

分类号: TP242.6

密级:

学校代码: 11065 学号: 2015020309

青岛大学 硕士学位论文

基于单目相机与 RGB-D 相机的 SLAM 研究与设计

Research And Design on SLAM Based on
Monocular And RGB-D Camera

安帅

指 导 教 师_____杨杰 副教授_____

学 科 专 业 名 称_____机械工程_____

培 养 单 位_____机电工程学院_____

论 文 答 辩 日 期_____2018 年 5 月 20 日_____

答 辩 委 员 会 主 席_____赵永瑞_____

摘要

SLAM(Simultaneous Localization And Mapping), 即即时定位与地图构建技术是全自主机器人、无人驾驶、自主无人机、AR 等领域的关键技术, 常用的 SLAM 传感器有激光雷达、单目、双目、RGB-D 等传感器。单目相机由于其价格低廉、便于携带、信息丰富、易于与其他信息融合等优点, 因此, 单目 SLAM 是视觉 SLAM 研究领域中的热点也是难点。此外, RGB-D 相机在室内比较稳定并且能够构建出稠密地图, 在室内 RGB-D SLAM 有一定的研究价值。在实际应用中, 例如通常无人机的飞行通过 GPS 进行定位, 其定位精度较差且在室内、隧道、洞穴、山谷等场景下会失效, 且一般通过人工控制进行避障, 不能够自主进行对环境的感知与避障。为了克服上述不足, 本文研究了单目 SLAM 算法, 并将其运用于无人机采集的视频序列当中, 对无人机的自主飞行提供了一定的理论基础。RGB-D 相机由于其深度信息容易获得, 易于搭建 SLAM 矿建, 本文利用 RGB-D 相机数据集进行了 SLAM 系统的设计与实验, 为后期系统的完善提供了大体框架, 以进行开发与拓展。

完善的 SLAM 系统包括前端里程计、后端优化、建图与回环检测四个部分。文中对 SLAM 系统的基本原理进行了较为深入地分析, 从前端设计、后端优化、回环检测与建图四个方面进行了数学原理的推导与程序的设计。

在前端设计中, 单目 SLAM 利用对极几何计算了本质矩阵 E 和单应性矩阵 H , 并分解这两个矩阵, 经过验证之后得到正确的位姿, 然后利用该位姿进行三角测量得到地图点的三维坐标, 再利用 PnP 的方法解算后续的位姿与地图点, RGB-D 相机的前端设计中则直接用 PnP 进行位姿估计与地图点的求解, 在基于直接法的前端设计中, 则是利用最小化光度误差(Photometric Error)进行迭代求解位姿与地图点; 进行后端优化时, 利用 Ceres 库与 G2O 库通过高斯牛顿、列文伯格——马库尔特等算法来进行迭代优化, 得到优化后的相机位姿与地图点; 建图部分则分别利用 ORB-SLAM2、LSD-SLAM 和 RGB-D 数据建立稀疏地图、稠密地图与占据栅格地图, 根据地图形式可达到进行定位、导航与避障等功能; 回环检测中利用基于机器学习中非监督学习的方法来检测已出现的图像, 并构造了基于 ORB 特征训练的字典, 然后进行了词带模型的建立并利用 TF-IDF 权值构造方法计算了图像的相似性, 以进行正确的回环检测。

关键词: 单目 SLAM; RGB-D SLAM; 非线性优化; 地图构建; 三维重建

Abstract

Simultaneous Localization And Mapping (SLAM) technology is the key technology in the field of autonomous robot, autonomous driving, autonomous UAV and AR. The most frequently used SLAM sensors include lidar, monocular camera, stereo camera, RGB-D camera and so on. It is regarded as a hot and difficult issue because of the advantage of the low cost, portability, rich information, easily integration with other sensor information and so on in the field of visual SLAM. In addition, the RGB-D camera is more stable indoors than outdoors and can build a dense map through the technology of SLAM. Therefore, indoor SLAM is also worth investigating. In practical applications, for example, the fly of UAV is generally localized by the GPS, but the localization accuracy is low and it would lead to invalidation in the scene of the indoors, tunnel, cave, valley etc. To overcome the faults above, the monocular SLAM algorithm has been researched in this thesis, and it was applied to the video sequences collected by the DJI UAV which offers theoretical bases for the autonomous flight of the UAVs.

A perfect SLAM system includes front-end design, back-end optimization, map building and loop detection. In this thesis, basic principles of SLAM system are deeply analyzed and mathematical principles are deduced and the program has been designed from these four aspects.

1. In the front-end design, the essential matrix E and the homography matrix H had been calculated exploiting the epipolar geometry for the monocular SLAM, and these two matrices had been decomposed to get the correct pose after verification, then 3D map points were solved out by triangulation exploiting this pose, the subsequent poses and map point would be estimated through Perspective-n-Point(PnP); While the pose and map point estimation of the RGB-D camera front-end was directly calculated through PnP algorithm; However, in the front-end design based on the direct method, the pose and the map point are iteratively solved by minimizing the Photometric Error. In this thesis, feature extraction and matching, pose estimation and triangulation and the other principles has been analyzed through the monocular UAV pictures, then the front-end system has been constructed based on RGB-D dataset, and it works and can be improved and expanded in the future.

2. In the back-end optimization, Levenberg-Marquardt algorithm and other algorithms has been used for iterative optimization by exploiting Ceres library and G2O library, so the camera pose estimation and the map point has been optimized, we found that Ceres library worked faster in the dataset we had chosen.

3. In the part of map building, the sparse map, dense map and occupancy grid map were built with the ORB-SLAM2, LSD-SLAM and RGB-D dataset respectively, functions of localization, navigation, and obstacle avoidance could be realized according to different map types, and the occupancy grid map has the advantage of small size memory occupation.

4. In the loop detection, emerged pictures were detected by unsupervised learning algorithm based on machine learning, and the dictionary based on ORB feature training was constructed, then bag of words model was constructed and the picture similarity was calculated by TF-IDF weights construction algorithm and the correct loop was detected.

Key Words: Monocular SLAM; RGB-D SLAM; Nonlinear Optimization; Mapping; 3-D Reconstruction

目录

第一章 绪论	1
1.1 课题背景及意义	1
1.2 国内外研究现状	1
1.2.1 激光 SLAM	2
1.2.2 单目 SLAM	3
1.2.3 双目 SLAM	4
1.2.4 RGB-D SLAM	4
1.3 SLAM 发展趋势	4
1.3.1 语义 SLAM	5
1.3.2 IMU+SLAM	5
1.4 论文结构	6
第二章 系统模型研究与算法分析	7
2.1 相机模型与标定	7
2.1.1 针孔相机模型	7
2.1.2 畸变模型及其校正	10
2.1.3 RGB-D 相机模型	16
2.2 相机刚体运动分析	16
2.2.1 旋转矩阵与变换矩阵	16
2.2.2 旋转向量	17
2.2.3 欧拉角	18
2.2.4 四元数	18
2.3 李群与李代数	20
2.3.1 群论	21
2.3.2 李群	21
2.3.3 李代数	21
2.4 非线性优化最小二乘法	21
2.4.1 非线性最小二乘	24
2.4.2 最速下降法	24
2.4.3 牛顿法	25
2.4.4 高斯牛顿法	26
2.4.5 列文伯格——马夸尔特法	26
2.4.6 Dogleg 算法	27
2.5 线性最小二乘法	27
2.6 本章小结	28
第三章 SLAM 系统研究与设计	29
3.1 传感器信息读取	29
3.2 视觉里程计前端设计	29
3.2.1 对应像素配准	29
3.2.2 基于特征点法的相机运动位姿估计	30
3.2.3 基于直接法的相机运动位姿估计	36
3.2.4 场景点位置估计	38

3.3 SLAM 后端优化.....	39
3.3.1 图优化简介.....	40
3.3.2 Bundle Adjustment.....	40
3.4 SLAM 建图.....	41
3.5 SLAM 回环检测.....	42
3.5.1 词袋模型与字典.....	42
3.5.2 回环判断.....	42
3.6 本章小结.....	43
第四章 系统设计实现与仿真分析.....	45
4.1 数据采集与数据集介绍.....	45
4.2 标定实验.....	45
4.2.1 普通摄像头标定.....	45
4.2.2 无人机摄像头标定.....	48
4.3 前端设计实验.....	48
4.4 后端优化实验.....	54
4.5 地图构建实验.....	56
4.5.1 稀疏重建.....	57
4.5.2 稠密重建.....	57
4.5.3 占据栅格地图.....	58
4.6 回环检测实验.....	59
4.6.1 词典训练.....	59
4.6.2 回环判断.....	60
4.7 本章小结.....	60
第五章 结论及展望.....	61
参考文献.....	63
攻读学位期间的研究成果.....	67
致谢.....	69
学位论文独创性声明.....	71
学位论文知识产权权属声明.....	71

第一章 绪论

1.1 课题背景及意义

人工智能是数学、计算机学、控制学、物理学等学科相结合的前沿技术科学，而该技术的表演者就有机器人、无人机、无人驾驶汽车、AR 技术等。扫地机器人、送货机器人以及智能仓储机器人等在逐渐普及，伴随智能导航避障的无人机也逐渐出现，无人驾驶汽车相继上路，AR 技术正与周围环境无缝贴合。上面这些现象的存在都离不开定位技术与建图技术，而定位技术与建图技术如今已经形成了一个新的概念——同时定位与地图构建。

同时定位与地图构建，英文全称 Simultaneous Localization And Mapping，简称 SLAM。该技术最早应用于机器人的自主定位与导航，机器人可以根据传感器获得周围环境的地图，来估计自身的位姿，并可以在机器人当前点与目标点之间检测障碍物以达到导航与避障的目的。而在如今的 SLAM 应用中，其体现出来的价值也越来越大，无人驾驶的定位技术现阶段则是利用激光雷达来进行实时定位与地图构建，其得益于高精度的激光雷达，但是缺点是价格太昂贵，随着固态激光雷达的出现，其价格很有可能直接下降到民用级，因此，激光雷达 SLAM 潜在价值仍然巨大。在全自主机器人的实现过程中，跟无人驾驶一样，机器人需要知道周围环境的情况，来进行路径规划，并进行导航与避障。单目 SLAM 以其价格低廉、体积小、便于与其他传感器融合、具有丰富特征等优点，因此被广泛研究，而 RGB-D 在室内效果较好、计算深度不依赖于场景纹理，且深度值已经由 TOF 原理或者结构光法计算出来，因此对于初学者来说，是较好的选择。

1.2 国内外研究现状

SLAM 技术最早由 Smith、Self 和 Cheeseman 提出并被认为是实现真正全自主移动机器人的关键技术^[1]。SLAM 系统是极为复杂的非线性非高斯系统，因此，在之后的很长一段时期内，都是以扩展卡尔曼滤波器（Extended Kalman Filter, EKF）的方法来进行 SLAM 的研究^[2]。由于 EKF 是建立在线性化误差以及服从高斯分布的噪声的假设之上的，其应用具有局限性，后来人们逐渐应用粒子滤波器（Particle Filter）等来进行 SLAM 中的状态估计^[3]。最近一段时间，人们发现基于非线性优化的方法比传统基于滤波器的方法得到的位姿更准确些，使得目前基于非线性优化的 SLAM 成为主流方向，也出现了非线性优化的开源库，例如基于图优化（Graph Optimization）的研究方法被广泛应用于 SLAM 中^[4]。在近期的 SLAM 的研究中，

学者们通过不断的尝试,最终形成了比较完善的视觉 SLAM 框架,即数据输入、里程计(前端)设计、后端优化与地图构建^[5]。

常用的 SLAM 传感器主要有激光雷达、单目、双目、RGB-D 相机等,激光雷达由于军事领域应用的需要,因此应用较早;而视觉传感器由于运算复杂,早期的 CPU 无法胜任大量的计算量,因此发展较为缓慢。随着计算机技术的发展以及 GPU 并行设备的出现,大大提升了计算速度。此外,大量工作人员不懈努力地改进算法,使得 SLAM 能够逐渐运用于普通 CPU 上,有的优秀 SLAM 算法甚至可以应用于手机终端。

1.2.1 激光 SLAM

激光雷达在 SLAM 中的运用可以分为二维激光 SLAM 与三维激光 SLAM,常用的几种激光 SLAM 分别是 Hector SLAM、Gmapping SLAM、Cartographer SLAM、KartoSLAM、CoreSLAM、LagoSLAM 等。

Hector SLAM 是由 Kohlbrecher 在 2011 年提出利用高斯牛顿法解决 Scan-matching 的算法,即激光雷达扫描的点云与地图点云进行匹配的问题^[6]。其优点是不需要额外的传感器(如里程计)并可运用于无人机或者自主机器人运动于较为复杂的环境,且其使用了多分辨率的形式来避免目标函数陷入局部最小值,且其状态估计可以与 IMU 进行融合,然后利用 EKF 进行滤波;缺点是由于该算法需要的激光雷达更新频率高、噪声低,因此不适合于快速运动的场景,而 IMU 在快速时结果才较为准确。

Cartographer SLAM 是 Google 公司的一个开源实时激光 SLAM 算法,可应用于 2D SLAM 也可以应用于 3D SLAM^[7]。其优点有速度快,利用了 SPA 与 BBS 算法,加速了整个工程的运行^[8];稳定度高,特别在与 IMU 进行数据融合之后,可以轻松应对于急转弯的情况;且该算法具有回环检测功能,可以消除累计误差。其缺点是需要一些激光雷达扫描得到的 Submap 来拼接成地图,在回环检测时用有误差的估计值来作为约束条件进行位姿优化,难免会长时间出现累计误差后使得误差不能完全消除。

Grisetti 等人于 2016 年提出了 Gmapping 算法,从实际应用上来讲是目前比较可靠的算法^[9]。Scan-matching 方法被用来估算自主机器人的位姿,并用梯度下降的迭代算法来求解最优位姿,该算法也采用了基于粒子滤波器的方法并考虑观测数据来减小自主机器人运动的不确定性^[10]。其优点是在走道、长廊等场景建图效果好,且不依赖于场景特征;缺点是不能应用于较复杂的非平坦区域,且没有回环检测,不能消除累计误差。实验也表明,在大场景中, Hector SLAM 的误差大于 Gmapping 算法,且在走廊的建模误差对比最为明显。

KartoSLAM 是 2010 年 Konolige 利用基于图优化的方法来进行非迭代 Kholesky 矩阵的稀疏分解求解最优值，由于图优化独特的优势，利用图优化的方法在大场景下有较大的优势^[11]。

Core SLAM 又称 Tiny SLAM，是由 Steux 与 Hamzaoui 等人于 2010 年提出的只有不到 200 行 C 代码的小型 SLAM 系统，其主要思想是将激光雷达扫描的数据用基于粒子滤波器的方式融合到定位系统中去^[12]。该算法的主要优点是代码量极少，易于初学者进行学习，缺点是没有回环检测等。

1.2.2 单目 SLAM

2007 年，A. J. Davison 提出了一个具有里程碑意义的实时单目 SLAM 系统：MonoSLAM^[13]。该 SLAM 系统以稀疏的特征点法作为前端里程计的设计，后端优化则利用基于 EKF 的滤波器来进行位姿优化，并设地图点服从高斯分布，在 SLAM 发展史上具有重要的意义。其缺点是不能在线运行该系统，且该特征点法容易跟踪丢失，导致系统失败。

同样在 2007 年，Y. Kameda 等人提出了 PTAM 算法，该算法同样具有重大意义，是第一个使用非线性优化的方法来进行后端优化，且该算法应用在了 AR 上，具有明确的实际意义；缺点是只能应用于小型场景，且容易跟踪丢失等^[14]。

2015-2016 年，R. Mur-Artal 等人提出了以单目为主的 ORB-SLAM 与可适用于单目、双目以及 RGB-D 相机的 ORB-SLAM2 系统，其前端采用改进的 ORB 特征为主的里程计设计，后端采用 Bundle Adjustment 的非线性优化方法来同时优化相机位姿与地图点，在回环检测部分利用 ORB 特征构建的词典，以及对每幅图像聚类成一个词袋，通过计算图像间的相似度来进行回环检测^[15]。该系统运行速率非常快，并且具有不易跟丢、程序易读等优点，是一个非常先进的单目 SLAM 系统^[16]。其不足之处在于特征点计算与匹配在程序中所占的时间较多，给 CPU 带来了比较沉重的负担。

2014-2016 年，J. Engel 等人分别提出了 LSD-SLAM 与 DSO 基于直接法的单目 SLAM，LSD-SLAM 是用于大场景的半稠密 SLAM，而 DSO 是稀疏的实时 SLAM，两个算法都是利用最小化光度误差的算法来计算相机位姿^[17-19]。LSD 算法在回环检测中仍然使用特征点法来进行，所以不是一个纯直接法的 SLAM，不过可以用来进行半稠密地图的构建。

2014 年，Foster 等人提出了 SVO 的半直接法单目 SLAM，这种算法在前端设计中仍然采用了特征提取的方法，只不过没有用对极几何的方法来计算位姿，而是利用最小化光度误差的方法来进行相机位姿的估计^[20]。其优势是速度极快，在 PC 端能以每秒 100 帧的速率进行计算；缺点是场景主要用于无人机，在平视相机中，效果不是太好。

1.2.3 双目 SLAM

2015 年, J. Engel 等人提出了 LSD-SLAM with Stereo, 该算法使用角点、边缘以及纹理具有高对比度的一类像素点来进行数据关联与图片对齐, 而且是一种基于关键帧的双目 SLAM 系统, 能够在普通 CPU 上实时高帧运行^[21]。

2016 年, P. L. Negre 等人提出了 Stereo SLAM, 该算法是基于 ROS 进行开发的, 主要用于水下机器人, 并用 G2O 库来进行图优化, 主要贡献是用关键点聚类的方法来进行回环检测, 能够在特征较少的环境提高校正率^[22]。

2017 年, R. Gomezjeda 等人提出了 PL-SLAM 算法, 这是一种利用点、线分割的双目 SLAM 算法, 其鲁棒性高, 能够用于各种场景包括低特征场景, 算法适用性较强^[23]。

1.2.4 RGB-D SLAM

2014 年, M. Labbé 等人提出了 RTAB-MAP (Real Time Appearance-Based Mapping) 算法, 主要用于 RGB-D 相机, 是一个比较完整的 SLAM 系统, 包括基于特征点法的前端设计、基于图优化的后端优化、基于词典的回环检测以及具有点云地图和 Mesh 三角地图的功能^[24]。支持如 Kinect1、Kinect2、Xtion 等 RGB-D 相机, 实时性强, 集成度高, 但是这也使得其很难进行二次开发。

2014 年, F. Endres 等人提出了 RGBD-SLAM-V2 方案, 作者从彩色图像中提取关键点并用深度图来确定彩色图中的点的 3D 位置, 然后用 RANSAC 算法来估计关联点之间的变换并用非线性优化方法来进行位姿图的优化^[25]。该算法能够在相机快速运动或者特征稀少 (Feature-poor) 的环境中运行, 鲁棒性高, 可以在线运行。

2015 年, N. Brunetto 等人提出了基于 Android 的 SLAM Dunk Android 算法, 是对传统移动端纯 RGB-D SLAM 算法的改进, 结合了 IMU 的位姿测量值, 框架应用了卡尔曼滤波器来细心选择状态变量, 并对 IMU 的值进行预处理以得到更简单有效的数据与 RGB-D 数据进行融合^[26]。这种算法展示了数量与质量的提升; 缺点是在空间位置需要进行两次积分中, 而加速度计的优势未体现出来。

2015-2016 年, D. Gutierrez-Gomez 等人提出了一种可以在 CPU 与 GPU 运行的基于直接和稠密的 RGBiD-SLAM 算法, 算法中 RGB-D 相机通过最小化当前帧相对于参考帧之间逐像素的重投影误差以及利用逆深度 (inverse depth, iD) 表达的几何误差^[27]。该系统还利用了词袋模型来进行回环检测并提升了重建效果, 也提供了根据场景元素的实时分割^[28]。

1.3 SLAM 发展趋势

SLAM 的发展趋势主要是根据应用情况来发展与进步的, 以后的 SLAM 会朝着轻量级、微型化发展或者朝着完备的地图信息的稠密化地图发展。轻量级应用主要

是用于手机移动端、AR、机器人等行业，这可以结合 IMU 进行数据融合来减轻对纯视觉 SLAM 应用的依赖；而完备地图主要是用于人机交互、稠密重建、丰富地图信息等方面，可以借助 GPU 来加速计算，也可用借助深度学习的方式来对 SLAM 场景中的物体进行语义标注。综上，SLAM 可能的发展趋势有语义 SLAM、IMU+SLAM、基于动态场景的 SLAM、基于线面特征的 SLAM、基于多机器人协作的 SLAM 等方向。

1.3.1 语义 SLAM

语义 SLAM 是指在 SLAM 运行过程中，其构建的地图只有纯粹的 3D 位置信息，而不知道地图中的这些点之间有什么关系，更不知道地图中的哪些点构成一个物体，也不知道构成什么物体，而这恰恰是人类与计算机的最大区别；计算机这样做就忽视了全局信息，而这些全局信息又极其重要，至少在数据关联、场景理解等有很高的应用价值。除此之外，SLAM 可协助进行物体识别，预测相机位姿，从而可以估计物体的运动，这样就可以自动进行语义标签的标注，进而节省人力标注，减轻人类负担。

语义 SLAM 近期的工作主要有 S. Yang 等人于 2016 年提出的 Pop-up SLAM 算法，该算法通过语义信息来提高在弱纹理环境下位姿估计信息与稠密地图的构建，在一定程度上提高了 SLAM 算法的普适性^[29]。

L. Bowman 等人于 2017 年提出了一种利用高级特征及语义标签进行 SLAM 的语义标识，也在闭环检测中有所贡献，在一些场合替代了原有的基于非监督学习方法的词袋模型，具有一定的实用性^[30]。

P. Luc 等人也在 2017 年提出了一种可以进行预测未来帧的语义分割信息的语义分割算法，主要用自动回归神经网络来迭代求解^[31]。

T. Pohlen 等人于 2016 年提出了一种用于自动驾驶汽车的语义图像分割算法，该算法用新提出的类 ResNet 网络架构进行定位和识别功能，能够达到实时的效果，具有优异的性能^[32]。

1.3.2 IMU+SLAM

另一个 SLAM 主要的发展趋势就是视觉 SLAM 与 IMU 的融合，IMU 与视觉 SLAM 可以进行互补。在相机快速运动的情况下，基于图像的 SLAM 会由于相机运动过快得到的图像间关联性太少而导致跟踪丢失，而在这种快速运动的情况下，IMU 恰恰可以获得比较精准的数据，此时，可以利用 IMU 的数据来补偿相机运动位姿以继续进行跟踪。当在相机运动过慢时，IMU 会由于设备本身的性质导致漂移很大，而视觉 SLAM 计算得到的相机位姿比较准确，因此，可以用 SLAM 计算得到的位姿来对 IMU 的测量数据进行纠正。

