如下图 2-4 所示。

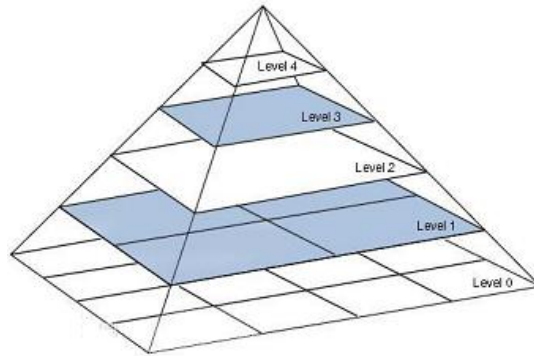


图 2-4 图像金字塔

对于高斯金字塔，生成过程是这样的：原始图像作为金字塔的最底层，

即 Level-0 层。然后对第 k 层的图像进行高斯滤波，然后以一定的采样率对其进行降采样，生成第 $k+1$ 层的图像，如此重复，直到生成的某一层的图像的分辨率达到了限定的最低分辨率。

利用生成的图像金字塔，可以描述图像的尺度概念。具体做法如下：在图像金字塔的每层都进行 FAST 特征点的检测，生成相应 BRIEF 特征描述子，在进行特征点匹配的时候，直接比较特征点的 BRIEF 特征描述子即可，并且由于 BRIEF 特征描述子是二值的，所以对于两个 BRIEF 向量，可以直接比较它们的汉明距离。

2.4. 立体视觉模型

本次工作所使用的传感器为双目相机和 IMU，相比于单目相机，双目相机可以直接获取特征点的深度值。但是有些时候，由于遮挡或者双目相机左右目的视角不同，一些特征点可能只在一个相机中被观测到，这些特征点的深度值就得靠多帧图像和对极约束去求得了。所以本节首先介绍一下双目测距的原理，然后介绍怎么用匹配到的特征点去求解相机的位姿。

2.4.1. 双目测距原理

首先需要说明的是双目相机首先得经过双目标定和立体校正后才能用于测距，对于双目标定和测距的介绍将在系统标定一章中介绍，所以本节假设相机已经立体校正。

经过立体校正后的相机是行对齐的，左右目的光心和光轴在同一平面上，并且两者具有统一的焦距值和内参数。

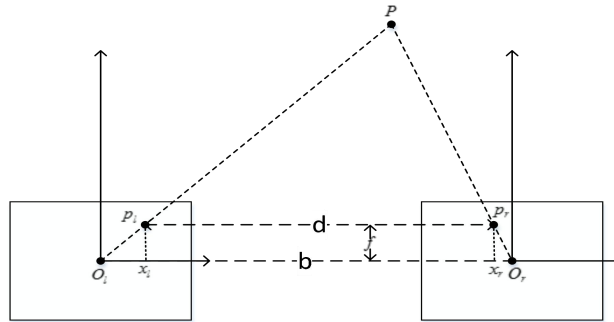


图 2-5 双目测距原理

双目测距的示意图如上图 2-5 所示，其中 O_l 和 O_r 分别是左目相机和右目

相机的光心，亦即相机坐标系的原点。 f 为相机的焦距， P 为空间一点，由于左右目的图像是行对齐，所以点 P 在左右目的投影在图像坐标系下有相同的 y 坐标，因此这里只用 x 坐标去计算点 P 的深度值。 b 为 O_l 和 O_r 之间的距离，这个距离被称作基线，可以由标定步骤得到， x_l 和 x_r 分别是投影点在左右目图像坐标系下的横坐标，可以得到的是 $d = b - (x_l - x_r)$ ，由三角形相似的原理可以得到下面式子（2-16）

$$\frac{d}{b} = \frac{b - (x_l - x_r)}{b} = \frac{z - f}{z} \quad (2-16)$$

亦即：

$$\frac{x_l - x_r}{b} = \frac{f}{z} \quad (2-17)$$

上面式子（2-17）中， f 表示焦距， z 表示点 P 的深度值。值得一提的是，在立体视觉中， $(x_l - x_r)$ 通常被称作视差，如果想利用双目相机获得深度图，通常首先获取视差图，再由视差图获取深度图。

2.4.2. 迭代最近点法

利用 2.3.1 的方法可以得到某一帧下图像上特征点的深度值，再利用相机模型可以得到这些特征点在相机坐标系下的 3D 坐标。对于相邻两帧图像，可以使用特征点匹配的方法找到这两帧图像的匹配关系，那么，就可以得到相邻两帧图像一系列匹配好的 3D 点了，利用这些匹配好的 3D 点，可以使用迭代最近点法（ICP, Iterative Closest Point）求解这两帧图像的相对位姿了。

现在，假设有 n 个匹配好的特征点对 $(p_1, p'_1), (p_2, p'_2) \dots (p_n, p'_n)$ ，待求得是两帧图像之间的相对旋转 R 和相对平移 t 。可以把这个问题当做一个优化问题去对待，那么首先构建一个损失函数。对于一个单独的点对 (p_i, p'_i) ，定义这样一个误差项：

$$e_i = p_i - (Rp'_i + t) \quad (2-18)$$

损失函数便可以这样去定义：

$$J = \frac{1}{2} \sum_{i=1}^n e_i^2 = \frac{1}{2} \sum_{i=1}^n \|p_i - (Rp'_i + t)\|^2 \quad (2-19)$$

现在的工作便是寻找相对旋转 R 和相对平移 t 使得损失函数 J 最小。

首先定义这组点对在每个图像坐标系下的质心：

$$p = \frac{1}{n} \sum_{i=1}^n (p_i), \quad p' = \frac{1}{n} \sum_{i=1}^n (p'_i) \quad (2-20)$$

现在，损失函数可以用如下形式去表示：

$$\begin{aligned} J &= \frac{1}{2} \sum_{i=1}^n \|p_i - Rp'_i - t - p + Rp' + p - Rp'\|^2 \\ &= \frac{1}{2} \sum_{i=1}^n \|p_i - p - R(p'_i - p') + (p - Rp' - t)\|^2 \\ &= \frac{1}{2} \sum_{i=1}^n (\|p_i - p - R(p'_i - p')\|^2 + \|(p - Rp' - t)\|^2) + \\ &\quad 2(p_i - p - R(p'_i - p'))^T (p - Rp' - t) \\ &= \frac{1}{2} \sum_{i=1}^n (\|p_i - p - R(p'_i - p')\|^2 + \|(p - Rp' - t)\|^2) \\ &= \frac{1}{2} \sum_{i=1}^n (\|p_i - p - R(p'_i - p')\|^2) + \frac{n}{2} \|(p - Rp' - t)\|^2 \\ &= J_1 + J_2 \end{aligned} \quad (2-21)$$

上式中 J_1 只和相对旋转 R 有关，所以可以先找到最优的旋转矩阵 R 使得 J_1 最小，然后只需让 $t = p - Rp'$ 便可以使 J_2 为 0。

现在的目标便是找到最优的旋转矩阵 R 使得 J_1 最小。首先定义一组变量：

$$q_i = p_i - p, \quad q'_i = p'_i - p' \quad (2-22)$$

上面 q_i 和 q'_i 被称作去质心坐标，将去质心坐标代到 J_1 可以得到：

$$\begin{aligned} J_1 &= \frac{1}{2} \sum_{i=1}^n \|q_i - Rq'_i\|^2 \\ &= \frac{1}{2} \sum_{i=1}^n (q_i^T q_i + q_i'^T R^T R q'_i - 2q_i^T R q'_i) \end{aligned} \quad (2-23)$$

上面式子中第一项和相对旋转矩阵 R 无关，所以不用考虑。对于第二项，由于 R 是旋转矩阵，所以 $R^T R = I$ ，所以第二项也与旋转矩阵无关，不用考虑。所以需要优化的部分变成了如下形式：

$$J_{13} = -\sum_{i=1}^n q_i^T R q'_i = -\sum_{i=1}^n \text{tr}(R q'_i q_i^T) = -\text{tr}\left(R \sum_{i=1}^n (q'_i q_i^T)\right) = -\text{tr}(RW) \quad (2-24)$$

对于上面式子的求解，需要对 W 矩阵进行奇异值分解：

$$W = U\Sigma V^T \quad (2-25)$$

当 W 矩阵是满秩矩阵时，相对旋转矩阵 R 可以由如下式子得到：

$$R = UV^T \quad (2-26)$$

得到相对旋转矩阵 R 后，再由 $t = p - Rp'$ 便可以得到相对平移矩阵 t 。至此，便是利用 ICP 方法求解两帧图像相对旋转矩阵和相对平移矩阵的步骤。

2.5. 回环检测回环校正

利用上面介绍的迭代最近点法，可以算出两帧图像之间的平移和旋转，在固定了第一帧图像的初始位姿之后，理论上便递推的算出后续各帧的位姿。但是这样做一个明显的缺点是会累计误差，对相邻两帧估计的位姿误差会累加到下一帧上，致使误差越来越大。

为了消除这个误差，可以利用这样一个信息：当系统两次经过同一个地点时，由于累计误差的影响，递推得到的位姿会差别很大，但是我们可以强制使这两次的位姿相同，然后校正其它帧的位姿，以此来降低累计误差的影响。要想实现这一思想，需要考虑两件事：一是如何准确地识别到系统两次经过了相同的地方；二是如何利用回环信息校正其它帧的位姿。下面来介绍如何解决这两件事。

2.5.1. 词袋模型和视觉字典

在本次工作中，使用词袋模型和视觉字典^[43]的方法对位置进行描述和识别。对于一帧图像，可以首先使用词袋模型的方法对该帧图像进行描述，然后再查询视觉字典找到与该帧图像相似的图像。下面介绍一下词袋模型和视觉字典。

对于这部分的介绍，需要用到上节介绍的特征点和特征点描述子的概念。要想构建视觉字典，首先得现在一帧图像上检测到足够多的特征点，并用适合的描述子对特征点进行描述。采集大量的图像，进行特征点的检测和描述子的计算，便可以得到大量的特征点和相应的描述子。利用描述子，对特征点进行 k -均值聚类，将这些特征点聚成 k 类，然后对于单独的每一类，再对其进行 k -均值聚类，也同样聚成 k 类，如此重复，会得到一个深度为 w 的字典树，把字典树的根节点叫做第 L_w 层，字典树的叶子节点叫做第 L_0 层，每个叶子节点都是若干特征点聚类的结果，被称作一个单词。需要注意的是，在

进行聚类的时候，需要对两个描述子进行相似性度量，由于使用的 ORB 特征点的描述子是一个二值的向量，所以使用下面式子来度量两个描述子之间的相似性：

$$s(v_1, v_2) = 1 - \frac{1}{2} \left| \frac{v_1}{|v_1|} - \frac{v_2}{|v_2|} \right| \quad (2-27)$$

有了视觉字典树之后，便可以把一帧图像转换为若干单词的集合。具体步骤如下：首先对待处理的一帧图像进行特征点的提取和描述子的生成，然后对每一个特征点，在字典树上查询其位置，获得其单词描述。这样，每个特征点都能够用一个视觉字典中的单词代替，把该帧图像的所有单词用一个直方图去统计，便可以获得一个描述该帧图像包含单词情况的一个向量，这就是词袋模型。

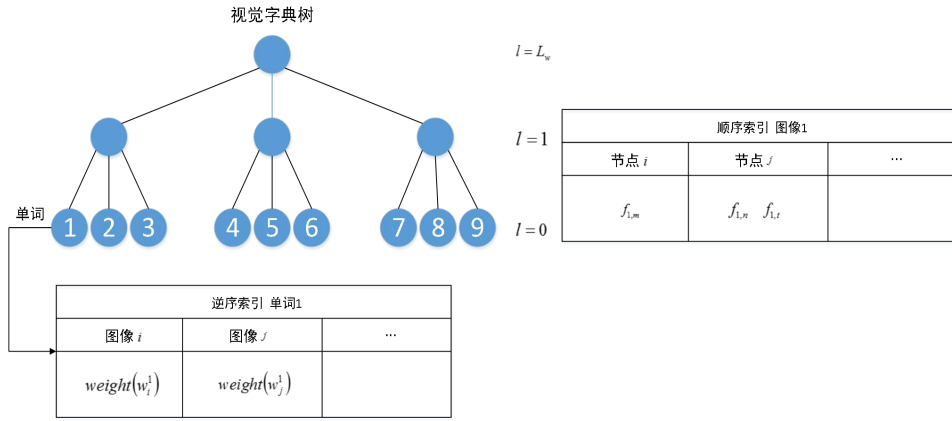


图 2-6 视觉字典树

在实际使用视觉字典树的时候，还会加上逆序索引和顺序索引。所谓逆序索引，就是对于视觉字典树的每个叶子节点，都会维持一个列表，该列表保存了包含该单词的图像以及此单词在该帧图像中的权重，单词的权重使用词频-逆文件频率（Term Frequency-Inverse Document Frequency, TF-IDF）去计算。所谓词频就是某一单词出现的频率，对于某一个单词 w ，用下面式子去计算其词频：

$$TF_w = \frac{n_w}{\sum_i n_i} \quad (2-28)$$

其中 n_i 是第 i 个单词出现的次数。而单词 w 逆文件频率则表征了包含该单词的图像占全部图像多少的一个参数，用下面式子计算单词 w 的逆文件频率：

$$IDF_w = \log \left(\frac{sum(image)}{1 + w(image)} \right) \quad (2-29)$$

其中 $sum(image)$ 表示全部图像的数量，而 $w(image)$ 表示包含单词 w 的图像的数量。利用词频-逆文件频率，可以得到逆序索引中单词在图像中的权重，具体计算式子如下式（2-30）所示：

$$weight(w) = TF_w * IDF_w \quad (2-30)$$

其中 $weight(w)$ 表示单词的权重。

而顺序索引则存储了图像的特征点和特征点在视觉字典树某一层所在的节点。顺序索引可以用来加速特征点的匹配，当需要匹配两帧图像的特征点的时候，可以只匹配位于同一节点的特征点，从而降低了需要匹配的特征点对数。

2.5.2. 回环检测

有了上面的视觉字典树和词袋模型，便可以对一个地点进行识别，从而检测出系统是否两次经过了相同的地方，下面是具体过程。

假设目前已经有一个建好的视觉字典树，以及从初始时刻到现在的图像建立起来的顺序索引和逆序索引，对于当前时刻 t 的图像 I_t ，首先提取当前帧的特征点，计算相应的特征描述子，得到描述该帧图像的词袋向量 v_t 。在视觉字典树上利用逆序索引找到与图像 I_t 拥有相同单词的一系列图像 v_{i1} , v_{i2} , ..., 作为回环检测的候选图像，同时使用式子（2-31）计算一个分数 $s(v_t, v_{i1})$ ，用来描述候选回环图像同当前图像的词袋向量的相似度。但是这个分数 $s(v_t, v_{i1})$ 会依赖于当前的图像 I_t ，这就很难用一个同一的阈值去筛选候选回环图像，为了解决这个问题，利用如下式子对分数进行归一化的处理：

$$\eta(v_t, v_{i1}) = \frac{s(v_t, v_{i1})}{s(v_t, v_{t-\Delta t})} \quad (2-31)$$

其中 $v_{t-\Delta t}$ 表示前一帧图像的词袋向量。这样归一化的前提是当前帧图像和前一帧图像非常相似，大多数情况下这一条件是满足的，但是也会有一些极端的情况，比如前后两帧有一个较大的旋转，此时 $s(v_t, v_{t-\Delta t})$ 会很小。针对这一问题，采取的方法是设置一个阈值 n_f ，只有前后两帧匹配到的特征点的数量超过这个阈值 n_f 时，才对当前帧进行回环检测。利用归一化处理过后的分数

$\eta(v_i, v_{t_j})$ 可以对候选回环图像进行筛选, 这里会设置一个阈值 α , 归一化分数低于 α 的候选回环图像会被从回环回环列表里剔除。

为了防止多个在时间上相近的图像都被选成候选回环, 可以把候选回环列表里面的候选回环图像按其时间戳分成多个群组, 用 T_i 表示多个时间上接近的候选回环, 这样多个候选回环组合 $(v_i, v_{t_{n_i}}), \dots, (v_i, v_{t_{m_i}})$ 表示成一个候选回环组合 (v_i, v_{T_i}) , 候选回环群的分分数等于群组内候选回环图像分分数累加:

$$H(v_i, V_{T_i}) = \sum_{j=n_i}^{m_i} \eta(v_i, v_{t_j}) \quad (2-32)$$

计算完各个候选回环群的分分数后, 选择分分数最大候选回环群组继续进行后续的步骤, 舍弃其它的候选回环群组。通常, 如果当前帧和某一候选回环图像是真的回环, 那么除了和回环图像很相似之外, 当前帧应该和回环图像邻近的图像也非常相似, 而上面计算候选回环群分分数的方法正式利用了这一特点。

选出了最佳候选回环群 $V_{T'}$, 还需要对其做一些验证。第一个验证是对于当前最佳回环对 $(v_i, V_{T'})$, 考察其之前的 k 个候选回环群 $(v_{i-\Delta t}, V_{T_1}), \dots, (v_{i-k\Delta t}, V_{T_k})$, 如果它们是时间上连续的, 则保留当前的回环群, 否则舍弃。对于保留的回环群, 在群里找一个分分数最高的回环对 $(v_i, v_{t'})$, 其中 $t' \in T'$, 保留这个回环对。第二个验证是几何上的验证。对于上面得到的回环对 $(v_i, v_{t'})$, 利用 RANSAC 算法算出这对图像的基础矩阵。再用计算得出的基础矩阵去区分图像对匹配到的特征点对, 根据误差的大小将其分为外点和内点。这里, 误差比较大的点称为外点, 认为其是匹配错误的点对, 误差比较小的点称为内点, 认为其是匹配正确的点对。当内点的数量大于一定阈值的时候, 便认为是得到了正确的回环, 接下来便是利用回环信息进行回环的校正。

