

分类号：TP273

单位代码：10363

密 级：公开

学 号：2210320112

题 目 基于点线融合特征的视觉惯性 SLAM 算法

英文并列

题 目 VISUAL INERTIAL SLAM ALGORITHM

BASED ON POINT-LINE FUSION FEATURES

学生姓名：吴桐

指导教师：黄宜庆（教授）

专 业：控制科学与工程

研究方向：模式识别与智能系统

论文答辩日期：2024 年 5 月 12 日

安徽工程大学学位论文原创性声明

本人郑重声明：我恪守学术道德，崇尚严谨学风。所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已明确注明和引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品及成果的内容。论文为本人亲自撰写，我对所写的内容负责，并完全意识到本声明的法律后果由本人承担。

学位论文作者签名：

日期：2024 年 5 月 29 日

安徽工程大学学位论文版权使用授权书

学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅或借阅。本人授权安徽工程大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保密 ☐，在 _____ 年解密后适用本版权书。

本学位论文属于

不保密 ☒。

学位论文作者签名：吴桐

指导教师签名：黄直欣

日期：2024 年 5 月 29 日

日期：2024 年 5 月 29 日

基于点线融合特征的视觉惯性 SLAM 算法

摘 要

随着科学技术的日益提升，机器人技术已渗透到人们日常生活中的方方面面。即时定位与地图构建(Simultaneous Localization and Mapping, SLAM)作为机器人的一项关键技术，已成为研究的热点内容。传统的视觉 SLAM 系统基于点特征信息进行位姿估计，或者是单纯的将线特征引入到系统，没有将点线特征进行融合，造成在楼道、室内走廊等光线弱的暗纹理环境中点特征信息不足、线特征杂糅甚至是特征信息丢失的情况，导致 SLAM 系统在此类环境中的位姿估计误差大，影响其定位精度。为改善此类问题，将线特征引入基于点特征的视觉惯性 SLAM 系统，改进点线提取算法，构建点线耦合残差模型并加入到后端滑动窗口优化中，给出了一种基于点线融合特征的视觉惯性 SLAM 算法。主要研究内容如下：

首先，在视觉前端特征信息处理过程中，引入线特征并增加约束条件，使点和线特征优势互补，给出了一种基于自适应点线特征提取的视觉惯性 SLAM 算法。采用 Shi-Tomasi 算法检测角点特征，用 KLT(Kanade-Lucas-Tomasi)算法执行点的跟踪。为解决提取线特征时存在多短和重叠的线段不能被有效剔除的问题，在传统的 LSD(Line segment Detector)算法中设定提取线特征的动态阈值，并增加短线拼接的方法提取场景中的线特征，在传统提取算法的基础上进一步提高线特征的利用率。实验证明，加入线特征后，丰富了特

征信息，提升了特征匹配精度。

其次，在前面工作的基础上，给出了基于动态点线耦合的单目视觉惯性 SLAM 算法。利用点与线特征位置的相似性，提取新的点线耦合特征并构造点线耦合残差模型，用点线耦合残差来搜索点与线特征之间的关系。得到点线特征信息的数据后，需要将视觉惯性测量单元(Inertial Measurement Unit, IMU)的预积分数据与点线特征数据进行视觉惯性对齐。滑动窗口的非线性状态估计器可以根据先验信息、IMU 残差、闭环约束、点残差、线残差以及构建的点线耦合残差构造联合优函数，计算滑动窗口中的帧的位置、速度、旋转与偏差。

最后，本论文基于视觉作业开源移动智能机器人平台搭建了视觉惯性 SLAM 系统。第一步对系统进行整体仿真验证，并在公开的 EuRoC 数据集中，将算法与传统的经典算法进行对比，通过 EVO 工具评测各算法的轨迹和定位误差。仿真实验证明，改进的算法要优于传统经典算法。第二步进行实物平台实验，把 SLAM 算法导入实物平台，并通过平台上的 INDEMIND 相机采集真实场景的信息来进行实时实物平台验证。经过实验可证明，本文给出的 SALM 算法在真实的复杂场景中具有良好性能，表明了所给出算法的可行性。

关键词：点线特征，短线拼接，自适应阈值，点线耦合，滑动窗口优化

VISUAL INERTIAL SLAM ALGORITHM BASED ON POINT-LINE FUSION FEATURES

ABSTRACT

With the increasing advancement of science and technology, robot technology has penetrated into every aspect of people's daily lives. Simultaneous Localization and Mapping (SLAM), as a key technology for robots, has become a hot research topic. The traditional vision SLAM system is based on point feature information for position estimation, or simply introduces line features into the system without fusion of point and line features, resulting in insufficient point feature information, hybridization of line features or even loss of feature information in dark textured environments such as hallways and indoor corridors with low light, leading to large position estimation errors of SLAM systems in such environments and affecting their positioning accuracy. In order to improve such problems, line features are introduced into the visual inertial SLAM system based on point features, the point and line extraction algorithm is improved, a point-line coupling residual model is constructed and added to the back-end sliding window optimization, and a point-line based Visual-inertial SLAM algorithm for fused features. The main research contents are as follows:

Firstly, a visual inertial SLAM algorithm based on adaptive point and line feature extraction is given by introducing line features and adding constraints to complement the advantages of point and line features during the processing of visual front-end feature information. The Shi-Tomasi algorithm is used to detect the corner point features, and the Kanade-Lucas-Tomasi (KLT) algorithm is used to perform the point tracking. In order to solve the problem that there are multiple short and overlapping line segments that can not be effectively eliminated when extracting the line features, the dynamic thresholds for extracting the line features are set in the traditional Line segment Detector (LSD) algorithm and the add the method of short line splicing to extract line features in the scene to further improve the utilization rate of line features on the basis of the traditional extraction algorithm. The experiment proves that adding line features enriches the feature information and improves the feature matching accuracy.