近期的工作有 2015 年 J. Huai 等人提出的 ORBSLAM_DWO 算法,该算法是利用非线性优化作为优化策略的双目——惯导 SLAM 系统^[33]。还有 M. Quan 等人于 2017 年提出的联合卡尔曼滤波基于地图的单目视觉惯导 SLAM 系统,用全新的紧耦合单目视觉惯导 SLAM 实现了高效的运动跟踪^[34]。

1.4 论文结构

本文主要研究了基于单目视觉与 RGB-D 相机的视觉 SLAM 系统,从原理分析、系统构建到程序实现都进行了较深入的分析。

论文具体结构如下:

第一章阐述了本课题研究的意义与背景,其应用价值极大,也调研了 SLAM 的国内外的研究现状,分别从激光 SLAM 与视觉 SLAM 进行了调研。

第二章介绍了 SLAM 系统构建所需的相关代数、几何、非线性优化等相关的知识,即论文中所需要用到的算法和理论,在第四章中需要用于进行算法的比较。

第三章介绍了视觉 SLAM 的基本架构并推导了视觉 SLAM 的一些基本原理并对视觉 SLAM 的前端里程计设计、后端优化、回环检测与建图都进行了比较深入的分析。

第四章主要是对第二章与第三章一些基本原理及过程的实现,利用 RGB-D 数据进行 SLAM 前端系统的设计,后端优化中利用 G2O 与 Ceres 库对所选数据进行了优化,达到了理想的效果,并用 Dogleg 算法代替 LM 算法,加快了 SLAM 的运行速率。建图部分则利用 ORB-SLAM、LSD-SLAM 分别建出了稀疏地图与半稠密地图,用八叉树地图建立了占据栅格地图,回环检测部分则利用词典模型建立 ORB 词典进行了分析。

第五章对论文进行了总结,并提出了视觉 SLAM 的展望。

第二章 系统模型研究与算法分析

2.1 相机模型与标定

视觉 SLAM 通常以单目相机、双目相机、RGB-D 相机作为输入设备，最终都可归结为对相机采集的图像数据进行处理。双目相机通过对采集得到的左右视图计算视差并根据基线宽度算出距离；RGB-D 相机通过红外相机结构光方法或者 TOF 方法测得距离并通过标定与彩色相机进行数据融合；单目 SLAM 较为复杂一些，需要进行多视图几何学的分析并通过三角化方法得到尺度不能固定的深度距离。三者共同点是都需要进行相机模型分析然后进行相机内参标定并通过坐标系的转换进行空间点的像素坐标、归一化坐标、三维空间坐标的相互转换，以达到对深度的估计与三维坐标的还原。

常用的相机模型有针孔相机模型、畸变模型、鱼眼相机模型、双目相机模型、RGB-D 相机模型、仿射模型、透视模型等。其中，对普通常用的摄像头而言，针孔相机模型足够准确描述这类图像的成像过程，而对于畸变较大的相机则需要采用畸变模型来消除图像在成像过程中由于镜头本身的影响带来的畸变，鱼眼相机模型、双目相机模型与 RGB-D 相机模型分别主要用于鱼眼相机、双目相机与 RGB-D 相机，仿射模型、透视模型在实际图像处理中也有用到，但是在文中不作考虑。

2.1.1 针孔相机模型

最常用的相机模型是针孔相机模型，它解释了现实场景中的物体投影到针孔之后的成像平面的成像原理，相关联的四个坐标系为图像像素直角坐标系、图像物理坐标系、相机坐标系与世界坐标系。

(1) 图像像素直角坐标系与图像物理坐标系及其转换

相机通过 CMOS(Complementary Metal-Oxide Semiconductor)或者 CCD(Charge Coupled Device)等感光元件接收来自外界的光线，然后通过各自的编码方式进行光线强度到数字信号的转换并压缩存储到闪存卡中。采集到的图像以标准电视信号的形式输入到计算机中，并以矩阵形式存储起来。

如图 2.1 所示，定义图像左上角为图像像素直角坐标系的原点 O_i ，则有图像像素直角坐标系 O_i-u-v ，其中，坐标 (u,v) 为像素坐标， u,v 分别代表图像矩阵的行数和列数，其值为该点像素的灰度，灰度值范围为 0 到 255 的整数，0 表示黑，255 表示白，以 1 个字节存储。

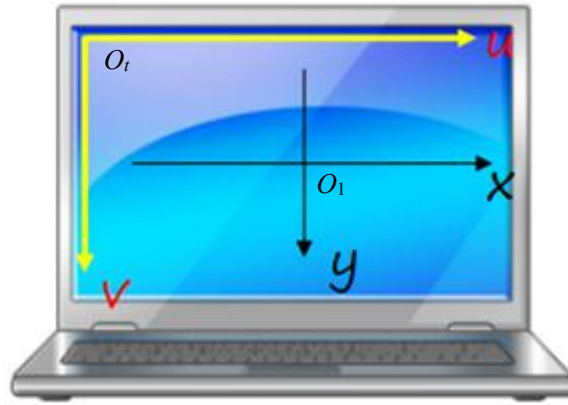


图 2.1 图像像素直角坐标系与图像物理坐标系

因为像素坐标系只定义了像素的位置，并没有定义出该像素与在图像中的位置关系与具体尺度关系，所以需要定义以 O_1 为坐标原点的且能以物理单位表示的图像物理坐标系 O_1-X-Y ，其中，不妨假设相机成像于相机前方，即成像平面位于相机光心 O_c 前方 f 距离处， f 为相机焦距，单位为米。

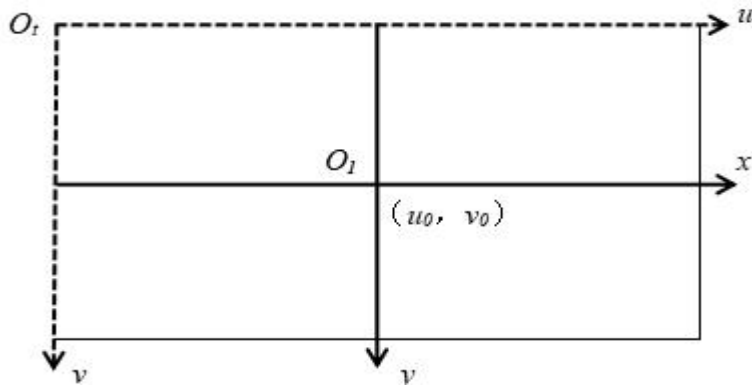


图 2.2 图像物理坐标系与图像像素坐标系之间的转换

如图 2.2 所示， O_1 在图像像素坐标系中的坐标为 (u_0, v_0) ，单个像素在 X 轴与 Y 轴方向的物理尺寸分别为 d_x 与 d_y ，单位为米/像素，则物理坐标系中一点与图像像素坐标系之间的关系为：

$$\begin{cases} u = \frac{x}{d_x} + u_0 \\ v = \frac{y}{d_y} + v_0 \end{cases} \quad 2-(1)$$

将图像像素坐标用齐次坐标表示为 $(u, v, 1)^T$ ，图像物理坐标系齐次坐标为 $(x, y, 1)^T$ ，用矩阵表达为：

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

(2) 相机坐标系、世界坐标系、图像物理坐标系与图像像素坐标系的转换

如图 2.3 所示，定义世界坐标系为 $O_w-X_w-Y_w-Z_w$ ，以相机光心 O_c 为原点的相机坐标系为 $O_c-X-Y-Z_c$ 。相机光心与图像物理坐标系原点 O_1 距离为焦距 f 。

对于相机坐标系为 $O_c-X-Y-Z_c$ 与图像物理坐标系 O_1-X-Y ，设相机坐标系中齐次坐标点 $(X_c, Y_c, Z_c, 1)$ 在图像物理坐标系的齐次坐标投影点为 $(x, y, 1)$ ，则根据三角形相似关系有：

$$\begin{cases} x = f \frac{X_c}{Z_c} \\ y = f \frac{Y_c}{Z_c} \end{cases} \quad 2-(3)$$

用矩阵表达为：

$$Z_c \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \quad 2-(4)$$

该式联立 2-(2) 式可得：

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} \triangleq K P_c \quad 2-(5)$$

其中 $f_x = \frac{f}{d_x}$ ， $f_y = \frac{f}{d_y}$ 分别表示焦距 f 在图像像素坐标系下 u 轴和 v 轴所占据

的像素个数，并定义矩阵 K 为相机内参矩阵（Camera Intrinsics）。

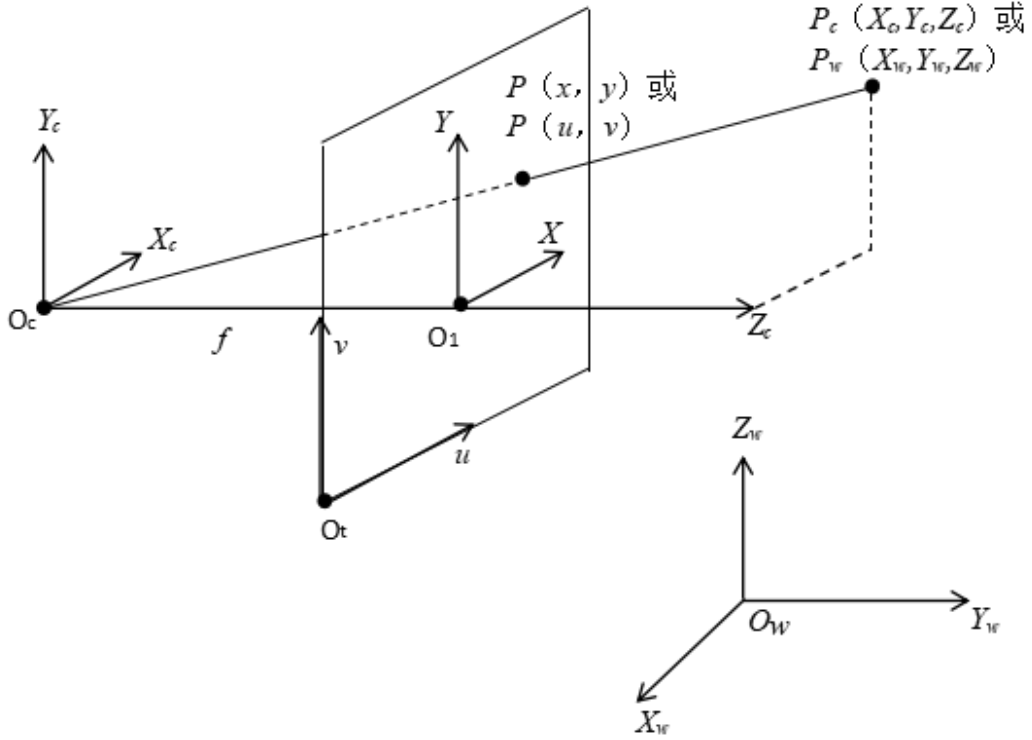


图 2.3 相机坐标系与世界坐标系及其转换

相机坐标系与世界坐标系的转换，包括旋转变换与平移变换，可将旋转变换与平移变换用变换矩阵 T_{cw} 与 T_{wc} 来表示，并将该变换矩阵定义为相机的外参数矩阵（Camera Extrinsic），简称为外参。记世界坐标系中有一点 $P_w = (X_w, Y_w, Z_w, 1)$ ，记世界坐标系到相机坐标系的变换矩阵为 T_{cw} ， P_w 经过坐标变换之后在相机坐标系下的坐标为 P_c ，则有：

$$P_c = T_{cw} P_w \quad 2-(6)$$

其中，外参矩阵为：

$$T_{cw} = \begin{bmatrix} R & t \\ \theta^T & 1 \end{bmatrix} \quad 2-(7)$$

2.1.2 畸变模型及其校正

为了得到更好的成像效果，通常会在相机前面加透镜来进行改善。但是加入镜头会带来新的误差，并称相机物镜在设计、制作、装配等工艺所导致的图像成像点偏离理想坐标点的误差为光学畸变，光学畸变分为径向畸变与偏心畸变。文中的畸变模型都是在相机归一化成像平面进行畸变模型分析。

(1) 径向畸变

径向畸变是由透镜形状而引起的，成像点会沿着径向方向偏离理想坐标位置。如图 2.4 所示，径向畸变可分为桶形畸变与枕形畸变，图中 (a) 为理想像素位置，(b) 为枕形畸变，原因是图像放大率会随着与光轴之间的距离的增大而增大，而 (c) 为桶形畸变，与枕形畸变相反。

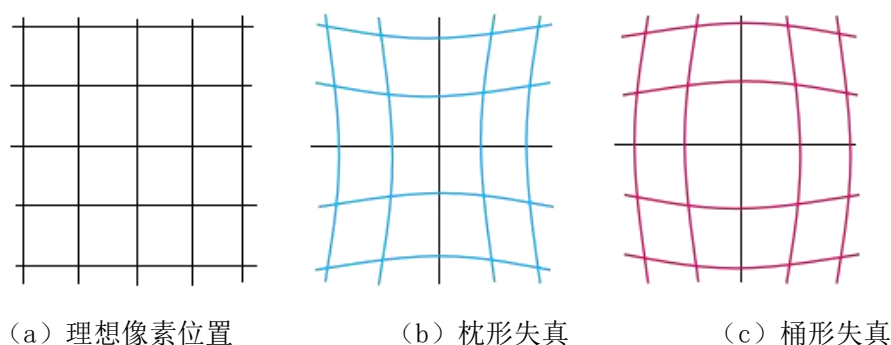


图 2.4 径向畸变

(2) 偏心畸变

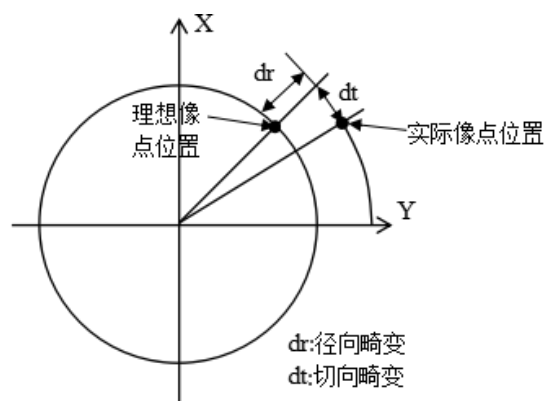


图 2.5 偏心畸变

如图 2.5 为偏心畸变，偏心畸变会使得成像点沿着径向方向和沿着垂直径向方向偏离理想成像坐标位置，其中沿着径向方向畸变的 d_r 叫做非对称畸变，垂直径向畸变的 d_t 称为切向畸变。

(3) 畸变校正模型

对于径向畸变，距离中心越远，畸变程度越大，因此，可以用多项式拟合的方法来逼近原来的真实成像位置。

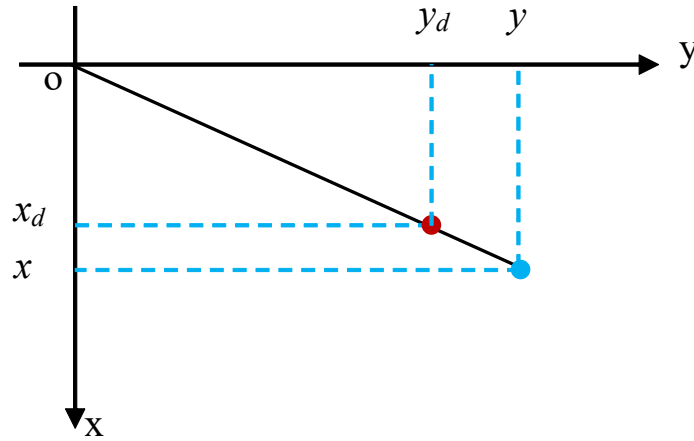


图 2.6 径向畸变校正模型

如图 2.6 所示，假设图像中的理想坐标点为 (x, y) ，畸变之后的坐标点为 $(x_{distorted}, y_{distorted})$ ，则有：

$$r_0 = \sum_{i=1}^n k_i i^r \quad 2-(8)$$

式中：

$$r_0 = \sqrt{x_{distorted}^2 + y_{distorted}^2}, \quad r = \sqrt{x^2 + y^2} \quad 2-(9)$$

其中， k_i 为畸变系数。通常可取如下式进行径向畸变的校正：

$$\begin{cases} x_{distorted} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \\ y_{distorted} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \end{cases} \quad 2-(10)$$

利用该式可根据原图中各个像素的位置用拟合得到畸变后的像素位置，若拟合出的像素坐标超出图像范围，则舍去；若拟合出的像素坐标在图像范围内，则将畸变校正图中的像素灰度值改为原图灰度值大小；若最终校正图像中的像素坐标位置未能按照原图像素坐标拟合出，则将其像素值置空，并且在处理该像素时跳过该点。对于畸变较小的图像区域， k_1 起主要作用，较大畸变处 k_2 起主要作用。因此对于普通摄像头可用 k_1 与 k_2 进行拟合，而畸变较大的图像可以加入 k_3 进行拟合。

对于偏心畸变，由于透镜装配的误差，致使透镜与成像平面不平行，从而引起了偏心畸变，此变形由径向变形分量与切向变形分量共同构成，可用如下数学模型：

$$\begin{cases} x_{distorted} = x + 2p_1xy + p_2(r^2 + 2x^2) \\ y_{distorted} = y + 2p_2xy + p_1(r^2 + 2y^2) \end{cases} \quad 2-(11)$$

其中， p_1 与 p_2 为切向畸变校正参数。

因此，对于相机归一化坐标点 \mathbf{P} ，可同时使用径向畸变和偏心畸变进行畸变校正，如下式所示：

$$\begin{cases} x_{distorted} = x(1 + k_1r^2 + k_2r^4 + k_3r^6) + x + 2p_1xy + p_2(r^2 + 2x^2) \\ y_{distorted} = y(1 + k_1r^2 + k_2r^4 + k_3r^6) + y + 2p_2xy + p_1(r^2 + 2y^2) \end{cases} \quad 2-(12)$$

(4) 相机标定

在很多情况下，相机的内参并没有给定出来，或者厂家给定的内参并不一定都准确，而我们为了应用针孔相机模型需要确定空间物体表面点的三维集合位置以及确定其在图像中对应的像素位置，就必须知晓相机的内参数。因此，需要采取一定行之有效地手段进行相机的标定工作，这是做好后续工作的前提，也是重点之所在。相机标定的工程就是利用已知的空间三维点坐标 (x,y,z) 以及其对应的图像像素坐标 (u,v) 来计算相机的几何特性参数以及光学特性参数等内部参数，也可计算出其空间点的坐标关系即外部参数。内参包括镜头焦距 f ，镜头畸变参数 k 、 p ，坐标扭曲因子 s ，图像坐标原点即主点 (u_0,v_0) 等参数。外部参数则包括世界坐标系到相机坐标系的旋转矩阵 \mathbf{R} 以及平移向量 \mathbf{t} 等参数。

常用的标定方法有线性标定方法、非线性优化标定方法、两步标定法等^[35]，其优点与缺点如表 2.1 所示，线性标定法优点是运算速度快，但是精度较低；非线性优化标定方法的标定精度高，但缺点是模型复杂且计算量较大；而我们最常用最经典的方法便是张正友的棋盘标定技术^[36]，因为张正友标定法易于实现，且参数求解简便，对于精度不是太高的情况，可以采用该标定法来进行标定^[37]。

表 2.1 常用标定技术

标定方法	优点	缺点
线性标定方法	运算速度快	标定的精度不高
非线性优化标定方法	标定精度高	模型复杂计算量过大
Tsai 的经典两步法	参数求解简易	标定成本高
张正友的标定方法		

由于本项目中主要用张正友标定法，故此处只介绍张氏标定法。如图 2.7 所示，根据相机针孔模型，其标定平面到成像平面的投影关系为：

$$s \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = K \begin{pmatrix} r_1 & r_2 & r_3 & t \end{pmatrix} \begin{pmatrix} X \\ Y \\ 0 \\ 1 \end{pmatrix} = K \begin{pmatrix} r_1 & r_2 & t \end{pmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \quad 2-(13)$$

其中, s 为尺度因子, K 为相机内参矩阵, 假设标定模板平面位于世界坐标系 $Z=0$ 的平面, 那么检测出的角点在世界坐标系中的坐标为 $M=(X,Y,0,1)^T$, $m=(u,v,1)^T$ 为标定板上检测出的该角点在成像平面上的投影坐标, (r_1, r_2, r_3) 为旋转矩阵。

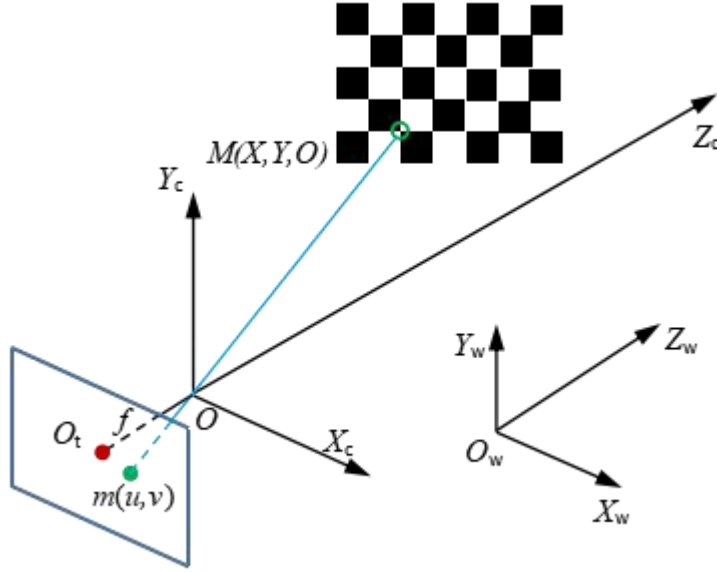


图 2.7 张正友棋盘标定法

记 $k(r_1, r_2, t)$ 为 H , 称之为单应性矩阵, 则有:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = H \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad 2-(14)$$