2.5.3. 回环校正

如图 2-6 所示, 回环校正所要做的事便是如何利用回环信息消除累计的误差。首先, 利用上节获得的回环信息, 会得到一个回环图像对以及它们匹配成功的内点, 回环校正的第一步便是回环融合。回环融合就是将回环对匹配到的特征点所相关联的地图点融合, 将重复的地图点剔除。然后利用回环检测得到的基础矩阵算出当前帧和回环帧之间的位姿转换关系, 然后将当前帧的位姿校正。最后利用 BA^[44] (Bundle Adjustment, 光束平差法) 将回环

内图像的所有位姿进行校正。

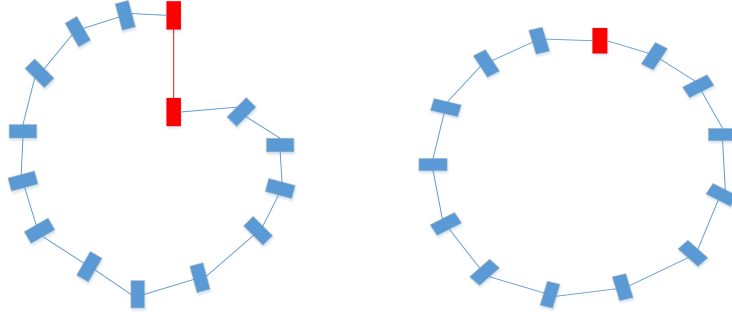
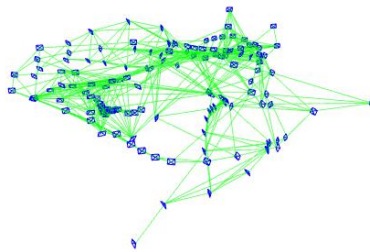


图 2-6 回环校正前（左）和回环校正后（右）

将回环内所有位姿校正这一步骤是利用 Essential Graph（本质图）完成的，所以在具体介绍所使用的算法之前，先介绍一下 Covisibility Graph（共视图）和 Essential Graph。

Covisibility Graph 是一种无向加权图。图的每个顶点表示某个时刻相机的状态，亦即位姿。图的边连接两个顶点，表示这两个状态下的相机能够共同观测到地图点，如图 2-8 所示便是一个共视图，其中蓝色的顶点表示相机状态点，绿色的线连接两个蓝色顶点，表示这两个时刻的相机能够共同观测到地图点。



2-8 共视图

共视图描述了各个时刻的图像帧以及同各个地图点之间的连接关系，但直接使用共视图进行位姿优化的话计算量会有些大，所以可以进行这样一个考虑：如果共视图中的两个顶点能够共同观测到更多地图点，说明它们有更强的连接关系。所以，利用这个特点，可以对共视图进行一些简化，设置一个合适的阈值，如果两个顶点能够观测到的地图点数量大于这个阈值，则它们之间的边会被保存，否则将被剔除。但是，若如果某一组全部顶点和另一组顶点的边全部都被剔除了，那么优化的时候这两组顶点就会被分开优化，不包含回环信息的那组相机的位姿将不会被得到优化，这是所不希望看到的。

所以，在剔除弱连接关系的边之前，先建立一个原来共视图的生成树，在此基础上在将一些不需要的边剔除，这样最终得到的也是一个原来共视图的生成树，并且顶点之间大多能够观测到足够多数量的地图点。用上面方法得到的加权无向图便是 Essential Graph。

回环的校正便是在本质图上进行的。现在设本质图内的有 n 帧图像，其 在世界坐标系下的位姿分别是 $T_1^w, \dots, T_i^w, \dots, T_n^w$ ，这些图像能够观测到 m 个 地图点，这些地图点在世界坐标系下的坐标分别是 $X_1^w, \dots, X_i^w, \dots, X_n^w$ ，需 要说明的是，对本质图的优化，只优化相机的位姿，不优化地图点的坐标。

假设相机顶点 i 能够观测到地图点 j ，定义如下一个误差项：

$$e_{ij} = x_{i,j} - \pi(T_i^w, X_j^w) \quad (2-33)$$

上面式子 (2-33) 中 $x_{i,j}$ 表示在图像帧 i 上关联地图点 j 的特征点的像素坐标， $\pi(T_i^w, X_j^w)$ 表示将地图点 j 利用相机模型投影到图像帧 i 上的像素坐标，其具 体表达式见式子 (2-34)：

$$\pi(T_i^w, X_j^w) = \begin{bmatrix} f_x * \frac{x_{i,j}^c}{z_{i,j}^c} + u_0 \\ f_y * \frac{y_{i,j}^c}{z_{i,j}^c} + v_0 \end{bmatrix} \quad (2-34)$$

$$\begin{bmatrix} x_{i,j}^c & y_{i,j}^c & z_{i,j}^c \end{bmatrix}^T = R_i^w X_j^w + t_i^w$$

其中 $\begin{bmatrix} x_{i,j}^c & y_{i,j}^c & z_{i,j}^c \end{bmatrix}^T$ 表示地图点 j 在相机顶点 i 相机坐标系下的坐标， $R_i^w \in SO(3)$ ，表示相机顶点 i 在世界坐标系下的旋转， $t_i^w \in R^3$ ，表示相机顶点 i 在世界坐标系下的平移， f_x 、 f_y 、 u_0 、 v_0 表示相机 i 的内参数。

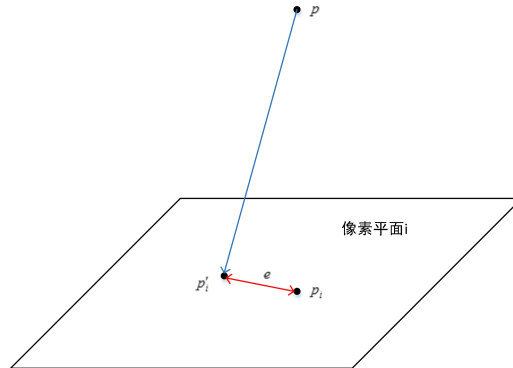


图 2-9 重投影误差

由式子 (2-33) 定义的误差叫做重投影误差，图 2-7 比较形象地解释了什么是重投影误差。对本质图的优化就是要优化所有地图点的重投影误差，基于这个思想，按如下形式定义损失函数：

$$C = \sum_{i,j} (b_{ij} \rho_h(e_{i,j}^T \Sigma_{ij} e_{i,j})) \quad (2-35)$$

上面式子中， b_{ij} 是一个标志量，当相机顶点 i 能够观测到地图点 j 是， b_{ij} 等于 1，否则 b_{ij} 等于 0。 $\rho_h(\cdot)$ 表示 Huber 损失函数， Σ_{ij} 表示协方差矩阵。

定义了误差函数之后，便可以使用高斯-牛顿法或者列文伯格-马尔夸特算法去优化相机的位姿 T_i^w 使得损失函数最小。利用这个方法，可以降低由于积分前后帧图像不断累积的误差。

2.6. 本章小结

本章叙述了本次工作对视觉信息的处理方法，包括相机模型，双目模型，双目测距原理以及特征点检测和描述的方法，利用这些可以相机坐标系下一系列特征点的三维坐标以及前后帧图像之间特征点之间的联系。然后利用迭代最近点法可以获得前后帧图像之间的相对位姿，对前后帧图像相对位姿进行不断的积分，可以获得各个图像帧相对于世界坐标系的位姿，最后通过回环检测和回环校正，可以降低由于积分累计下来的误差。

本章给出了计算重投影误差的方法，为后续的非线性优化提供了视觉信息的误差项，本章还给出了利用初始角估计相机位姿的方法，利用该方法可以对系统进行一些初始化参数的估计。本章最后给出的回环检测和回环校正的方法则可以消除累计非误差，提高系统的估计精度。

第 3 章 IMU 信息预处理

3.1. 引言

利用上一章叙述的视觉方法已经可以构成一个完整的 SLAM 系统，但是，纯视觉的 SLAM 方法有诸多缺点，比如当快速移动或者视觉特征缺失的时候会跟丢，在重复纹理的场景下会发生特征点的误匹配，导致位姿解算误差较大等等。所以，本次工作决定使用 IMU 的信息与视觉信息进行融合，充分利用这两种传感器互补的特点，提高 SLAM 系统的鲁棒性。

本章着重叙述怎么去对 IMU 的信息进行一些预处理^[45]，包括对 IMU 建立噪声模型和运动学模型，对 IMU 进行预积分等等，来解决 IMU 信息和视觉信息频率不匹配的问题，为后续的视觉信息和 IMU 信息融合奠定基础。

3.2. IMU 建模

3.2.1. IMU 噪声模型

IMU 是由陀螺仪和加速度计组成的，陀螺仪可以测量自身的角速度，加速度计可以测量自身的加速度。对于加速度计和陀螺仪的测量数据，都建模成三个部分的叠加：

$$\begin{aligned}\tilde{a}(t) &= a(t) + b^a(t) + n^a(t) \\ \tilde{\omega}(t) &= \omega(t) + b^g(t) + n^g(t)\end{aligned}\tag{3-1}$$

其中 $\tilde{a}(t)$ 和 $\tilde{\omega}(t)$ 分别表示加速度计和陀螺仪的原始测量的加速度和角速度， $a(t)$ 和 $\omega(t)$ 分别表示真实的加速度测量值和角速度测量值， $b^a(t)$ 和 $b^g(t)$ 分别表示加速度计和陀螺仪的 bias（随机游走）， $n^a(t)$ 和 $n^g(t)$ 分别表示加速度计和陀螺仪的噪声。

对于加速度计和陀螺仪的噪声，本次工作将其都建模成高斯白噪声。对于加速度计和陀螺仪的噪声 $n^a(t)$ 和 $n^g(t)$ ，现在统一将其写成 $n(t)$ ，对 $n(t)$ 建模如下式（3-2）所示：

$$\begin{aligned}E[n(t)] &= 0 \\ E[n(t_1)n(t_2)] &= \sigma^2 \delta(t_1 - t_2)\end{aligned}\tag{3-2}$$

上式 (3-2) 中 σ 为噪声强度, 加速度计和陀螺仪的噪声强度可以通过标定或者查 IMU 的数据手册获得。在进行实际的程序实现时, 获得的数据都是离散的测量数据, 所以要将上面的式子离散化, 下面式子 (3-3) 是离散化后的结果:

$$\begin{aligned} n_d[k] &= \sigma_d w[k] \\ w[k] &\sim N(0,1) \\ \sigma_d &= \sigma \frac{1}{\sqrt{\Delta t}} \end{aligned} \quad (3-3)$$

其中 σ_d 是离散的噪声强度, Δt 是系统的采样时间。

对于加速度计和陀螺仪的随机游走, 本次工作将其建模成一个布朗运动, 即一个由高斯白噪声驱动的值。如下式 (3-4) 所示:

$$\dot{b}(t) = \sigma_b w(t) \quad (3-4)$$

其中, σ_b 为随机游走的强度, 也是由标定步骤或者查阅 IMU 的数据手册获得。同样, 需要对其进行离散化, 会得到如下离散化的形式:

$$\begin{aligned} b_d[k] &= b_d[k-1] + \sigma_{bd} w[k] \\ w[k] &\sim N(0,1) \\ \sigma_{bd} &= \sigma_b \sqrt{\Delta t} \end{aligned} \quad (3-5)$$

其中 σ_{bd} 为离散的随机游走强度, Δt 是系统的采样时间。

3.2.2. IMU 运动学模型

上节介绍了 IMU 噪声模型, 这节将着重介绍 IMU 的运动学模型。需要说明的是, IMU 测量的角速度和加速度所在的坐标系与参数方程都是 IMU 坐标系, 在本次工作中, 用字母 B 表示 IMU 坐标系, 用字母 W 表示世界坐标系, 用 R_{WB} 表示 IMU 坐标系到世界坐标系的旋转矩阵, 用 p_{WB} 表示 IMU 坐标系到世界坐标系的平移矩阵。将上节介绍的噪声模型表示在各个坐标系下, 并将重力向量考虑在内, 可以得到下面的式子:

$${}_B \tilde{a}_{WB}(t) = R_{WB}^T ({}_W a_{WB}(t) - {}_W g) + b^a(t) + n^a(t) \quad (3-6)$$

$${}_B \tilde{\omega}_{WB}(t) = {}_B \omega_{WB}(t) + b^g(t) + n^g(t) \quad (3-7)$$

上面式子中, ${}_W g$ 表示世界坐标系下的重力向量。

对 IMU 姿态更新的运动学推导有多种方法, 比如四元数法, 旋转矢量法,

旋转矩阵法等等。本次工作使用旋转矩阵法来推导 IMU 的运动学模型。首先，可以建立一个连续时间状态下的模型，如下面式子（3-8）所示：

$$\begin{aligned}\dot{R}_{WB} &= R_{WB} \hat{\omega}_{WB} \\ {}_W \dot{v}_{WB} &= {}_W a_{WB} \\ {}_W \dot{p}_{WB} &= {}_W v_{WB}\end{aligned}\quad (3-8)$$

对上面式子进行离散化，可以得到离散形式的运动学方程：

$$\begin{aligned}R_{WB}(t + \Delta t) &= R_{WB}(t) \text{Exp}({}_B \omega_{WB}(t) \Delta t) \\ {}_W v_{WB}(t + \Delta t) &= {}_W v_{WB}(t) + {}_W a_{WB}(t) \Delta t \\ {}_W p_{WB}(t + \Delta t) &= {}_W p_{WB}(t) + {}_W v_{WB}(t) \Delta t + \frac{1}{2} {}_W a_{WB}(t) \Delta t^2\end{aligned}\quad (3-9)$$

上面便是离散形式的 IMU 运动学方程。将上节建立的噪声模型代到运动学模型中，同时为了书写的简化略去坐标系的下标，可以得到完整的 IMU 模型的方程：

$$\begin{aligned}R(t + \Delta t) &= R(t) \text{Exp}((\tilde{\omega}(t) - b^g(t) - n^{gd}(t)) \Delta t) \\ v(t + \Delta t) &= v(t) + g \Delta t + R(t)(\tilde{a}(t) - b^a(t) - n^{ad}(t)) \Delta t \\ p(t + \Delta t) &= p(t) + v(t) \Delta t + \frac{1}{2} g \Delta t^2 + \frac{1}{2} R(t)(\tilde{a}(t) - b^a(t) - n^{ad}(t)) \Delta t^2\end{aligned}\quad (3-10)$$

至此，IMU 初步建模完毕。

3.3. IMU 预积分

由于 IMU 的测量频率比较高，而视觉传感器的测量频率相对较低，所以，当这两种传感器进行信息融合时，首先考虑的是如何处理测量频率的问题。一个简单的想法是在每次视觉获得测量数据时，对 IMU 的数据进行一次积分，并利用上节建立的 IMU 模型，获取当时 IMU 的位姿，将这个位姿结果记录下来然后和视觉得到的位姿结果融合。但是，这样做的一个明显的缺点就是当前图像帧时刻得到的 IMU 位姿会依赖前面图像帧时刻的位姿，当前面的位姿或者随机游走的 bias 值因为优化或者回环检测而发生变化时，不得不重新进行积分的操作，否则无法将再次利用 IMU 信息来对两个图像帧的位姿进行约束。

所以，本次工作将使用将 IMU 预积分的方法解决两个传感器频率不匹配的问题。预积分的思想就是只对两个图像时刻的 IMU 信息进行积分，获得相

对位姿值，并将这个积分结果保存，用于后续联合优化。



图 3-1 预积分前（左）和预积分后（右），其中方块代表相机信息节点，圆点代表 IMU 信息节点

图 3-1 形象的说明了这个思想。通过预积分，可以使两个传感器的信息频率保持一致，方便后续的信息融合。所以，本节将着重介绍对 IMU 预积分的方法，包括对测量量的预积分，噪声传播方程以及对随机游走值 bias 得更新等，为后续的视觉和 IMU 的信息融合奠定基础。

3.3.1. 预积分 IMU 测量值

因为预积分的思想是把两个时刻之间的 IMU 观测信息积分成两个时刻的相对位姿值，所以首先要做的是对两个时刻间的加速度值和角速度值积分。现在假设有两个图像帧时刻 t_i 和 t_j ，且有 $t_j > t_i$ ，和这两个时刻之间的 IMU 的测量的角速度 $\omega(t_i), \dots, \omega(t_k), \dots, \omega(t_{j-1})$ 以及加速度 $a(t_i), \dots, a(t_k), \dots, a(t_{j-1})$ ，为了书写简便，以下将使用下标表示当时时刻的变量，即将测量的角速度值写成 $\omega_i, \dots, \omega_k, \dots, \omega_{j-1}$ ，将加速度值写成 $a_i, \dots, a_k, \dots, a_{j-1}$ 。于是，式子 (3-10) 可以写成下面的形式：

$$\begin{aligned}
 R_j &= R_i \prod_{k=i}^{j-1} \text{Exp}((\tilde{\omega}_k - b_k^g - n_k^{gd}) \Delta t) \\
 v_j &= v_i + g \Delta t_{ij} + \sum_{k=i}^{j-1} R_k (\tilde{a}_k - b_k^a - n_k^{ad}) \Delta t \\
 p_j &= p_i + \sum_{k=i}^{j-1} v_k \Delta t + \frac{1}{2} g \Delta t_{ij}^2 + \frac{1}{2} \sum_{k=i}^{j-1} R_k (\tilde{a}_k - b_k^a - n_k^{ad}) \Delta t^2
 \end{aligned} \tag{3-11}$$