Secondly, based on the previous work, a monocular visual inertial SLAM algorithm based on dynamic point-line coupling is given. Using the similarity of the position of point and line features, new point-line coupling features are extracted and the point-line coupling residual model is constructed, and the point-line coupling residual is used to search the relationship between point and line features. After obtaining the data of point and line feature information, it is necessary to visually inertially align

the pre-integrated data from the visual inertial measurement unit (IMU) with the point and line feature data. The nonlinear state estimator for the sliding window can construct a joint optimization function based on a priori information, IMU residuals, closed-loop constraints, point and line residuals, and constructed point-line coupling residuals to compute the position, velocity, rotation, and deviation of the frames in the sliding window.

Finally, this thesis builds a visual inertial SLAM system based on the vision operations open source mobile intelligent robot platform. The first step is to conduct overall simulation verification of the system, compare the algorithm with traditional classic algorithms in the public EuRoC data set, and evaluate the trajectory and positioning errors of each algorithm through the EVO tool. Simulation experiments prove that the improved algorithm is better than the traditional classic algorithm. The second step is to carry out the physical platform experiment, import the SLAM algorithm into the physical platform, and collect the information of the real scene through the INDEMIND camera on the platform to carry out the real-time physical platform verification. After the experiments it can be proved that the SALM algorithm given in this thesis has good performance in real complex scenes, which shows the feasibility of the given algorithm.

Wu Tong (Control Science and Engineering)

Supervised by Huang Yiqing

KEY WORDS: point and line features, short line splicing, adaptive

thresholding, point and line coupling, sliding window optimization

目 录

摘 要	I
ABSTRACT.....	III
插图清单	IX
表格清单	XI
第 1 章 绪论	1
1.1 课题的研究背景及意义	1
1.2 国内外 SLAM 研究现状	2
1.2.1 视觉 SLAM 研究现状	2
1.2.2 视觉惯性 SLAM 研究现状	4
1.2.3 点线视觉惯性 SLAM 研究现状	6
1.3 本文研究内容及组织结构	7
第 2 章 视觉惯性 SLAM 算法基础理论分析.....	9
2.1 机器人的刚体运动	9
2.1.1 相机旋转与平移	9
2.1.2 四元数	10
2.2 李群和李代数	12
2.3 相机数学理论	13
2.3.1 小孔成像原理	14
2.3.2 图像畸变矫正	16
2.4 IMU 数学模型.....	17
2.5 本章小结	18
第 3 章 基于自适应点线特征提取的视觉惯性 SLAM 算法.....	19
3.1 LSD 线特征提取	19
3.1.1 线特征提取原理	19
3.1.2 设定线特征检测的自适应阈值	21
3.1.3 短线拼接	22
3.2 线特征描述匹配	24

3.3 线特征数学表述	26
3.3.1 Plücker 坐标	26
3.3.2 正交表示法	28
3.4 点线提取算法实验与分析	28
3.5 本章小结	30
第 4 章 基于动态点线耦合的单目视觉惯性 SLAM 算法.....	31
4.1 算法框架	31
4.2 点线耦合	32
4.2.1 提取点线耦合特征	32
4.2.2 点线耦合残差模型	33
4.3 联合优化求解	34
4.3.1 IMU 残差模型.....	34
4.3.2 点残差模型	35
4.3.3 线残差模型	36
4.4 后端滑动窗口优化	37
4.5 点线耦合特征提取实验与分析	39
4.6 本章小结	40
第 5 章 基于 ROS 的视觉惯性 SLAM 算法仿真与平台验证	41
5.1 仿真实验验证	41
5.1.1 SLAM 算法整体运行结果	42
5.1.2 仿真实验评估分析	43
5.2 实物平台实验验证	47
5.3 本章小结	49
第 6 章 总结与展望	50
6.1 总结	50
6.2 展望	51
参考文献	52
作者攻读学位期间发表的学位论文及参加的科研项目	58
致 谢	59

插图清单

图 1-1 多元化智能机器人	1
图 1-2 经典视觉 SLAM 算法时间线	3
图 1-3 视觉惯性 SLAM 发展	5
图 2-1 坐标变换示意图	9
图 2-2 小孔成像原理图	14
图 2-3 真实、对称的成像图示	15
图 2-4 径向畸变两种类型	16
图 2-5 切向畸变来源示意图	17
图 3-1 LSD 线段支持域示意图	19
图 3-2 LSD 算法流程图	21
图 3-3 短线拼接示意图	23
图 3-4 短线拼接算法图	23
图 3-5 LBD 描述线段支持域	24
图 3-6 二进制描述子	26
图 3-7 Plücker 坐标示意图	27
图 3-8 线残差模型	27
图 3-9 改进前后线特征的提取效果	29
图 3-10 大型工厂环境中三种算法位姿轨迹	29
图 3-11 小型室内环境中三种算法位姿轨迹	30
图 4-1 视觉惯性 SLAM 系统框架结构	31
图 4-2 点特征搜索线特征示意图	33
图 4-3 点线耦合残差模型	33
图 4-4 点残差模型	35
图 4-5 线残差测量模型	37
图 4-6 滑动窗口优化示意图	38
图 4-7 特征提取算法在 MH_05 验证	39
图 4-8 特征提取算法在 V1_03 验证	39