其中, 令 H 为:

$$H = (h_1, h_2, h_3) = \lambda K(r_1, r_2, t) \quad 2-(15)$$

H 为齐次矩阵, 可用文中 3.2.2 节基于特征点法的相机运动位姿估计方法来解算该矩阵, 需要 4 个对应点来求解。根据旋转矩阵为正交矩阵的性质得:

$$\begin{cases} \mathbf{r}_1^T \mathbf{r}_2 = 0 \\ \|\mathbf{r}_1\| = \|\mathbf{r}_2\| = 1 \end{cases} \quad 2-(16)$$

则每个对应点可获得两个方程：

$$\begin{cases} \mathbf{h}_1^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{h}_2 = 0 \\ \mathbf{h}_1^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{h}_1 = \mathbf{h}_2^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{h}_2 \end{cases} \quad 2-(17)$$

相机有 5 个未知参数，因此，当图像数量有 3 张以上时，可以用文中 2.5 节线性最小二乘的解法解得 \mathbf{H} 。

令 \mathbf{B} 为：

$$\mathbf{B} = \mathbf{A}^{-T} \mathbf{A}^{-1} = \begin{pmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{pmatrix} \quad 2-(18)$$

则可求得如下结果：

$$\begin{cases} v_0 = (B_{11}B_{13} - B_{11}B_{23}) / (B_{11}B_{12} - B_{12}^2) \\ \lambda = B_{33} - [B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23})] / B_{11} \\ \alpha = \sqrt{\lambda / B_{11}} \\ \beta = \sqrt{\lambda B_{11} / (B_{11}B_{12} - B_{12}^2)} \\ \gamma = -B_{12}\alpha^2\beta / \lambda \\ u_0 = v_0\gamma / \beta - B_{13}\alpha^2 / \lambda \end{cases} \quad 2-(19)$$

由该结果也可求得：

$$\begin{cases} \mathbf{r}_1 = \lambda_0 \mathbf{A}^{-1} \mathbf{h}_1 \\ \mathbf{r}_2 = \lambda_0 \mathbf{A}^{-1} \mathbf{h}_2 \\ \mathbf{t} = \lambda_0 \mathbf{A}^{-1} \mathbf{h}_3 \end{cases} \quad 2-(20)$$

通常，由于拍摄图像噪声的干扰，可用最小二乘法来求解。构造目标函数为：

$$\min \sum_{i=1}^m \sum_{j=1}^n \|\mathbf{m}_{ij} - \hat{\mathbf{m}}(\mathbf{A}, k_1, k_2, \mathbf{R}_i, \mathbf{t}_i, \mathbf{M})\|_2^2 \quad 2-(21)$$

其中, m_{ij} 为标定板上的角点像素坐标, \hat{m}_0 函数为相机成像模型, 可利用该式来求解各内参以及外参。

2.1.3 RGB-D 相机模型

RGB-D 相机如 Kinect 1 代与 Kinect 2 代分别是以结构光的原理和 TOF (Time-of-Flight) 的原理来测量场景点到相机之间距离的。

结构光法有一个结构光发射器与一个结构光接收器, 主要原理是通过 RGB-D 相机向物体发射一束红外光线, 然后相机通过返回的结构光的图案来计算到物体的距离; TOF 的原理则较为简单, 有一个脉冲光发射器和一个脉冲光接收器, 通过计算光束的飞行时间来测得两者之间的距离。

得到了这个深度距离之后, RGB-D 相机会通过深度图的坐标系位姿与 RGB 相机的位姿进行空间坐标的转换, 这个可以通过 RGB-D 相机标定来完成。通过标定的 RGB-D 相机则可以获得每一个像素的像素坐标与其深度距离, 然后可根据相机模型获得其空间三维坐标。

2.2 相机刚体运动分析

2.2.1 旋转矩阵与变换矩阵

相机的运动可以认为是一个刚体运动, 可由旋转变换与平移变换合成, 这种变换也称为欧式变换, 其特点是同一个向量在欧式变换前后的长度和大小都不会发生变化。旋转矩阵用于描述两个坐标系之间的旋转关系, 而平移矩阵用于描述两个坐标系原点的位置关系, 两者合起来称作以齐次坐标表示的变换矩阵。

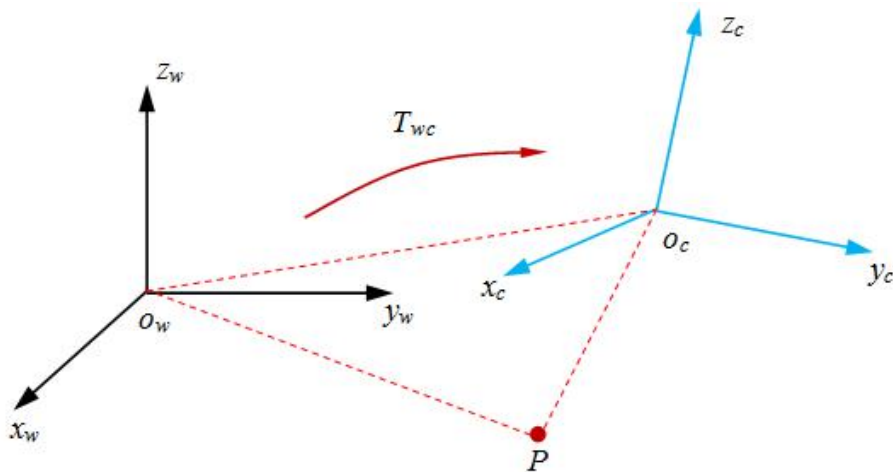


图 2.8 相机坐标系与世界坐标系坐标变换

如图 2.8 所示, 对于空间点 P , 坐标系 $O_w-x_w-y_w-z_w$ 与 $O_c-x_c-y_c-z_c$ 所表示的坐标不一样, 但是可以由变换矩阵进行相互转换。设 P_w 为点 P 在世界坐标系 $O_w-x_w-y_w-z_w$

中的表示， P_c 为点 P 在相机坐标系 $O_c-x_c-y_c-z_c$ 的表示，世界坐标系到相机坐标系的表示为 T_{cw} ，则有：

$$P_c = T_{cw} P_w \quad 2-(22)$$

P_c 为世界坐标系中的一点 P_w 经过变换矩阵 T_{cw} 变换之后后在相机坐标系中的表示。其中，有：

$$T_{cw} = \begin{bmatrix} R_{cw} & t_{cw} \\ \theta^T & 1 \end{bmatrix} \quad 2-(23)$$

2.2.2 旋转向量

旋转矩阵能够完美地表达刚体的旋转，但是缺点也很明显，其一是用 9 个量表示 3 个自由度的旋转比较冗余；其二是旋转矩阵本身是一个强约束的矩阵，必须满足正交矩阵以及行列式为 1 的性质，因此在求解以旋转矩阵或者含有旋转矩阵的优化问题时，约束使之变得比较困难。所以我们可以用另一种方式来表达旋转，即旋转向量。这是一种用三维向量表达的旋转，方向与旋转轴一致且长度等于旋转角的三维向量被定义为旋转角，也被称为轴角（Axis-Angle）。用这种方式定义的旋转可以将旋转矩阵从 9 个自由度降低到 3 个自由度，同时可将 16 个自由度的变换矩阵的自由度降低到 6 个。

旋转向量也可以通过罗德里格斯公式转换为旋转向量^[38]：

$$R = \cos \theta I + (1 - \cos \theta) n n^T + \sin \theta n^{\wedge} \quad 2-(24)$$

其中， n^{\wedge} 表示单位向量 n 的反对称矩阵， θ 为旋转角的大小。

同样可以通过罗德里格斯公式将旋转矩阵转换为旋转向量，对于 θ ，并利用迹函数的性质有：

$$\text{tr}(R) = \cos \theta \text{tr}(I) + (1 - \cos \theta) \text{tr}(n n^T) + \sin \theta \text{tr}(n^{\wedge}) = 1 + 2 \cos \theta \quad 2-(25)$$

因此，可得 θ ：

$$\theta = \arccos\left(\frac{\text{tr}(\mathbf{R}) - 1}{2}\right) \quad 2-(26)$$

求得 θ 之后，可以通过旋转轴上的向量在绕该轴旋转之后不发生改变的性质得到方程：

$$\mathbf{R}\mathbf{n} = \mathbf{n} \quad 2-(27)$$

因此，可以求解该线性方程，易求得转轴即为旋转矩阵 \mathbf{R} 特征值 1 所对应的特征向量，进而求得该特征向量然后归一化就得到了旋转轴。

2.2.3 欧拉角

虽然旋转向量有降低旋转矩阵冗余度的优点，且没有约束，但是其表达方式不太直观，难以理解该旋转是如何进行的。为了解决这个问题人们提出了用欧拉角来描述旋转，欧拉角用三个独立的转角来形象地描述旋转关系，其分解方式有多种，如 XYZ、ZYX 等表达方式。XYZ 定义为先绕 X 轴旋转，再绕 Y 轴旋转，最后绕 Z 轴进行旋转，其他表达方式以此类推。通常用的一种旋转方式是 ZYX，也就是航空领域常用的 rpy (roll-pitch-yaw，滚转-俯仰-偏航)。其定义方式为绕 Z 轴旋转，得偏航角 yaw，绕 Y 轴旋转得俯仰角 pitch，绕 X 轴旋转得滚转角 roll。

欧拉角 $(\mathbf{r}, \mathbf{p}, \mathbf{y})^T$ 可以形象直观地表达出旋转，自由度也只有 3 个，但是，其重大缺点便是会遇到万向锁 (Gimbal Lock) 问题，即当旋转轴绕 Y 轴旋转时，若俯仰角为直角，会使得第三次旋转与第一次旋转为同一个旋转，使其自由度下降，这种现象称为旋转的奇异性问题。因此，在很多情况，仍然满足不了应用，因为此时的欧拉角不适合进行迭代与插值，不过可以直观地验证自己算法的正确与否。

2.2.4 四元数

J. Stuelpnagel 等人于 1964 年已经证明，不存在不带奇异性的三维向量来表达旋转^[39]，为了紧凑地、不带奇异性地表达三维旋转，Hamilton 扩展了一种可用于表达三维旋转的复数——四元数 (Quaternion)^[40]。优点是不带奇异，不存在万向锁的问题，自由度也只有 4 个；缺点是计算复杂，不如欧拉角直观。

一个四元数有一个实部和三个虚部，定义为：

$$\mathbf{q} = q_0 + q_1i + q_2j + q_3k \quad 2-(28)$$

可记为：

$$\mathbf{q} = [q_0, q_1, q_2, q_3] \quad 2-(29)$$

我们可以用单位四元数来表述刚体在三维空间中的任意一个旋转，用四元数乘以 i 则为绕 i 轴旋转 180 度；若绕 i 轴旋转 360 度，会得到与原来相反的刚体；若需要用四元数经过旋转得到与原来一样的位置，则需要绕该轴旋转 720 度。

同样，类似旋转向量与旋转矩阵可通过罗德里格斯公式进行转换，也可以用一定的方式将四元数与旋转向量、四元数与旋转矩阵进行相互转换。

旋转向量与四元数的转换关系如下：

$$\mathbf{n} = [n_x, n_y, n_z]^T \quad 2-(30)$$

进行了 θ 角的旋转后，这个旋转的四元数形式为：

$$\mathbf{q} = \left[\cos \frac{\theta}{2}, n_x \sin \frac{\theta}{2}, n_y \sin \frac{\theta}{2}, n_z \sin \frac{\theta}{2} \right] \quad 2-(31)$$

反之，也可以从单位四元数中就算出旋转轴与夹角：

$$\begin{cases} \theta = 2 \arccos q_0 \\ [n_x, n_y, n_z]^T = [q_1, q_2, q_3]^T / \sin \frac{\theta}{2} \end{cases} \quad 2-(32)$$

设四元数 $q = q_0 + q_1 i + q_2 j + q_3 k$ ，可证明旋转矩阵与四元数的关系如下，四元数 \mathbf{q} 到旋转矩阵 \mathbf{R} 的转换为：

$$\mathbf{R} = \begin{bmatrix} 1-2q_2^2-2q_3^2 & 2q_1q_2+2q_0q_3 & 2q_1q_3-2q_0q_2 \\ 2q_1q_2-2q_0q_3 & 1-2q_1^2-2q_3^2 & 2q_2q_3+2q_0q_1 \\ 2q_1q_3+2q_0q_2 & 2q_2q_3-2q_0q_1 & 1-2q_1^2-2q_2^2 \end{bmatrix} \quad 2-(33)$$

同样，若令旋转矩阵 $\mathbf{R} = \{m_{ij}\}, i, j \in [1, 2, 3]$ ，则旋转矩阵 \mathbf{R} 到四元数 \mathbf{q} 的变换为：

$$\begin{cases} q_0 = \frac{\sqrt{\text{tr}(\mathbf{R})+1}}{2} \\ q_1 = \frac{m_{23}-m_{32}}{4q_0} \\ q_2 = \frac{m_{31}-m_{13}}{4q_0} \\ q_3 = \frac{m_{12}-m_{21}}{4q_0} \end{cases} \quad 2-(34)$$

四元数 q 也可以将一个三维空间点 $p = [x, y, z] \in R^3$ 进行旋转，该空间点用四元数表示为：

$$p = [0, x, y, z] = [0, v] \quad 2-(35)$$

可证明，用四元数 q 旋转之后的空间点 p' 为：

$$p' = qpq^{-1} \quad 2-(36)$$

转换后的 p' 与原来的点 p 都属于纯虚四元数，虚部的 3 个分量表示空间点三维坐标。

2.3 李群与李代数

由前面知道，表达空间三维旋转的方式有旋转矩阵、旋转向量（轴角）、欧拉角、四元数等，各自的表达方式有各自的优缺点，在应用当中得依照实际情况来定夺采取何种表达方式较为有利。虽然表达旋转的方式有上述几种，但是在视觉 SLAM 中，解决的关键问题就是根据观测数据来求取相机位姿。相机位姿包含旋转矩阵 R 与平移矩阵 t ，观测数据为图像像素，由空间点经过针孔相机模型投影得到。在实际求取图像位姿时有非线性优化方法以及基于滤波器的方法，近来，视觉 SLAM 中采用非线性优化方法得到的位姿估计比基于滤波器的方法估计的位姿要准确^[41]。文中也会采用基于非线性优化的方法来求取相机位姿，但是用非线性优化的方法需要对优化变量进行求导，而旋转矩阵自身是有约束的，在优化时会引入其本身带来的约束，使得优化更加复杂，并且会增加很多的计算量，而通过引入李群、李代数的数学概念，可以将该问题从有约束的优化问题转换为无约束的优化问题。

2.3.1 群论

群是一种定义在一个非空集合和一种运算符号上的一种代数结构，设该非空集合为 G ，运算符记为“ \cdot ”，群的定义需要满足以下条件：

- (1) 封闭性： $\forall g_1, g_2 \in G, g_1 \cdot g_2 \in G$.
- (2) 结合律： $\forall g_1, g_2, g_3 \in G, (g_1 \cdot g_2) \cdot g_3 = g_1 \cdot (g_2 \cdot g_3)$.
- (3) 单位元： $\exists g_0 \in G, \forall g \in G, g_0 \cdot g = g \cdot g_0 = g$.
- (4) 逆元： $\forall g \in G, \exists g^{-1} \in G, \text{s.t. } g \cdot g^{-1} = g^{-1} \cdot g = g_0$.

群在各种数学问题中经常出现，常见的群有整数加法 $(\mathbb{Z}, +)$ ，去掉 0 后的有理数的乘法 $(\mathbb{Q} \setminus 0, \cdot)$ 等等。矩阵相关的常见群有一般线性群 $GL(n)$ 、特殊正交群 $SO(n)$ 、特殊欧式群 $SE(n)$ 等，其中一般线性群指 n 阶可逆矩阵对矩阵乘法构成群，特殊正交群为 n 维旋转矩阵群，特殊欧式群即 n 维欧式变换群。

2.3.2 李群

具有连续、光滑性质的群叫做李群 (Lie Group)，是 1870 年由挪威数学家 S. Lie 提出的概念，广泛应用于物理学、几何学与拓扑学中^[42]。文中所用的旋转矩阵群 $SO(3)$ 、变换群 $SE(3)$ 与相似变换群 $Sim(3)$ 都属于李群，在文中的旋转矩阵、变换矩阵中应用广泛，表示为后端优化变量。

2.3.3 李代数

每一个李群都有其对应的李代数，李代数表达了李群的局部性质。对一个集合 V ，一个数域 F 以及一个二元运算符 $[\cdot, \cdot]$ ，若满足如下性质：

- (1) 封闭性：对 $\forall X, Y \in V$ ，有： $[X, Y] \in V$ 。
- (2) 双线性：对 $X, Y, Z \in V$ ，有： $[aX + bY, Z] = a[X, Z] + b[Y, Z]$ ， $[Z, aX + bY] = a[Z, X] + b[Z, Y]$ 。
- (3) 由于自反性：对 $\forall X \in V$ ，有： $[X, X] = 0$ 。
- (4) 雅克比等价：对： $\forall X, Y, Z \in V$ ，有： $[X, [Y, Z]] + [Z, [X, Y]] + [Y, [Z, X]] = 0$ 。则记 $(V, F, [\cdot, \cdot])$ 为一个李代数，记做 \mathfrak{g} ，其中 $[\cdot, \cdot]$ 叫做李括号。

2.4 非线性优化最小二乘法

在视觉 SLAM 中，其系统由一个运动方程与一个观测方程组成，即有如下方程式：

$$\begin{cases} \mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{w}_k \\ \mathbf{z}_{k,j} = h(\mathbf{y}_j, \mathbf{x}_k) + \mathbf{v}_{k,j} \end{cases} \quad 2-(37)$$

其中, $\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{w}_k$ 为运动方程, 描述了在输入 \mathbf{u}_k , 且噪声为 \mathbf{w}_k 时, 相机位姿 \mathbf{x}_{k-1} 如何运动到 \mathbf{x}_k 的; $\mathbf{z}_{k,j} = h(\mathbf{y}_j, \mathbf{x}_k) + \mathbf{v}_{k,j}$ 为观测方程, 此处 $\mathbf{z}_{k,j}$ 为相机在位姿 \mathbf{x}_k 处对空间点 \mathbf{y}_j 的观测值, 在视觉 SLAM 中一般为图像像素坐标, \mathbf{y}_j 为路标点, 即空间中的三维场景点坐标, 其噪声项为 $\mathbf{v}_{k,j}$ 。设运动方程与观测方程的噪声项均服从零均值的高斯分布:

$$\mathbf{w}_k \sim N(0, \mathbf{R}_k), \quad \mathbf{v}_{k,j} \sim N(0, \mathbf{Q}_{k,j}) \quad 2-(38)$$

其中, $\mathbf{R}_k, \mathbf{Q}_{k,j}$ 为协方差矩阵。

在没有提供获取相机位姿的一般 SLAM 系统中, 是没有运动方程的, 因此, 只有观测方程:

$$s\mathbf{z}_{k,j} = \mathbf{K} \exp(\zeta^{\wedge}) \mathbf{y}_j \quad 2-(39)$$

而 SLAM 系统中, 我们的最终目标就是求相机的位姿及其观测到的空间三维点的集合, 因此, 可以将位姿与空间三维点放到一个变量 \mathbf{x} 中, 则有新的待估计状态变量为 $\mathbf{x} = \{\mathbf{x}_1, \dots, \mathbf{x}_N, \mathbf{y}_1, \dots, \mathbf{y}_M\}$, 并应用概率学来进行理论的可行性分析, 已知观测数据 \mathbf{z} , 则状态变量 \mathbf{x} 的条件概率分布为:

$$\mathbf{P}(\mathbf{x} | \mathbf{z}) \quad 2-(40)$$

有了此方程之后, 我们可以利用贝叶斯公式将其展开为:

$$\mathbf{P}(\mathbf{x} | \mathbf{z}) = \frac{\mathbf{P}(\mathbf{z} | \mathbf{x}) \mathbf{P}(\mathbf{x})}{\mathbf{P}(\mathbf{z})} \quad 2-(41)$$

其中, $\mathbf{P}(\mathbf{x} | \mathbf{z})$ 为后验概率, $\mathbf{P}(\mathbf{x})$ 为先验概率, $\mathbf{P}(\mathbf{z} | \mathbf{x})$ 为似然。而 SLAM 系统的目的是根据此方程求出后验概率的分布, 但是实际情况是先验概率未知, 因此, 可以求解状态变量 \mathbf{x} 的后验概率最大化:

$$\mathbf{x}_{MAP}^* = \arg \max \mathbf{P}(\mathbf{x} | \mathbf{z}) = \arg \max \mathbf{P}(\mathbf{z} | \mathbf{x}) \mathbf{P}(\mathbf{x}) \quad 2-(42)$$

由于先验概率未知, 求解 \mathbf{x} 的后验概率最大可以转化为求解 \mathbf{x} 的最大似然估计

(Maximize Likelihood Estimation) :

$$\mathbf{x}_{MLE}^* = \arg \max \mathbf{P}(\mathbf{z} | \mathbf{x}) \quad 2-(43)$$

其物理意义是在已知观测值时，求出最适合该观测值的状态变量 \mathbf{x} 。

由式 2-(37)知，观测方程为：

$$\mathbf{z}_{k,j} = h(\mathbf{y}_j, \mathbf{x}_k) + \mathbf{v}_{k,j} \quad 2-(44)$$

且噪声项服从高斯分布为：

$$\mathbf{v}_k \sim N(0, \mathbf{Q}_{k,j}) \quad 2-(45)$$

则观测方程的条件概率分布为：

$$\mathbf{P}(\mathbf{z}_{j,k} | \mathbf{x}_k, \mathbf{y}_j) \sim N(h(\mathbf{y}_j, \mathbf{x}_k), \mathbf{Q}_{k,j}) \quad 2-(46)$$

其展开式为：

$$\mathbf{P}(\mathbf{z}_{j,k}) = \frac{1}{\sqrt{(2\pi)^N |\mathbf{Q}_{k,j}|}} \exp\left(-\frac{1}{2} (\mathbf{z}_{j,k} - h(\mathbf{x}_k, \mathbf{y}_j))^T \mathbf{Q}_{k,j}^{-1} (\mathbf{z}_{j,k} - h(\mathbf{x}_k, \mathbf{y}_j))\right) \quad 2-(47)$$