现在，利用上面的式子结果，可以利用预积分的思想将两帧图像时刻的旋转值、速度值和平移值从 IMU 的测量信息里面分离，这样在这些量优化的时候可以避免重新积分 IMU 的测量信息，提升计算效率。定义下面变量为两个图像帧时刻 t_i 和 t_j 的相对值：

$$\begin{aligned}
 \Delta R_{ij} &= R_i^T R_j = \prod_{k=i}^{j-1} \text{Exp}((\tilde{\omega}_k - b_k^g - n_k^{gd}) \Delta t) \\
 \Delta v_{ij} &= R_i^T (v_j - v_i - g \Delta t_{ij}) = \sum_{k=i}^{j-1} \Delta R_{ik} (\tilde{a}_k - b_k^a - n_k^{ad}) \Delta t \\
 \Delta p_{ij} &= R_i^T \left(p_j - p_i - v_i \Delta t_{ij} - \frac{1}{2} g \Delta t_{ij}^2 \right) = \sum_{k=i}^{j-1} \left[\Delta v_{ik} \Delta t + \frac{1}{2} R_{ik} (\tilde{a}_k - b_k^a - n_k^{ad}) \Delta t^2 \right]
 \end{aligned} \tag{3-12}$$

其中, $\Delta R_{ik} = R_i^T R_k$, $\Delta v_{ik} = v_k - v_i$ 。可以看出, 上面定义的几个相对值亦即积分结果只与 t_i 和 t_j 之间的 IMU 测量值、随机游走 bias 值和噪声有关, 已经不依赖前面时刻的位姿值。

由于随机游走在短时间内变化不大, 所以在两个图像帧时刻之间, 认为它不变, 即:

$$\begin{aligned}
 b_i^g &= b_{i+1}^g = \dots = b_{j-1}^g \\
 b_i^a &= b_{i+1}^a = \dots = b_{j-1}^a
 \end{aligned} \tag{3-13}$$

为了将噪声的影响从上面积分得到的相对值中分离出来, 可以对上面的相对值做一些处理, 将其写成一个仅由测量值和随机游走 bias 得到的测量相对值和一个由噪声引起的扰动误差项叠加的形式。基于这个思想, 本次工作做了如下处理, 首先是对相对旋转量 ΔR_{ij} 的处理:

$$\begin{aligned}
 \Delta R_{ij} &= \prod_{k=i}^{j-1} \text{Exp}((\tilde{\omega}_k - b_i^g - n_k^{gd}) \Delta t) \\
 &= \prod_{k=i}^{j-1} [\text{Exp}((\tilde{\omega}_k - b_i^g) \Delta t) \text{Exp}(-J_r^k n_k^{gd} \Delta t)] \\
 &= \Delta \tilde{R}_{ij} \prod_{k=i}^{j-1} \text{Exp}(-\Delta \tilde{R}_{k+1,j}^T J_r^k n_k^{gd} \Delta t) \\
 &= \Delta \tilde{R}_{ij} \text{Exp}(-\delta \phi_{ij})
 \end{aligned} \tag{3-14}$$

上面式子中, $\Delta \tilde{R}_{ij} = \prod_{k=i}^{j-1} \text{Exp}((\tilde{\omega}_k - b_i^g) \Delta t)$, 是预积分的旋转测量值, 其噪声扰动为 $\delta \phi_{ij}$ 。 $J_r^k = J_r^k(\tilde{\omega}_k - b_i^g)$, 是右雅克比矩阵。右雅克比矩阵按下面式子(3-15)定义方式去计算:

$$J_r(\phi) = I - \frac{1 - \cos(\|\phi\|)}{\|\phi\|^2} \phi^\wedge + \frac{\|\phi\| - \sin(\|\phi\|)}{\|\phi\|^3} (\phi^\wedge)^2 \tag{3-15}$$

其中 $\hat{\phi}$ 是向量 ϕ 的斜切矩阵，斜切矩阵的计算方法如下式（3-16）所示：

$$\hat{\phi} = \begin{bmatrix} 0 & -\phi_3 & \phi_2 \\ \phi_3 & 0 & -\phi_1 \\ -\phi_2 & \phi_1 & 0 \end{bmatrix} \quad (3-15)$$

类似的，对相对旋转量和相对平移量也做类似上面方式的处理：

$$\begin{aligned} \Delta v_{ij} &= \sum_{k=i}^{j-1} \Delta R_{ik} (\tilde{a}_k - b_k^a - n_k^{ad}) \Delta t \\ &= \sum_{k=i}^{j-1} [\Delta \tilde{R}_{ik} (I - \delta \hat{\phi}_{ik}) (\tilde{a}_k - b_i^a) \Delta t - \Delta \tilde{R}_{ik} n_k^{ad} \Delta t] \\ &= \sum_{k=i}^{j-1} \Delta \tilde{R}_{ik} (\tilde{a}_k - b_i^a) \Delta t + \sum_{k=i}^{j-1} [\Delta \tilde{R}_{ik} (\tilde{a}_k - b_i^a)^\wedge \delta \tilde{\phi}_{ik} \Delta t - \Delta \tilde{R}_{ik} n_k^{ad} \Delta t] \\ &= \Delta \tilde{v}_{ij} - \delta v_{ij} \end{aligned} \quad (3-16)$$

$$\begin{aligned} \Delta p_{ij} &= \sum_{k=i}^{j-1} \left[\Delta v_{ik} \Delta t + \frac{1}{2} R_{ik} (\tilde{a}_k - b_k^a - n_k^{ad}) \Delta t^2 \right] \\ &= \sum_{k=i}^{j-1} \left[\frac{3}{2} R_{ik} (\tilde{a}_k - b_i^a - n_k^{ad}) \Delta t^2 \right] \\ &= \sum_{k=i}^{j-1} \left[\frac{3}{2} \tilde{R}_{ik} (I - \delta \hat{\phi}_{ik}) (\tilde{a}_k - b_i^a) \Delta t^2 \right] - \sum_{k=i}^{j-1} \left[\frac{3}{2} \tilde{R}_{ik} n_k^{ad} \Delta t^2 \right] \\ &= \sum_{k=i}^{j-1} \left[\frac{3}{2} R_{ik} (\tilde{a}_k - b_i^a) \Delta t^2 \right] + \sum_{k=i}^{j-1} \left[\frac{3}{2} \tilde{R}_{ik} (\tilde{a}_k - b_i^a)^\wedge \delta \phi_{ik} \Delta t^2 - \frac{3}{2} \tilde{R}_{ik} n_k^{ad} \right] \\ &= \Delta \tilde{p}_{ij} - \delta p_{ij} \end{aligned} \quad (3-17)$$

上面式子得到的结果分别被称为预积分速度测量值和预积分位置测量值，它们的噪声项分别是 δv_{ij} 和 δp_{ij} 。

结合式子（3-14）、（3-16）、（3-17）的结果，最终可以得到预积分观测模型：

$$\begin{aligned} \Delta \tilde{R}_{ij} &= R_i^T R_j \text{Exp}(\delta \phi_{ij}) \\ \Delta \tilde{v}_{ij} &= R_i^T (v_j - v_i - g \Delta t_{ij}) + \delta v_{ij} \\ \Delta p_{ij} &= R_i^T \left(p_j - p_i - v_i \Delta t_{ij} - \frac{1}{2} g \Delta t_{ij}^2 \right) + \delta p_{ij} \end{aligned} \quad (3-18)$$

对于噪声项 $\delta \phi_{ij}$ 、 δv_{ij} 和 δp_{ij} 的推导，将在后续章节展开介绍。为了便于编程实

现，在下面先给出递推形式的预积分观测模型的公式：

$$\begin{aligned}\Delta\tilde{R}_{ik+1} &= \Delta\tilde{R}_{ik} \text{Exp}(\tilde{\omega}_k - b_i^g) \\ \Delta\tilde{v}_{ik+1} &= \Delta\tilde{v}_{ik} + \Delta\tilde{R}_{ik}(\tilde{a}_k - b_i^a)\Delta t \\ \Delta\tilde{p}_{ik+1} &= \Delta\tilde{p}_{ik} + \Delta\tilde{v}_{ik}\Delta t + \frac{1}{2}\Delta\tilde{R}_{ik}(\tilde{a}_k - b_i^a)\Delta t^2\end{aligned}\quad (3-19)$$

上面便是 IMU 的预积分观测模型，最终把两帧之间的 IMU 信息积分成了一个测量信息得到的预积分值以及由噪声引起的噪声项之和，下面开始分析噪声项。

3.3.2. 噪声的传播

在 3.2.1 节中，IMU 观测量的噪声被建模成了高斯白噪声的形式，并可以通过标定的方法获取噪声的强度。但是，通过 3.3.1 节的预积分，噪声项变成了一系列噪声值的累加，其噪声的强度以及协方差矩阵已经和原来原始观测量的噪声强度不同，在本节中，将分析预积分过程中噪声是如何传播的。

首先是旋转量的噪声项：

$$\text{Exp}(-\delta\phi_{ij}) = \prod_{k=i}^{j-1} \text{Exp}(-\Delta\tilde{R}_{k+1j}^T J_r^k n_k^{gd} \Delta t) \quad (3-20)$$

对上面式子 (3-20) 的两边同时取 Log 运算，可以得到：

$$\delta\phi_{ij} = -\text{Log}\left(\prod_{k=i}^{j-1} \text{Exp}(-\Delta\tilde{R}_{k+1j}^T J_r^k n_k^{gd} \Delta t)\right) \quad (3-22)$$

这里，要引入 Log 运算的一阶近似：

$$\text{Log}(\text{Exp}(\phi)\text{Exp}(\delta\phi)) \approx \phi + J_r^{-1}(\phi)\delta\phi \quad (3-23)$$

其中 $J_r^{-1}(\phi)$ 表示右雅克比矩阵的逆，右雅克比矩阵的计算方式如式 (3-15)。

对式子 (3-23) 重复使用 Log 运算的一阶近似，可以得到：

$$\delta\phi_{ij} = \sum_{k=i}^{j-1} \Delta\tilde{R}_{k+1j}^T J_r^k n_k^{gd} \Delta t \quad (3-24)$$

由于 n_k^{gd} 是零均值的高斯白噪声，而 $\delta\phi_{ij}$ 是 n_k^{gd} 的线性加和，所以 $\delta\phi_{ij}$ 也是零均值的高斯白噪声。

对于预积分速度和预积分平移的噪声项，也做类似的分析：

$$\delta v_{ij} = -\sum_{k=i}^{j-1} \left[\Delta \tilde{R}_{ik} (\tilde{a}_k - b_i^a)^\wedge \delta \tilde{\phi}_{ik} \Delta t - \Delta \tilde{R}_{ik} n_k^{ad} \Delta t \right] \quad (3-25)$$

从上面式子 (3-24) 可以看出, 由于 $\delta \phi_{ik}$ 、 n_k^{ad} 是服从零均值的高斯分布, 所以 δv_{ij} 和 δp_{ij} 也是一系列零均值的高斯白噪声的加和, 所以也服从零均值的高斯分布, 有了这样结果, 便可以使用非线性最小二乘优化的方法对这些噪声进行处理了。对于 $\delta \phi_{ij}$ 、 δv_{ij} 和 δp_{ij} 可以写成下面的形式:

$$\begin{bmatrix} \delta \phi_{ij}^T & \delta v_{ij}^T & \delta p_{ij}^T \end{bmatrix}^T \sim N(0_{9 \times 1}, \Sigma_{ij}) \quad (3-26)$$

在进行实际的实现的时候, 通常会使用累加或者累成的方式计算上面的结果, 所以为了便于编程实现和计算协方差矩阵, 对于噪声的传播, 也需要推导出其递推形式。首先是预积分旋转量的噪声项:

$$\begin{aligned} \delta \phi_{ij} &= \sum_{k=i}^{j-1} \Delta \tilde{R}_{k+1j}^T J_r^k n_k^{gd} \Delta t \\ &= \sum_{k=i}^{j-2} \Delta \tilde{R}_{k+1j}^T J_r^k n_k^{gd} \Delta t + \Delta \tilde{R}_{jj}^T J_r^{j-1} n_{j-1}^{gd} \Delta t \\ &= \sum_{k=i}^{j-2} \left(\Delta \tilde{R}_{k+1j-1} \Delta \tilde{R}_{j-1j} \right)^T J_r^k n_k^{gd} \Delta t + J_r^{j-1} n_{j-1}^{gd} \Delta t \\ &= \Delta \tilde{R}_{j-1j}^T \sum_{k=i}^{j-2} \Delta \tilde{R}_{k+1j-1}^T J_r^k n_k^{gd} \Delta t + J_r^{j-1} n_{j-1}^{gd} \Delta t \\ &= \Delta \tilde{R}_{j-1j}^T \delta \phi_{ij-1} + J_r^{j-1} n_{j-1}^{gd} \Delta t \end{aligned} \quad (3-27)$$

从上面式子可以看出, 当前时刻的噪声值等于前一时刻的噪声值乘上一个旋转矩阵, 然后再线性叠加噪声强度。利用这样的结果, 我们可以很方便的就算出协方差矩阵。利用类似的思想, 接下来是预积分速度和预积分平移的噪声项的处理:

$$\begin{aligned} \delta v_{ij} &= -\sum_{k=i}^{j-1} \left[\Delta \tilde{R}_{ik} (\tilde{a}_k - b_i^a)^\wedge \delta \tilde{\phi}_{ik} \Delta t - \Delta \tilde{R}_{ik} n_k^{ad} \Delta t \right] \\ &= -\sum_{k=i}^{j-2} \left[\Delta \tilde{R}_{ik} (\tilde{a}_k - b_i^a)^\wedge \delta \tilde{\phi}_{ik} \Delta t - \Delta \tilde{R}_{ik} n_k^{ad} \Delta t \right] \\ &\quad - \Delta \tilde{R}_{ij-1} (\tilde{a}_{j-1} - b_i^a)^\wedge \delta \tilde{\phi}_{ij-1} \Delta t + \Delta \tilde{R}_{ij-1} n_{j-1}^{ad} \Delta t \\ &= \delta v_{ij-1} - \Delta \tilde{R}_{ij-1} (\tilde{a}_{j-1} - b_i^a)^\wedge \delta \tilde{\phi}_{ij-1} \Delta t + \Delta \tilde{R}_{ij-1} n_{j-1}^{ad} \Delta t \end{aligned} \quad (3-28)$$

$$\begin{aligned}
 \delta p_{ij} &= -\sum_{k=i}^{j-1} \left[\frac{3}{2} \tilde{R}_{ik} (\tilde{a}_k - b_k^a)^\wedge \delta \phi_{ik} \Delta t^2 - \frac{3}{2} \tilde{R}_{ik} n_k^{ad} \Delta t^2 \right] \\
 &= \sum_{k=i}^{j-1} \left[\delta v_{ik} \Delta t - \frac{1}{2} \tilde{R}_{ik} (\tilde{a}_k - b_k^a)^\wedge \delta \phi_{ik} \Delta t^2 + \frac{1}{2} \tilde{R}_{ik} n_k^{ad} \Delta t^2 \right] \\
 &= \sum_{k=i}^{j-2} \left[\delta v_{ik} \Delta t - \frac{1}{2} \tilde{R}_{ik} (\tilde{a}_k - b_k^a)^\wedge \delta \phi_{ik} \Delta t^2 + \frac{1}{2} \tilde{R}_{ik} n_k^{ad} \Delta t^2 \right] \\
 &\quad + \delta v_{ij-1} \Delta t - \frac{1}{2} \tilde{R}_{ij-1} (\tilde{a}_j - b_i^a)^\wedge \delta \phi_{ij-1} \Delta t^2 + \frac{1}{2} \tilde{R}_{ij-1} n_{j-1}^{ad} \Delta t^2 \\
 &= \delta p_{ij-1} + \delta v_{ij-1} \Delta t - \frac{1}{2} \tilde{R}_{ij-1} (\tilde{a}_j - b_i^a)^\wedge \delta \phi_{ij-1} \Delta t^2 + \frac{1}{2} \tilde{R}_{ij-1} n_{j-1}^{ad} \Delta t^2
 \end{aligned} \tag{3-29}$$

将上面的式子整理一下写成矩阵的形式：

$$\begin{bmatrix} \delta \phi_{ik+1} \\ \delta v_{ik+1} \\ \delta p_{ik+1} \end{bmatrix} = \begin{bmatrix} \Delta \tilde{R}_{kk+1}^T & 0_{3 \times 3} & 0_{3 \times 3} \\ -\Delta \tilde{R}_{ik} (\tilde{a}_k - b_i^a)^\wedge \Delta t & I_{3 \times 3} & 0_{3 \times 3} \\ -\frac{1}{2} \Delta \tilde{R}_{ik} (\tilde{a}_k - b_i^a)^\wedge \Delta t^2 & I_{3 \times 3} \Delta t & I_{3 \times 3} \end{bmatrix} \begin{bmatrix} \delta \phi_{ik} \\ \delta v_{ik} \\ \delta p_{ik} \end{bmatrix} + \begin{bmatrix} J_r^k \Delta t & 0_{3 \times 3} \\ 0_{3 \times 3} & \Delta \tilde{R}_{ik} \Delta t \\ 0_{3 \times 3} & \frac{1}{2} \Delta \tilde{R}_{ik} \Delta t^2 \end{bmatrix} \begin{bmatrix} n_k^{gd} \\ n_k^{ad} \end{bmatrix} \tag{3-30}$$

将上面式子 (3-30) 可以简写成下面式子 (3-31)：

$$n_{ik+1}^\Delta = A n_{ik}^\Delta + B n_k^d \tag{3-31}$$

其中 $n_{ik}^\Delta = [\delta \phi_{ik} \quad \delta v_{ik} \quad \delta p_{ik}]^T$ ， $n_k^d = [n_k^{gd} \quad n_k^{ad}]$ 。那么，两帧图像帧时刻之间 IMU 信息噪声的协方差矩阵 Σ_{ij} 就可以按照下面方式去计算：

$$\Sigma_{ik+1} = A \Sigma_{ik} A^T + B \Sigma_n B^T \tag{3-32}$$

其中 $\Sigma_n \in R^{6 \times 6}$ ，是 IMU 的原始测量噪声矩阵，用下面式子 () 的方式去计算：

$$\Sigma_n = \begin{bmatrix} I_{3 \times 3} \sigma_{gd} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_{3 \times 3} \sigma_{ad} \end{bmatrix} \tag{3-33}$$