图 4-9 特征提取算法在 V2_03 验证	39
图 5-1 数据集不同场景图	41
图 5-2 本文算法在 MH_05_difficult 运行轨迹	42
图 5-3 本文算法在 V1_03_difficult 运行轨迹	43
图 5-4 本文算法在 V2_03_difficult 运行轨迹	43
图 5-5 三种算法在 MH_04 序列的三维绝对轨迹误差图	44
图 5-6 三种算法在 V1_03 序列的三维绝对轨迹误差图	44
图 5-7 三种算法在 V2_03 序列的三维绝对轨迹误差图	44
图 5-8 三种算法在两序列的实时绝对位姿误差曲线图	45
图 5-9 三种算法在 V1_03 序列绝对位姿误差分布	45
图 5-10 三种算法在 V2_03 序列绝对位姿误差分布	46
图 5-11 实物平台	47
图 5-12 基于实物平台的实验	48
图 5-13 基于实物平台的算法运行轨迹	48

表格清单

表 5-1 数据集传感器参数	41
表 5-2 不同算法的位姿估计误差	46

第 1 章 绪论

1.1 课题的研究背景及意义

科技在进步，社会在发展。在当前时代，人们致力于解放双手，提高生产效率，降低人力成本，在机器人研究领域已深耕细作数十年，并在多方面实现了自动化、无人化。比如，可以无死角不留痕迹清扫细小颗粒垃圾的家庭扫地机器人，如图 1-1(a)所示；在餐厅中有可以代替人工给顾客送餐的服务型送餐机器人，如图 1-1(b)所示；在军事领域中，机器人可以代替扫雷兵排雷，减少不必要的伤亡，如图 1-1(c)所示；在工厂的生产线上，机器人能代替工人完成手动装配任务，实现装配机械化作业，例如汽车部件装配等，如图 1-1(d)所示。机器人的身影已出现在了当今的各行各业中，在人们身边随处可见。近几年国家出台了多项关于鼓励机器人产业发展的政策，资金投入规模庞大，对机器人产业的重视尤为突出。因此，机器人的研究在当前意义重大。



(a) 扫地机器人



(b) 送餐机器人



(c) 扫雷机器人



(d) 装配机器人

图 1-1 多元化智能机器人

智能机器人的功能受感知与认知、运动与操作、控制与规划、人机交互以及学习与智能这五大关键技术的影响，作为感知与认知中的核心 SLAM 技术^[1-2]一直都是国内外学者研究的热点课题。SLAM 技术可以让机器人依靠自身搭载的传感器而无需全球定位系统就能够在未知环境中完成其运动位姿估计和地图构建。根据传感器类型的分类，SLAM 可以分为视觉 SLAM、激光雷达 SLAM、惯性导航 SLAM、多传感器融合 SLAM 等。虽然激光雷达 SLAM 技术成熟，可以精准实现测距和定位，但是由于其成本昂贵的原因未能进入大众视野，目前只用于高端自动驾驶领域。反之，视觉 SLAM 技术得益于其能够通过可视化的图像提供相当丰富的信息，且凭借成本低、体积小、质量轻等优势，在无人驾驶^[3-4]、虚拟现实^[5-6]等领域中得到了广泛的应用，使得基于该技术生产的产品走向平民化。由于基于视觉的 SLAM 技术具有更佳的实用价值，使得国内外研究者更倾向于对该项技术的研究。

1.2 国内外 SLAM 研究现状

SLAM 这一概念最早于 1986 年由 Smith 等人^[7]提出，自那时起，这一技术经历了持续发展，并被划分为三个主要阶段。在早期阶段（1986 年—2004 年），主要研究的是 SLAM 问题的定义、基于概率框架的建模和求解方法；中期阶段（2004 年—2015 年），深入研究 SLAM 问题的一些性质，比如稀疏性、收敛性和一致性等，更多样和更高效的算法也被相继提出；当前阶段（2015 年—至今），考虑算法的鲁棒性、可扩展性、资源约束下的高效性以及高层语义认知任务导向等。SLAM 系统主要包括前端和后端两部分，前端负责从传感器数据中提取特征、数据关联（如特征提取和回环检测）等工作，而后端负责执行最大后验估计、滤波等任务。

1.2.1 视觉 SLAM 研究现状

机器人要实现功能，需要以其感知未知环境为基础，这就需要利用各种传感器进行辅助。然而，相机作为机器人的“眼睛”去感知外界环境无疑是一个不错的选择，这也就是常称的视觉 SLAM。视觉 SLAM 根据相机特性的不同可以分为单目视觉 SLAM，双目视觉 SLAM 以及 RGB-D 视觉 SLAM。图 1-2 展示了经典视觉 SLAM 算法的发展时间线。