对式 2-(47)两边同时取负对数，得：

$$-\ln(P(\mathbf{z}_{j,k})) = \frac{1}{2} \ln((2\pi)^N |\mathbf{Q}_{k,j}|) + \frac{1}{2} (\mathbf{z}_{j,k} - h(\mathbf{x}_k, \mathbf{y}_j))^T \mathbf{Q}_{k,j}^{-1} (\mathbf{z}_{j,k} - h(\mathbf{x}_k, \mathbf{y}_j)) \quad 2-(48)$$

现在，求状态变量 $\mathbf{x}_k, \mathbf{y}_j$ 的最大似然估计相当于求下式：

$$(\mathbf{x}_k, \mathbf{y}_j)^* = \arg \min((\mathbf{z}_{j,k} - h(\mathbf{x}_k, \mathbf{y}_j))^T \mathbf{Q}_{k,j}^{-1} (\mathbf{z}_{j,k} - h(\mathbf{x}_k, \mathbf{y}_j))) \quad 2-(49)$$

定义观测值与观测模型的误差为：

$$\mathbf{e}_{y,j,k} = \mathbf{z}_{k,j} - h(\mathbf{x}_k, \mathbf{y}_j) \quad 2-(50)$$

则 2-(49)式等价于:

$$(\mathbf{x}_k, \mathbf{y}_j)^* = \arg \min_{\mathbf{y},j,k} \mathbf{e}_{y,j,k}^T \mathbf{Q}_{k,j}^{-1} \mathbf{e}_{y,j,k} \quad 2-(51)$$

问题则转变为了最小二乘问题，即求取满足式 2-(51)的最佳状态变量 $\mathbf{x}_k, \mathbf{y}_j$ ，由于观测方程即为针孔相机模型，那么可以用李代数来表示变换矩阵，将有约束的优化问题转换为无约束的优化问题，由于目标函数是非线性的，因此，该问题变为非线性最优化问题，非线性最优化问题可用最速下降法、牛顿法、高斯牛顿法与列文伯格——马夸尔特法等方法来求解，考虑到问题的非凸性与工程问题的稳定性，常常使用高斯牛顿法与列文伯格——马夸尔特法来进行迭代优化，求取最佳的优化变量值。

2.4.1 非线性最小二乘

设最小二乘问题为:

$$\min \frac{1}{2} \|f(\mathbf{x})\|_2^2 \quad 2-(52)$$

其中， \mathbf{x} 为 n 维的向量， $f(\mathbf{x})$ 为 m 维的非线性函数，由于其非线性，所以不能直接用求导的方式来得到极值，一般用迭代的方法，先按照一定的策略选取一个初始值，然后进行迭代计算，并计算 $f(\mathbf{x})$ 的值；若增量足够小，则停止迭代。具体步骤如下：

- (1) 按照一定策略选取一个初始值 \mathbf{x}_0 ，并设置迭代次数为 1；
- (2) 确定迭代方向 \mathbf{n} ；
- (3) 用寻优的策略确定步长 λ ；
- (4) 计算是否满足停止条件（常用的停止条件为函数值 $f(\mathbf{x})$ 足够小或者增量 Δx 足够小），若条件满足，则停止迭代， \mathbf{x}_0 为所求值；若不满足，则迭代次数加 1，并令 $\mathbf{x}_0 = \mathbf{x}_0 + \lambda \mathbf{n}$ ，返回第（2）步开始计算。

步长与迭代方向的选取有多种方法，常用策略有最速下降法、牛顿法、高斯牛顿法与列文伯格——马夸尔特法等^[43]。

2.4.2 最速下降法

将目标函数 $\|f(\mathbf{x})\|_2^2$ 在 \mathbf{x} 进行一阶泰勒展开，得：

$$\|f(\mathbf{x} + \Delta\mathbf{x})\|_2^2 \approx \|f(\mathbf{x})\|_2^2 + J(\mathbf{x})\Delta\mathbf{x} \quad 2-(53)$$

其中, $J(\mathbf{x}) = \left(\frac{\partial \|f(\mathbf{x})\|_2^2}{\partial x_1} \frac{\partial \|f(\mathbf{x})\|_2^2}{\partial x_2} \dots \frac{\partial \|f(\mathbf{x})\|_2^2}{\partial x_n} \right)$ 为目标函数的 $\|f(\mathbf{x})\|_2^2$ 关于 \mathbf{x} 的雅克比矩阵。利用迭代的方式来求 2-(53) 式的最小值, 则迭代时其增量方向为梯度的负方向, 即 $-J(\mathbf{x})^T$ 若取步长为 $\lambda=1$, 则增量为:

$$\Delta\mathbf{x} = -J(\mathbf{x})^T \quad 2-(54)$$

这种方法可求得最快的下降方式, 因此称为最速下降法, 其优点是计算速度非常快, 能够在凸优化问题中快速求解梯度方向, 然后以负梯度方向进行一定的步长进行迭代求解。

2.4.3 牛顿法

若将目标函数 $\|f(\mathbf{x})\|_2^2$ 在 \mathbf{x} 进行二阶泰勒展开, 得:

$$\|f(\mathbf{x} + \Delta\mathbf{x})\|_2^2 \approx \|f(\mathbf{x})\|_2^2 + J(\mathbf{x})\Delta\mathbf{x} + \frac{1}{2} \Delta\mathbf{x}^T H \Delta\mathbf{x} \quad 2-(55)$$

其中, $J(\mathbf{x})$ 为目标函数的 $\|f(\mathbf{x})\|_2^2$ 关于 \mathbf{x} 的雅克比矩阵, H 为 Hessian 矩阵:

$$H = \begin{pmatrix} \frac{\partial^2 \|f(\mathbf{x})\|_2^2}{\partial x_1^2} & \frac{\partial^2 \|f(\mathbf{x})\|_2^2}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 \|f(\mathbf{x})\|_2^2}{\partial x_1 \partial x_n} \\ \frac{\partial^2 \|f(\mathbf{x})\|_2^2}{\partial x_2 \partial x_1} & \frac{\partial^2 \|f(\mathbf{x})\|_2^2}{\partial x_2^2} & \dots & \frac{\partial^2 \|f(\mathbf{x})\|_2^2}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 \|f(\mathbf{x})\|_2^2}{\partial x_n \partial x_1} & \frac{\partial^2 \|f(\mathbf{x})\|_2^2}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 \|f(\mathbf{x})\|_2^2}{\partial x_n^2} \end{pmatrix} \quad 2-(56)$$

为求目标函数的增量, 则可以将二阶泰勒展开对 $\Delta\mathbf{x}$ 进行求导并令其为 0, 可得:

$$J(\mathbf{x})^T + H\Delta\mathbf{x} = 0 \quad 2-(57)$$

由此式容易求得目标函数的增量，这种方法叫做牛顿法。

2.4.4 高斯牛顿法

在最速下降法与高斯法中，都是将非线性函数 $f(\mathbf{x})$ 的平方进行一阶和二阶泰勒展开，而高斯牛顿法则直接将非线性函数 $f(\mathbf{x})$ 先进行泰勒展开：

$$f(\mathbf{x} + \Delta\mathbf{x}) = f(\mathbf{x}) + \mathbf{J}(\mathbf{x})\Delta\mathbf{x} \quad 2-(58)$$

为了求得 $\Delta\mathbf{x}$ ，可以构建一个最小二乘问题：

$$\min_{\Delta\mathbf{x}} \frac{1}{2} \|\mathbf{f}(\mathbf{x}) + \mathbf{J}(\mathbf{x})\Delta\mathbf{x}\|_2^2 \quad 2-(59)$$

将 2-(59) 式展开并对 $\Delta\mathbf{x}$ 进行求导，令该式为 0，则可得方程组：

$$\mathbf{J}(\mathbf{x})^T \mathbf{J}(\mathbf{x})\Delta\mathbf{x} = -\mathbf{J}(\mathbf{x})\mathbf{f}(\mathbf{x}) \quad 2-(60)$$

其中，2-(60) 式称为高斯牛顿方程，或者正规方程。定义变量如下：

$$\begin{cases} \mathbf{H} = \mathbf{J}(\mathbf{x})^T \mathbf{J}(\mathbf{x}) \\ \mathbf{g} = -\mathbf{J}(\mathbf{x})\mathbf{f}(\mathbf{x}) \end{cases} \quad 2-(61)$$

则有增量方程：

$$\mathbf{H}\Delta\mathbf{x} = \mathbf{g} \quad 2-(62)$$

可用解线性方程组的方法来求解该方程，得到 $\Delta\mathbf{x}$ ，并求取最终的优化变量 \mathbf{x} 。

2.4.5 列文伯格——马夸尔特法

高斯牛顿法是在目标函数的二阶泰勒展开点附近进行近似的，这种方法只能在此展开点附近有效，为了让增量求解更加鲁棒，列文伯格与马夸尔特法选取了一种信赖区域算法^[44]。当增量在信赖区域内时，认为近似有效，否则，近似无效。可用下式来判断近似好坏：

$$\rho = \frac{f(\mathbf{x} + \Delta\mathbf{x}) - f(\mathbf{x})}{J(\mathbf{x})\Delta\mathbf{x}} \quad 2-(63)$$

其中，分母为真实函数的变化值，分母为估计的近似值。当 ρ 较小时，则说明真实变化值太小，需减小信赖区域的半径；当 ρ 较大时，则说明估计值较小，需要扩大信赖区域的半径；当 ρ 近似为 1 时，则说明近似较好。

由于是将增量限定在一定区域之内的，在求解非线性优化问题时，相当于引入了约束，可以引入拉格朗日乘子将一个非线性无约束优化问题转换成非线性有约束问题，如下式：

$$\min_{\Delta\mathbf{x}} = \frac{1}{2} \|f(\mathbf{x}) + J(\mathbf{x})\Delta\mathbf{x}\|_2^2 + \frac{\lambda}{2} \|D\Delta\mathbf{x}\|_2^2 \quad 2-(64)$$

2.4.6 Dogleg 算法

Dogleg 算法是高斯牛顿法与最速下降法的另一个混合方法，不用阻尼项，而是用信赖域的方法来进行步长范围的保障，设其下降方向为：

$$\mathbf{h}_{sd} = -F'(\mathbf{x}_k) = -J_k^T \mathbf{f}_k \quad 2-(65)$$

则步长可根据该算法得：

$$\alpha = -\frac{\mathbf{h}_{sd}^T J(\mathbf{x})^T \mathbf{f}(\mathbf{x})}{\|J(\mathbf{x})\mathbf{h}_{sd}\|^2} = \frac{\|\mathbf{g}\|^2}{\|J(\mathbf{x})\mathbf{g}\|^2} \quad 2-(66)$$

则用高斯牛顿法则时取：

$$\mathbf{x}_{k+1} - \mathbf{x}_k = \mathbf{h}_{gn} \quad 2-(67)$$

用最速下降法则时取：

$$\mathbf{x}_{k+1} - \mathbf{x}_k = \alpha \mathbf{h}_{sd} \quad 2-(68)$$

2.5 线性最小二乘法

线性最小二乘法较为简单，设有线性方程：

$$\mathbf{Ax} = \mathbf{b} \quad 2-(69)$$

其中 \mathbf{A} 为系数矩阵， \mathbf{x} 为变量， \mathbf{b} 为向量。构建最小二乘问题的目标函数为：

$$\min_{\mathbf{x}} (\|\mathbf{Ax} - \mathbf{b}\|_2^2) \quad 2-(70)$$

该问题实质则是多变量的二次优化问题，将该目标函数展开并对 \mathbf{x} 求导得：

$$\frac{\partial (\|\mathbf{Ax} - \mathbf{b}\|_2^2)}{\partial \mathbf{x}} = 2\mathbf{A}^T \mathbf{Ax} - 2\mathbf{A}^T \mathbf{b} \quad 2-(71)$$

整理得：

$$\mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} \quad 2-(72)$$

2-(72)式即为所求最小二乘解。

2.6 本章小结

本章主要阐述了视觉 SLAM 的一些必备的几何、代数、优化理论等知识，分析了相机模型与成像投影像素位置的关系，推导了相机标定模型，并在第四章用张正友标定法进行相机标定，研究了刚体运动的各种表示方法，包括旋转矩阵、平移矩阵、变换矩阵、四元数、欧拉角等，并用李群与李代数进行表示，然后进行了非线性优化的推导，比较了四种优化算法，将用于 SLAM 的后端优化之中。

第三章 SLAM 系统研究与设计

随着近年来许许多多研究者们的辛勤付出,视觉 SLAM 系统架构也渐趋稳定和统一,目前较为成熟的 SLAM 系统架构分为前端设计、后端优化、回环检测、建图重构^[45]。

3.1 传感器信息读取

SLAM 系统信息数据的源传感器主要有激光雷达、单目相机、双目相机、RGB-D 相机等。激光雷达所采集的数据为场景的点云数据,即场景的 3D 坐标;单目相机获取的是彩色图像;双目相机采集的数据为两个相对位置不变的相机所拍摄的两视角图像;RGB-D 相机获取的是场景的彩色图像以及彩色图像对应的深度图。SLAM 系统经过对传感器获取的数据进行校正、滤波等操作后作为系统的输入。

3.2 视觉里程计前端设计

视觉里程计前端设计有基于特征点法的前端与基于直接法的前端。前端主要是计算机视觉领域中对图像的处理,经过特征匹配或者光流法等方法寻找两幅图像中同一个空间点的投影,找到对应关系之后,即可运用几何关系与相机模型计算出初始相机位姿估计,然后通过计算出来的相机位姿估计并利用三角化原理计算空间点坐标。

3.2.1 对应像素配准

对于空间一点 P ,经过不同方位的相机(一般为同一相机)得到两幅图像后,由于图像以灰度值存储图像像素,并且计算机也不能像人一样很轻易知道图像中哪些点为同一个场景,因此,在前面较长一段时间内,很多学者都研究了大量的算法来解决这一问题,常见的思路有利用图像中比较有代表性的点或者用利用整幅图像,利用图像的运动特征来找到对应点,分别为特征点法和光流法。

图像的特征点是指图像中一些“有特征”的点,这些点比较特殊,比如角点等,另外,图像中还有很大一部分没有特征的点,这些点通常是在大面积颜色变化较少且比较平整的地方。由于受到光线变化、动态物体、尺度等问题的影响,很多特征还需要根据应用场景的需求具有光度不变性、尺度不变性、方向不变性等性质,以达到鲁棒性的要求。特征点法的匹配有两个步骤,特征提取与特征匹配,特征提取得到特征点在图像中的像素位置,甚至有方向、尺度等信息,然后在特征匹配过程中计算各个特征点的描述子,选取两幅图像中描述子最近的两个点作为一对匹配点。

常用的特征点算法有 SIFT、SURF、ORB、DAISY 等特征点^[46-49]。这些特征点

大都具有光度不变性、尺度不变性、方向不变性等性质，因此常用于特征匹配中。根据这些算法提取了图像特征点的关键点的像素位置之后，根据实际情况，也可定义关键点方向、尺度等信息。例如可以利用灰度质心法来设计关键点方向信息，也可以利用图像金字塔模型来定义图像的尺度。

特征匹配是指在两幅图像中，通过一定的方式来求得两幅图像中同一个空间点的投影，而描述一个点可以利用该点周围像素的信息来进行辅助描述，然后按照一定的方式存储为向量，该向量就称为描述子(Descriptor)。得到了描述子之后就可以运用匹配算法来进行匹配，常用的匹配算法有暴力匹配、FLANN 等算法^[50]。匹配的准则是计算两个描述子的距离如汉明距离、欧氏距离、马氏距离等，选取距离最近的两个点作为匹配的点。

3.2.2 基于特征点法的相机运动位姿估计

(1) 对极几何求解

1) 对极约束求解

如图 3.1 所示，空间点 P 经过相机从两个不同角度拍摄，分别得到图像 I_1 的像素 p_1 图像 I_2 的像素 p_2 ，点 P 与 p_1 与 p_2 构成的平面叫做极平面，连接两个相机的光心的连线 O_1O_2 ，此线叫做基线，与两幅图像 I_1 与 I_2 分别交于点 e_1 与点 e_2 ， e_1 与 e_2 叫做极点，其中，连线 p_1e_1 与连线 p_2e_2 分别为图像 I_1 与 I_2 的极线。

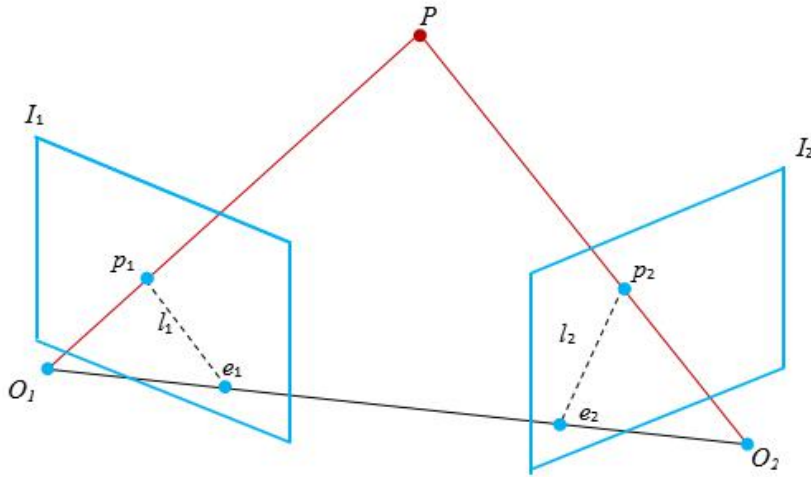


图 3.1 对极几何约束

设在第一个相机坐标系下点 P 的空间位置为：

$$P = (X, Y, Z)^T \quad 3-(1)$$

若已知像素 p_1, p_2 位置，由针孔相机模型知：

$$s_1 \mathbf{p}_1 = \mathbf{K} \mathbf{P}, s_2 \mathbf{p}_2 = \mathbf{K} (\mathbf{R} \mathbf{P} + \mathbf{t}) \quad 3-(2)$$

使用归一化坐标表示，则有：

$$\mathbf{x}_1 = \mathbf{K}^{-1} \mathbf{p}_1, \mathbf{x}_2 = \mathbf{K}^{-1} \mathbf{p}_2 \quad 3-(3)$$

$\mathbf{x}_1, \mathbf{x}_2$ 为归一化像素平面的坐标。

易得：

$$\mathbf{x}_2 = \mathbf{R} \mathbf{x}_1 + \mathbf{t} \quad 3-(4)$$

两边同左乘 \mathbf{t}^\wedge ，得：

$$\mathbf{t}^\wedge \mathbf{x}_2 = \mathbf{t}^\wedge \mathbf{R} \mathbf{x}_1 \quad 3-(5)$$

两边再左乘 \mathbf{x}_2^T ：

$$\mathbf{x}_2^T \mathbf{t}^\wedge \mathbf{x}_2 = \mathbf{x}_2^T \mathbf{t}^\wedge \mathbf{R} \mathbf{x}_1 = 0 \quad 3-(6)$$

代入 $\mathbf{p}_1, \mathbf{p}_2$ ，有：

$$\mathbf{p}_2^T \mathbf{K}^{-T} \mathbf{t}^\wedge \mathbf{R} \mathbf{K}^{-1} \mathbf{p}_1 = 0 \quad 3-(7)$$

因此有对极约束：

$$\begin{cases} \mathbf{x}_2^T \mathbf{t}^\wedge \mathbf{R} \mathbf{x}_1 = 0. \\ \mathbf{p}_2^T \mathbf{K}^{-T} \mathbf{t}^\wedge \mathbf{R} \mathbf{K}^{-1} \mathbf{p}_1 = 0. \end{cases} \quad 3-(8)$$

定义本质矩阵 \mathbf{E} ：

$$\mathbf{E} = \mathbf{t}^\wedge \mathbf{R} \quad 3-(9)$$

基础矩阵 F :

$$F = K^{-T} E K^{-1} \quad 3-(10)$$

则对极约束可简化为:

$$x_2^T E x_1 = p_2^T F p_1 = 0 \quad 3-(11)$$

对极约束描述了两个像素点的空间位置关系, 然后就可以根据空间的两个匹配点先求出 E 或 F , 然后分解得到 R 、 t , 常用的方法有八点法。

可证明, 根据本质矩阵的性质得到本质矩阵的奇异值为 $[\sigma, \sigma, 0]^T$, 而平移与旋转一共有 6 个自由度, 因此, E 具有 6 个自由度, 如果利用尺度等价的性质, 可得到 E 具有 5 个自由度。因此, 可以用 5 对点来求解本质矩阵 E , 而用 5 对点来求解比较困难, 通常考虑矩阵 E 的线性性质使用八点法——(Eight-Point-Algorithm) 来求解 E ^[51]。

设空间点 P 在两个相机坐标系下的归一化坐标为

$$x_1 = [u_1, v_1, 1]^T, x_2 = [u_2, v_2, 1]^T \quad 3-(12)$$

由对极约束得:

$$(u_1, v_1, 1) \begin{pmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & e_9 \end{pmatrix} \begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} = 0 \quad 3-(13)$$

令 e 为:

$$e = [e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9]^T. \quad 3-(14)$$

将 3-(13) 式展开可得:

$$[u_1 u_2, u_1 v_2, u_1, v_1 u_2, v_1 v_2, v_1, u_2, v_2, 1] \cdot e = 0 \quad 3-(15)$$

将 8 对匹配点对带入可得:

$$\begin{pmatrix} u_1^1 u_2^1 & u_1^1 v_2^1 & u_1^1 & v_1^1 u_2^1 & v_1^1 v_2^1 & v_1^1 & u_2^1 & v_2^1 & 1 \\ u_1^2 u_2^2 & u_1^2 v_2^2 & u_1^2 & v_1^2 u_2^2 & v_1^2 v_2^2 & v_1^2 & u_2^2 & v_2^2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_1^8 u_2^8 & u_1^8 v_2^8 & u_1^8 & v_1^8 u_2^8 & v_1^8 v_2^8 & v_1^8 & u_2^8 & v_2^8 & 1 \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_7 \\ e_8 \\ e_9 \end{pmatrix} = 0 \quad 3-(16)$$

3-(16)式可以用奇异值分解（SVD 分解）的方法得到：

$$\mathbf{E} = \mathbf{U} \mathbf{D} \mathbf{V}^T \quad 3-(17)$$

其中， \mathbf{U} ， \mathbf{V} 为正交矩阵 \mathbf{D} 为奇异值矩阵，由 \mathbf{D} 的性质得知，

$$\mathbf{D} = \text{diag}(\sigma, \sigma, 0) \quad 3-(18)$$

而用八点法得到的 \mathbf{D} 可能具有式 3-(18)的性质，因此需要进行调整，设 \mathbf{E} 的奇异值为 $\sigma_1 \geq \sigma_2 \geq \sigma_3$ ，则可取

$$\mathbf{E} = \mathbf{U} \text{diag}\left(\frac{\sigma_1 + \sigma_2}{2}, \frac{\sigma_1 + \sigma_2}{2}, 0\right) \mathbf{V}^T \quad 3-(19)$$