其中 σ_{gd} 和 σ_{ad} 分别是陀螺仪和加速度计的离散噪声强度。

3.3.3. 随机游走 bias 的更新

在前面 3.3.1 节中提到，将 IMU 信息进行预积分后，得到预积分结果已经与前面时刻的位姿、IMU 信息无关，至于噪声项和陀螺仪、加速度计的 bias 有关。在 3.3.2 节中则分析了预积分环节中噪声的传播，明确了噪声项对预积分项的影响。在本节中，将分析一下加速度计和陀螺仪的 bias 对预积分项

的影响，以及 bias 的更新是如何影响位姿状态量更新的。

现在假设对 bias 的值进行了一次更新： $\hat{b}_i \leftarrow \bar{b}_i + \delta b_i$ ，现在的任务便是分析这时 $\Delta \tilde{R}_{ij}$ 、 $\Delta \tilde{v}_{ij}$ 和 $\Delta \tilde{p}_{ij}$ 是如何更新的。

首先是预积分旋转量的更新，预积分旋转量只和陀螺仪的 bias 有关，和加速度计的 bias 无关，可以通过对 $\Delta \tilde{R}_{ij}$ 一阶展开，求取 bias 更新时 $\Delta \tilde{R}_{ij}$ 的改变量：

$$\begin{aligned}
 \Delta \tilde{R}_{ij}(\hat{b}_i) &= \prod_{k=i}^{j-1} [Exp((\tilde{\omega}_k - \hat{b}_i^g) \Delta t)] \\
 &= \prod_{k=i}^{j-1} [Exp((\tilde{\omega}_k - \bar{b}_i^g - \delta b_i^g) \Delta t)] \\
 &= \prod_{k=i}^{j-1} [Exp((\tilde{\omega}_k - \bar{b}_i^g) \Delta t) Exp(-J_r^k \delta b_i^g \Delta t)] \\
 &= \Delta \tilde{R}_{ij}(\bar{b}_i) \prod_{k=i}^{j-1} [Exp(-\Delta \tilde{R}_{k+1,j}^T(\bar{b}_i) J_r^k \delta b_i^g \Delta t)] \\
 &= \Delta \tilde{R}_{ij}(\bar{b}_i) Exp\left(\sum_{k=i}^{j-1} [-\Delta \tilde{R}_{k+1,j}^T(\bar{b}_i) J_r^k \Delta t] \delta b_i^g\right) \\
 &= \Delta \tilde{R}_{ij}(\bar{b}_i) Exp\left(\frac{\partial \Delta \tilde{R}_{ij}}{\partial b_i^g} \delta b_i^g\right)
 \end{aligned} \tag{3-34}$$

在上面式子 (3-34) 表示出了陀螺仪 bias 更新时预积分旋转项的更新方法，接下来继续介绍预积分速度项和预积分平移项的更新方法：

$$\begin{aligned}
 \Delta \tilde{v}_{ij}(\hat{b}_i) &= \sum_{k=i}^{j-1} \Delta \tilde{R}_{ik}(\bar{b}_i) Exp\left(\frac{\partial \Delta \tilde{R}_{ik}}{\partial b_i^g}\right) (\tilde{a}_k - \bar{b}_i^a - \delta b_i^a) \Delta t \\
 &\approx \sum_{k=i}^{j-1} \Delta \tilde{R}_{ik}(\bar{b}_i) \left(I + \left(\frac{\partial \Delta \tilde{R}_{ik}}{\partial b_i^g} \delta b_i^g \right)^\wedge \right) (\tilde{a}_k - \bar{b}_i^a - \delta b_i^a) \Delta t \\
 &\approx \sum_{k=i}^{j-1} \Delta \tilde{R}_{ik}(\bar{b}_i) (\tilde{a}_k - \bar{b}_i^a) \Delta t + \sum_{k=i}^{j-1} \Delta \tilde{R}_{ik}(\bar{b}_i) (-\delta b_i^a) \Delta t + \sum_{k=i}^{j-1} \Delta \tilde{R}_{ik}(\bar{b}_i) \left(\frac{\partial \Delta \tilde{R}_{ik}}{\partial b_i^g} \delta b_i^g \right)^\wedge (\tilde{a}_k - \bar{b}_i^a) \Delta t \\
 &= \Delta \tilde{v}_{ij}(\bar{b}_i) + \left(\sum_{k=i}^{j-1} -\Delta \tilde{R}_{ik}(\bar{b}_i) \Delta t \right) \delta b_i^a + \left(\sum_{k=i}^{j-1} -\Delta \tilde{R}_{ik}(\bar{b}_i) (\tilde{a}_k - \bar{b}_i^a)^\wedge \frac{\partial \Delta \tilde{R}_{ik}}{\partial b_i^g} \Delta t \right) \delta b_i^g \\
 &= \Delta \tilde{v}_{ij}(\bar{b}_i) + \frac{\partial \Delta \tilde{v}_{ij}}{\partial b_i^a} \delta b_i^a + \frac{\partial \Delta \tilde{v}_{ij}}{\partial b_i^g} \delta b_i^g
 \end{aligned} \tag{3-35}$$

$$\begin{aligned}
 \Delta \tilde{p}_{ij}(\hat{b}_i) &= \sum_{k=i}^{j-1} \frac{3}{2} \Delta \tilde{R}_{ik}(\bar{b}_i) \text{Exp} \left(\frac{\partial \Delta \tilde{R}_{ik}}{\partial b^g} \delta b_i^g \right) (\tilde{a}_k - \bar{b}_i^a - \delta b_i^a) \Delta t^2 \\
 &\approx \sum_{k=i}^{j-1} \frac{3}{2} \Delta \tilde{R}_{ik}(\bar{b}_i) \left(I + \left(\frac{\partial \Delta \tilde{R}_{ik}}{\partial b^g} \delta b_i^g \right)^\wedge \right) (\tilde{a}_k - \bar{b}_i^a - \delta b_i^a) \Delta t^2 \\
 &\approx \Delta \tilde{p}_{ij}(\bar{b}_i) + \sum_{k=i}^{j-1} \frac{3}{2} \Delta \tilde{R}_{ik}(\bar{b}_i) (-\delta b_i^a) \Delta t^2 + \sum_{k=i}^{j-1} \frac{3}{2} \Delta \tilde{R}_{ik}(\bar{b}_i) \left(\frac{\partial \Delta \tilde{R}_{ik}}{\partial b^g} \delta b_i^g \right)^\wedge (\tilde{a}_k - \bar{b}_i^a) \Delta t^2 \\
 &= \Delta \tilde{p}_{ij}(\bar{b}_i) + \left(\sum_{k=i}^{j-1} -\frac{3}{2} \Delta \tilde{R}_{ik}(\bar{b}_i) \Delta t^2 \right) \delta b_i^a + \left(\sum_{k=i}^{j-1} -\frac{3}{2} \Delta \tilde{R}_{ik}(\bar{b}_i) (\tilde{a}_k - \bar{b}_i^a)^\wedge \frac{\partial \Delta \tilde{R}_{ik}}{\partial b^g} \Delta t^2 \right) \delta b_i^g \\
 &= \Delta \tilde{p}_{ij}(\bar{b}_i) + \frac{\partial \Delta \tilde{p}_{ij}}{\partial b^a} \delta b_i^a + \frac{\partial \Delta \tilde{p}_{ij}}{\partial b^g} \delta b_i^g
 \end{aligned} \tag{3-36}$$

以上便是预积分旋转项、预积分速度项以及预积分平移项对 **bias** 的一阶展开，当 **bias** 发生微小的改变时，可以利用上面的展开项对预积分项进行相应的改变，最后，将上面结果整理一下，可以得到如下的结论：

$$\begin{aligned}
 \Delta \tilde{R}_{ij}(\hat{b}_i) &= \Delta \tilde{R}_{ij}(\bar{b}_i) \text{Exp} \left(\frac{\partial \Delta \tilde{R}_{ij}}{\partial b^g} \delta b_i^g \right) \\
 \Delta \tilde{v}_{ij}(\hat{b}_i) &= \Delta \tilde{v}_{ij}(\bar{b}_i) + \frac{\partial \Delta \tilde{v}_{ij}}{\partial b^a} \delta b_i^a + \frac{\partial \Delta \tilde{v}_{ij}}{\partial b^g} \delta b_i^g \\
 \Delta \tilde{p}_{ij}(\hat{b}_i) &= \Delta \tilde{p}_{ij}(\bar{b}_i) + \frac{\partial \Delta \tilde{p}_{ij}}{\partial b^a} \delta b_i^a + \frac{\partial \Delta \tilde{p}_{ij}}{\partial b^g} \delta b_i^g
 \end{aligned} \tag{3-37}$$

其中：

$$\begin{aligned}
 \frac{\partial \Delta \tilde{R}_{ij}}{\partial b^g} &= \sum_{k=i}^{j-1} -\Delta \tilde{R}_{k+1j}^T(\bar{b}_i) J_r^k \Delta t \\
 \frac{\partial \Delta \tilde{v}_{ij}}{\partial b^a} &= \sum_{k=i}^{j-1} -\Delta \tilde{R}_{ik}(\bar{b}_i) \Delta t \\
 \frac{\partial \Delta \tilde{v}_{ij}}{\partial b^g} &= \sum_{k=i}^{j-1} -\Delta \tilde{R}_{ik}(\bar{b}_i) (\tilde{a}_k - \bar{b}_i^a)^\wedge \frac{\partial \Delta \tilde{R}_{ik}}{\partial b^g} \Delta t \\
 \frac{\partial \Delta \tilde{p}_{ij}}{\partial b^a} &= \sum_{k=i}^{j-1} -\frac{3}{2} \Delta \tilde{R}_{ik}(\bar{b}_i) \Delta t^2 \\
 \frac{\partial \Delta \tilde{p}_{ij}}{\partial b^g} &= \sum_{k=i}^{j-1} -\frac{3}{2} \Delta \tilde{R}_{ik}(\bar{b}_i) (\tilde{a}_k - \bar{b}_i^a)^\wedge \frac{\partial \Delta \tilde{R}_{ik}}{\partial b^g} \Delta t^2
 \end{aligned} \tag{3-38}$$

3.4. 本章小结

本章介绍了本次工作所用到的关于处理 IMU 信息的方法，包括对 IMU 建立噪声模型和运动学模型，以及对 IMU 测量到的加速度和角速度进行预积分。利用预积分，可以解决 IMU 信息和视觉信息频率不相同的问题，同时解决了位姿优化时需要重新积分的问题。然后，还对 IMU 信息的误差传播进行了分析，并给出了协方差矩阵的递推求法。最后，还分析了随机游走 bias 对预积分项的影响，为后续的视觉信息和 IMU 信息融合奠定了基础。

第 4 章 视觉和 IMU 信息融合

4.1. 引言

第 2 章给出了相机的模型和处理视觉信息的方法, 第 3 章给出了 IMU 的噪声模型和运动学模型, 以及对 IMU 进行预积分的方法, 本章将在前两章的基础上对 IMU 和视觉信息利用非线性优化进行融合。首先, 将介绍一种用于系统初始化的方法, 用来估计重力向量、陀螺仪初始的随机游走 bias 以及初始的速度, 然后将介绍如何利用图优化的方法优化视觉的重投影误差和 IMU 的残差以获取更精确的位姿。之后将介绍如何将获得的地图保存, 以及如何在有地图的情况下进行定位。另外, 本次工作还使用了 IMU 的信息去预测位姿代替原来 ORB SLAM 的恒速模型, 但是因原理并不复杂, 所以不再展开叙述。

4.2. 系统初始化

系统初始化的目的在于求解一些系统的初始值, 包括初始的 bias 和速度, 以及重力向量。由于加速度计的 bias 本身很小, 并且受重力的影响很大, 所以本次工作初始化的时候只估计陀螺仪的初始 bias, 不估计加速度计的初始 bias, 认为加速度计的初始 bias 是 0。

初始化的过程就是一个视觉信息和 IMU 信息松耦合的过程, 初始化的时候, 会选取若干个图像帧, 用第 2 章介绍的方法计算出它们的位姿, 并利用第 3 章介绍的方法对相邻两帧图像之间的 IMU 信息进行预积分, 获取预积分旋转、速度和位移。然后再利用这两个传感器的信息解算得到陀螺仪的初始 bias、初始速度和重力向量。下面介绍具体的方法。

4.2.1. 估计初始的陀螺仪 bias

在本次初始化的过程中, 需要先选取 N 帧图像, 利用第 2 介绍的方法计算出相机坐标系相对于世界坐标系的平移 p_{WC}^i 和旋转 R_{WC}^i , 这里使用下标 C 表示相机坐标系, 再利用下面的式子得到 IMU 坐标系相对于世界坐标系的平移和旋转:

$$\begin{aligned} R_{WB}^i &= R_{WC}^i R_{BC}^T \\ p_{WB}^i &= p_{WC}^i - R_{WB}^i p_{BC} \end{aligned} \quad (4-1)$$

其中 p_{BC} 和 R_{BC} 分别是相机坐标系相对于 IMU 坐标系的平移和旋转，这两个参数称为相机和 IMU 的外参数，可以通过标定的方法获得，这个在后面章节会有所介绍，这里先假设已经获得这两个参数。

获取了 N 个图像帧时刻 IMU 坐标系相对于世界坐标系相对于世界坐标系的平移和旋转后，可以在利用第 3 章介绍的方法对相邻两帧图像之间的 IMU 数据进行预积分获得预积分旋转 $\Delta \tilde{R}_{ii+1}$ 、预积分速度 Δv_{ii+1} 和预积分平移项 Δp_{ii+1} 。利用下面式子 (4-2) 获取初始的陀螺仪 bias:

$$b_g = \arg \min_{b_g} \sum_{i=1}^{N-1} \left\| \text{Log} \left(\left(R_{BW}^{i+1} \right)^T R_{WB}^i \Delta \tilde{R}_{ii+1} \text{Exp} \left(\frac{\partial R_{ii+1}}{\partial b_g} b_g \right) \right) \right\|^2 \quad (4-2)$$

需要说明的是，在初始化阶段进行预积分计算的时候，代入的 bias 量都是 0。

对于 (4-2) 式的求解可以使用高斯-牛顿法。在本次工作的 C++ 实现中，使用的是 g2o 非线性优化库。将待优化量 b_g 作为唯一的顶点，每两帧之间的数据可以构成一个边，求解出初始的陀螺仪 bias。

4.2.2. 估计初始速度和重力向量

对于初始化过程中 N 个图像帧时刻的速度，这里用 $v_1, \dots, v_i, \dots, v_N$ 表示，需要说明的是，这些速度都表示的是在世界坐标系下 IMU 坐标系的速度。根据预积分计算的方法，可以有下面式子 (4-3)：

$$\begin{aligned} \Delta v_{ij} &= R_i^T (v_j - v_i - g \Delta t_{ij}) \\ \Delta p_{ij} &= R_i^T \left(p_j - p_i - v_i \Delta t_{ij} - \frac{1}{2} g \Delta t_{ij}^2 \right) \end{aligned} \quad (4-3)$$

对式子 (4-3) 稍微做一下改变，可以得到：

$$\begin{aligned} v_i - v_{i+1} + g \Delta t_{ii+1} &= R_{WB}^i \Delta v_{ii+1} \\ \Delta t_{ii+1} v_i + \frac{1}{2} \Delta t_{ii+1}^2 g &= p_i + R_{WB}^i \Delta p_{ii+1} - p_{i+1} \end{aligned} \quad (4-4)$$

将式子 (4-3) 写成一个矩阵的形式：

$$\begin{bmatrix} I_{3 \times 3} & -I_{3 \times 3} & \Delta t_{ii+1} \\ \Delta t_{ii+1} & 0_{3 \times 3} & \frac{1}{2} \Delta t_{ii+1}^2 \end{bmatrix} \begin{bmatrix} v_i \\ v_{i+1} \\ g \end{bmatrix} = \begin{bmatrix} R_{WB}^i \Delta v_{ii+1} \\ p_i + R_{WB}^i \Delta p_{ii+1} - p_{i+1} \end{bmatrix} \quad (4-5)$$

对于 N 个图像帧时刻，一共可以得到 $N-1$ 个式子 (4-5)，将其写成一个大矩阵的形式：

$$\begin{bmatrix} I_{3 \times 3} & -I_{3 \times 3} & 0_{3 \times 3} & \cdots & \cdots & 0_{3 \times 3} & \Delta t_{12} \\ \Delta t_{12} & 0_{3 \times 3} & 0_{3 \times 3} & \cdots & \cdots & 0_{3 \times 3} & \frac{1}{2} \Delta t_{12}^2 \\ 0_{3 \times 3} & I_{3 \times 3} & -I_{3 \times 3} & 0_{3 \times 3} & \cdots & 0_{3 \times 3} & \Delta t_{23} \\ 0_{3 \times 3} & \Delta t_{23} & 0_{3 \times 3} & 0_{3 \times 3} & \cdots & 0_{3 \times 3} & \frac{1}{2} \Delta t_{23}^2 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0_{3 \times 3} & \cdots & \cdots & 0_{3 \times 3} & I_{3 \times 3} & -I_{3 \times 3} & \Delta t_{N-1N} \\ 0_{3 \times 3} & \cdots & \cdots & 0_{3 \times 3} & \Delta t_{N-1N} & 0_{3 \times 3} & \frac{1}{2} \Delta t_{N-1N}^2 \end{bmatrix}_{(2N-2) \times (N+1)} \cdot \begin{bmatrix} v_1 \\ v_2 \\ \cdots \\ v_N \\ g \end{bmatrix}_{(N+1) \times 1} = \begin{bmatrix} R_{WB}^1 \Delta v_{12} \\ p_1 + R_{WB}^1 \Delta p_{12} - p_2 \\ R_{WB}^2 \Delta v_{23} \\ p_2 + R_{WB}^2 \Delta p_{23} - p_3 \\ \cdots \\ R_{WB}^{N-1} \Delta v_{N-1N} \\ p_{N-1} + R_{WB}^{N-1} \Delta p_{N-1N} - p_N \end{bmatrix}_{(2N-2) \times 1} \quad (4-6)$$