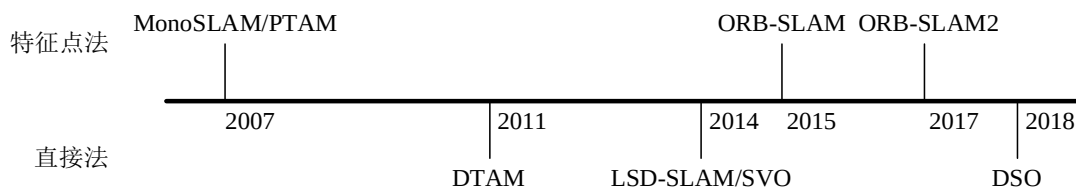


图 1-2 经典视觉 SLAM 算法时间线

首个纯单目视觉 SLAM 系统 Mono-SLAM 于 2007 年由 Davision 等人^[8]提出。这也是首个提取 Shi-Tomasi 角点^[9]的系统，通过主动搜索投影椭圆中的匹配点特征和测量平滑相机一般 3D 运动模型（从视频流捕获动态先验信息），并采用一种自顶向下的方法解决单目特征初始化问题，使一个自由运动的相机实现实时定位和建图。由于其视觉后端采用了扩展卡尔曼滤波器(Extended Kalman Filter, EKF)^[10]进行位姿状态估计，计算复杂程度大，消耗了大量资源。同年，Klein 等人^[11]在算法结构上进行了创新并设计出了 PTAM(Parallel Tracking And Mapping)算法，首次提出将姿态跟踪与地图构建分开作为两个独立线程，并在后端采用光速平差法(Bundle Adjustment, BA)^[12]进行非线性优化的方案，实现了跟踪与建图并行优化，提升了系统的定位精度与实时性，但是 PTAM 没有闭环检测线程，不可避免会产生高功耗，因此增加了成本，不宜应用在大范围场景中。虽然 PTAM 在当时的使用场景存在局限性，但是其采用的方案为后期的研究提供了参考，当今大部分视觉 SLAM 系统都是在其框架基础上进行改进与创新的。如 Mur-Artal 等人^[13]于 2015 年设计的 ORB-SLAM，在 PTAM 算法基础上增加了闭环检测算法，由双线程扩展到了三线程，能执行多线程任务，且采用适者生存的方法对地图点和关键帧进行选择，剔除了冗余关键帧，增强了系统长时间运行的能力，提高了系统的鲁棒性。由于 ORB-SLAM 采用的单目相机，致使其从场景中获取到的信息有限，无法测量空间点的深度信息。后来该团队在 ORB-SLAM 的基础上进行改进，提出了功能更强的 ORB-SLAM2 算法^[14]，该算法增加了对双目和 RGB-D 相机的支持，通过使用远和近双目点和单目观测，能够有效补齐单目相机提取信息有限的短板，其结果比之前的 SLAM 精度更好，被誉为是最全面基于点特征的视觉 SLAM 系统。

上文所述的几种视觉 SLAM 皆以特征点法来进行研究的，该方法导致 SLAM 系统需要耗费更多的时间去完成特征识别和匹配，且需要以场景中高质量特征信息为支撑，否则极易造成图像信息丢失。针对该类问题，有学者提出

了在视觉前端运用直接法的方案，该方法直接利用图像的明暗对比度估计相机运动，缩减特征识别与匹配时间。首个基于直接法的系统——稠密跟踪与建图 (Dense Tracking and Mapping, DATM) 由 Newcombe 等人^[15]提出，该系统结合了深度相机直接获取图像的深度信息，但由于其需要额外的 GPU 参与来完成大量计算，增加了系统的开发成本。Engel 等人^[16]提出了一种基于直接法的大范围单目 SLAM (Large-scale Direct Monocular SLAM, LSD-SLAM)，该算法采用随机深度初始化策略对像素随机深度初始化，并对每个像素深度独立计算，用卡尔曼滤波更新深度估计，构建高精度的半稠密 3D 环境地图。该算法可直接运行在 CPU 上，无需 GPU 参与计算，但是该算法在高速运动场景中的效果表现差。苏黎世大学的 Forster 等人^[17]于 2014 年提出了名为快速半视觉里程计算法 (Fast Semi-Direct Monocular Visual Odometry, SVO)，采取了特征点法与直接法相结合的方法，在定位上取得了很好的效果，SVO 在位姿估计中的思路很新颖，运行速度非常快，即使是在普通 PC 上，其更新速度也能达到 100 Frame/s，后来该团队对 SVO 进行了升级，迭代版本的更新速度是上一代的 4 倍。Engel 于 2018 年提出了直接稀疏里程计 (The Direct Sparse Odometry, DSO)^[18]方法，该方法引入了光度模型，对图像像素进行均匀采样并计算。通过结合光度模型与集合模型，能够联合估计相机位姿和像素点的深度等参数，使系统的定位精度与鲁棒性得到了提高。

近年来，深度学习在学术界引起了广泛热议。一些学者提出将深度学习技术融入到传统的 SLAM 算法中，旨在解决传统 SLAM 算法难以应对的问题。Yu 等人^[19]基于 ORB-SLAM2 系统提出了 DS-SLAM 系统以解决动态环境问题，在系统中引入了语义分割网络算法 SegNet^[20]，并将其与对极几何约束方法结合，以减小动态环境对系统定位的影响。Zhan 等人^[21]提出的 DF-VIO 算法将传统方法与深度学习方法相结合，采用卷积神经网络来估计相机深度，并提取匹配特征信息。Wang 等人^[22]提出了 DSP-SLAM 系统，该系统通过语义实例分割检测对象，并联合新颖的二阶优化算法，采用特定类别的深度形状嵌入来估计对象的形状和姿态^[23]。

1.2.2 视觉惯性 SLAM 研究现状

纯视觉 SLAM 算法仅依靠相机对场景中的图像处理实现系统的定位和建图。

然而，该方法存在局限性，受环境影响的程度大，特别是在暗纹理环境中，容易导致特征丢失，进而引起定位不准确的问题。为了更好优化 V-SLAM 算法，研究者们在其框架中引入了视觉惯性里程计(Visual-Inertial Odometry, VIO)^[24]，这是一个将低成本的 IMU 集成到视觉里程计(Visual Odometry, VO)的系统，通过系统中的视觉传感器（如单目相机）和 IMU 相结合^[25-错误:未找到引用源。]，在提取特征信息方面取得了更好的效果，并被广泛应用到实际场景中，有助于提高 SLAM 系统在复杂环境下的性能表现。图 1-3 展示了视觉惯性 SLAM 算法的发展时间线。