常常可取 $\text{diag}(1,1,0)$ ，以简化运算，这是利用了尺度等价性质。

2) 共面点成像单应性矩阵求解

另一种已知两幅二维图像求得相机位姿的方法是利用单应性矩阵，并通过直接线性变换法（Direct Linear Transform, DLT）得到。不过这种方法使用与场景点在同一平面上，通常无人机俯视相机拍摄的场景较多。

同样，设空间点经过相机从两个不同角度拍摄，分别得到图像 \mathbf{I}_1 的像素 \mathbf{p}_1 图像 \mathbf{I}_2 的像素 \mathbf{p}_2 ，设多个场景点位于统一个平面 \mathbf{P} 上，则有：

$$\mathbf{n}^T \mathbf{P} + d = 0. \quad 3-(20)$$

代入式 $p_2 = K(RP + t)$ ，得：

$$p_2 = K \left(R - \frac{tn^T}{d} \right) K^{-1} p_1. \quad 3-(21)$$

则得到了一个关于两幅图像像素 p_1 与 p_2 之间的一个方程，即

$$p_2 = Hp_1 \quad 3-(22)$$

其中， H 被定义为单应性矩阵：

$$H = K \left(R - \frac{tn^T}{d} \right) K^{-1} \quad 3-(23)$$

该矩阵描述了两幅图像的内参、旋转、平移以及一批场景点的共同平面的关系。求解该矩阵用 DLT 方法即可，即将 3-(21) 式展开：

$$\begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} \quad 3-(24)$$

取 $h_9=1$ 则有：

$$\begin{cases} h_1 u_1 + h_2 v_1 + h_3 - h_7 u_1 u_2 - h_8 v_1 u_2 = u_2 \\ h_4 u_1 + h_5 v_1 + h_6 - h_7 u_1 v_2 - h_8 v_1 v_2 = v_2 \end{cases} \quad 3-(25)$$

即一对匹配对可以得到两个方程，由于 H 矩阵的自由度为 8，因此可以用 4 对匹配对可进行求解 H 。方程如下：

$$\begin{pmatrix} u_1^1 & v_1^1 & 1 & 0 & 0 & 0 & -u_1^1 u_2^1 & -v_1^1 v_2^1 \\ 0 & 0 & 0 & u_1^1 & v_1^1 & 1 & -u_1^1 v_2^1 & -v_1^1 v_2^1 \\ u_1^2 & v_1^2 & 1 & 0 & 0 & 0 & -u_1^2 u_2^2 & -v_1^2 v_2^2 \\ 0 & 0 & 0 & u_1^2 & u_1^2 & 1 & -u_1^2 v_2^2 & -v_1^2 v_2^2 \\ u_1^3 & v_1^3 & 1 & 0 & 0 & 0 & -u_1^3 u_2^3 & -v_1^3 v_2^3 \\ 0 & 0 & 0 & u_1^3 & u_1^3 & 1 & -u_1^3 v_2^3 & -v_1^3 v_2^3 \\ u_1^4 & v_1^4 & 1 & 0 & 0 & 0 & -u_1^4 u_2^4 & -v_1^4 v_2^4 \\ 0 & 0 & 0 & u_1^4 & u_1^4 & 1 & -u_1^4 v_2^4 & -v_1^4 v_2^4 \end{pmatrix} \begin{pmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \end{pmatrix} = \begin{pmatrix} u_2^1 \\ v_2^1 \\ u_2^2 \\ v_2^2 \\ u_2^3 \\ v_2^3 \\ u_2^4 \\ v_2^4 \end{pmatrix} \quad 3-(26)$$

求解出单应性矩阵之后，同样可以将该矩阵进行分解，以求得相机的旋转矩阵 \mathbf{R} 与平移矩阵 \mathbf{t} 。

利用单应性矩阵的意义在于当相机有纯旋转以及场景点位于同一平面时作为对极约束的替代方案，因为在这两种情况下，对极约束下的本质矩阵 \mathbf{E} 的自由度会降低，会有退化现象存在。而在实际情况中，会同时计算这两个矩阵，最终使用重投影误差较小的作为结果。

(2) PnP 问题求解

PnP (Perspective-n-Point) 即透视 n 点问题，是求解 3D-2D 点对运动的方法，描述的是已知 n 个空间点及其投影位置时，如何估计相机的位姿。PnP 问题求解方法有：P3P、直接线性变换、EPnP、UPnP 等等^[52,53]。此外，还有非线性优化的方式，构建最小二乘问题并迭代求解，即 Bundle Adjustment。下面用直接线性变换法与 Bundle Adjustment 分别进行求解。

1) 直接线性变换法

对空间点 \mathbf{P} ，齐次坐标为 $\mathbf{P} = (X, Y, Z, 1)^T$ 。在图像 I_1 中，投影到归一化平面齐次坐标为 $\mathbf{x}_1 = (u_1, v_1, 1)^T$ ，设 \mathbf{R} ， \mathbf{t} 未知，定义增广矩阵 $[\mathbf{R} | \mathbf{t}]$ 为一个 3×4 的矩阵：

$$s \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} = \begin{pmatrix} t_1 & t_2 & t_3 & t_4 \\ t_5 & t_6 & t_7 & t_8 \\ t_9 & t_{10} & t_{11} & t_{12} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad 3-(27)$$

消去 s ，可得：

$$\begin{cases} u_1 = \frac{t_1 X + t_2 Y + t_3 Z + t_4}{t_9 X + t_{10} Y + t_{11} Z + t_{12}} \\ u_1 = \frac{t_5 X + t_6 Y + t_7 Z + t_8}{t_9 X + t_{10} Y + t_{11} Z + t_{12}} \end{cases} \quad 3-(28)$$

定义 \mathbf{T} 的行向量:

$$\mathbf{t}_1 = (t_1, t_2, t_3, t_4)^T, \mathbf{t}_2 = (t_5, t_6, t_7, t_8)^T, \mathbf{t}_3 = (t_9, t_{10}, t_{11}, t_{12})^T \quad 3-(29)$$

则有:

$$\begin{cases} \mathbf{t}_1^T \mathbf{P} - \mathbf{t}_3^T \mathbf{P} u_1 = 0 \\ \mathbf{t}_2^T \mathbf{P} - \mathbf{t}_3^T \mathbf{P} v_1 = 0 \end{cases} \quad 3-(30)$$

因此, 一个点提供了两个关于 \mathbf{t} 的线性方程, 假设一共有 N 个特征点, 则有:

$$\begin{pmatrix} \mathbf{P}_1^T & 0 & -u_1 \mathbf{P}_1^T \\ 0 & \mathbf{P}_1^T & -v_1 \mathbf{P}_1^T \\ \vdots & \vdots & \vdots \\ \mathbf{P}_N^T & 0 & -u_N \mathbf{P}_N^T \\ 0 & \mathbf{P}_N^T & -v_N \mathbf{P}_N^T \end{pmatrix} \begin{pmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \\ \mathbf{t}_3 \end{pmatrix} = 0. \quad 3-(31)$$

\mathbf{t} 维度为 12, 因此需要至少 6 对 3D-2D 匹配对来求解, 当匹配点大于 6 对点时, 可以最小二乘法来求解该超定线性方程。

直接求得 \mathbf{T} 中的 \mathbf{R} 不一定满足旋转矩阵的内在约束, 此时, 可用 QR 分解完成, 转换成结果接近旋转性质和平移两部分。

2) Bundle Adjustment

Bundle Adjustment 算法, 简称 BA, 即光束平差法, 该算法是相机模型方程与投影方程中所有的未知参数都当做优化变量来同时进行非线性优化的方法, 优化的变量有李代数表示的位姿 ξ 与空间点 \mathbf{P} 的三维坐标, 即使得其重投影误差最小的位姿与空间点即为该算法的最终解。

3.2.3 基于直接法的相机运动位姿估计

另一类估计相机位姿的方法为直接法, 直接法估计位姿的前提条件为灰度不变性的假设: 即场景点在图像中成像并且在下一帧中, 对应场景点在图像中的像素灰度不变。不过此假设的应用范围会受到限制, 在光照变化较大、闪光下以及材质不

同都会影响到位姿估计的准确性。其优点为对特征点要求较低，即使在低纹理场景下也能实用；缺点是对内参、相机运动速度比较敏感。

跟据灰度不变假设，可以推导直接法求解相机位姿的具体过程，最终可将问题建模成为一个最小化光度误差来求解。假设空间点：

$$\mathbf{P} = (X, Y, Z)^T \quad 3-(32)$$

其在相邻两帧中的像素坐标分别为 \mathbf{p}_1 与 \mathbf{p}_2 ，跟据针孔相机模型以及李代数表达位姿可得如下关系式：

$$\begin{cases} Z_1 \mathbf{p}_1 = Z_1 \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}_1 = \mathbf{K} \mathbf{P} \\ Z_2 \mathbf{p}_2 = Z_2 \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}_2 = \mathbf{K} (\mathbf{R} \mathbf{P} + \mathbf{t}) = \mathbf{K} (\exp(\hat{\boldsymbol{\xi}}) \mathbf{P}) \end{cases} \quad 3-(33)$$

\mathbf{p}_1 与 \mathbf{p}_2 分别在两幅图像中的灰度值为 $I_1(\mathbf{p}_1)$ 与 $I_2(\mathbf{p}_2)$ ，定义这两个像素的亮度误差为：

$$\mathbf{e} = I_1(\mathbf{p}_1) - I_2(\mathbf{p}_2) \quad 3-(34)$$

可构建最小二乘问题为：

$$\min_{\boldsymbol{\xi}} f(\boldsymbol{\xi}) = \|\mathbf{e}\|_2^2 = \sum_{i=1}^N \mathbf{e}_i^T \mathbf{e}_i. \quad 3-(35)$$

为了利用高斯牛顿法或者列文伯格——马库尔特法迭代求解目标函数 3-(35) 的最优解，则需要将该式进行展开得：

$$\begin{aligned} \mathbf{e}(\boldsymbol{\xi} \oplus \delta \boldsymbol{\xi}) &= I_1\left(\frac{1}{Z_1} \mathbf{K} \mathbf{P}\right) - I_2\left(\frac{1}{Z_2} \mathbf{K} \exp(\hat{\boldsymbol{\xi}}) \mathbf{P} + \mathbf{u}\right) \\ &\approx I_1\left(\frac{1}{Z_1} \mathbf{K} \mathbf{P}\right) - I_2\left(\frac{1}{Z_2} \mathbf{K} \exp(\hat{\boldsymbol{\xi}}) \mathbf{P}\right) - \frac{\partial I_2}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \mathbf{q}} \frac{\partial \mathbf{q}}{\partial \delta \boldsymbol{\xi}} \delta \boldsymbol{\xi} \\ &= \mathbf{e}(\boldsymbol{\xi}) - \frac{\partial I_2}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \mathbf{q}} \frac{\partial \mathbf{q}}{\partial \delta \boldsymbol{\xi}} \delta \boldsymbol{\xi} \end{aligned} \quad 3-(36)$$

其中， \mathbf{q} 与 \mathbf{u} 分别为扰动分量在相机 2 下的相机坐标与像素坐标：

$$\begin{cases} \mathbf{q} = \delta \hat{\boldsymbol{\zeta}}^{\wedge} \exp(\hat{\boldsymbol{\zeta}}^{\wedge}) \mathbf{P} \\ \mathbf{u} = \frac{1}{Z_2} \mathbf{K} \mathbf{q} \end{cases} \quad 3-(37)$$

式 3-(36)最后为链式法则，各项偏导皆好求，得雅克比为：

$$\mathbf{J} = -\frac{\partial \mathbf{I}_2}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \delta \hat{\boldsymbol{\zeta}}} \quad 3-(38)$$

然后可以利用非线性优化方法求出其最小二乘解， \mathbf{R} 与 \mathbf{t} 。

3.2.4 场景点位置估计

(1) 单目场景点位置估计

已知了两幅图像的相机位姿之后，则可以利用三角测量原理估算出该空间点到相机坐标系原点的深度。

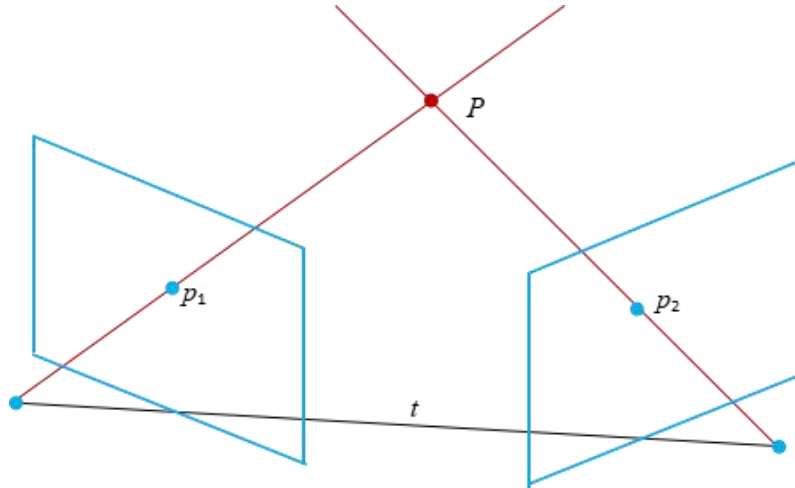


图 3.2 三角测量

如图 3.2 所示，空间点 \mathbf{P} 在两幅图像的投影为 \mathbf{p}_1 、 \mathbf{p}_2 ，相机 2 到相机 1 的旋转矩阵为 \mathbf{R}_{21} ，平移矩阵为 \mathbf{t}_{21} ，设 \mathbf{x}_1 与 \mathbf{x}_2 分别为 \mathbf{P} 在相机 1 与相机 2 的归一化坐标，则有：

$$s_1 \mathbf{x}_1 = s_2 \mathbf{R}_{12} \mathbf{x}_2 + \mathbf{t}_{12} \quad 3-(39)$$

由于 \mathbf{R}_{21} 与 \mathbf{t}_{21} 为已知，且上式可化为：

$$s_2 \mathbf{x}_1^T \mathbf{R}_{12} \mathbf{x}_2 + \mathbf{x}_1^T \mathbf{t}_{12} = 0 \quad 3-(40)$$

由于受到噪声影响，该式一般不为 0，因此可以解其最小二乘解，其为一个超静定非线性最小二乘问题，比较好解，带入公式得：

$$s_2 = -\left[(\mathbf{x}_1^T \mathbf{R}_{12} \mathbf{x}_2)^T \mathbf{x}_1^T \mathbf{R}_{12} \mathbf{x}_2 \right]^{-1} (\mathbf{x}_1^T \mathbf{R}_{12} \mathbf{x}_2) \mathbf{x}_1^T \mathbf{t}_{12} \quad 3-(41)$$

由公式 3-(41)得出 s_2 , s_1 即可较易求得。 \mathbf{P} 点在相机 1 坐标系下的坐标为 $s_1 \mathbf{x}_1$ ，转换为世界坐标系下的坐标为：

$$\mathbf{P}_{wc} = \mathbf{T}_{wc} \mathbf{P}_c = s_1 \mathbf{T}_{wc} \mathbf{x}_1 \quad 3-(42)$$

(2) RGB-D 场景点位置估计

经过标定之后的 RGB-D 相机，其每次获得的图像像素点与其深度都可以理解为从彩色图像相机光心到空间点的距离，设一幅图像某点像素为 (u, v) ，该点到空间点 \mathbf{P} 距离为 d ，相机内参为 \mathbf{K} ，则有：

$$d \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{K} \mathbf{P} \quad 3-(43)$$

易得 \mathbf{P} 点的空间坐标为：

$$\mathbf{P} = d \mathbf{K}^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \quad 3-(44)$$

设相机坐标系到世界坐标系的变换矩阵为 \mathbf{T}_{wc} ，可得 \mathbf{P} 点在世界坐标系中的坐标为：

$$\mathbf{P}_w = \mathbf{T}_{wc} \mathbf{P} \quad 3-(45)$$

3.3 SLAM 后端优化

SLAM 后端优化主要有基于滤波器优化的方法和基于非线性优化的方法。考虑分析整个地图中的位姿估计与所有空间点的位置估计，并对所有估计进行非线性优化，从而使相机位姿与空间点位置以最优的方式满足所有有效的观测方程模型。

3.3.1 图优化简介

图优化理论是图论与优化方法的一个结合，即用图论的方式来描述一个优化问题，主要用于非线性优化问题中，在图优化中，常用的优化库为 G2O 图优化库，该库是由 R. Kümmerle 在 2010 年提出主要用于解决非线性优化问题的^[41]。在 G2O 库中，如求目标函数为最小二乘的最优化问题时，分别以图论中的顶点（Vertex）和边（Edge）来表示目标函数待优化的变量和误差项，根据优化变量的数量，可以构建一元边、二元边等图优化问题，使用该库时，需要继承其基类 BaseVertex 与 BaseEdge，然后定义好待优化变量以及误差项，带入 G2O 构造问题进行求解。

3.3.2 Bundle Adjustment

Bundle Adjustment(BA)，即光束平差法^[54]，其原理是根据一幅图像中各个像素对应的空间点进行的捆集调整，即经过各种位姿估计方法得到的相机位姿并经过三角化等方法得到的空间点的深度信息后，再进行反投影得到新的像素，新像素与实际得到的像素做差，然后利用非线性优化方法使得该差最小的过程。

设第 i 个相机对应的相机外参为 \mathbf{R}_i , \mathbf{t}_i ，其对应的李代数为 ξ_i 空间三维点为 \mathbf{p}_i ，则可以根据观测模型得到观测误差为：

$$\mathbf{e}_i = \mathbf{z}_i - h(\xi_i, \mathbf{p}_i) \quad 3-(46)$$

把所有的观测量误差考虑进来，构建一个全局的关于优化变量的 ξ_i 与 \mathbf{p}_i 的代价函数 (Cost Function) 为：

$$\frac{1}{2} \sum_{i=1}^m \sum_{j=1}^n \|\mathbf{e}_i\|_2^2 \quad 3-(47)$$

现在的目的则是根据这个全局的代价函数求出整体的最优位姿估计与空间点的位置估计，为了求解该方程，可以利用高斯牛顿法与列文伯格——马库尔特方法进行迭代求解。此时，记所有的优化变量为：

$$\mathbf{x} = (\xi_1, \dots, \xi_m, \mathbf{p}_1, \dots, \mathbf{p}_n)^T \quad 3-(48)$$

则目标函数可记为：

$$\frac{1}{2} \sum_{i=1}^m \sum_{j=1}^n \|e_{ij} + J_{ij} \Delta x_{ij}\|_2^2 \quad 3-(49)$$

其中, J_{ij} 是整个误差项对整体变量的雅克比矩阵。由于第 i 幅图像观测到的第 j 个空间点只有一个投影方程, 因此在该优化变量求偏导数时, 雅克比 J_{ij} 中有大量的导数块为 0 , 如下式所示:

$$J_{ij} = (0_{2 \times 6}, \dots, 0_{2 \times 6}, \frac{\partial e_{ij}}{\partial \xi_i}, 0_{2 \times 6}, \dots, 0_{2 \times 3}, \dots, 0_{2 \times 3}, \frac{\partial e_{ij}}{\partial p_j}, \dots, 0_{2 \times 3}) \quad 3-(50)$$

得到了 J_{ij} 之后, 则可以利用高斯牛顿法或者列文伯格——马夸尔特方法得到系数矩阵 H :

$$H = \sum_{i,j} J_{ij}^T J_{ij} \quad 3-(51)$$

之后便可以利用增量线性方程:

$$H \Delta x = g \quad 3-(52)$$

进行求解。由于 H 矩阵的维度极高, 一般有几十万个维度, 因此直接求解该方程比较困难, 而考虑到 J_{ij} 的稀疏性, 因而得到的 H 矩阵也是有一定规律和稀疏性的, 常用的方法为 Schur 消元, 即 Marginalization(边缘化)^[55]。

3.4 SLAM 建图

SLAM 所得到的地图可以建模成稀疏地图与稠密地图, 该地图即为前面空间点 p 的集合。地图常用的用途为定位、导航、避障、三维重建以及交互等。定位功能通常用于机器人、无人驾驶、无人机等, 一般来说稀疏地图即可完成较好的定位; 导航与避障需要稠密地图来作为基础, 因为导航与避障功能需要知道地图中哪些地方可通过、哪些地方是障碍需要绕过、哪些障碍可以根据机器人的性能进行绕过; 三维重建一般用于完成对特定目标进行详细重建, 以尽量达到原三维物体的原貌, 这种情况绝大部分为稠密地图; 交互功能主要用于虚拟购物、AR、以及未来我相信必然会存在的全息 AR 投影聊天, 这种地图作用的实现需要定位也需要较为完整的 3D 模型, 因此一般情况也是稠密地图。

3.5 SLAM 回环检测

在 SLAM 系统中，由于各种因素如相机模型非最佳、标定误差、误匹配等问题的存在导致估计得到的位姿不够准确，而在估计当前帧的位姿时，又依赖于上一帧的位姿与空间点，这样的估计本身就是建立在前面很多图像位姿的不确定性上的，因此误差会越来越大，漂移也会越来越远。为了解决这个问题，通常根据实际应用情况来决定用何种方法来纠偏，如果用于固定场景，例如扫地机器人、服务机器人等通常就在固定的室内进行移动，则可以先让机器人在该场景中“走一圈”，这种情况就会存在“回环”，而我们可以巧妙的利用该回环来进行位姿重估计，即可减小全局的误差。回环检测的核心问题就是让机器人知道什么时候出现了回环，常用的回环检测算法利用机器学习中非监督学习的方法来进行检测，比如先建立一个词袋模型以及训练出一个字典，然后按照一定的方式选择一些特定的关键帧来进行相似度的判断，相似度较高者，判断为回环^[56]。