对于上面方程的求解，可以利用 SVD 奇异值分解的方法，求得的解便是初始速度和重力向量。

对于重力向量，可以通过标定的方法事先得到其模的大小，但是式子 (4-6) 并没有利用这一信息。如果直接将重力大小约束添加到式子 (4-6) 中，那么该式将变成一个非线性的方程，变得不容易求解。为此，本次工作初始化的时候采用下面的方式去运用重力大小这一约束：首先假设通过标定步骤得到的重力大小为 G ，那么，重力向量可以被参数化成这样：

$$g = G\bar{g} + \omega_1 b_1 + \omega_2 b_2 \quad (4-7)$$

其中 \bar{g} 是重力向量的方向， b_1 和 b_2 是和重量向量方向 \bar{g} 垂直的两个相互垂直的单位向量，即 \bar{g} 、 b_1 和 b_2 两两相互垂直。 ω_1 、 ω_2 分别是 b_1 和 b_2 方向上的扰动。对于 b_1 和 b_2 的求解，可以使用如下方法：

$$b_1 = \begin{cases} \text{normalize}(\bar{\hat{g}} \times [1 & 0 & 0]) & \text{if } \bar{\hat{g}} \neq [1 & 0 & 0] \\ \text{normalize}(\bar{\hat{g}} \times [0 & 0 & 1]) & \text{if } \bar{\hat{g}} \equiv [1 & 0 & 0] \end{cases} \quad (4-8)$$

$$b_2 = \bar{\hat{g}} \times b_1$$

那么，现在求解重力向量和初始速度的步骤变成这样：首先求解方程（4-6）得到重力的方向，然后通过式子（4-8）求解 b_1 和 b_2 ，得到 b_1 和 b_2 后，利用式子（4-7）得到参数化的 g ，将其再代到方程（4-6）中重新求解，如此迭代，直到 g 收敛，便可求得重力向量 g 和初始的速度。系统的初始化完成。

4.3. 非线性优化

上节介绍的初始化工作求解出了初始的陀螺仪 bias、初始速度和重力向量，接下来便可以进行位姿的解算工作了。对于位姿的解算，本次工作还是采用求解帧与帧相对位姿然后积分的方法求得与第一帧的相对位姿。如果将第一帧的相机坐标系或者 IMU 坐标系作为世界坐标系的话，就可以得到在各个图像帧时刻相机坐标系或者 IMU 坐标系在世界坐标系下的位姿。

4.3.1. 状态量和误差项

对于相邻两帧之间的位姿解算，本次工作使用非线性优化的方法融合 IMU 信息和视觉特征信息，使得位姿的求解工作更具鲁棒性。首先，先定义需要优化求解的状态量：

$$\theta_i = \{R_{WB}^i, v_{WB}^i, p_{WB}^i, b_i^g, b_i^a\} \quad (4-9)$$

θ_i 便是在图像帧 i 需要求解和优化的状态量， R_{WB}^i 、 v_{WB}^i 和 p_{WB}^i 表示图像帧 i 时刻 IMU 坐标系相对于世界坐标系的旋转、速度和位移， b_i^g 和 b_i^a 分别表示图像帧 i 时刻陀螺仪和加速度计的随机游走 bias。

下面开始分析误差项。误差项包括四个部分：单目点的误差项、双目点的误差项、IMU 残差项和 bias 误差项，下面分别对这四种误差项进行说明。

首先是单目点的误差项。所谓单目点，就是只在左右目相机一个里面被检测到的特征点，由于无法进行双目匹配和测距，所以这类特征点只有二维坐标，无法获取三维坐标。对于这种特征点，本次使用 2.5.3 节介绍的重投影误差作为其误差项。即：

$$E_{mono}^i = \sum \rho((x_{mono} - \pi(X_C))^T \Sigma_x (x_{mono} - \pi(X_C))) \quad (4-10)$$

$$X_C = R_{CB} R_{BW}^i (X_W - p_{WB}^i) + p_{CB}$$

其中 $\rho(\cdot)$ 表示 Huber 损失函数。

而双目点便是双目匹配成功的特征点，这类特征点可以通过双目测距的原理获取深度，所以在度量误差的时候也把深度信息考虑在内，对于某一个双目点，假设其在左目上的图像坐标为 $[u_{stereo-l} \ v_{stereo-l}]^T$ ，其在右目上的横坐标为 $u_{stereo-r}$ ，与其关联的地图点在世界坐标系世界坐标系下的坐标为 X_W ，那么这个特征点的误差项用下面式子（4-11）去计算：

$$e_{stereo}^i = \begin{bmatrix} u_{stereo-l} - \left(\frac{x_C \cdot f_x}{z_C} + c_x \right) \\ v_{stereo-l} - \left(\frac{y_C \cdot f_y}{z_C} + c_y \right) \\ u_{stereo-r} - \left(x_C - \frac{b \cdot f_x}{z_C} \right) \end{bmatrix}_{3 \times 1} \quad (4-11)$$

$$X_C = [x_C \ y_C \ z_C]^T = R_{CW}^i X_W + p_{CW}^i$$

$$R_{CW}^i = R_{CB} (R_{WB}^i)^T \quad p_{CW}^i = p_{CB} - (R_{WB}^i)^T p_{WB}^i$$

上面式子中， f_x 、 f_y 、 c_x 、 c_y 为相机的内参数，由标定步骤获得， b 为双目相机的基线长，也是通过标定步骤获得的。

上面是单个双目点的误差项，对所有双目点的误差项求和可以得到在图像帧 i 时刻双目点的误差项：

$$E_{stereo}^i = \sum \rho(e_{stereo}^i \Sigma e_{stereo}^i) \quad (4-12)$$

其中 $\rho(\cdot)$ 表示 Huber 损失函数。

然后便是 IMU 残差项。IMU 残差项包括 3 个部分：预积分旋转残差、预积分速度残差和预积分平移残差，预积分残差项的意义是利用 IMU 的信息约束前后两帧的状态量，预积分旋转残差项表征利用 IMU 预积分旋转量约束前后两帧的旋转矩阵，预积分速度残差项是利用 IMU 预积分速度量约束前后两帧的速度，预积分平移残差项表征利用 IMU 预积分旋转量约束前后两帧的平移矩阵。将这三个残差项写成一个向量的形式：

$$\begin{aligned}
 e_{preimu}^{i-1,i} &= [e_{\Delta R}^{i-1,i} \quad e_{\Delta v}^{i-1,i} \quad e_{\Delta p}^{i-1,i}]^T \\
 e_{\Delta R}^{i-1,i} &= \text{Log} \left(\left(\Delta \tilde{R}_{i-1,i}(\bar{b}_{i-1}^g) \text{Exp} \left(\frac{\partial \Delta \tilde{R}_{i-1,i}}{\partial b^g} \delta b^g \right) \right)^T R_{i-1}^T R_i \right) \\
 e_{\Delta v}^{i-1,i} &= R_{i-1}^T (v_i - v_{i-1} - g \Delta t_{i-1,i}) - \left[\Delta \tilde{v}_{ij}(\bar{b}_{i-1}^g, \bar{b}_{i-1}^a) + \frac{\partial \Delta \tilde{v}_{i-1,i}}{\partial b^g} \delta b^g + \frac{\partial \Delta \tilde{v}_{i-1,i}}{\partial b^a} \right] \\
 e_{\Delta p}^{i-1,i} &= R_{i-1}^T \left(p_i - p_{i-1} - v_{i-1} \Delta t_{i-1,i} - \frac{1}{2} g \Delta t_{i-1,i}^2 \right) - \left[\Delta \tilde{p}_{ij}(\bar{b}_{i-1}^g, \bar{b}_{i-1}^a) + \frac{\partial \Delta \tilde{p}_{i-1,i}}{\partial b^g} \right]
 \end{aligned} \quad (4-13)$$

所以，IMU 残差可以用下面式子去表示：

$$E_{preimu}^{i-1,i} = \rho \left((e_{preimu}^{i-1,i})^T \Sigma_{preimu}^{i-1,i} (e_{preimu}^{i-1,i}) \right) \quad (4-14)$$

其中 $\rho(\cdot)$ 表示 Huber 损失函数， $\Sigma_{preimu}^{i-1,i}$ 为 3.3.2 节推导出的协方差矩阵。

最后一个误差项是 bias 的误差。因为在 3.2.1 节中，陀螺仪和加速度计的 bias 被建模成了白噪声驱动项，所以，可以利用前一帧 bias 去约束当前帧的 bias：

$$\begin{aligned}
 E_b^{i-1,i} &= \rho \left((e_b^{i-1,i})^T \Sigma_b^{i-1,i} (e_b^{i-1,i}) \right) \\
 e_b^{i-1,i} &= \begin{bmatrix} b_i^g - b_{i-1}^g \\ b_i^a - b_{i-1}^a \end{bmatrix} \\
 \Sigma_b^{i-1,i} &= \begin{bmatrix} I_{3 \times 3} \sigma_{bgd} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_{3 \times 3} \sigma_{bad} \end{bmatrix}
 \end{aligned} \quad (4-15)$$

其中 $\rho(\cdot)$ 表示 Huber 损失函数， $\Sigma_b^{i-1,i}$ 为协方差矩阵。

上面便是本次工作所使用的状态量和误差项，可以通过优化误差项来求解最优的状态量 θ_i ：

$$\theta_i = \arg \min_{\theta} (E_{mono}^i + E_{stereo}^i + E_{preimu}^{i-1,i} + E_b^{i-1,i}) \quad (4-16)$$

对于上面式子 (4-16) 求解，可以使用高斯牛顿法或者列文伯格-马夸尔特方法，这将在后面章节介绍。需要说明的是，在优化式子 (4-16) 的时候，第 $i-1$ 帧时刻的状态量是作为先验信息的，在优化过程中保持不变。

4.3.2. IMU 残差项的雅克比矩阵

当使用高斯牛顿法或者列文伯格-马夸尔特方法求解式子 (4-16) 时，需

要用到 IMU 残差对状态量的雅克比矩阵，本章节的主要任务便是求解这些雅克比矩阵。

首先是预积分旋转残差项的雅克比矩阵。由式子 (4-13) 可以看出，预积分旋转残差项只和状态量中的 R_i 、 b_i^g 有关，与 p_i 、 v_i 以及 b_i^a 无关。首先先求解预积分旋转残差项对 R_i 的雅克比矩阵：

$$e_{\Delta R}^{i-1,i}(R_i \text{Exp}(\delta\phi_i)) = \text{Log} \left(\left(\Delta\tilde{R}_{i-1,i}(\bar{b}_{i-1}^g) \text{Exp} \left(\frac{\partial \Delta\tilde{R}_{i-1,i}}{\partial b^g} \delta b^g \right) \right)^T R_{i-1}^T R_i \text{Exp}(\delta\phi_i) \right) \quad (4-17)$$

在此，引入下面式子 (4-18)：

$$\text{Log}(\text{Exp}(\phi) \text{Exp}(\delta\phi)) \approx \phi + J_r^{-1}(\phi) \delta\phi \quad (4-18)$$

那么，可以得到：

$$e_{\Delta R}^{i-1,i}(R_i \text{Exp}(\delta\phi_i)) = e_{\Delta R}^{i-1,i}(R_i) + J_r^{-1}(e_{\Delta R}^{i-1,i}(R_i)) \delta\phi_i \quad (4-19)$$

预积分旋转残差项对 R_i 的雅克比矩阵求解完毕，接下来是对 b_i^g 的雅克比矩阵：

$$\begin{aligned} e_{\Delta R}^{i-1,i}(\delta b_i^g + \bar{\delta} b_i^g) &= \text{Log} \left(\left(\Delta\tilde{R}_{i-1,i}(\bar{b}_{i-1}^g) \text{Exp} \left(\frac{\partial \Delta\tilde{R}_{i-1,i}}{\partial b^g} (\delta b_i^g + \bar{\delta} b_i^g) \right) \right)^T R_{i-1}^T R_i \right) \\ &\approx \text{Log} \left(\left(\Delta\tilde{R}_{i-1,i}(\bar{b}_{i-1}^g) \text{Exp} \left(\frac{\partial \Delta\tilde{R}_{i-1,i}}{\partial b^g} \delta b_i^g \right) \text{Exp} \left(J_r \left(\frac{\partial \Delta\tilde{R}_{i-1,i}}{\partial b^g} \delta b_i^g \right) \frac{\partial \Delta\tilde{R}_{i-1,i}}{\partial b^g} \bar{\delta} b_i^g \right) \right)^T R_{i-1}^T R_i \right) \\ &= \text{Log} \left(\text{Exp} \left(-J_r \left(\frac{\partial \Delta\tilde{R}_{i-1,i}}{\partial b^g} \delta b_i^g \right) \frac{\partial \Delta\tilde{R}_{i-1,i}}{\partial b^g} \bar{\delta} b_i^g \right) \left(\Delta\tilde{R}_{i-1,i}(\bar{b}_{i-1}^g) \text{Exp} \left(\frac{\partial \Delta\tilde{R}_{i-1,i}}{\partial b^g} \delta b_i^g \right) \right)^T R_{i-1}^T R_i \right) \\ &= \text{Log} \left(\text{Exp} \left(-J_r \left(\frac{\partial \Delta\tilde{R}_{i-1,i}}{\partial b^g} \delta b_i^g \right) \frac{\partial \Delta\tilde{R}_{i-1,i}}{\partial b^g} \bar{\delta} b_i^g \right) \text{Exp}(e_{\Delta R}^{i-1,i}(\delta b_i^g)) \right) \\ &= \text{Log} \left(\text{Exp}(e_{\Delta R}^{i-1,i}(\delta b_i^g)) \text{Exp} \left(-\text{Exp}(e_{\Delta R}^{i-1,i}(\delta b_i^g))^T J_r \left(\frac{\partial \Delta\tilde{R}_{i-1,i}}{\partial b^g} \delta b_i^g \right) \frac{\partial \Delta\tilde{R}_{i-1,i}}{\partial b^g} \bar{\delta} b_i^g \right) \right) \\ &= e_{\Delta R}^{i-1,i}(\delta b_i^g) - J_r^{-1}(e_{\Delta R}^{i-1,i}(\delta b_i^g)) \text{Exp}(e_{\Delta R}^{i-1,i}(\delta b_i^g))^T J_r \left(\frac{\partial \Delta\tilde{R}_{i-1,i}}{\partial b^g} \delta b_i^g \right) \frac{\partial \Delta\tilde{R}_{i-1,i}}{\partial b^g} \bar{\delta} b_i^g \end{aligned} \quad (4-20)$$

上面的结果便是预积分旋转残差项对 b_i^g 的雅克比矩阵，最后，将结果整理一下，可以写出预积分旋转残差项对状态量 θ_i 的雅克比矩阵：

$$\begin{aligned}
 \frac{\partial e_{\Delta R}^{i-1,i}}{\partial \phi_i} &= J_r^{-1}(e_{\Delta R}^{i-1,i}(R_i)) \\
 \frac{\partial e_{\Delta R}^{i-1,i}}{\partial v_i} &= 0 \\
 \frac{\partial e_{\Delta R}^{i-1,i}}{\partial p_i} &= 0 \\
 \frac{\partial e_{\Delta R}^{i-1,i}}{\partial b_i^g} &= -J_r^{-1}(e_{\Delta R}^{i-1,i}(\delta b_i^g)) \text{Exp}(e_{\Delta R}^{i-1,i}(\delta b_i^g))^T J_r \left(\frac{\partial \Delta \tilde{R}_{i-1,i}}{\partial b^g} \delta b_i^g \right) \frac{\partial \Delta \tilde{R}_{i-1,i}}{\partial b^g} \\
 \frac{\partial e_{\Delta R}^{i-1,i}}{\partial b_i^a} &= 0
 \end{aligned} \tag{4-21}$$

接下来是预积分速度残差项：

$$\begin{aligned}
 e_{\Delta v}^{i-1,i}(v_i + \delta v_i) &= R_{i-1}^T(v_i + \delta v_i - v_{i-1} - g\Delta t_{i-1,i}) - \left[\Delta \tilde{v}_{ij}(\bar{b}_{i-1}^g, \bar{b}_{i-1}^a) + \frac{\partial \Delta \tilde{v}_{i-1,i}}{\partial b^g} \delta b^g + \frac{\partial \Delta \tilde{v}_{i-1,i}}{\partial b^a} \delta b^a \right] \\
 &= e_{\Delta v}^{i-1,i}(v_i) + R_{i-1}^T \delta v_i
 \end{aligned} \tag{4-22}$$

$$\begin{aligned}
 \frac{\partial e_{\Delta v}^{i-1,i}}{\partial \phi_i} &= 0 \\
 \frac{\partial e_{\Delta v}^{i-1,i}}{\partial v_i} &= R_{i-1}^T \\
 \frac{\partial e_{\Delta v}^{i-1,i}}{\partial p_i} &= 0 \\
 \frac{\partial e_{\Delta v}^{i-1,i}}{\partial b_i^g} &= -\frac{\partial \Delta \tilde{v}_{ij}}{\partial b_i^g} \\
 \frac{\partial e_{\Delta v}^{i-1,i}}{\partial b_i^a} &= -\frac{\partial \Delta \tilde{v}_{ij}}{\partial b_i^a}
 \end{aligned} \tag{4-23}$$