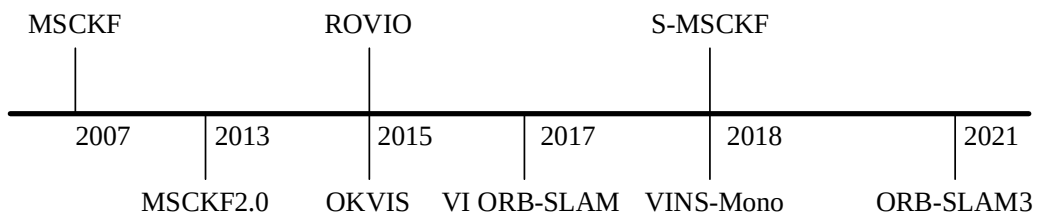


图 1-3 视觉惯性 SLAM 发展

Mourikis 等人^[27]在 2007 年提出 MSCKF 算法，此算法融合了 EKF，采用紧耦合的方法把相机采集的图像与 IMU 信息结合，实现对图像的匹配与更新，并在更新阶段把图像引入后端滑动窗口优化，降低了系统的计算量。Li 等人^[28]在 KSCKF 的基础上提出了 MSCKF2.0 算法，该算法通过确保其线性化系统模型具有正确的可观测性，以及执行相机到 IMU 校准参数的在线估计，经过测试，其精度和一致性高于迭代滑动窗口固定滞后平滑器。Leutenegger 等人^[29]提出了利用非线性优化技术实现基于关键帧的视觉惯性里程测量(OKVIS)的方法，该方法结合地标和惯性项重投影误差，制定了一个严格的概率成本函数，通过边缘化将优化限制在关键帧的有界窗口内，确保操作的实时性。Bloesch 等人^[30]也将 EKF 融入到算法，提出了 ROVIO 系统，通过直接使用图像像素强度误差，以纯机器人中心表示全滤波器状态，共同估计相机特性与 IMU 加法偏差，实现精确的跟踪性能与极高的鲁棒性。Mur-Artal 等人^[31]在之前开发的 ORB-SLAM 系统的基础上加入了 IMU，提出了视觉惯性 ORB-SLAM(Visual Inertial ORB-SLAM, VI ORB-SLAM)系统，该系统在视觉前端采用 IMU 预积分方法，对数据进行测量预处理，且利用 IMU 与相机的融合数据对系统进行位姿估计，通过在数据集测试，其定位精度要优于 ORB-SLAM，但是系统在运行之初要进行复杂的初始

化是该系统的缺点所在。Qin 等人^[32]基于单目相机并结合 IMU 设计了 VINS-Mono 系统,该系统采用了紧耦合的方式^[33],通过联合点特征与 IMU 数据进行位姿估计,加入闭环检测线程^[34],对机器人位姿重定位,使系统的定位精度达到了很高的水平,同时提升了系统的鲁棒性。Sun 等人^[35]提出一种基于滤波器的立体视觉惯性里程测量法,该方法使用了立体多态约束卡尔曼滤波器(S-MSCKF),计算成本与先进的单目解决方案相媲美,鲁棒性得到显著提高。Qin^[36]于 2019 年对 VINS-Mono 进行了拓展升级,提出了 VINS-Fusion 系统,该系统增加了对双目相机的支持,并融入了 GPS 定位^[37-38]。Campos 等人^[39]在 2021 年发布了开源 ORB-SALM3 系统,该系统为首个能够使用单目、立体和 RGB-D 相机,利用针孔和鱼镜头模型执行视觉、视觉惯性与多地图的 SLAM 系统,紧密集成了 IMU,在 IMU 初始化期间能够完全依赖于最大后验估计,在各种场景中都能保证良好的鲁棒性。

1.2.3 点线视觉惯性 SLAM 研究现状

视觉 SLAM 系统经过数十年的发展,已逐渐趋于稳定。然而,这些系统大多都是基于点的提取和跟踪来估计运动姿态,这一方法在暗纹理环境中会面临极大挑战,因为缺乏丰富的特征点会导致特征信息提取不足,进而降低了系统的定位精度。为了克服这一缺陷,研究人员们将目光转向了线特征。考虑将线特征引入基于点特征的 SLAM 系统中。相比于点特征,线特征包含更丰富的信息。在缺乏点特征的人造环境中,线特征能为系统提供更多的建图信息,从而提高系统的鲁棒性。这一新的方法不仅可以增强系统对低纹理环境的适应能力,还能提高定位的准确性和可靠性。

Grompone Von Gioi 等人^[40]提出了线段检测器(Line Segment Detector, LSD),利用梯度方向值对环境中的线段进行高效检测,是目前线特征检测采取的主流方法之一。2015 年,Zhang 等人^[41]将直线作为特征观察,利用立体 SLAM 构建了三维线性地图,通过不同的方法来参数化空间中的直线,并使用相机新观察到的线进行初始化和 3D 投影。Pumarola 等人^[42]在 ORB-SLAM 算法的基础上加入线特征,提出实时点线视觉 SLAM 算法,该算法使用了新的初始化策略,可以仅通过检测三个连续帧中的线对应关系来完成初始化。He 等人^[43]在 VINS-Mono 的系统框架上作了修改,提出了一种基于点线特征融合的紧耦合视觉里程