3.5.1 词袋模型与字典

一个字典有很多单词，假设一个字典中有 n 个单词，一个词袋中有 n 个单词中的 m 个，则词袋可以用一个 n 维向量来表示，其中，单词若存在就在该单词在 n 维向量中相应的位置记为 1，反之记为 0。SLAM 中所用的字典是经过训练而得，例如可以通过特征点来进行训练，先将图片进行特征提取，然后经过机器学习中的聚类方法将这些特征进行聚类，常用聚类方法有 K-means、K-means++ 等^[57,58]。我们用 K-means 算法来进行聚类，每层聚类的时候将这些特征聚成 K 类，然后对每一个节点再次聚成 K 类，并进行深度为 d 的聚类。在进行 d 层聚类之后就得到了叶子结点，所有的叶子结点就是所有的单词，而这些单词就构成了字典。当在进行单词查找时，只需与聚类中心进行比较，选取距离最小的作为该单词的当前聚类类型，只需进行 d 次聚类即可得到该单词最相近的单词类型。查找的目的是构建一个词袋，确定图像中是否含有该单词，并用一个特征向量来进行表示。

3.5.2 回环判断

设一系列图像中的单词总数为 N ，则对于一幅图像 A ，可以将该图像的的词袋定义为 v_A ：

$$v_A = \{(w_1, \eta_1), (w_2, \eta_2), \dots, (w_N, \eta_N)\} \quad 3-(53)$$

其中， w_i 为单词，其下标序列中单词的位置， η_i 为该单词的权重。单词的位置即为词典中叶子节点的位置，其权重利用了文献检索中的译频率——逆文档频率算法

(Term Frequency-Inverse Document Frequency, TF-IDF)^[59], 某个单词 w_i 权值 η_i 可定义为:

$$\eta_i = TF_i \times IDF_i \quad 3-(54)$$

单词的重要性程度是随着它在一幅图像中出现的次数成正比增加, 而随着它在所有图像中出现的次数成反比下降。若设一幅图像 A 中单词 w_i 的数量为 n_{Ai} , 该幅图像中的单词总数为 N_{Ai} , 则有

$$TF_i = \frac{n_{Ai}}{N_{Ai}} \quad 3-(55)$$

若所有图像中单词总数为 N , 该单词 w_i 的在所有图像中的数量为 N_i , 则有:

$$IDF_i = \frac{N_i}{N} \quad 3-(56)$$

同理可记图像 B 的词袋为 \mathbf{v}_B , 则判断 SLAM 系统中是否存在回环, 则可以设定一个阈值, 若两幅图像的相似度超过这个阈值, 则判断这两幅图像为回环, 否则, 不是回环^[60]。具体的计算公式可用如 3-(58)式定义的 L_1 范数进行计算^[61]:

$$s(\mathbf{v}_A - \mathbf{v}_B) = 2 \sum_{i=1}^N \left(\left| \mathbf{v}_{A_i} \right| + \left| \mathbf{v}_{B_i} \right| - \left(\left| \mathbf{v}_{A_i} \right| - \left| \mathbf{v}_{B_i} \right| \right) \right) \quad 3-(57)$$

3.6 本章小结

本章主要分析了视觉 SLAM 的基本架构, 包括传感器数据的读取、前端视觉里程计的设计、后端优化与回环检测等。前端可分为基于特征点法的前端与基于直接法的前端, 都进行了数学的推导, 后端主要是用高斯牛顿法或者列文伯格——马夸尔特算法来进行优化, 建图部分根据应用可建立成不一样的地图, 而回环检测分析了词袋模型以及词典的建立。

第四章 系统设计实现与仿真分析

4.1 数据采集与数据集介绍

实验的数据集采用德国慕尼黑工业大学的 TUM 的公开数据集^[62]，其用于视觉 SLAM 的公开数据集包含了单目视觉里程计数据集、单目全向镜头 SLAM 数据集、RGB-D SLAM 数据集等。其单目视觉里程计数据集包含有 50 个序列，每个序列都提供了该序列的帧数、视频时间、帧率、最短曝光时间以及最长曝光时间，并且每个序列的开始位置与结束位置相互重合，这样在检测算法的精度时，就不需要利用真实相机轨迹来进行测评，只需要初始位置与结束位置的偏差即可，这样就能计算出来最大的漂移为多少。

另一部分数据则是利用无人机拍摄的视频和普通相机拍摄的视频进行 SLAM 系统的实验分析。

4.2 标定实验

Matlab 的 Camera Calibrator 模块的标定精度比 OpenCV 高，因此，在后续的实验过程中都用 Matlab 来进行标定，下面分别对需要采集数据的无人机相机与普通摄像头进行了标定，并对其重投影误差、径向畸变、切向畸变等进行了分析，选取重投影误差最小的相机模型作为最终的标定结果。而 TUM 数据集已经标定好，因此不需要进行额外的标定。

4.2.1 普通摄像头标定

所用摄像头为 Logitech 公司的普通摄像头，其好处是便于移动，且可无死角旋转，用该摄像头采集了 16 张各个方位标定板的图像，如图 4.1 所示。

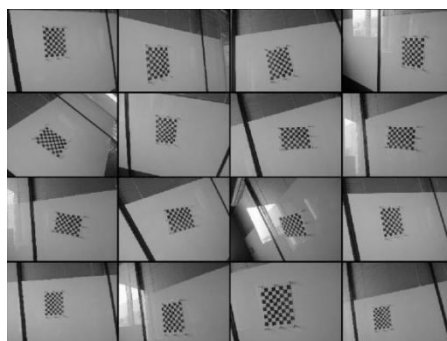


图 4.1 普通摄像头标定图采集

所用标定板为边长 25mm 的方格，以 9 个角点的边为 x 轴，以 7 个角点的边为

y 轴，然后分别采集各个旋转方向的标定图像进行标定。如图 4.2 所示，为部分标定图检测的角点以及坐标系。

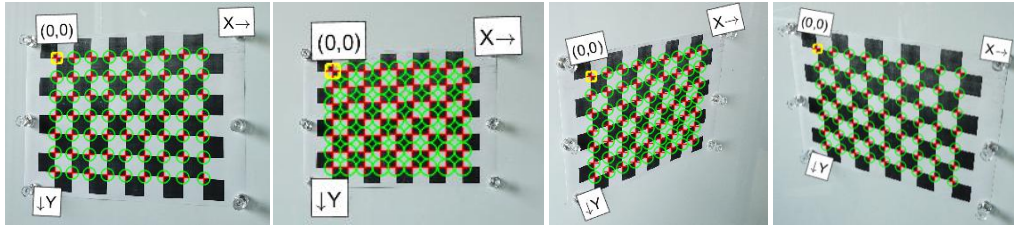


图 4.2 部分标定图角点检测

利用公式 2-(12)进行标定，可以分别以四种模式来进行标定，如图 4.3 所示，展示了相机相对于标定板的空间位置情况。

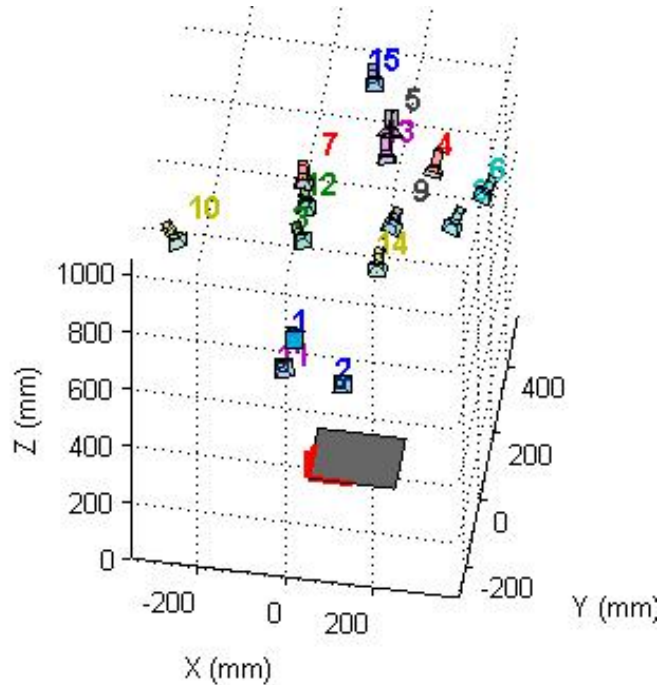


图 4.3 相机相对于标定板空间位置分布

如图 4.4 所示，为切向畸变模型的标定，且只标定径向畸变的前两个参数，即标定 2-(12)中的 k_1 、 k_2 、 p_1 与 p_2 ，该图表示各幅图像在标定出的内参与外参下，得到的重投影误差分布图，且 16 幅图像的平均重投影误差为 0.17 个像素，由于第 7 幅图像的重投影误差较大，因此在实际标定过程中可将该图片去掉以提高标定精度，引起该较大误差的原因有可能是光线影响，也有可能是拍摄等问题，在标定过程中应当避免在反光情况下进行标定。

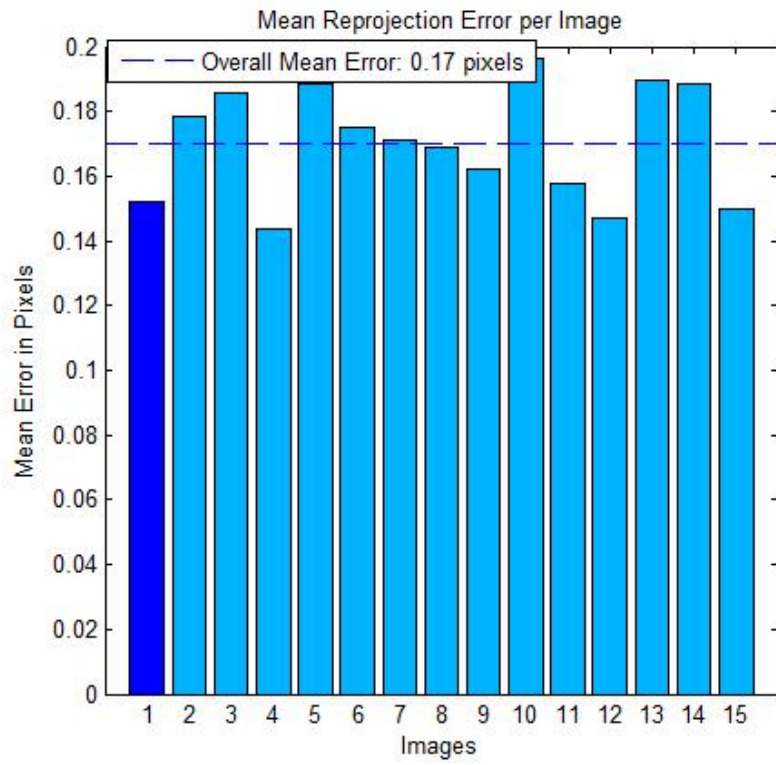


图 4.4 切向畸变模型标定平均重投影误差分布图

每一张图片的重投影误差如图 4.5 所示，可以看出，该误差分布可以用均值为 0 的高斯模型来进行模拟。

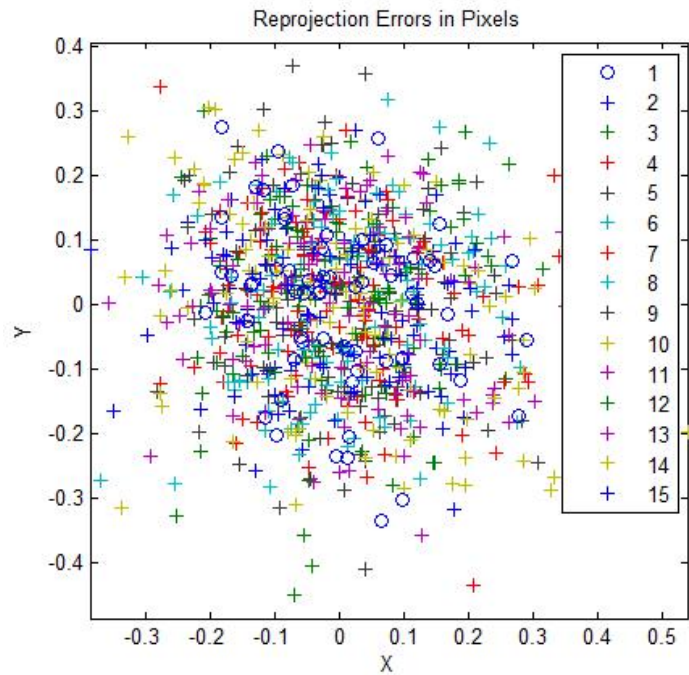


图 4.5 切向畸变模型标定重投影误差分布图

如表 4.1 所示, 利用四种模式分别来标定该摄像头, 即分别标定 2-(12)中的径向畸变参数与切向畸变参数, 取标定所得重投影误差较小者作为最终标定结果。

表 4.1 四种模式下相机标定内参与重投影误差

标定模式		k_1 、 k_2	k_1 、 k_2 、 k_3	k_1 、 k_2 、 p_1 、 p_2	k_1 、 k_2 、 k_3 、 p_1 、 p_2
径向畸变	k_1	0.0913	0.0901	0.0895	0.0797
	k_2	-0.1817	-0.1567	-0.1112	0.0741
	k_3	0	-0.1420	0	-0.9186
切向畸变	p_1	0	0	-0.0073	-0.0074
	p_2	0	0	-0.0042	0.0043
相机焦距	f_x	643.6025	643.6177	645.4208	645.6178
	f_y	643.3351	643.3474	645.1917	645.3611
相机主点	c_x	389.0151	389.0192	395.4208	395.5170
	c_y	316.3828	316.3796	305.8700	305.7142
平均重投影误差/像素		0.17027	0.17026	0.16889	0.16881

4.2.2 无人机摄像头标定

如表 4.2 所示, 为无人机四种标定模式下的内参与重投影误差, 其中, 标定了 k_1 、 k_2 、 k_3 、 p_1 、 p_2 的模式下所得平均重投影误差最小, 约为 0.507 个像素。

表 4.2 无人机四种模式标定内参与重投影误差

标定模式		k_1 、 k_2	k_1 、 k_2 、 k_3	k_1 、 k_2 、 p_1 、 p_2	k_1 、 k_2 、 k_3 、 p_1 、 p_2
径向畸变	k_1	0.1918	0.3979	0.2117	0.3907
	k_2	-0.5206	-3.3115	-0.6464	-3.2122
	k_3	0	7.2883	0	6.9199
切向畸变	p_1	0	0	-0.0090	-0.0014
	p_2	0	0	-0.0061	0.0022
相机焦距	f_x	3106.4	3086.1	3100.0	3085.1
	f_y	3095.3	3079.0	3089.8	3078.1
相机主点	c_x	1979.7	1978.2	1942.7	1990.0
	c_y	1136.4	1119.7	1079.6	1112.2
平均重投影误差/像素		0.6206	0.5091	0.6173	0.5070

4.3 前端设计实验

(1) 软件架构

SLAM 系统的执行需要一个合适的软件架构, 本系统用面向对象的设计思想来进行设计, 面向对象设计的主要特点是封装、继承、多态。根据 SLAM 系统前端的原理及最终的实现目的, 可以将其进行归类并分别进行实现。基于特征点法的前端主要是进行特征提取、特征匹配、位姿估计、三角测量、初始化、PnP 位姿优化、

建立局部地图等内容。特征提取(Feature Extraction)是从一幅图像帧中提取, 特征提取包括提取图像的关键点(Keypoint)与其描述子(Descriptor); 特征匹配(Feature Matching)是将两幅已提取特征及描述子的图像帧进行描述子的相似性的比对, 匹配程度越高的两个像素(Pixel)点视为来自同一个地图点(Map Point); 位姿估计(Pose Estimation)是通过对极几何的方法将两幅未知深度的图像进行位姿结算或者用 PnP 的方法进行计算; 三角测量(Triangulation)可以通过已经计算出来的两幅图像之间的位姿估计通过几何的方法算出空间点到两幅图像相机中心的深度(Depth)距离; 初始化(Initializing)是指单目 SLAM 中在开始时由于图像中没有深度信息而用对极几何的方法算法计算经过一定方式选择的具有鲁棒性的两帧之间的相机位姿, 然后三角化测量其深度信息, 并通过光束平差法(Bundle Adjustment)优化初始化位姿及三维点的过程; PnP 位姿优化(PnP Pose Optimization) 用迭代优化的方法进行优化, 例如用高斯牛顿法或者列文伯格——马库尔特算法进行迭代优化得到较好的位姿; 建立局部地图(Local Mapping)是指将三角测量得到的地图点全部转化为世界坐标存储起来, 并可以用该局部地图进行 PnP 的位姿计算。

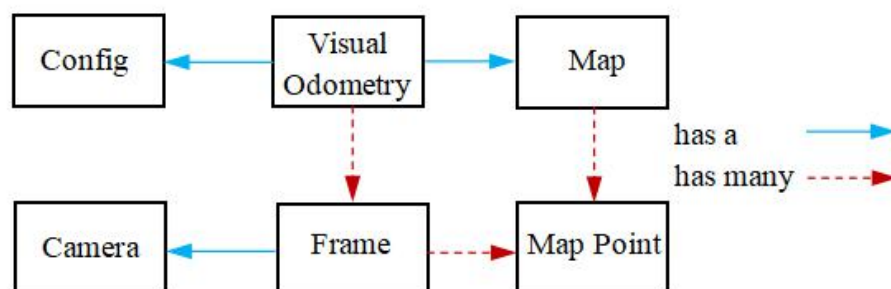


图 4.6 软件架构类关系图

根据上面 SLAM 的过程可设计系统用到的并可以抽象为类的有 Camera 类、Frame 类、Map 类、MapPoint 类、VisualOdometry 类等, Camera 类的属性有相机内参、尺度因子等, 方法有像素坐标系、相机坐标系与世界坐标系之间的相互转换等。Frame 类属性有图像帧的位姿、ID、时间戳、特征点、三维点、深度图、词袋(回环检测用)等, 方法有创建帧、深度查询、获取相机主点、判断空间点是否在图像视野范围内等; MapPoint 类属性有点在世界坐标系中的位置、法向量、ID、在图像帧中用特征提取算法被看到的帧数、用位姿估计算法投影回来仍在图像上的点的次数, 其方法有创建地图点等; Map 类用于对地图点的管理, 其属性有存储地图点的 ID 与对应地图点的哈希表, 存储关键帧 ID 与关键帧的哈希表等, 方法有插入关键帧、插入地图点、删除地图点等; VisualOdometry 类的是视觉里程计的主要类, 包含了视觉里程计的主要属性和方法, 属性主要有视觉里程计的状态、图像帧、地图变量、特征点、描述子、相机位姿以及视觉里程计运行需要的参数, 其主要方法有特征提

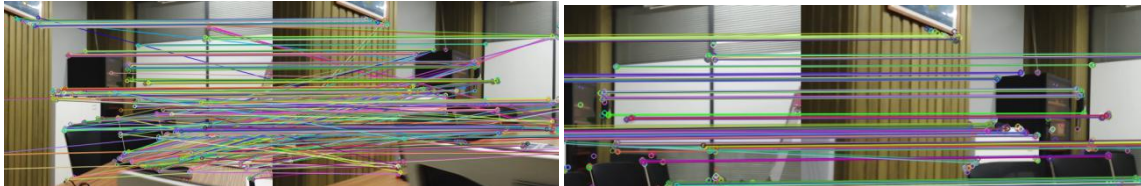
取、特征匹配、PnP 位姿估计、对极几何位姿估计、三角测量、添加关键帧等；Config 类主要是一些参数文件，常用于经常改参数的情况，以免重新进行编译浪费时间，主要实现读取参数文件中参数的功能。各类之间关系图如图 4.6 所示。

(2) 特征提取与匹配

取 TUM 数据集 rgbd_dataset_freiburg1_xyz 数据集进行试验，如图 4.7 所示，为特征提取与特征匹配情况，所用特征为 ORB 特征，图 4.7 (a) 为 ORB 特征提取，(b) 与 (c) 分别为特征匹配的原始匹配对与经过消除误匹配之后的匹配。



(a) 特征提取



(b) 原始特征匹配

(c) 消除误匹配

图 4.7 ORB 特征提取与特征匹配

(3) 对极几何位姿估计

表 4.3 对极约束求相机位姿

对极 约束 矩阵	t	R	H	E	$t^{\wedge}R$
值	$\begin{pmatrix} -0.39 \\ -0.30 \\ 0.87 \end{pmatrix}$	$\begin{pmatrix} 1 & 0.006 & -0.004 \\ -0.006 & 1 & -0.006 \\ 0.004 & 0.006 & 1 \end{pmatrix}$	$\begin{pmatrix} 1.06 & -0.02 & -13.9 \\ 0.01 & 1.01 & -4.62 \\ 8.56 & -3.02 & 1 \end{pmatrix}$	$\begin{pmatrix} 0.003 & -0.62 & -0.21 \\ 0.62 & 0.005 & 0.27 \\ 0.21 & -0.27 & 0 \end{pmatrix}$	$\begin{pmatrix} 0.004 & -0.87 & -0.30 \\ 0.87 & 0.01 & 0.38 \\ 0.30 & -0.38 & 0.001 \end{pmatrix}$

利用对极约束进行位姿估计，其对应像素的对极几何约束如表 4.3 所示。表中，为利用八点法进行的位姿估计，分解本质矩阵 E 得到旋转矩阵 R 与平移向量 t ，其单应性矩阵为 H ， $t^{\wedge}R$ 为进行结果验证，若接近 E ，则表明位姿估计准确。

为了表明所计算的结果满足极线约束，即满足公式 3-(11)，将所求得旋转矩阵 R 与平移向量 t 代入该式，如表 4.4 所示，其结果接近于零，因此位姿估计正确。

表 4.4 对极约束验证

匹配像素对	
1	0.00252812
2	-0.0016637
...	...
26	0.00080090
44	0.00016390

(4) PnP 位姿估计

单目进行初始化之后，就得到了前两帧关键帧之间的位姿与三维地图点，然后，可通过后续帧与当前帧得到的特征匹配利用 PnP 算法进行位姿估计，本实验用了 OpenCV 提供的 EPnP 算法来求解，得到的旋转矩阵与平移向量如表 4.5 所示。

表 4.5 PnP 位姿估计

位姿项	估计值
R	$\begin{pmatrix} 1 & 0.006 & -0.004 \\ -0.006 & 1 & -0.006 \\ 0.004 & 0.006 & 1 \end{pmatrix}$
t	$\begin{pmatrix} -0.39 \\ -0.30 \\ 0.87 \end{pmatrix}$

(5) 视觉里程计效果

运用 TUM 图像序列进行视觉里程计的测试，数据集中一共有 793 张图片，未经过局部位姿优化的视觉里程计会导致相机位姿与空间点估计不准确，特别是算得的空间点在进行反投影后会没有一个 PnP 内点，导致了算法的失败，运行了 12 帧之后，算法就会终止运行使得整个系统失败，其运行结果如表 4.6 所示，所计算得到的 PnP 内点个数都为 0 个，良好匹配对也从 253 个逐渐减少到 154 个，且其每一帧的视觉里程计的用时从约为 0.03 秒。

表 4.6 未优化视觉里程计运行参数

帧数	关键帧(个)	良好匹配对 (对)	PnP 内点 (个)	视觉里程计用时 (s)
1	0	0	0	0
2	0	253	0	0.136737
3	0	251	0	0.034344
...
10	0	146	0	0.032535
11	0	141	0	0.032055
12	0	154	0	0.031976

为了让系统更加鲁棒，而不是经过几帧就跳出循环，因此需要对相机位姿进行优化，其结果如表 4.7 部分局部 BA 优化之后的里程计运行参数表I所示，序列同样使用了 793 张图片进行测试，在每次估计视觉里程计之后对新估计的位姿进行重投影误差的位姿优化，优化后的视觉里程计能够持续运行下去，效果有明显提升。在整个视觉里程计中，如表中所示，特征提取、描述子计算与特征匹配占整个视觉里程计的计算都在 92%以上，因此，寻找图像中的对应点消耗了大量的时间，而这个在直接法中将有所改进。

表 4.7 部分局部 BA 优化之后的里程计运行参数表 I

帧数	良好匹配对 (对)	PnP 内点 (个)	特征点提取时间 (s)	描述子耗时 (s)	特征匹配时间 (s)	视觉里程计用时 (s)	特征计算 / 里程计 (%)
1	0	0	0	0	0	0	0
2	253	248	0.009143	0.00872	0.008865	0.028635	93.34
3	251	244	0.008458	0.00693	0.00761	0.024441	94.083
4	232	226	0.007126	0.00557	0.006936	0.021046	93.27
5	226	222	0.006273	0.00491	0.00633	0.018717	93.55
6	209	205	0.006794	0.00607	0.00629	0.020481	93.50
7	210	209	0.006765	0.00557	0.006531	0.020039	94.12
8	160	158	0.005774	0.00509	0.006065	0.017949	94.31
9	148	141	0.006157	0.00533	0.006182	0.018734	94.29
10	146	143	0.006386	0.00540	0.006029	0.018696	95.29
11	141	135	0.008095	0.00748	0.008999	0.026553	92.54
...