最后是预积分平移残差项的雅克比矩阵：

$$\begin{aligned}
 e_{\Delta p}^{i-1,i}(p_i + \delta p_i) &= R_{i-1}^T \left(p_i + \delta p_i - p_{i-1} - v_{i-1} \Delta t_{i-1,i} - \frac{1}{2} g \Delta t_{i-1,i}^2 \right) \\
 &\quad - \left[\Delta \tilde{p}_{ij}(\bar{b}_{i-1}^g, \bar{b}_{i-1}^a) + \frac{\partial \Delta \tilde{p}_{i-1,i}}{\partial b^g} \delta b^g + \frac{\partial \Delta \tilde{p}_{i-1,i}}{\partial b^a} \delta b^a \right] \\
 &= e_{\Delta p}^{i-1,i}(p_i) + R_{i-1}^T \delta p_i
 \end{aligned} \tag{4-24}$$

$$\begin{aligned}
 \frac{\partial e_{\Delta p}^{i-1,i}}{\partial \phi_i} &= 0 \\
 \frac{\partial e_{\Delta p}^{i-1,i}}{\partial v_i} &= 0 \\
 \frac{\partial e_{\Delta p}^{i-1,i}}{\partial p_i} &= R_{i-1}^T \\
 \frac{\partial e_{\Delta p}^{i-1,i}}{\partial b_i^g} &= -\frac{\partial \Delta \tilde{p}_{ij}}{\partial b_i^g} \\
 \frac{\partial e_{\Delta p}^{i-1,i}}{\partial b_i^a} &= -\frac{\partial \Delta \tilde{p}_{ij}}{\partial b_i^a}
 \end{aligned} \tag{4-25}$$

至此边求解出了预积分残差项对状态量 θ_i 的雅克比矩阵。

4.3.3. LM 算法

上面章节定义了一些需要优化求解的状态量和相应的误差项，接下来的目标便是寻找优化算法对上面的问题进行求解。本次工作采用 LM 算法（Levenberg-Marquardt，列文伯格马尔夸特方法）求解式子（4-16）获取最优的 θ_i ，本节将介绍具体一下 LM 算法。

LM 算法是基于高斯牛顿法的，在高斯牛顿法的基础上改进了因步长太大可能导致的迭代不收敛的问题，所以，这里将先介绍一下高斯牛顿法。首先，对于式子（4-16）将其简化为一个如下的最小二乘问题：

$$\min_x \frac{1}{2} \|f(x)\|_2^2 \tag{4-26}$$

其中 $x \in R^n$ 是待优化的状态变量， $f(\cdot)$ 表示某一非线性函数，在本次工作中代表误差项和残差项。

首先，将 $f(x)$ 在 \bar{x} 处进行一阶泰勒展开：

$$f(x) = f(\bar{x} + \Delta x) = f(\bar{x}) + J(\bar{x})\Delta x \tag{4-27}$$

其中 $J(\bar{x})$ 是 $f(x)$ 在 \bar{x} 处的雅克比矩阵。于是，式子（4-27）的最小二乘问题可以变成下面的线性最小二乘问题：

$$\Delta x^* = \arg \min_{\Delta x} \frac{1}{2} \|f(\bar{x}) + J(\bar{x})\Delta x\|^2 \tag{4-28}$$

为了求解这个方程，对右边的目标函数展开：

$$\begin{aligned}\frac{1}{2}\|f(\bar{x}) + J(\bar{x})\Delta x\|^2 &= \frac{1}{2}(f(\bar{x}) + J(\bar{x})\Delta x)^T(f(\bar{x}) + J(\bar{x})\Delta x) \\ &= \frac{1}{2}(\|f(\bar{x})\|_2^2 + 2f(\bar{x})^T J(\bar{x})\Delta x + \Delta x^T J(\bar{x})^T J(\bar{x})\Delta x)\end{aligned}\quad (4-29)$$

因为要优化的参数变成了 Δx ，所以将上面式子对 Δx 求导，可以得到下面的方程组：

$$J(\bar{x})^T J(\bar{x})\Delta x = -f(\bar{x})J(\bar{x})^T \quad (4-30)$$

上面的方程称为高斯牛顿方程，它是一个线性方程组，待求解的变量为 Δx 。对高斯牛顿方程进行一些简写：

$$\begin{aligned}H\Delta x &= g \\ H &= J(\bar{x})^T J(\bar{x}) \\ g &= -f(\bar{x})J(\bar{x})^T\end{aligned}\quad (4-31)$$

现在的问题变成了如何求解牛顿高斯方程，首先，可以使用下面的算法：

表 4-1 高斯牛顿法

算法 1
1. $H\Delta x = g$ 给出一个迭代的初值 x_0 ； 2. 计算雅克比矩阵 $J(x)$ 和误差函数 $f(x)$ ； 3. 求解方程； 4. 如果到达某一迭代次数或者 Δx 足够小，则迭代停止，否则令 $x \leftarrow x + \Delta x$ ，返回第 2 步继续进行迭代。

上面的算法被称为高斯牛顿法，通过高斯牛顿法可以求解式子（4-26）形式的非线性最小二乘问题。

但是，高斯牛顿法也存在一些问题。比如他要求 H 矩阵是正定可逆的，但实际计算的数据可能并不满足这一要求，此时再使用高斯牛顿法可能导致算法不收敛的情况。并且，就算 H 矩阵是正定可逆这一条件满足，算法求解出的步长 Δx 可能出现太大的情况，导致算法不收敛。

由于高斯牛顿法上面的缺点，本次工作使用 LM 算法。在 LM 算法中，给每次的增量 Δx 添加一个信赖域，当增长的步长 Δx 在这个信赖域里面的时候，认为这个 Δx 是有效的，否则认为其是无效的，可能会使迭代发散。基于这个思想，对算法 1 进行修改，得到下面的算法：

表 4-2 LM 算法

算法 2

1. 给出一个初值 x_0 以及初始信赖域半径 μ_0 ;

2. 求解最优化问题:

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

3. 按下面式子计算 ρ :

$$\rho = \frac{f(x + \Delta x) - f(x)}{J(x)\Delta x}$$

4. 更新信赖域半径 μ 。如果 $\rho > \frac{3}{4}$, 则 $\mu = 2\mu$, 如果 $\rho < \frac{1}{4}$, 则 $\mu = 0.5\mu$;

5. 检查是否满足迭代终止条件, 若满足则迭代停止, 否则令: $x \leftarrow x + \Delta x$, 返回第 2 步。

上面的算法 2 就是 LM 算法。需要说明的是, 第 3 步中的 ρ 其实是对一阶泰勒展开近似程度的一个度量, ρ 的分子是误差函数实际的改变量, 分母是一阶泰勒展开的该变量。而第 4 步则是根据近似程度的好坏来调整信赖域半径。

另外还需要说明的是第 2 步的最优化问题。可以使用拉格朗日乘子法将其转化为一个无约束优化的问题:

$$\min_{\Delta x} \frac{1}{2} \|f(x) + J(x)\Delta x\|^2 + \frac{\lambda}{2} \|D\Delta x\|^2 \quad (4-32)$$

其中 λ 为拉格朗日乘子。那么, 高斯牛顿方程变成了下面形式的增量方程:

$$(H + \lambda D^T D)\Delta x = g \quad (4-33)$$

如果将 $D = I$ 的话, 那么增量方程则变成:

$$(H + \lambda I)\Delta x = g \quad (4-34)$$

以上便是 LM 算法。通常, LM 算法能比高斯牛顿法提供更稳定、更准确的解。

4.4. 地图保存和复用

虽然一个 SLAM 系统能够在定位的时候同时生成地图, 但是如果一个系统多次在相同的环境内运行, 那么把地图保存下来不仅可以节省很多计算资

源，还可以提高定位的精度和鲁棒性。本节提供了一种地图保存、加载和复用的方法，首先会先介绍一下地图保存的方法以及需要保存的数据，然后再介绍一下如何使用保存下来的地图。

4.4.1. 地图保存

本节将主要介绍一下本次工作中地图是如何保存的。本节工作将分为两部分展开叙述，首先将介绍一下需要保存的数据，然后将介绍保存文件的方法。

本次工作需要保存的数据有三种：关键帧、地图点和视觉字典的逆序索引。所谓关键帧，就是根据某一标准筛选出来的相对来说比较重要的图像帧，这里使用 ORB SLAM 的标准去筛选关键帧和地图点，并将这些关键帧和地图点作为地图的一部分保存下来。在后续利用地图的时候可以根据关键帧数据去匹配地图点，优化位姿状态量。对于关键帧，要保存的东西由很多，这里将关键帧保存的数据一一列举下来：

表 4-3 关键帧需要保存的数据

关键帧数据保存	
关键帧序号	每个关键帧创立的时候都会分配一个唯一的标号。
特征点	在该帧图像上检测到的特征点的一些信息。包括图像坐标、深度值、描述符以及尺度信息等。
关联的地图点	该帧图像的特征点关联到的地图点。
相机参数	包括相机内参，双目外参。这个主要用来将特征点变换到相机坐标系下。
IMU 参数	包括 IMU 内参与与相机的外参。
位姿	包括在世界坐标系下的旋转矩阵和平移矩阵。
BoW 向量	即对该帧的词袋描述。也包括各个特征点的顺序索引，用于特征点匹配和图像查询。
共视图和生长树	即该帧关键帧与其它关键帧的连接关系。包括与之在共视图上连接的关键帧，在生长树上与之连接的父节点和子节点等。

上面便是关键帧需要保存的数据。除了关键帧，另一个比较重要的数据便是地图点，在后面利用地图的时候，会将当前图像上的特征点同地图上的地图点进行匹配，通过特征点-地图点匹配对来求解位姿状态量。下面列出地

图点需要保存的数据：

表 4-4 地图点需要保存的数据

地图点数据保存	
地图点序号	每个地图点创立的时候都会分配一个唯一的标号。
坐标	在世界坐标系下的三维坐标。
被观测列表	这个存储了该地图点被哪些关键帧观测到了，以及在该关键帧上与之关联的地图点。
描述符	在视觉字典树中的单词描述，可用于与特征点快速匹配。
观测视角	该数据描述了该地图点被观测到时被观测视角的平均值。
参考帧	记录了第一个观测到该地图点的关键帧。
被观测次数	该数据记录了该地图点被关键帧观测到的次数。

除了地图点和关键帧，另外一个需要储存的数据是视觉词典的逆序索引，逆序索引主要用来将当前帧和地图上的关键帧匹配，并利用视觉字典树加速特征点的匹配。关于逆序索引的介绍前面章节已经给出，这里不再赘述其保存方法。

除此之外，还需要保存的数据包括共视图、本质图等一些描述关键帧之间连接关系的变量，在此不再一一赘述。

以上这些便是需要储存的数据。对于这些数据的储存，本次工作使用序列化的技术将其存为二值化文件。所谓序列化就是将程序中的对象转换形式以方便存储，将其存储在某一区域。以后，可以通过反序列化重构该对象。

本次工作利用 Boost 库中的 **Serialization** 部分对关键帧、地图点、视觉词典的逆序索引以及共视图、本质图等进行序列化，相比于将其存入一个 txt 文件，使用序列化不仅降低了内存消耗，还大大降低了数据的保存和加载所需的时间。

4.4.2. 地图复用

上节介绍了如何进行地图的保存，对于保存的地图，可以使用反序列化的技术将其加载到内存中，然后便可以使用其保存的数据。本节将介绍一下如何使用加载进来的内存。

在前面的章节已经介绍了如何利用非线性优化的方法融合 IMU 和视觉信息来获取相邻帧之间的相对位姿，在通过积分获取当前帧的位姿。在本节

已经有地图的情况下，将在介绍两种获取当前帧位姿的方法。

第一种是利用关键帧。下面给出利用关键帧获取当前位姿的算法：

表 4-5 利用保存地图的关键帧定位

算法 3
<ol style="list-style-type: none"> 1. 获取当前帧图像，提取特征点，计算视觉词袋向量； 2. 将当前帧和参考帧进行特征点匹配，获取与当前帧特征点关联的地图点； 3. 利用光束平差法求解当前帧的位姿； 4. 获取与参考帧共视的关键帧，并将与这些关键帧关联的地图点投影到当前帧上面，获取更多的特征点-地图点匹配； 5. 重新对该帧进行优化，并把与当前帧共视的关键帧设为参考帧供下帧图像使用；

第二种方法类似于回环检测，直接利用视觉字典树获取匹配的关键帧，然后解算出位姿，下面是具体的算法：

表 4-6 利用保存地图的逆序索引定位

算法 4
<ol style="list-style-type: none"> 1. 获取当前帧图像，提取特征点，计算视觉词袋向量； 2. 利用视觉字典的逆序索引获取与当前帧相似的关键帧集合； 3. 将当前帧和这些关键帧进行特征点匹配，获取与当前帧特征点关联的地图点； 4. 利用 PnP 算法求解出当前帧的位姿，并将与该帧共视最多的关键帧设为参考帧； 5. 利用共视图，获取与参考帧共视的关键帧，并将与这些关键帧关联的地图点投影到当前帧上面，获取更多的特征点-地图点匹配； 6. 重新对该帧进行优化，并把与当前帧共视的关键帧设为参考帧供下帧图像使用；

当使用一个固定的地图时，算法 3 和算法 4 所求解出的位姿都是没有累计误差的，这一点要由于利用积分相邻帧相对位姿的算法。并且，算法 3 和算法 4 有一些重叠的地方，比如都要先提取特征点，最后都要投影一个局部地图，这使得完全可以先进行这些重叠操作，再选择使用哪一种算法获取位姿。

在本次工作中，会先进行特征点的提取和计算当前帧的词袋向量的工

作。然后利用算法 3 来获取当前帧位姿，如果解算失败，比如迭代发散或者匹配特征点得到的内点数较少，则再次利用算法 4 进行位姿的求解工作，这样大大增强了算法鲁棒性。

4.5. 本章小结

本章介绍了视觉信息和 IMU 信息融合的方法，其中，系统初始化求解了初始的陀螺仪的初始 bias、初始速度和重力向量，为后续的求解位姿奠定了基础。然后介绍了如何利用非线性优化的方法求解相邻帧之间的相对位姿，这其中通过联合优化视觉信息的误差和 IMU 的预积分残差以及 bias 的误差来融合 IMU 和视觉的位姿信息。最后则介绍了地图保存以及复用的方法，利用保存下来的地图，可以获得更准确的位姿信息，并且也会减少很多计算量。

另外需要说明的是，由于跟踪是否跟取决于初始的匹配得到的特征点的多少，所以本章介绍的将 IMU 信息加入优化的方法在实际使用的时候只用作位姿预测、初步的位姿解算和状态量求解，而在后续利用共视图匹配更多地图点、局部线程优化和全局优化的时候均没有使用 IMU 约束，这样做的目的是在使用 IMU 提升跟踪的稳定性，但不降低优化的稳定性和精度，在后续的实验中将验证这一思想。

第 5 章 实验结果与分析

5.1. 引言

本章将对上面介绍的算法进行实际的实验，本次所进行的实验包括 3 个部分：标定实验、使用数据集定位建图的实验和使用小觅双目相机定位建图的实验。

在标定实验中，首先将对双目相机进行标定，然后会对相机和 IMU 进行联合标定。本次工作使用的标定工具是 `kalibr`，在进行 IMU 和相机联合标定的时候，其使用的方法类似于 3、4 章叙述的建模和优化处理方法，只是状态量变成了 IMU 和相机的外参矩阵。由于在本次工作中，标定是一件重要但非重点的工作，所以这里不再详细介绍其原理，只是给出标定的结果以证明本次标定工作的准确性。

在使用数据集定位建图的实验中，因为数据集有 `ground truth`，所以本次工作将会对定位的结果和实际的 `ground truth` 进行对比，测定一下定位的误差。除此之外，本次工作还会将本次工作的定位效果同双目的 ORB SLAM 进行比较，给出比较的结果。

在使用小觅双目相机定位建图的实验中，将使用小觅双目相机对一些真实的环境进行定位和建图，证明在廉价的传感器上面算法依然能够稳定的工作。

5.2. 标定实验

本节将介绍一下本次工作所使用的标定方法和实验。本节的组织结构如下：首先将介绍一下所使用的实验平台，然后将介绍一下双目相机的标定，最后将介绍一下如何进行联合标定获取双目相机和 IMU 的外参。

5.2.1. 实验平台

本次实验所采用的相机是小觅双目相机，实物如图 5-1 所示。小觅相机是一款集成 IMU 的双目相机，并且其相机和 IMU 做了硬件同步，时间差小于 1ms，是一款非常适合 SLAM 系统的相机。

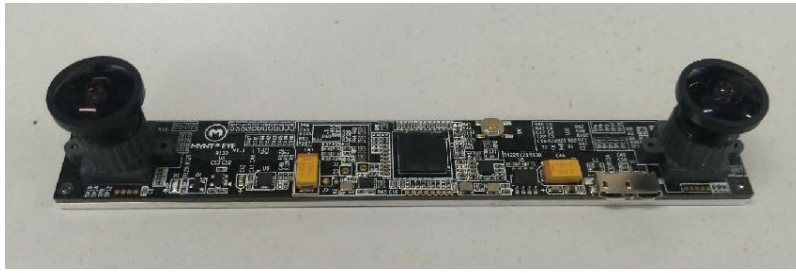


图 5-1 小觅双目相机

下面为小觅双目相机的一些比较重要的参数：

表 5-1 小觅双目相机的一些参数

型号	MYNT EYE S1010
图像帧率	50fps
水平视角	96°
垂直视角	61°
对角线视角	120°
基线长	120.0mm
焦距	2.6mm
分辨率	752*480
运动感知	6 轴 IMU
IMU 帧率	500HZ
同步精度	<1ms
色彩模式	灰图度
扫描模式	全局快门
深度图适用距离	0.8-10m
接口	USB 3.0

本次所使用计算平台之一为 Intel Core i7-6700K, 4 核, 主频为 4.00GHZ, 内存为 32GB, 所使用的系统为 Ubuntu 14.04。另一个计算平台为 Intel Core i5-3210M, 4 核, 主频为 2.5GHz, 内存为 4GB, 所使用的系统为 Ubuntu16.04。另外, 本次工作使用 ROS (Robot Operating System, 机器人操作系统) 来进行传感器数据的读取和预处理。

5.2.2. 双目相机标定

本次工作中相机和 IMU 的联合标定使用的工具是 kalibr。kalibr 是一款开源的能够标定双目相机和 IMU 的软件。kalibr 使用重投影误差作为度量标