计(PL-VIO), 该系统是使用 OpenCV 中的 LSD 线段提取算法来检测线特征, 但是因为它具有高的计算成本, 严重影响了 PL-VIO 系统的性能。Zhao 等人^[44]提出了一种实现线辅助 VO/VSLAM 的精确姿态跟踪方法, 通过良好线条切割, 从每条三维线中提取信息量最大的子段, 用于姿态优化计算, 以提高姿态估计精度。Gomez-Ojeda 等人^[45]于 2019 年在 ORB-SLAM2 系统框架的基础上提出了单目视觉 SLAM 系统 PL-SLAM, 该算法结合了周围环境线的特征, 将线条特征融入到地图初始化过程中。同年, Yang 等人^[46]提出了紧密耦合的 VINS 系统, 此系统利用点和线的退化运动分析三维线的三角测量, 以提高系统的定位精度, 但是该方法的缺陷在于线地标不包括在状态向量中。Zou 等人^[47]提出一种新颖的 StructVIO 方法, 该方法采用人造环境中的线特征结构规律性, 并使用亚特兰大世界模型进行描述, 使系统更精确和稳定, 但是该方法没有考虑非结构线特征。Wen 等人^[48]基于点线特征提出了 PLS-VIO 系统, 通过改进匹配过滤策略, 定位精度相对于传统的算法有一定提升。方琪等人^[49]将线段分组方法加入到 SLAM 系统中, 对潜在的同方向线段进行合并^[50], 得到高质量的长线段, 但是该方法存在线段误匹配率, 对系统的定位精度产生了影响。刘建军等人^[51]提出了基于点线特征的快速单目惯性 SLAM 算法, 但是对系统的定位精度提升的不够明显, 付煜等人^[52]用点线融合的方法提取了丰富的线特征, 但是提取的线特征质量差。Jia 等人^[53]提出的一种基于点线特征的视觉里程计和全球卫星导航系统多点定位技术, 通过引入边缘片段检测的线特征算法来提高弱纹理下姿态估计的准确性。

1.3 本文研究内容及组织结构

本文开展了基于点线融合特征的视觉惯性 SLAM 算法的研究, 目的是解决传统 SLAM 算法在暗纹理等复杂环境中定位不佳的问题。本文内容共分为 6 个章节, 各章节内容安排如下:

第 1 章 绪论。首先对本论文所研究的课题背景及其意义进行了阐述与分析, 紧接着分别分析了视觉 SLAM、视觉惯性 SLAM 以及基于点线特征的视觉惯性 SLAM 的研究现状。最后介绍了本文的研究内容及组织结构。

第 2 章 视觉惯性 SLAM 算法基础理论。首先分析了旋转与平移以及四元数的数学模型, 并用其表示机器人的刚体运动。此外, 还对描述机器人运动过程

进行了数学推导。接着对 SLAM 中常用的李群和李代数进行了说明。然后深入研究了相机的成像原理，并对相机产生的径向畸变与切向畸变问题进行了分析，解释了其产生的原因，并给出了矫正模型。最后对 IMU 的数学模型进行了分析。

第 3 章 基于自适应点线特征提取的视觉惯性 SLAM 算法。首先对 LSD 线特征提取算法及线特征匹配所需的 LBD 描述子原理进行了分析。然后给出了一种自适应线特征提取方法，通过设定线特征提取自适应阈值，剔除多而过短的线段，避免因线段冗余而影响线特征的匹配。接着通过短线拼接的方法将剩余的短线合并成长线，提高线段的利用率。最后用 Plücker 坐标和正交表示法对线特征线特征进行了数学描述。

第 4 章 基于动态点线耦合的单目视觉惯性 SLAM 算法。本章首先搭建了 SLAM 系统整体的结构框架，并对系统框架的各个模块的功能进行了详细分析。然后定义了一种新的点线耦合特征，并对该特征信息进行提取和构建点线耦合残差模型。最后将点线耦合残差模型、点线残差、闭环信息以及前端的预处理信息集成到后端的滑动窗口中，构造了最小化成本函数。

第 5 章 基于 ROS 的视觉惯性 SLAM 算法仿真与平台验证。首先分析了本文算法进行仿真实验时所要用到的 EuRoC 数据集以及本文算法的评估指标（绝对轨迹误差）。然后对本文中的点线特征提取算法在不同的场景中进行了实验展示。在数据集中整体运行了本文算法，得到算法在不同数据集中的运行轨迹，并采用 EVO 工具对其轨迹进行了评估。通过与两种经典算法进行完整比较和详细分析，评估了本文算法的性能。最后对实物平台的 IMU 相机进行了标定并记录标定结果，并将本文算法导入实物平台进行实验验证。

第 6 章 总结与展望。主要目的是对研究内容和成果进行归纳与总结，同时探讨本文的不足之处，并说明在今后对本文会做哪些改进。

第 2 章 视觉惯性 SLAM 算法基础理论分析

视觉惯性 SLAM 算法主要依赖相机与 IMU 结合的方式，使机器人能够在未知环境中自主完成自身运动位姿估计和地图构建。众所周知，算法的设计与实现需要有极强的数学理论及模型作为支撑。因此，在本章中，将详细探讨视觉惯性 SLAM 算法所涉及的相关数学基础理论^[54]。

2.1 机器人的刚体运动

2.1.1 相机旋转与平移

机器人在未知环境中进行位姿估计时，依靠的是通过坐标系来判断自身所处的位置。因此，建立坐标系是至关重要的。在 SLAM 中，需要建立三种坐标系，分别为世界坐标系 W 、相机坐标系 C 以及 IMU 坐标系 I 。在跟踪和定位时，这三种坐标系会经历一个相互变换的过程，以 W 为参照，对 C 与 I 中的数据进行融合，求解机器人在空间中的位姿。其中，把 W 与 C 之间的变换定义为欧式变换，两者的变换关系如下图。