能将该里程计延续下去的另一个原因就是建立了一个地图，该地图存储的是以世界坐标系为基础坐标的三维点，其优点是，当地图中有个别帧出现误差或者追踪失败时，由于地图仍然存在，可以将后续帧与地图进行 PnP 的位姿估算，从而使得里程计能继续运行。

如表 4.8 所示，为部分局部 BA 优化与采用世界坐标来存储地图策略之后的里程计运行参数表II，其旋转矩阵 R 与平移向量 t 都是经过优化之后的，且为了保证

存储空间不至于太大，可以给地图数量设定一个阈值，但又不至于太少，以便保证地图在长时间跟踪丢失的情况下仍然有地图点与新来的某良好关键帧进行 PnP 的计算。地图点数量的设置需要考虑特征提取的数量与特征匹配对数量的多少来相应的进行设置，若特征提取数量在 500 个左右，考虑到图像间的最小重叠率在 30% 的理想情况，并假设特征分布均匀，则有特征匹配 150 对。考虑特征匹配存在误匹配的情况，假设特征匹配正确率为 80%，则可有 120 对正确匹配对，此时，我们可以用 RANSAC 或者选匹配最近的 8 个点对进行八点法的位姿估计^[63]。

表 4.8 部分局部 BA 优化之后的里程计运行参数表 II

帧数	旋转矩阵 R	平移向量 t	地图点数量
1	0	0	0
2	$\begin{pmatrix} 0.999972 & -0.00273623 & 0.00697135 \\ 0.0027608 & 0.99999 & -0.00351744 \\ -0.00696166 & 0.00353659 & 0.99997 \end{pmatrix}$	$\begin{pmatrix} 0.00124764 \\ -0.00905056 \\ -0.0131247 \end{pmatrix}$	424
3	$\begin{pmatrix} 0.99983 & -0.00125806 & 0.018408 \\ 0.00177116 & 0.99961 & -0.0278844 \\ -0.0183657 & 0.0279122 & 0.999442 \end{pmatrix}$	$\begin{pmatrix} 0.00256082 \\ -0.00650978 \\ -0.0392371 \end{pmatrix}$	424
...
10	$\begin{pmatrix} 0.998914 & 0.000139659 & 0.0465917 \\ 0.00354327 & 0.996872 & -0.0789548 \\ -0.046457 & 0.0790342 & 0.995789 \end{pmatrix}$	$\begin{pmatrix} -0.0003817 \\ -0.0110577 \\ -0.129542 \end{pmatrix}$	410
11	$\begin{pmatrix} 0.999151 & -0.00680301 & 0.0406424 \\ 0.0105661 & 0.9956 & -0.0931056 \\ -0.0398302 & 0.093456 & 0.994826 \end{pmatrix}$	$\begin{pmatrix} 0.0010861 \\ -0.00379868 \\ -0.157752 \end{pmatrix}$	372
...

如图 4.8 所示，为表 4.8 中前 10 帧的轨迹图，每个点代表的是该点在世界坐标系中的坐标，可以在图中看出，我们的相机进行了螺旋上升，与我们的实际实验情况相符，证明了程序的可行性，不过由于仍然存在漂移的问题，在长时间情况下难免由于漂移过大而跟踪丢失，因此仍需要后续的全局优化。

图中的每个点的三维坐标的坐标系都是以第一帧作为参考坐标系进行计算的，也就是第一帧所在的相机坐标系为世界坐标系，后续帧所观察到的空间点需要进过相机坐标系到世界坐标系的变换以得到该点在世界坐标系中的表示。其中，各坐标轴的方向以第一帧相机坐标系的方向为基准，且以两帧之间的平移向量为参考单位来进行后续帧之间平移向量的长度的计算。

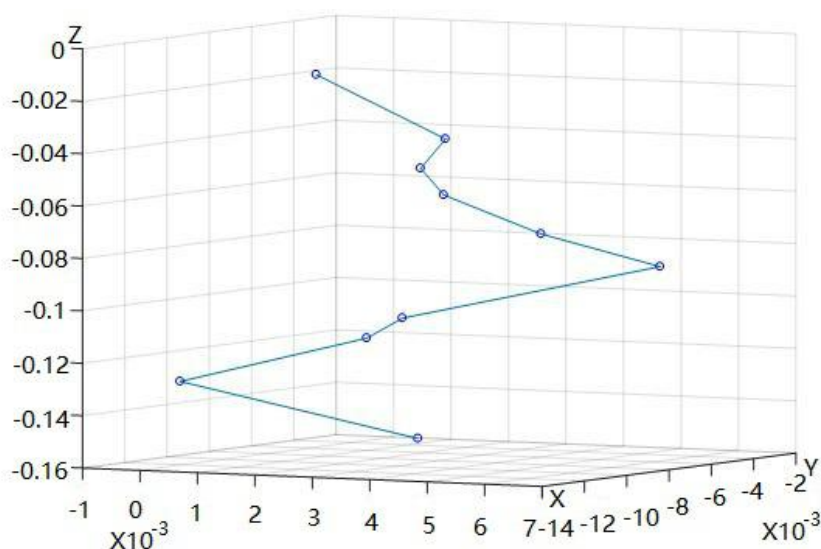


图 4.8 优化后的视觉里程计运行轨迹图

4.4 后端优化实验

一般的视觉 SLAM 系统是仅有观测方程构成的，如表 4.9 所示，取书中提供的观测值数据来进行后端优化实验，一共有 16 帧图片，22106 个地图点，83718 个观测值，其用于迭代的 H 矩阵将是 22122 维度的一个方阵，为了解得迭代式子的增量方程，则需要用到 H 矩阵的稀疏性，用 Schur 消元的方法来加速计算^[64]。

表 4.9 部分图像帧对地图点的观测值

图像帧 ID	地图点 ID	像素坐标 u	像素坐标 v
0	0	3.859900e+02	3.871200e+02
1	0	3.844000e+01	4.921200e+02
2	0	6.679200e+02	1.231100e+02
7	0	5.991800e+02	4.079300e+02
12	0	7.204300e+02	3.143400e+02
13	0	1.151300e+02	5.548999e+01
0	1	3.838800e+02	1.529999e+01
1	1	5.597500e+02	1.061500e+02
...

在实际编程时也通常利用前人编写好的优化库来进行编程，常用的优化库有 G2O 与 Google 公司的优化库 Ceres 库^[65]，将上述问题构建成 Bundle Adjustment 问题，则 G2O 与 Ceres 都采用列文伯格——马夸尔特算法进行迭代优化，其运行结果分别如表 4.10 与表 4.11 所示。

表 4.10 G2O 库进行 Bundle Adjustment 的优化结果

迭代次数	拉格朗日乘子 λ	单次迭代时间 (s)	累计时间 (s)
0	9370.114177	0.302989	0.302989
1	3123.371392	0.258655	0.561644
2	0.561644	0.187948	0.937621
\vdots	\vdots	\vdots	\vdots
17	0.001948	0.185674	3.54755
18	0.000649	0.185651	3.7332
19	0.000216	0.186157	3.91935

由表 4.10 与表 4.11 可知, G2O 库的迭代速度比 Ceres 库要慢将近一倍多, 用 G2O 库进行的列文伯格——马夸尔特算法需要迭代 19 次才能达到目标函数的最小值, 其拉格朗日乘子 λ 为 0.000216, 几乎可以忽略不计。而用 Ceres 库进行的列文伯格——马夸尔特算法迭代次数只需要进行 8 次就能得到目标函数的最小值, 且目标函数的误差也接近于 0, 为 1.74×10^{-09} , 步长约为 1.98×10^{-04} , 也可满足迭代停止条件。因此两种算法都是可靠的, 但是本次实验中, 表明 Ceres 库比 G2O 库有更快的优化速率。

表 4.11 Ceres 库进行 Bundle Adjustment 的优化结果

迭代次数	目标函数误差	步长 λ	单次迭代时间(s)	累计时间
0	0.00e+00	0.00e+00	1.58e-01	2.70e-01
1	3.99e+06	2.40e+03	1.76e-01	4.46e-01
2	1.47e+05	1.01e+03	1.52e-01	5.98e-01
3	3.23e+04	2.64e+02	1.53e-01	7.51e-01
4	5.58e+02	8.66e+01	1.52e-01	9.03e-01
5	4.66e+00	1.02e+01	1.54e-01	1.06e+00
6	1.85e-03	4.60e-01	1.52e-01	1.21e+00
7	1.01e-06	8.99e-03	1.52e-01	1.36e+00
8	1.74e-09	1.98e-04	1.52e-01	1.51e+00

为了让算法收敛更快, 如图 4.9 所示, 分别利用列文伯格——马夸尔特 (LM) 算法与 Dogleg 算法进行 Bundle Adjustment 计算与分析, 发现 Dogleg 算法比 LM 算法收敛速度快, 这在有大量图像数据需要处理的 SLAM 系统中具有极大意义。在

G2O 图优化程序框架下，使用 LM 算法迭代 19 次耗时约 4.6 秒，而使用 Dogleg 算法耗时约 3.9 秒，节省 0.7 秒。在 Ceres 框架下，LM 算法耗时约 1.4 秒，Dogleg 算法耗时约 1.5 秒，节省 0.1 秒。所节省的时间在计算资源异常宝贵的 CPU 上是非常重要的，文中所用 Dogleg 算法在速度上具有一定的优势，具有一定的实用价值。



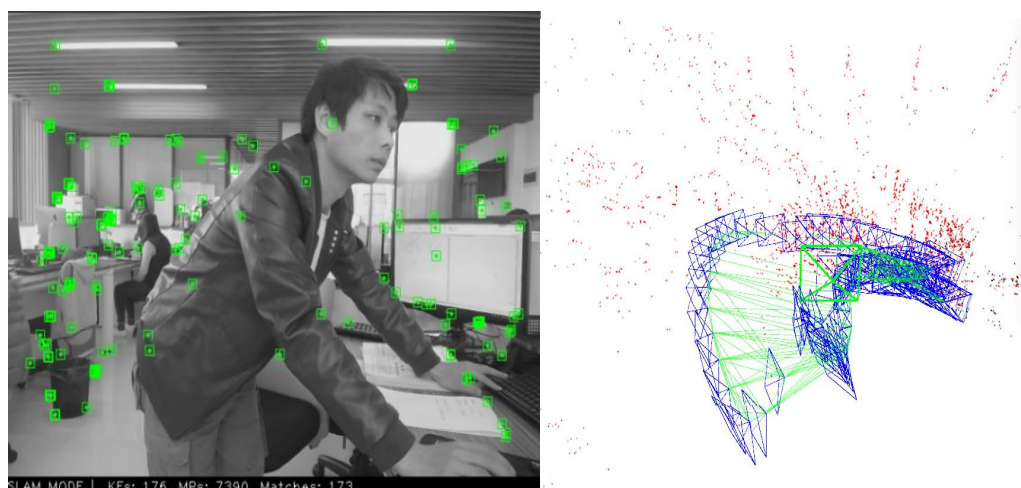
图 4.9 LM 算法与 Dogleg 算法收敛速度对比

4.5 地图构建实验

SLAM 的地图形式根据地图点的稠密程度分为稠密地图与稀疏地图。稠密地图可用 SFM 稠密重建的方法来进行稠密重建，不过代价是计算量指数级的提升导致无法在 CPU 上实时运行，甚至无法在一般的 GPU 上实时地建出稠密地图；稠密地图也可以通过直接法来获取，这样直接将地图点与位姿点当成优化变量通过非线性优化的方法来减小光度误差，这种方法的好处是可以在设备不是要求很高的情况下进行实时稠密重建，缺点是不能在快速运动以及光照变化明显的地方进行稠密重建。稀疏地图通常用提取特征的方法来获取，其主要用于定位功能，优点是可以进行实时的计算，可以实时跟踪相机运动。根据实际应用情况，还可以将地图建立成拓扑地图、占据栅格地图等形式。

4.5.1 稀疏重建

稀疏重建的地图可以用特征点法计算而来，也可以用直接法计算而来，稀疏地图可以用于快速定位也可以用于对建好后的地图进行重定位，对于固定场景的情况下也可以用于导航。如图 4.10 所示，为在室内进行的特征点法稀疏地图 SLAM 的运行结果图，图 4.10（a）中为 ORB 特征点跟踪，（b）为关键帧轨迹图与局部稀疏地图，其轨迹方向与实际运动情况相符。



（a）特征跟踪 （b）关键帧与稀疏地图

图 4.10 ORB-SLAM 稀疏地图与定位运行效果图

4.5.2 稠密重建

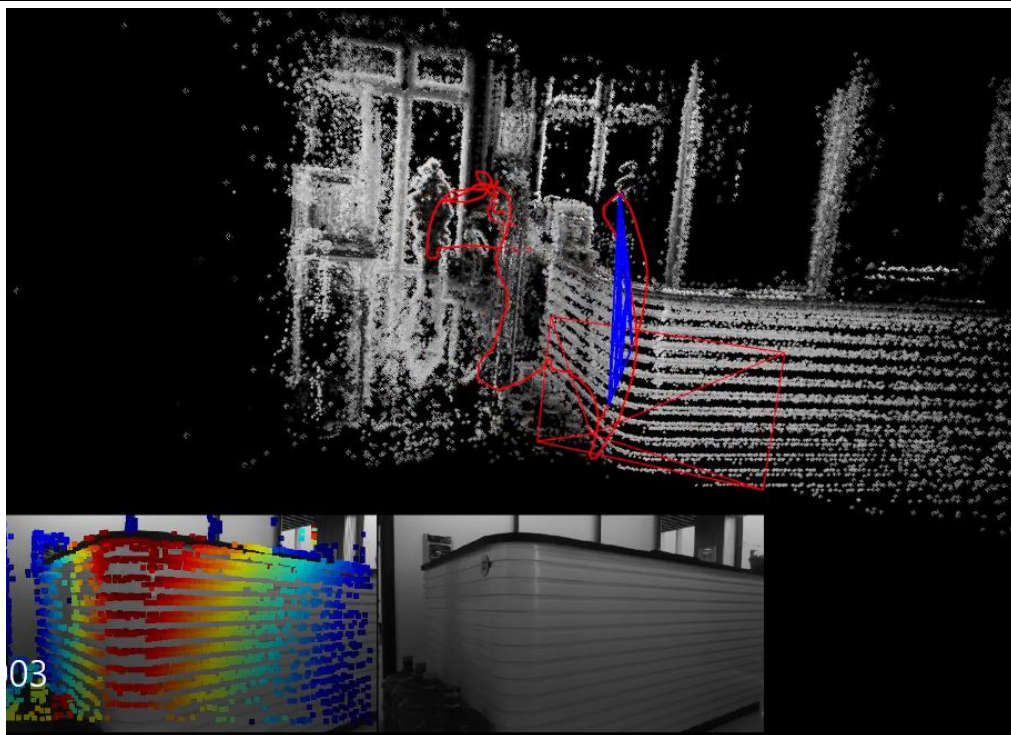
SLAM 地图的稠密形式主要用于重建、人机交互、避障等功能，实现的主要形式为直接法。

如图 4.11 所示，为手持无人机拍摄的视频序列进行的 DSO 算法的直接法半稠密重建^[19]。其中：

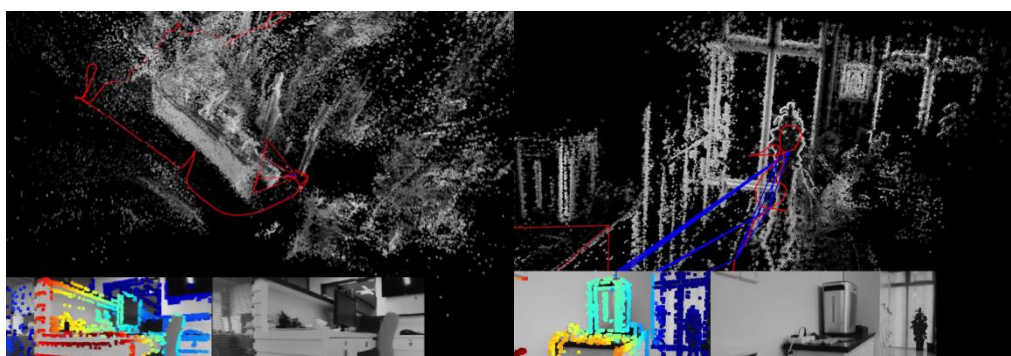
- （a）为前台、绿萝走道等，效果较好；
- （b）为前台部分，前台轮廓纹理比较清晰；
- （c）为窗边部分，可以看出，绿萝轮廓也比较清晰。

（a）、（b）与（c）三幅图的左下角的左图为计算图像深度信息，并进行实时更新，右图则为原图的灰度图。

从图中可知，建图较为清晰，且能够观察出无人机周围环境的半稠密轮廓三维模型，且定位精度较高，后端优化也较为准确，能够进行无人机的避障与导航需求。虽然有这些优点，但是在长时间大规模的情景下，地图构建仍然会出现漂移的问题，并且导致整个地图在后期过程中无法进行校正，导致失效。



(a) 绿萝走道



(b) 前台

(c) 窗边

图 4.11 DSO 稠密重建效果图

4.5.3 占据栅格地图

占据栅格地图主要用八叉树地图的形式表示，如图 4.12 所示，左右两图都是同一个点云地图进行八叉树划分后得到的地图，分辨率为 5cm^3 ，左右图是该点云的不同视角的情况，其中颜色是表征地图点的远近信息的，也可以表示出高度信息。占据栅格地图主要用于导航、避障与路径规划，用八叉树图表示的优点是可以节省大量的存储空间，因为只要是在一定空间内的点云可以聚成一个栅格立方体，这样就可以通过设置栅格的分辨率来控制八叉树地图的内存占据情况。

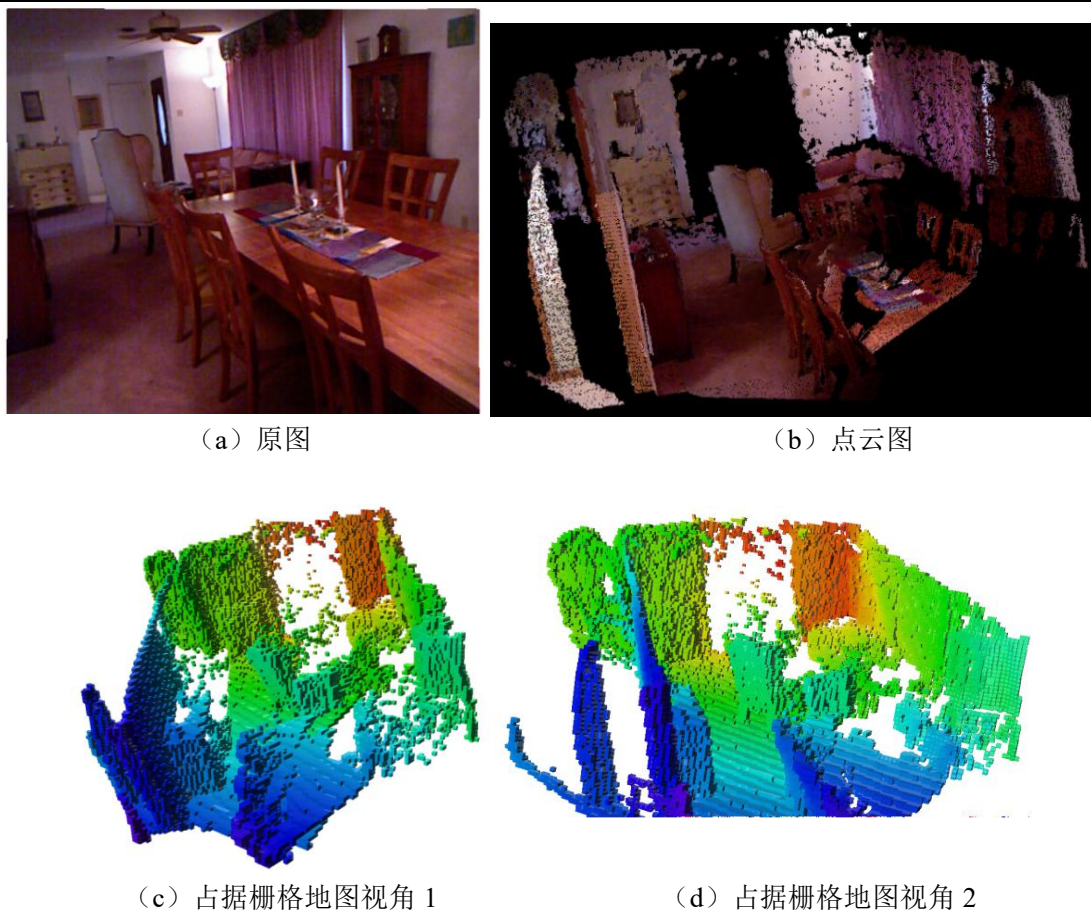


图 4.12 八叉树地图

4.6 回环检测实验

4.6.1 词典训练

表 4.13 词典训练结果摘选

单词 ID	叶子节点 ID	父节点 ID	TF-IDF 值	描述子
280	362	317	2.3025850929940459e+00	0 32 23 94 107 ...
600	1259	1247	2.3025850929940459e+00	0 32 223 221 37 ...
1000	692	2	0	0 32 1 172 90 ...
2000	2494	2479	2.3025850929940459e+00	0 32 36 31 230 ...
3000	3729	3698	2.3025850929940459e+00	0 32 252 85 83 ...
4000	4964	4903	2.3025850929940459e+00	0 32 180 79 238 ...