定结果好坏的标准，如图 5-2 便是本次双目标定过程左目相机和右目相机的重投影误差，可以看出，重投影误差基本上都在 ± 1 个像素之间。

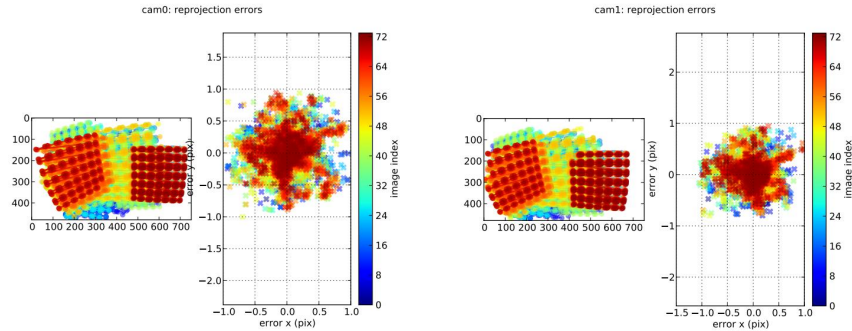


图 5-2 双目标定过程左目相机（左）和右目相机（右）的重投影误差

表 5-2 给出了详细的重投影误差值，可以看出，重投影误差很小，这个结果表明了标定结果的正确性。

表 5-2 小觅相机双目标定的误差

误差项	数值（单位：像素值 ² ）
左目重投影误差平方	$[-0.000006, 0.000000] \pm [0.236346, 0.226996]$
右目重投影误差平方	$[0.000006, -0.000000] \pm [0.238436, 0.207813]$

在通过标定获取双目相机的参数之后，可以利用获得的参数对双目相机进行立体校正，本次工作使用 OpenCV 库进行双目校正，图 5-3 和图 5-4 显示了立体校正前后的差别。经过立体校正过后的相机不仅去除了畸变，还进行了行对齐的操作，进行行对齐过后的双目可以加速立体匹配。

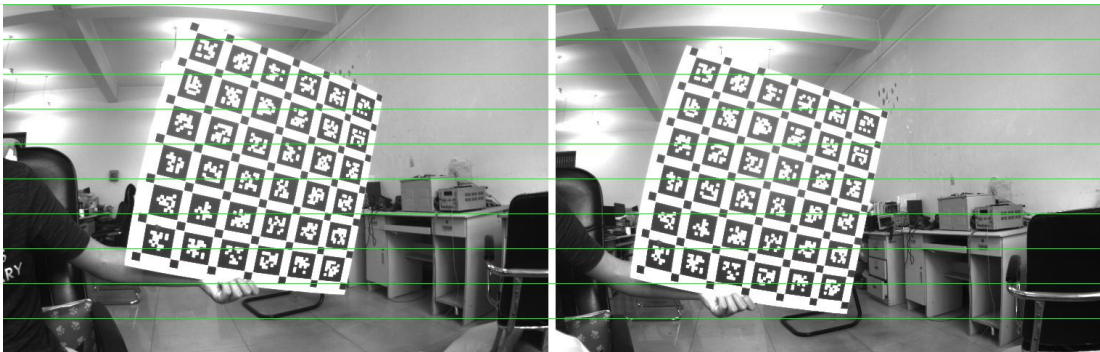


图 5-3 双目校正之前

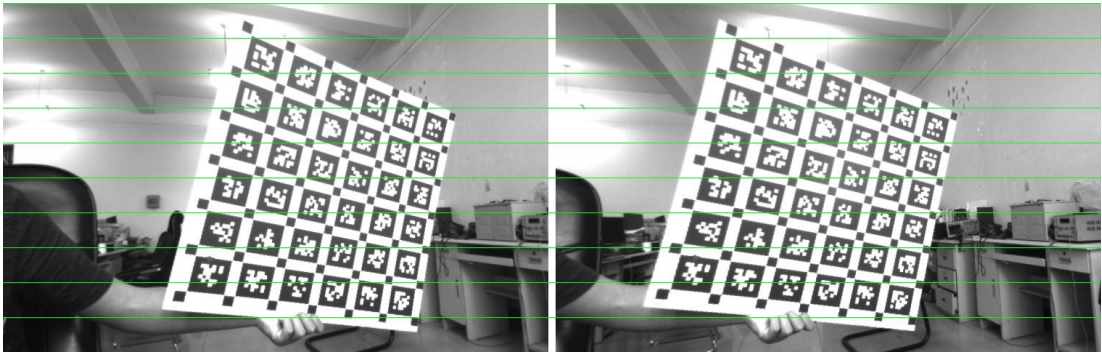


图 5-4 双目校正之后

上图中，蓝色绿色线条为辅助线，利用辅助线可以看出，图 5-3 经过校正得到的 5-4 左右双目图像已经进行了行对齐。由立体校正前后的对比可以看出，本次工作中的双目标定工作进行的非常成功。

5.2.3. 相机和 IMU 联合标定

对于相机和 IMU 的联合标定，依然使用 `kalibr` 去完成。类似于第 3 章对 IMU 建模的方法，`kalibr` 也会对 IMU 建立噪声模型和运动学模型，然后样条基函数会对 IMU 的数据进行拟合。标定过程中，`kalibr` 会使用 `apriltag` 作为视觉基准，利用非线性优化的方法优化视觉的重投影误差、角速度误差、加速度误差和 `bias` 误差，最后得到所需要的 IMU 和相机的外参数以及重力向量大小。除此之外，`kalibr` 还考虑了相机和 IMU 不同步的情况，因此，还可以使用 `kalibr` 获得相机和 IMU 之间的时间延迟。

本次工作中使用 `kalibr` 对小觅双目相机进行了外参的标定，图 5-5 是标定过程中估计的 `bias`，上中下三条曲线分别是 x 轴、 y 轴和 z 轴的分量。图 5-6 表示的是估计的角速度和加速度和用 IMU 数据和样条基函数得到的图像帧时刻角速度和加速度之间的误差，可以看出，角速度误差基本小于 0.02 rad/s ，加速度的误差基本小于 0.004 m/s^2 。图 5-7 表示的是左右目的重投影误差，可以看出，重投影误差大多在 ± 0.5 个像素之间，基本上都在 ± 1 个像素之间，这些数据从侧面反映了标定结果的准确性。

另外，从获得的陀螺仪的随机游走曲线和加速度计的随机游走曲线可以看出，随机游走值的变化没有规律，并且前后变化值没有太大的变化，所以第 3 章中将随机游走建模成一个白噪声驱动项的做法是合理的。

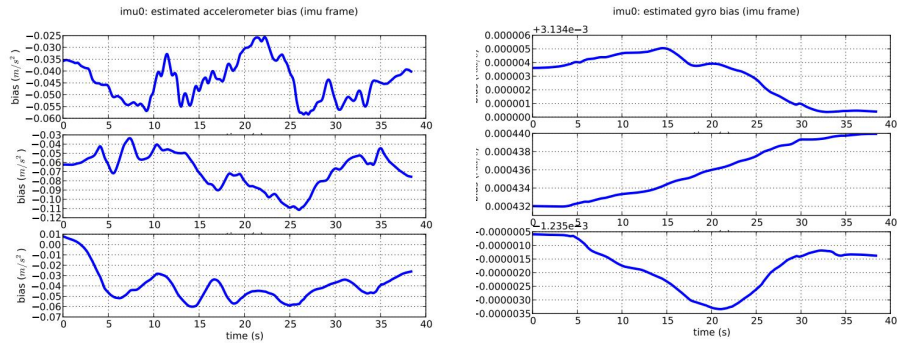


图 5-5 联合标定过程中估计的加速度计的 bias（左）和陀螺仪的 bias（右）

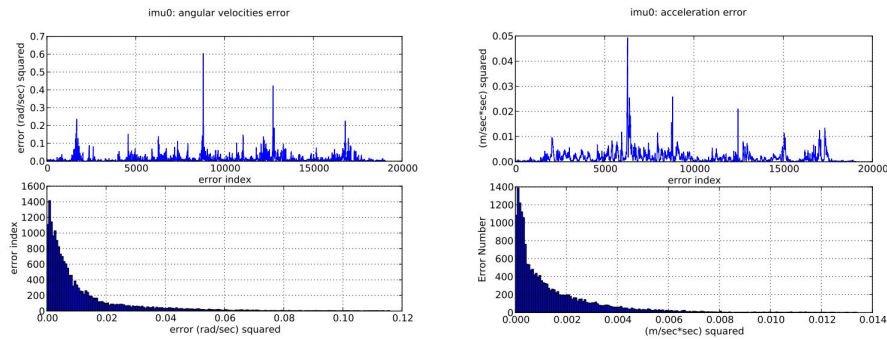


图 5-6 联合标定过程中的角速度误差（左）和加速度误差（右）

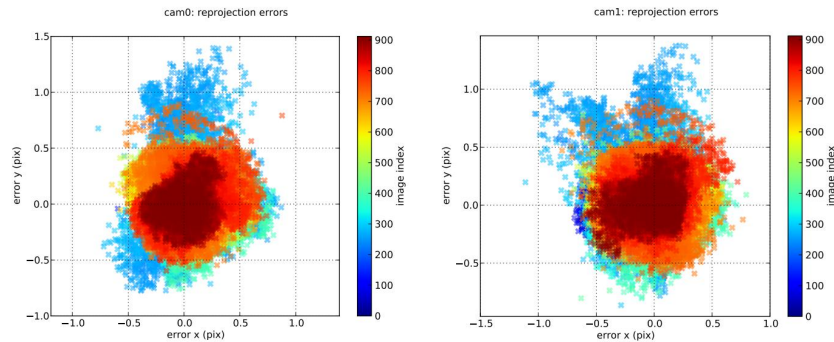


图 5-7 联合标定过程中的左目相机的重投影误差（左）和右目相机的重投影误差（右）

表 5-3 是小觅双目相机和 IMU 联合标定的一些误差：

表 5-3 小觅相机双目和 IMU 联合标定的误差

误差项	数值
左目重投影误差平方（单位：像素 ² ）	0.0572818226644
右目重投影误差平方（单位：像素 ² ）	0.0554352158043
角速度误差平方（单位：(rad/s) ² ）	0.0136255628652
加速度误差（单位：(m ² /s) ² ）	0.00166423202111

由上面的标定结果可以看出，联合标定的重投影误差在 0.5 个像素值左

右，加速度误差在 0.115rad/s 左右，加速度误差在 $0.04\text{m}^2/\text{s}$ 左右，这些误差值都很小，从而可以看出本次工作的联合标定工作也进行的非常成功。

5.3. 使用 EUROC 数据集定位和建图

本节将利用 Euroc 数据集对本次工作的程序进行一个验证, Euroc 数据集是利用一台无人机为平台, 如图下图 5-8 所示:

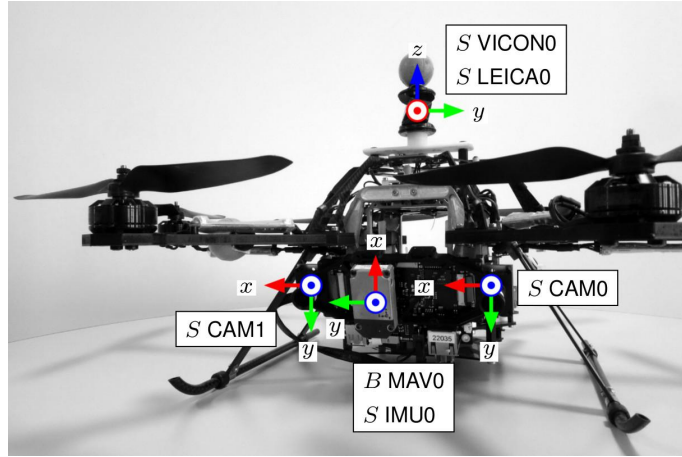


图 5-8 Euroc 数据集使用的采集平台

搭载的传感器为一个双目相机和一个 6 轴的 IMU。双目相机型号为 MT9V034，全局快门单通道图像，分辨率为 752×480 ，帧率为 20fps 。IMU 和相机数据同步，型号为 ADIS16448，帧率为 200Hz 。除此之外，Euroc 数据集使用了 Vicon 光学动作捕捉系统和 Leica MS50 激光跟踪仪来获取 ground truth 来衡量算法的性能。

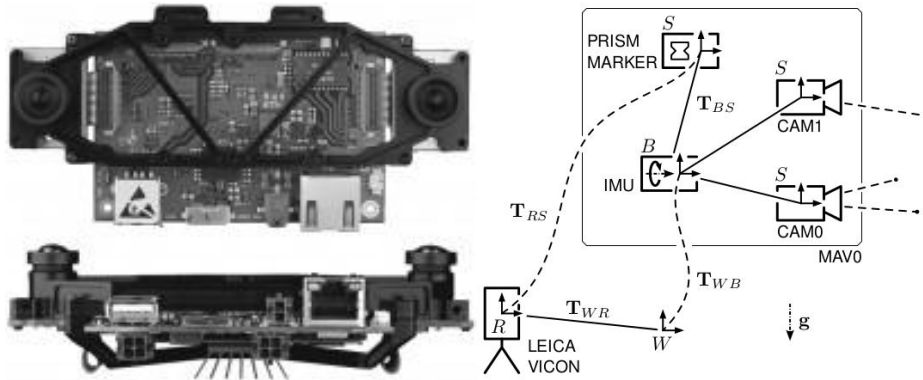


图 5-9 左: VI-Sensor; 右: 传感器和 ground truth 仪器概述

下面图 5-10 为利用 Euroc 数据集中的 MH_01_easy 保存的地图，蓝点表示关键帧，红点和黑点为地图点，绿线表示共视图。

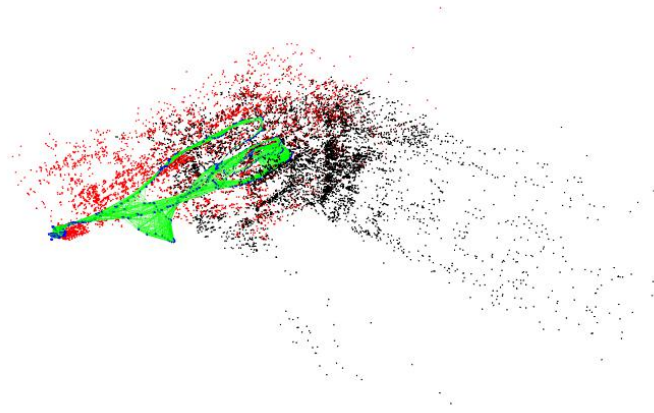


图 5-10 保存的 Euroc 数据集中 MH_01_easy 的地图

对于地图的大小，表格 5-4 则给出了 Euroc 数据集中各个 rosbag 的保存的地图的大小。

表 5-4 Euroc 数据集保存地图的大小

数据集	地图大小
MH_01_easy	139.4M
MH_02_easy	121.6M
MH_03_medium	124.2M
MH_04_difficult	81M
MH_05_difficult	102.7M
V1_01_easy	35.3M
V1_02_medium	43.6M
V1_03_difficult	53.6M
V2_01_easy	35.0M
V2_02_medium	68.1M
V2_03_difficult	76.7M

因为本次工作是在 ORB SLAM 的工作上进行改进的，所以本次工作对改进前后的算法进行了对比，使用的性能指标为 RMSE（root-mean-square error，均方根误差），RMSE 的计算方法如下面式子（5-1）所示：

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{p}_i - p_i)^2} \quad (5-1)$$

其中 \hat{p}_i 为算法得到的位置值， p_i 为数据集提供的真实的位置值， N 为需要比较的位置的数量。

使用的工具为 evo (<https://github.com/MichaelGrupp/evo>)。图 5-11、5-12、

5-13 为使用 i7 处理器的算法修改前后的误差和 RMSE 的情况：

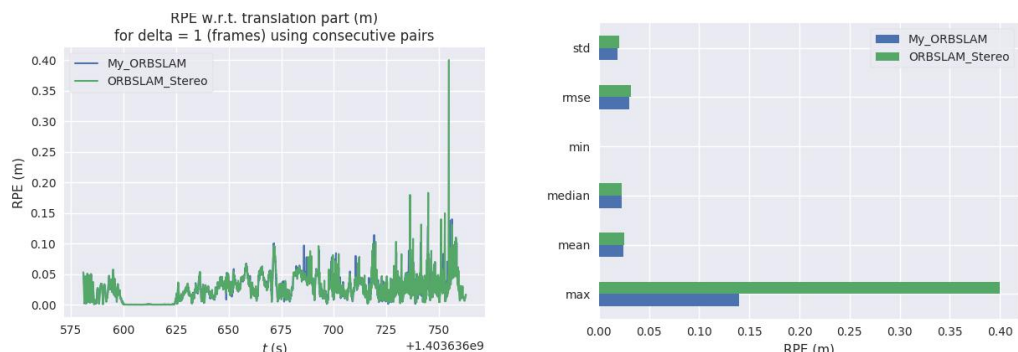


图 5-11 使用 MH_01_easy 测试的算法修改前后对比图

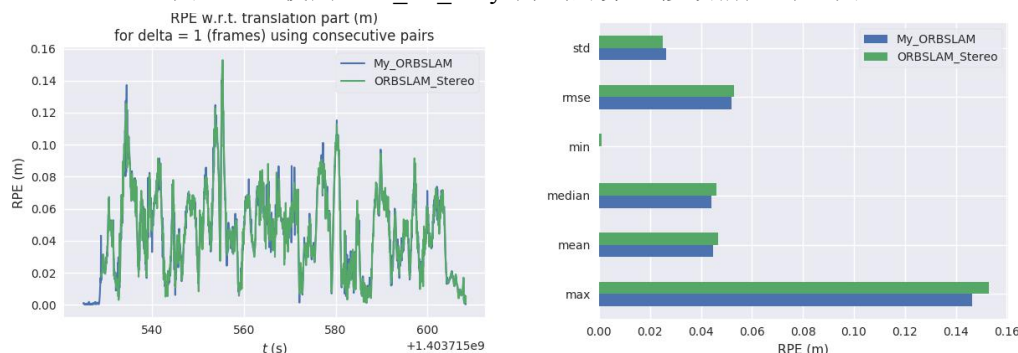


图 5-12 使用 V1_02_medium 测试的算法修改前后对比图

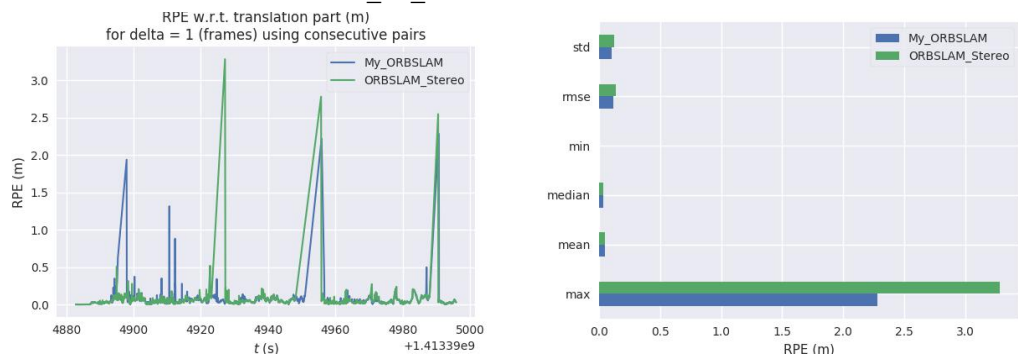


图 5-13 使用 V2_03_difficult 测试的算法修改前后对比图

上图 5-13、5-14 和 5-15 中，左侧图为改进前后两种算法的位置误差随时间的变化曲线，右侧两种算法误差的标准差、RMSE、最小误差、误差中值、平均误差和最大误差的一个详细对比。从这三幅图可以看出，在使用 i7 处理器的情况下，使用 MH_01_easy、V1_02_medium、V2_03_difficult 三个数据集本文算法的精度要高于原来的双目 ORB SLAM。