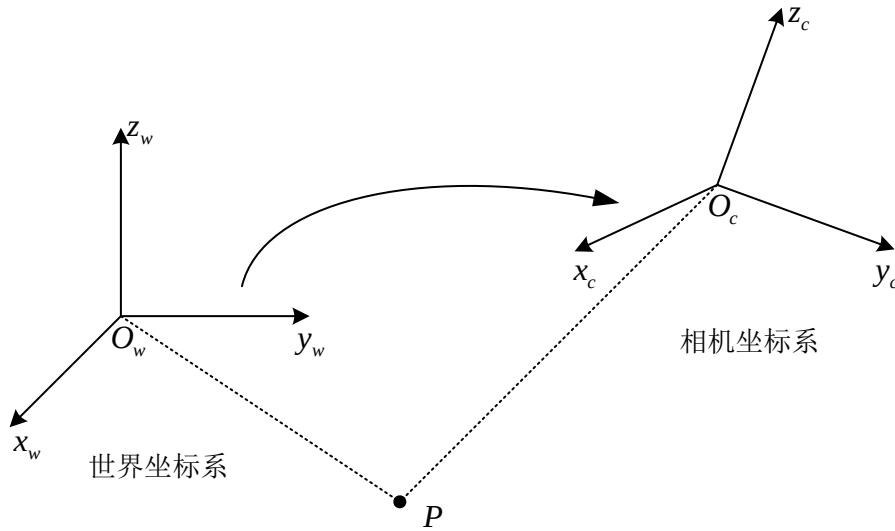


图 2-1 坐标变换示意图

欧式变换涉及到旋转与平移两个过程。在旋转过程中，假设两不同坐标系下的一组单位正交基分别为 (e_1, e_2, e_3) 、 (e'_1, e'_2, e'_3) ，空间向量 a 在这两坐标系下对应的坐标分别为 $[a_1, a_2, a_3]^T$ 与 $[a'_1, a'_2, a'_3]^T$ ，此向量并没有随着旋转而产生改变，则根据坐标定义有如下关系：

$$(e_1, e_2, e_3) \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = (e'_1, e'_2, e'_3) \begin{bmatrix} a'_1 \\ a'_2 \\ a'_3 \end{bmatrix} \quad (2-1)$$

为了体现两坐标系的关系，需要对上式进行处理，把公式(2-1)等号两边同时左乘 $[e_1^T, e_2^T, e_3^T]^T$ ，得到如下等式：

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} e_1^T e'_1 & e_1^T e'_2 & e_1^T e'_3 \\ e_2^T e'_1 & e_2^T e'_2 & e_2^T e'_3 \\ e_3^T e'_1 & e_3^T e'_2 & e_3^T e'_3 \end{bmatrix} \begin{bmatrix} a'_1 \\ a'_2 \\ a'_3 \end{bmatrix} \stackrel{def}{=} \mathbf{R} \mathbf{a}' \quad (2-2)$$

将上式中间的 3×3 矩阵 $\mathbf{R}$ 称为旋转矩阵，由两组正交基之间的内积组成，代表了相机旋转信息。由于正交基向量长度为1，那么 $|\mathbf{R}|=1$ ，根据这个特殊性质，可定义 n 维旋转矩阵的集合为：

$$SO(n) = \{ \mathbf{R} = \mathbb{R}^{n \times n} \mid \mathbf{R} \mathbf{R}^T = \mathbf{I}, \det(\mathbf{R}) = 1 \} \quad (2-3)$$

式中： $SO(n)$ 表示特殊正交群，表示 n 维空间的旋转。

除了旋转，欧式变换还有平移。空间中的向量 $\mathbf{a}$ 经历一次旋转和一次平移 $\mathbf{t}$ 之后，得到 $\mathbf{a}'$ ，将旋转与平移表示在一起，有：

$$\mathbf{a}' = \mathbf{R} \mathbf{a} + \mathbf{t} \quad (2-4)$$

式中： $\mathbf{t}$ 是平移向量。通过公式(2-4)可以完整描述一个欧式空间坐标变换。

2.1.2 四元数

由于旋转矩阵中有9个元素，这在描述3个自由度的旋转时显得冗余。但是根据数学知识可知，在复平面上的向量可以用复数集表示。复平面的旋转可以使用复数的乘法运算来描述。若把一个复向量乘 i ，则等同于把此复向量逆时针旋转了90度。因此，机器人在空间中的运动还可以用类似与复数形式的四元数表示。设四元数为 $\mathbf{q}$ ，由1个实部与3个虚部构成，其表达式为：

$$\mathbf{q} = q_0 + q_1 \mathbf{i} + q_2 \mathbf{j} + q_3 \mathbf{k} \quad (2-5)$$

式中： $\mathbf{i}, \mathbf{j}, \mathbf{k}$ 表示 $\mathbf{q}$ 的3个虚部，三者满足以下关系：

$$\begin{cases} \mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = -1 \\ \mathbf{ij} = \mathbf{k}, \mathbf{ji} = -\mathbf{k} \\ \mathbf{jk} = \mathbf{i}, \mathbf{kj} = -\mathbf{i} \\ \mathbf{ki} = \mathbf{j}, \mathbf{ik} = -\mathbf{j} \end{cases} \quad (2-6)$$

有时为了表示的便捷，用如下式子表达：

$$\mathbf{q} = [s, \mathbf{v}]^T, s = q_0 \in \mathbb{R}, \mathbf{v} = [q_1, q_2, q_3]^T \in \mathbb{R}^3 \quad (2-7)$$