利用 ORB 特征进行字典的训练, 取 10 张 TUM 数据集的图片进行为 10、深度为 5 的树的建立, 这样可以最多聚类成 100000 个单词的词袋, 并利用开源 BoW 库来编程训练, 这 10 张图片中生成了 4983 个单词, 取权重形式为 TF-IDF 的值作为单词的权重值, 如表 4.12 所示为取部分单词进行展示。

4.6.2 回环判断

为了得到更好的效果，回环检测利用训练的更大的词典来进行图像词典的评分计算，新的词典有 2900 幅图片，得到的词典有 99566 个单词，图像间的相似性评分利用式 3.51 进行计算，取每一帧图像为检测目标，得到结果如图 4.13 所示，图中横坐标表示每一幅图像与其他图像进行相似性评分之后得到的评分最高的三幅图片，纵坐标为该图片与柱状图上端所标记数字的图片 ID 的相似度值的大小。从图中可知第 10 帧图像与第 1 帧图像相似度最高，相似度约为 0.032，故该帧与第一帧构成回环。

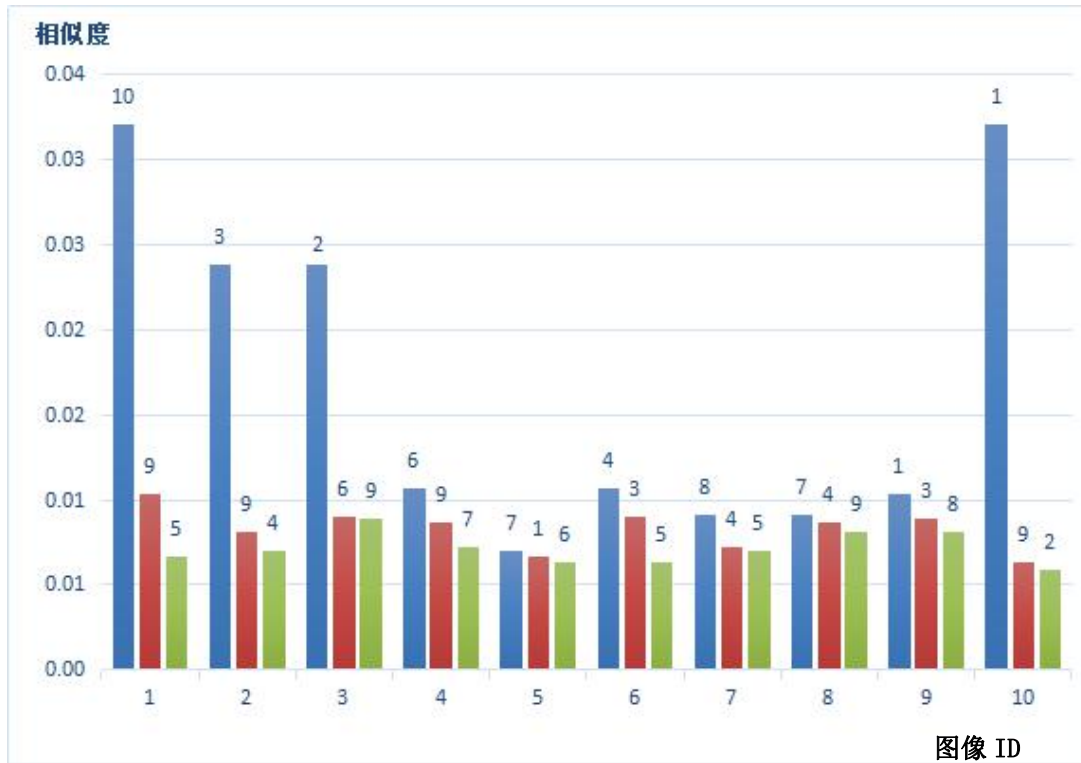


图 4.13 图像相似度匹配图

4.7 本章小结

本章主要对第二章与第三章的理论进行实现，分别标定了无人机相机与 Logitech 公司的普通相机，并用特征点法进行了前端设计，然后分别用 Ceres 库与 G2O 库的列文伯格——马夸尔特算法进行了后端优化，并用稀疏地图、稠密地图与栅格地图构建了不同形式的地图，最后在回环检测用 10 张图片进行了验证。

第五章 结论及展望

SLAM 具有广阔的研究以及应用前景,尤其是单目 SLAM,其有望应用在小型移动设备上,而 RGB-D 由于其深度信息易得而广泛用于 SLAM 基础理论的研究,易于上手,因此是较为合适初学者的传感器。文中对 SLAM 的原理进行了较为深入的分析,并搭建了基于 RGB-D 相机的 SLAM 简单系统,主要有如下结论:

(1) 搭建了基于 RGB-D 的 SLAM 系统的前端里程计的设计,初始设计中的位姿估计误差较大,即使进行了局部位姿优化,也会在较短时间跟踪丢失,因为前端是用上一帧的位姿以及地图点来进行 PnP 的位姿估算计算下一帧的,因此,若中间存在任何一个问题,都会导致算法的失败。而改用了存储地图点之后,将新来的图像帧与整个地图点进行 PnP 的位姿估算,这样就避免了出现中间帧效果不好导致跟踪失败的情况的发生。

(2) 进行了后端优化的实验,用 16 张图片的观测值构建 Bundle Adjustment 问题,并分别用 Ceres 库与 G2O 图进行了基于列文伯格—马夸尔特算法的迭代优化,发现 Ceres 的迭代次数比 G2O 库少,时间也快 1 倍左右。并用 Dogleg 算法与 LM 算法进行比较,发现 Dogleg 算法比 LM 算法收敛速度要快。

(3) 利用 ORB-SLAM、LSD-SLAM 分别建出了稀疏地图与半稠密地图,用八叉树地图建立了占据栅格地图,回环检测部分则利用词典模型建立 ORB 字典进行了分析。分析了各种地图形式在不同场景下的表达形式,发现用点云表达的地图内存占据较大,只能进行重建,而占据栅格地图如八叉树图可进行导航且内存占据极小。

SLAM 仍然需要在很多方面进行改进才能应用于实际系统,如下:

(1) 在 SLAM 初始化过程以及跟踪过程中,基于特征点法用的 RANSAC 算法对噪声比较敏感,以后的工作可尝试选择匹配点描述子距离最近的点来进行位姿估算,也许效果会更好。

(2) 单目由于存在尺度性问题,因此以后的工作可以尽量结合 IMU 进行 SLAM 的研究,并尽量与实际工作结合,朝轻量化、小型化的方向发展。

(3) 单目 SLAM 与 RGB-D SLAM 含有丰富的图像信息,在考虑诸如野外场景下的地图构建时,为了避免如机器人将草丛视为障碍或者浮萍视为平坦区域的误判断情况,可以考虑深度学习与 SLAM 的结合,减小误判断发生的概率。

参考文献

- [1] Smith R, Self M, Cheeseman P. Estimating Uncertain Spatial Relationships in Robotics[J]. Autonomous Robot Vehicles, 1988, 5 (5) : 435-461.
- [2] Davison A J, Reid I D, Molton N D, et al. MonoSLAM: real-time single camera SLAM[J]. IEEE Trans Pattern Anal Mach Intell, 2007, 29 (6) : 1052-1067.
- [3] Grisetti G, Stachniss C, Burgard W. Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters[J]. IEEE Transactions on Robotics, 2007, 23 (1) : 34-46.
- [4] Hauke Strasdat, J. M. M. Montiel, Andrew J. Davison. Visual SLAM: Why filter?[J]. Image & Vision Computing, 2012, 30 (2) : 65-77.
- [5] Cadena C, Carlone L, Carrillo H, et al. Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age[J]. IEEE Transactions on Robotics, 2016, 32 (6) : 1309-1332.
- [6] Kohlbrecher S, Stryk O V, Meyer J, et al. A flexible and scalable SLAM system with full 3D motion estimation[C]// IEEE International Symposium on Safety, Security, and Rescue Robotics. IEEE, 2011: 155-160.
- [7] Hess W, Kohler D, Rapp H, et al. Real-time loop closure in 2D LIDAR SLAM[C]// IEEE International Conference on Robotics and Automation. IEEE, 2016: 1271-1278.
- [8] Carrasco P L N, Bonin-Font F, Codina G O. Stereo Graph-SLAM for Autonomous Underwater Vehicles[M]// Intelligent Autonomous Systems 13. Springer International Publishing, 2016: 351-360.
- [9] Balasuriya B L E A, Chathuranga B A H, Jayasundara B H, et al. Outdoor robot navigation using Gmapping based SLAM algorithm[C]// Moratuwa Engineering Research Conference. IEEE, 2016: 403-408.
- [10] Olson E B. Real-time correlative scan matching[C]// IEEE International Conference on Robotics and Automation. IEEE, 2009: 4387-4393.
- [11] Konolige K, Grisetti G, Kümmerle R, et al. Efficient Sparse Pose Adjustment for 2D mapping[J]. 2010: 22-29.
- [12] Steux B, Hamzaoui O E. CoreSLAM: a SLAM Algorithm in less than 200 lines of C code[J]. 2010.
- [13] Davison A J, Reid I D, Molton N D, et al. MonoSLAM: real-time single camera SLAM[J]. IEEE Trans Pattern Anal Mach Intell, 2007, 29 (6) : 1052-1067.
- [14] Kameda Y. Parallel Tracking and Mapping for Small AR Workspaces (PTAM) Augmented Reality[J]. Journal of the Institute of Television Engineers of Japan, 2012, 66 (1) : 45-51.
- [15] Mur-Artal R, Montiel J M M, Tardós J D. ORB-SLAM: A Versatile and Accurate Monocular SLAM System[J]. IEEE Transactions on Robotics, 2015, 31 (5) : 1147-1163.
- [16] Mur-Artal R, Tardós J D. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras[J]. IEEE Transactions on Robotics, 2016, 33 (5) : 1255-1262.

-
- [17] Engel J, Sturm J, Cremers D. Semi-dense Visual Odometry for a Monocular Camera[C]// IEEE International Conference on Computer Vision. IEEE, 2014: 1449-1456.
 - [18] Engel J, Schöps T, Cremers D. LSD-SLAM: Large-Scale Direct Monocular SLAM[M]// Computer Vision – ECCV 2014. Springer International Publishing, 2014: 834-849.
 - [19] Engel J, Koltun V, Cremers D. Direct Sparse Odometry[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2016, PP (99) : 1-1.
 - [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] Engel J, Stückler J, Cremers D. Large-scale direct SLAM with stereo cameras[C]// Ieee/rsj International Conference on Intelligent Robots and Systems. IEEE, 2015: 1935-1942.
 - [22] Negre P L, Bonin-Font F, Oliver G. Cluster-based loop closing detection for underwater slam in feature-poor regions[C]// IEEE International Conference on Robotics and Automation. IEEE, 2016: 2589-2595.
 - [23] Gomezojeda R, Moreno F A, Scaramuzza D, et al. PL-SLAM: a Stereo SLAM System through the Combination of Points and Line Segments[J]. 2017.
 - [24] Labbé M, Michaud F. Online global loop closure detection for large-scale multi-session graph-based SLAM[C]// Ieee/rsj International Conference on Intelligent Robots and Systems. IEEE, 2014: 2661-2666.
 - [25] Endres F, Hess J, Sturm J, et al. 3-D Mapping With an RGB-D Camera[J]. IEEE Transactions on Robotics, 2014, 30 (1) : 177-187.
 - [26] Brunetto N, Salti S, Fioraio N, et al. Fusion of Inertial and Visual Measurements for RGB-D SLAM on Mobile Devices[C]// IEEE International Conference on Computer Vision Workshop. IEEE, 2015: 148-156.
 - [27] Gutierrez-Gomez D, Mayol-Cuevas W, Guerrero J J. Inverse depth for accurate photometric and geometric error minimisation in RGB-D dense visual odometry[C]// IEEE International Conference on Robotics and Automation. IEEE, 2015: 83-89.
 - [28] Gutierrez-Gomez D, Mayol-Cuevas W, Guerrero J J. Dense RGB-D visual odometry using inverse depth[M]. North-Holland Publishing Co. 2016.
 - [29] Yang S, Song Y, Kaess M, et al. Pop-up SLAM: Semantic monocular plane SLAM for low-texture environments[C]// Ieee/rsj International Conference on Intelligent Robots and Systems. IEEE, 2016: 1222-1229.
 - [30] Bowman S L, Atanasov N, Daniilidis K, et al. Probabilistic data association for semantic SLAM[C]// IEEE International Conference on Robotics and Automation. IEEE, 2017: 1722-1729.
 - [31] Luc P, Neverova N, Couprie C, et al. Predicting Deeper into the Future of Semantic Segmentation[J]. 2017: 648-657.
 - [32] Pohlen T, Hermans A, Mathias M, et al. Full-Resolution Residual Networks for Semantic Segmentation in Street Scenes[J]. 2016: 3309-3318.

-
- [33] Huai J, Toth C K, Grejner-Brzezinska D A. Stereo-inertial odometry using nonlinear optimization[C]// International Technical Meeting of the Satellite Division of the Institute of Navigation. 2015.
 - [34] Quan M, Piao S, Tan M, et al. Map-Based Visual-Inertial Monocular SLAM using Inertial assisted Kalman Filter[J]. 2017.
 - [35] Tsai R Y. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses[M]// Radiometry. Jones and Bartlett Publishers, Inc. 1992.
 - [36] Zhang Z. A Flexible New Technique for Camera Calibration[J]. Tpami, 2000, 22 (11) : 1330-1334.
 - [37] 马颂德, 张正友. 计算机视觉: 计算理论与算法基础[M]. 北京: 科学出版社, 1998.
 - [38] Leonhard Euler, " Problema algebraicum ob affectiones prorsus singulares memorabile " , Commentatio 407 Indicis Enestoemiani, Novi Comm. Acad. Sci. Petropolitanae 15 (1770) , 75-106.
 - [39] Stuelpnagel J. On the Parametrization of the Three-Dimensional Rotation Group[J]. Siam Review, 1964, 6 (4) : 422-430.
 - [40] 程小红. 哈密顿与四元数[J]. 数学通报, 2006, 45 (6) : 57-59.
 - [41] Kümmerle R, Grisetti G, Strasdat H, et al. G2o: A general framework for graph optimization[C]// IEEE International Conference on Robotics and Automation. IEEE, 2011: 3607-3613.
 - [42] Varadarajan V S. Lie groups, Lie algebras, and their representations[J]. 1974, 659 (5) : xiv, 351.
 - [43] 陈宝林. 最优化理论与算法[M]. 北京: 清华大学出版社, 1989.
 - [44] Fan J Y, Yuan Y X. On the Quadratic Convergence of the Levenberg-Marquardt Method without Nonsingularity Assumption[J]. Computing, 2005, 74 (1) : 23-39.
 - [45] 刘浩敏, 章国锋, 鲍虎军. 基于单目视觉的同时定位与地图构建方法综述[J]. 计算机辅助设计与图形学学报, 2016, 28 (6) : 855-868.
 - [46] Lowe D G, Lowe D G. Distinctive Image Features from Scale-Invariant Keypoints[J]. International Journal of Computer Vision, 2004, 60 (2) : 91-110.
 - [47] Bay H, Tuytelaars T, Gool L V. SURF: Speeded Up Robust Features[J]. Computer Vision & Image Understanding, 2006, 110 (3) : 404-417.
 - [48] Rublee E, Rabaud V, Konolige K, et al. ORB: An efficient alternative to SIFT or SURF[J]. 2011, 58 (11) : 2564-2571.
 - [49] Tola E, Lepetit V, Fua P. DAISY: An Efficient Dense Descriptor Applied to Wide-Baseline Stereo[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2010, 32(5):815-30.
 - [50] Muja M. Fast approximate nearest neighbors with automatic algorithm configuration[C]// International Conference on Computer Vision Theory and Application Vissapp. 2009: 331-340.
 - [51] Hartley R. In defense of the 8-point algorithm[J]. IEEE Iccv, 1995, 19 (6) : 580-593.

-
- [52] Lepetit V, Moreno-Noguer F, Fua P. EPnP: An Accurate $O(n)$ Solution to the PnP Problem[J]. International Journal of Computer Vision, 2009, 81 (2) : 155-166.
 - [53] Penate-Sanchez A, Andrade-Cetto J, Moreno-Noguer F. Exhaustive Linearization for Robust Camera Pose and Focal Length Estimation[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2013, 35 (10) : 2387-2400.
 - [54] Wu C, Agarwal S, Curless B, et al. Multicore bundle adjustment[C]//Computer Vision and Pattern Recognition. IEEE, 2011: 3057-3064.
 - [55] Sibley G, Matthies L, Sukhatme G. Sliding window filter with application to planetary landing[J]. Journal of Field Robotics, 2010, 27 (5) : 587-608.
 - [56] Gao X, Zhang T. Robust RGB-D simultaneous localization and mapping using planar point features[J]. Robotics & Autonomous Systems, 2015, 72: 1-14.
 - [57] Lloyd S. Least squares quantization in PCM[J]. IEEE Trans, 1982, 28 (2) : 129-137.
 - [58] Arthur D, Vassilvitskii S. k-means++: the advantages of careful seeding[C]// Eighteenth Acm-Siam Symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics, 2007: 1027-1035.
 - [59] Robertson S. Understanding inverse document frequency: on theoretical arguments for IDF[J]. Journal of Documentation, 2013, 60 (5) : 503-520.
 - [60] D. Nister; H. Stewenius, Scalable Recognition with a Vocabulary Tree[C]//Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, 2006: 2161 - 2168.
 - [61] Nister D, Stewenius H. Robust Scalable Recognition with a Vocabulary Tree[J]. Proc Cvpr, 2006, 2 (10) : 2161-2168.
 - [62] Sturm J, Engelhard N, Endres F, et al. A benchmark for the evaluation of RGB-D SLAM systems[C]// Ieee/rsj International Conference on Intelligent Robots and Systems. IEEE, 2012: 573-580.
 - [63] Fischler M A, Bolles R C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography[M]. ACM, 1981.
 - [64] 高翔, 张涛, 颜沁睿, 刘毅. 视觉 SLAM 十四讲: 从理论到实践[M], 北京: 电子工业出版社, 2017.
 - [65] S. Agarwal, and K. Mierle, and Others, " Ceres Solver. " [http: //ceres-solver. org](http://ceres-solver.org).

攻读学位期间的研究成果

- [1] 安帅, 杨杰, 乔羽, 李鹏. 基于单目增量式稀疏 SFM 研究与实现[J]. 青岛大学学报(自然科学版), 2018(02).

致谢

时间过得飞快，转眼硕士三年也即将过去，有幸来到青岛大学，从本科到硕士的母校，一切都倍感亲切，在这三年中，收货颇多。

在此，首先感谢我的导师——杨老师，感谢当年的知遇之恩，感谢在学习上的指导，感谢在未来中的引导，感谢对我如此的信任与关怀，感谢无私的帮助！杨老师是我非常尊重又极其喜欢的老师，每当有困惑时，总是给我信心，像家人一样，为自己点明未来的灯。遇到杨老师，是极大的幸运！

在此，还要感谢李老师，不愧为大师级的教授，感谢教会我学习、研究的方法，包括查阅文献，资料筛选等等，对我终身受益，还要感谢在生活中的关怀与精神的鼓舞，再次感谢！

在此，还要感谢纪老师，在生活中给予了很多帮助，也教会为人处世的准则，更是着眼于未来规划人生，追逐梦想。非常感谢！

在此，还要感谢哈工大的张老师，于老师，张老师的眼界高远，是凡人所不能及！与张老师谈话，句句新奇，无不让人振奋！于老师像是一位姐姐一样，年轻漂亮，且认真负责，在学习上给了我极大帮助。感谢上海卫星设计院的施老师，其学术上的严谨，让我受益良多。感谢体育学院的方老师，感谢在生活中的帮助，如好朋友一般的关心。

在此，还要感谢帮助我的同学老师与朋友，感谢在生活中、学习上等方面的帮助，感谢乔羽、李鹏、于海龙、高涵、邹云龙、武支友、杨志宏等等同学的帮助与鼓励，万分感谢！

最后，感谢父母，感谢养育之恩，感谢支持，感谢！

学位论文独创性声明

本人声明，所呈交的学位论文系本人在导师指导下独立完成的研究成果。文中依法引用他人的成果，均已做出明确标注或得到许可。论文内容未包含法律意义上已属于他人的任何形式的研究成果，也不包含本人已用于其他学位申请的论文或成果。

本人如违反上述声明，愿意承担由此引发的一切责任和后果。

论文作者签名：安帅

日期：2018年5月5日

学位论文知识产权权属声明

本人在导师指导下所完成的学位论文及相关的职务作品，知识产权归属学校。学校享有以任何方式发表、复制、公开阅览、借阅以及申请专利等权利。本人离校后发表或使用学位论文或与该论文直接相关的学术论文或成果时，署名单位仍然为青岛大学。

本学位论文属于：

保密 ☐，在 年解密后适用于本声明。

不保密 ☒。

（请在以上方框内打“√”）

论文作者签名：安帅

日期：2018年5月5日

导师签名：杨杰

日期：2018年5月5日

（本声明的版权归青岛大学所有，未经许可，任何单位及任何个人不得擅自使用）