之后，本文又使用整个 EuRoC 数据集在 i7 CPU 的环境下对算法修改前后进行了一个对比实验，下面表格 5-3 则给出了利用全部 Euroc 数据集测试

的情况：

表 5-5 使用 i7 处理器全部 Euroc 数据集测试的算法修改前后对比（RMSE）

数据集	原始双目 ORB SLAM	本文改进的 ORB SLAM
MH_01_easy	0.032203	0.030388
MH_02_easy	0.029669	0.029532
MH_03_medium	0.074395	0.073244
MH_04_difficult	0.070104	0.070192
MH_05_difficult	0.062705	0.062237
V1_01_easy	0.025987	0.027458
V1_02_medium	0.053073	0.051858
V1_03_difficult	0.048691	0.046433
V2_01_easy	0.019068	0.018898
V2_02_medium	0.042645	0.042772
V2_03_difficult	0.126167	0.117855

在上面表格 5-5 中，误差小的数据已经用黑体标注。从上面表格可以看出，EuRoC 一共 11 个子数据集，在 i7 CPU 的处理器下，两个算法都能够跑完整个数据集，有 8 个数据集算法修改后精度有所提升，有三个数据集精度有所下降，所以可以得出结论在利用本文算法引入 IMU 信息后系统的精度有一定程度上的提升。对于上面表格中三个精度下降的数据集，在此对其的原因进行了分析：由于双目 ORB SLAM 的本身进行了多次优化，其对视觉的信息利用的比较彻底。精度已经很高了，而 IMU 信息的引入使得优化过程出现了一些不稳定性，虽然这也是本次系统设计之初便着重考虑的一件事，但是还会有一定程度的影响。

另外，本次工作还使用了 i5 处理器对算法修改前后进行了实验结果的对比。对于 i5 处理器，ORB SLAM 并不能实时处理 Euroc 数据集，但是程序总是处理最新的图像，所以程序会隔若干帧图像再处理一帧图像。由于这个原因，使用 i5 CPU 的处理器进行对比实验不仅可以得到算法在处理性能受限的情况下的性能效果，还相当于模拟了快速移动时的场景，下表 5-6 是对比的实验结果：

表 5-6 使用 i5 全部 Euroc 数据集测试的算法修改前后对比 (RMSE)

数据集	原始双目 ORB SLAM	本文改进的 ORB SLAM
MH_01_easy	0.100549	0.075148
MH_02_easy	0.088619	0.068898
MH_03_medium	-	0.227523
MH_04_difficult	0.235487	0.148765
MH_05_difficult	0.181076	0.136439
V1_01_easy	0.062219	0.061291
V1_02_medium	-	0.104187
V1_03_difficult	-	0.101624
V2_01_easy	0.042133	0.035623
V2_02_medium	-	0.100325
V2_03_difficult	-	-

在上表 5-4 中,“-”表示程序不能完整的跑完整个数据集,误差小的数值已经用黑体标注。从上面表格可以看出, EuRoC 的 11 子数据集, 原始的双目 ORB SLAM 能够完整的跑完 6 个, 而本文改进的算法可以跑完 10 个, 并且在都能跑完的 6 个数据集中, 本文的算法精度都要比原始的 ORB SLAM 精度要高, 由此可以得出结论: 在使用 i5 CPU 下, 本文改进的算法比原始的算法无论是精度还是鲁棒性上都有很大的提升。

5.4. 使用小觅双目相机定位和建图

最后, 本次工作还使用了 5.2 节所介绍的小觅双目相机进行了建图和定位的实验, 因为没有真值做比较, 所以本节实验只是为了证明在实际的廉价的传感器上本文提出的算法依然能够正常、稳定的工作, 不做任何关于精度上的结论。

本次工作在哈尔滨工业大学多个地点进行了实验, 包括科学园室外、科学园 2B 栋内部、活动中心等地。在此给出哈工大活动中心的一楼大厅进行的实验, 地图实际大小约 15m*15m, , 最后保存的地图大小是 185M, 下图 5-14 是建立的地图和定位的轨迹:

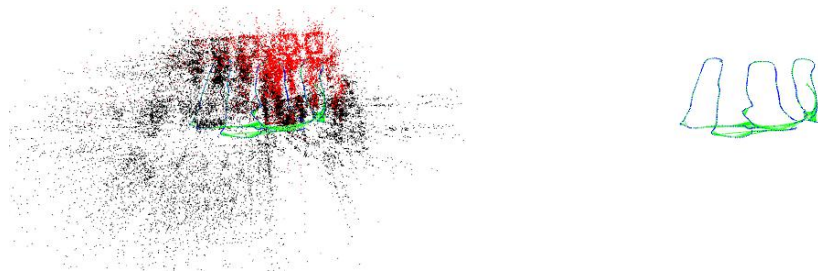


图 5-14 使用小觅双目相机在哈工大活动中心一楼大厅运行效果图，左：建立的地图；
右：定位的轨迹

本次实验特意手持小觅双目相机走了一个闭合的曲线，从定位的轨迹来看，曲线已经闭合，定位的效果很理想。

5.5. 本章小结

本章主要介绍并进行了一些实验。首先是标定实验，包括双目的标定和校正以及双目相机和 IMU 的联合标定，通过标定，获取了双目自身的内参和外参信息，以及双目相机和 IMU 的相对外参。另外，通过标定还获取了重力的大小，这些信息都为相机和 IMU 的信息融合奠定了基础。

之后，本章使用了 EuRoC 数据集对本文的算法进行了实验的验证。首先给出的是建图的结果，可以看出地图的大小都在可以接受的范围内。之后则给出了本文改进的算法同原始双目 ORB SLAM 的对比实验，从实验结果可以看出，融合了 IMU 信息后，系统的精度和鲁棒性得到了提高。

结论

SLAM 是室内移动机器人定位的主要方式之一，而视觉 SLAM 则由于其成本低廉、信息量大而具有广阔的发展前景。本文主要聚焦于将 IMU 的信息融合到视觉 SLAM 系统里面，以提高其鲁棒性和稳定性，本文的主要贡献如下：

(1) 综述了视觉 SLAM 的基础知识，对 IMU 的进行了建模，推导了 IMU 预积分的方法和其方便编程实现的递推形式；

(2) 提出了一种将 IMU 的信息融合到 ORB SLAM 的方法，因为本文的方法只利用 IMU 的信息进行了相机位姿预测和初步的优化，没有过多的使用 IMU 的信息，所以本算法对 IMU 的要求并不高，可以使用低廉的 IMU 就能达到很不错的效果，甚至对 IMU 和相机同步都没有太高的要求；

(3) 提出了一种保存和复用地图的算法，可以将生成的地图保存并重复利用。

当然，本文还有很多方向可以改进，比如可以尝试利用 GPU 对程序进行加速，尝试融入更多的传感器信息等等。

参考文献

- [1] A. J. Davison. Real-time simultaneous localisation and mapping with a single camera[C]. Computer Vision, 2003. Proceedings. 9th IEEE International Conference on. IEEE,2003:1403-1410.
- [2] G. Klein, D. Murray. Parallel tracking and mapping for small ar workspace[C]. Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on. IEEE,2007:225-234.
- [3] J. Engel, T. Schops, D. Cremers. Lsd-slam: Large-scale direct monocular slam[C]. Computer Vision-ECCV, 2014:834-849. Springer,2014.
- [4] C. Forster, M.Pizzoli, D.Scaramuzza. Svo: Fast semi-direct monocular visual odometry[C]. Robotics and Automation (ICRA), 2014 IEEE International Conference on. IEEE,2014:15-22.
- [5] M. Labbe, F. Michaud. Online global loop closure detection for large-scale muti-session graph-based slam[C]. 2014 IEEE/RSJ International Conference on Robot and System. IEEE,2014:2661-2666.
- [6] R. Mur-Artal, J. Montiel, J. D. Tardos. Orb-slam: a versatile and accurate monocular slam system[J]. IEEE Transactions on Robotics, 2015:1147-1163.
- [7] R. Mur-Artal, J. D. Tardos. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras[J]. IEEE Transactions on Robotics, 2017:1255-1262.
- [8] E. Rublee, V. Rabaud, K. Konolige, G. Bradski. ORB: an efficient alternative to SIFT or SURF[C]. IEEE International Conference on Computer Vision. IEEE,2012:2564-2571.
- [9] C. Kerl, J. Sturm, D. Creamers. Dense visual slam for rgb-d cameras[C]. 2013 IEEE/RSJ International Conference on Intelligent Robots and System. IEEE,2013:2100-2106.
- [10] J. Engel, V. Koltun, D. Cremers. Direct sparse odometry[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence. IEEE,2016:1-1.
- [11] Richard A. Newcombe, Steven J. Lovegrove, Andrew J. Davison. DTAM: Dense Tracking and Mapping in Real-Time[C]. IEEE International Conference on Computer Vision. IEEE,2011:2320-2327.
- [12] A. Martinelli. Closed-form solution of visual-inertial structure from

- motion[J]. International Journal of Computer Vision, vol.106, no.2, 2014:138-152.
- [13] R. E. Kalman , R. S. Bucy. New Results in Linear Filtering and Prediction Theory[J]. J. Basic Eng, vol.83, 1961:95-108.
- [14] Anastasios I. Mourikis, Stergios I. Roumeliotis. A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation[C]. IEEE International Conference on Robotics and Automation. IEEE,2007:3565-3572.
- [15] Mingyang Li, Anastasios I. Mourikis. High-precision, consistent EKF-based visual-inertial odometry[J]. The International Journal of Robotics Research, vol. 32, 2014:690-711.
- [16] M. Bloesch, S. Omari, M. Hutter, R. Siegwart. Robust visual inertial odometry using a direct efk-based approach[C]. Intelligent Robots and System (IROS), 2015 IEEE/RSJ International Conference on. IEEE,2015:298-304.
- [17] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, P. Furgale. Keyframe-based visual - inertial odometry using nonlinear optimization[J]. International Journal of Robotics Research, vol.34, 2015:314-334.
- [18] S. Leutenegger, M. Chli, and R.Y. Siegwart. BRISK: Binary robust invariant scalable keypoints[C]. International Conference on Computer Vision. IEEE, 2012:2548-2556.
- [19] Martin A. Fischler, Robert C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography[J]. Communications of the ACM , vol.24. 1981: 381 - 395.
- [20] C. Forster, L. Carlone, F. Dellaert, D. Scaramuzza. IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation[J]. Robotics: Science and Systems (RSS), 2015.
- [21] R. Mur-Artal, J. D. Tardos. Visual-Inertial Monocular SLAM with Map Reuse[J]. IEEE Robotics mation Letters, vol.2, 2017:796-803.
- [22] Shaojie Shen, et al. Autonomous Aerial Navigation Using Monocular Visual-Inertial Fusion[J]. Journal of Field Robotics, vol.35,2017.
- [23] Rosten E, Drummond T. Machine learning for high-speed corner detection[C]. European Conference on Computer Vision. Springer, Berlin, Heidelberg, 2006:430-443.
- [24] Calonder M, Lepetit V, Strecha C, et al. BRIEF: Binary Robust Independent

- Elementary Features[C]. European Conference on Computer Vision. 2010:778-792.
- [25] Blöchliger F, Fehr M, Dymczyk M, et al. Topomap: Topological Mapping and Navigation Based on Visual SLAM Maps[J]. 2017.
- [26] Fankhauser P, Bloesch M, Krüsi P, et al. Collaborative navigation for flying and walking robots[C]. IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2016.
- [27] Burri M, Oleynikova H, Achtelik M W, et al. Real-time visual-inertial mapping, re-localization and planning onboard MAVs in unknown environments[C]. IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2015:1872-1878.
- [28] Bürki M, Gilitschenski I, Stumm E, et al. Appearance-based landmark selection for efficient long-term visual localization[C]. IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2016.
- [29] Fehr M, Dymczyk M, Lynen S, et al. Reshaping our model of the world over time[C]. IEEE International Conference on Robotics and Automation. IEEE, 2016:2449-2455.
- [30] Dymczyk M, Lynen S, Bosse M, et al. Keep it brief: Scalable creation of compressed localization maps[C]. IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2015.
- [31] Dymczyk M, Lynen S, Cieslewski T, et al. The gist of maps - summarizing experience for lifelong localization[C]. IEEE International Conference on Robotics and Automation. IEEE, 2015:2767-2773.
- [32] Oleynikova H, Burri M, Lynen S, et al. Real-time visual-inertial localization for aerial and ground robots[C]. IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2015:3079-3085.
- [33] Cieslewski T, Lynen S, Dymczyk M, et al. Map API - scalable decentralized map building for robots[C]. IEEE International Conference on Robotics and Automation. IEEE, 2015:6241-6247.
- [34] Hinzmann T, Schneider T, Dymczyk M, et al. Monocular Visual-Inertial SLAM for Fixed-Wing UAVs Using Sliding Window Based Nonlinear Optimization[J]. 2016:569-581.
- [35] Hinzmann T, Schneider T, Dymczyk M, et al. Robust map generation for fixed-wing UAVs with low-cost highly-oblique monocular cameras[C]. IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2016.

- [36] Dymczyk M, Schneider T, Gilitschenski I, et al. Erasing bad memories: Agent-side summarization for long-term mapping[C]. IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2016.
- [37] Merzic H, Stumm E, Dymczyk M, et al. Map quality evaluation for visual localization[C]. IEEE International Conference on Robotics and Automation. IEEE, 2017.
- [38] Gehrig M, Stumm E, Hinzmann T, et al. Visual place recognition with probabilistic voting[C]. IEEE International Conference on Robotics and Automation. IEEE, 2017:3192-3199.
- [39] 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.
- [40] Krystian Mikolajczyk, Cordelia Schmid. Scale & Affine Invariant Interest Point Detectors[J]. International Journal of Computer Vision, 2004, 60(1):63 - 86.
- [41] Bay H, Ess A, Tuytelaars T, et al. Speeded-Up Robust Features (SURF)[J]. Computer Vision & Image Understanding, 2008, 110(3):346-359.
- [42] David G. Lowe, David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints[J]. International Journal of Computer Vision, 2004, 60(2):91-110.
- [43] Tardos J D. Bags of Binary Words for Fast Place Recognition in Image Sequences[M]. IEEE Press, 2012.
- [44] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, et al. Bundle Adjustment — A Modern Synthesis[J]. Vision Algorithms : Theory & Practice, 2000, 1883(1883):298-372.
- [45] 秦永元. 惯性技术[M]. 科学出版社,2006:287-389.

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《融合 IMU 信息的双目视觉 SLAM 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:徐宽 日期:2017 年 6 月 25 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:徐宽 日期:2017 年 6 月 25 日

导师签名:王世 日期:2017 年 6 月 25 日

致 谢

两年的时间转瞬即逝，我也即将结束我的硕士生涯。在此，首先要感谢的是我的导师——刘志远老师，不仅仅是因为刘老师悉心指导了我本次的毕业设计工作，更是因为在实验室的三年时光里，刘老师给了我极大的研究自由，让我按照自己的兴趣去学习和探索自己喜欢的东西，这也体现了刘老师因材施教的育人方法和海纳百川的胸襟。另外，刘老师对待科研事业严谨的态度和热爱运动的生活习惯也都深深的影响了我，再次感谢刘志远老师。

其次，感谢实验室的刘海峰老师、赵林辉老师、井后华老师和周洪亮老师在学术和科研上给我的指导和生活中帮助。感谢赵洁老师每次在实验室举行活动的时候都能尽心尽力的组织，让我们在实验室度过愉快的时光。

对于论文的撰写，还要感谢何朕老师和周洪亮老师作我的论文评阅人，为我的论文提出宝贵建议。

同时也要感谢实验室的师兄师姐给予我的帮助。感谢实验室的同学、我的室友以及生活中的好友，感谢你们的陪伴。

最后还要感谢我的家人：我的父母和姐姐。感谢你们二十几年的的养育和陪伴，你们是我永远的港湾。

在本文的最后，我想留一些话给将来的自己：当你再次翻开这篇文章的时候，不知已经到了何年何月，正在处于怎样的境地。如果你现在事事如意，那么希望你能珍惜现在幸福的生活，珍惜自己的家人和朋友；而如果你现在正处于人生的低谷，那么希望你能回想起硕士毕业时那个无所畏惧、一心向前的自己，重新捡起对生活的信心和勇气，开创一番新的境地。