式中： s 与 $\mathbf{v}$ 分别代表 $\mathbf{q}$ 的实部和虚部。

四元数可以表示一个空间点的旋转。假设有一个空间点 $\mathbf{p} = [x, y, z] \in \mathbb{R}^3$ 经过一个确定的单位四元数 $\mathbf{q}$ 旋转后变为 $\mathbf{p}'$ ，可以先用四元数的形式来表示这个空间点 $\mathbf{p}$ ，如下式：

$$\mathbf{p} = [0, x, y, z]^T = [0, \mathbf{v}]^T \quad (2-8)$$

这样处理是为了将四元数的三个虚部与空间坐标轴一一对应，那么 $\mathbf{p}'$ 就可以表示为：

$$\mathbf{p}' = \mathbf{q}\mathbf{p}\mathbf{q}^{-1} \quad (2-9)$$

上式中得到 $\mathbf{p}'$ 的虚部即为空间点 $\mathbf{p}$ 旋转后的点坐标。

机器人的空间旋转表示方法多种多样，具体用哪种形式表达需要根据实际应用需求来选择，接下来本文给出旋转矩阵与四元数的转换关系。

对于公式(2-5)表达的四元数，与之对应的旋转矩阵 $\mathbf{R}$ 为：

$$\mathbf{R} = \begin{bmatrix} 1-2q_2^2-2q_3^2 & 2q_1q_2+2q_0q_3 & 2q_1q_3-2q_0q_2 \\ 2q_1q_2-2q_0q_3 & 1-2q_1^2-2q_3^2 & 2q_2q_3-2q_0q_1 \\ 2q_1q_3+2q_0q_2 & 2q_2q_3+2q_0q_1 & 1-2q_1^2-2q_2^2 \end{bmatrix} \quad (2-10)$$

对上述公式进行简写，简写后的表达式为：

$$\mathbf{R} = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix} \quad (2-11)$$

根据公式(2-11)，可把四元数 $\mathbf{q}$ 表示为：

$$\mathbf{q} = \frac{\sqrt{\text{tr}(\mathbf{R})+1}}{2} + \frac{p_{23}-p_{32}}{4q_0} \mathbf{i} + \frac{p_{31}-p_{13}}{4q_0} \mathbf{j} + \frac{p_{12}-p_{21}}{4q_0} \mathbf{k} \quad (2-12)$$

式中： $\text{tr}(\mathbf{R})$ 为 $\mathbf{R}$ 的迹。

若旋转时绕向量 $\mathbf{n} = [n_x, n_y, n_z]^T$ 旋转了 θ ，则与之对应的四元数 $\mathbf{q}$ 为：

$$\mathbf{q} = [n_x, n_y, n_z]^T \cdot \sin \frac{\theta}{2} \quad (2-13)$$

2.2 李群和李代数

在 SLAM 系统中，后端构建的优化函数需要用到旋转矩阵作为优化参数，但是旋转矩阵有正交条件，会引入额外的约束，导致计算变得复杂，这时候就需要引入李群李代数，将约束问题转化为无约束问题，简化求解过程。下面首先阐述关于群的概念。

在数学中，群是指一种具有封闭性、结合律、存在单位元和每个元素存在逆元的二元运算的代数结构。群的集合通常用符号 $G(G \neq \emptyset)$ 表示，二元运算记为“ $\cdot$ ”，其运算未必是通常意义下数的乘法，其结果称为积。群的具体运算满足以下条件：

- (1) 封闭性： $\forall a, b \in G, a \cdot b \in G$
- (2) 结合律： $\forall a, b, c \in G, (a \cdot b) \cdot c = a \cdot (b \cdot c)$
- (3) 单位元： $\exists a_0 \in G, s.t. \forall a \in G, a_0 \cdot a = a \cdot a_0 = a$
- (4) 逆元： $\forall a \in G, \exists a^{-1} \in G, s.t. a \cdot a^{-1} = a_0$

在群概念的基础上，衍生出了李群的概念，其实李群就是一种特殊的群，是具有光滑性质的群。机器人在空间中是做连续的运动，因此其运动可以看作是李群，SLAM 中常见的李群是 $SO(3)$ 与 $SE(3)$ ，在机器人位姿估计时二者是必不可少的。公式(2-14)为两个群的表示。

$$\begin{cases} SO(3) = \{ \mathbf{R} = \mathbb{R}^{3 \times 3} \mid \mathbf{R}\mathbf{R}^T = \mathbf{I}, \det(\mathbf{R}) = 1 \} \\ SE(3) = \left\{ T = \begin{bmatrix} \mathbf{R} & t \\ \mathbf{0}^T & 1 \end{bmatrix} \in \mathbb{R}^{4 \times 4} \mid \mathbf{R} \in SO(3), t \in \mathbb{R}^3 \right\} \end{cases} \quad (2-14)$$

任意的一个李群都有一个李代数对应，李代数是李群局部性质的反映，准确的表述为李群元素在单位元的正切空间。其定义如下：

给定数域 F 上的线性空间 V ，在 V 定义一个二元运算 $[\cdot, \cdot]$ ，对于 $\forall \mathbf{X}, \mathbf{Y} \in V$ ，运算结果记为 $[\mathbf{X}, \mathbf{Y}]$ ，称为李括号。若运算满足以下几点性质，则称 $(V, F, [\cdot, \cdot])$ 为一个李代数。

- (1) 封闭性： $\forall \mathbf{X}, \mathbf{Y} \in V, [\mathbf{X}, \mathbf{Y}] \in V$
- (2) 双线性： $\forall \mathbf{X}, \mathbf{Y}, \mathbf{Z} \in V, a, b \in F$ ，则：

$$[a\mathbf{X} + b\mathbf{Y}, \mathbf{Z}] = a[\mathbf{X}, \mathbf{Z}] + b[\mathbf{Y}, \mathbf{Z}], [\mathbf{Z}, a\mathbf{X} + b\mathbf{Y}] = a[\mathbf{Z}, \mathbf{X}] + b[\mathbf{Z}, \mathbf{Y}]$$

(3) 反对称性: $\forall \mathbf{X}, \mathbf{Y} \in \mathbf{V}, [\mathbf{X}, \mathbf{Y}] + [\mathbf{Y}, \mathbf{X}] = 0$

(4) Jacobi 结合性: $\forall \mathbf{X}, \mathbf{Y}, \mathbf{Z} \in \mathbf{V}, [\mathbf{X}, [\mathbf{Y}, \mathbf{Z}]] + [\mathbf{Z}, [\mathbf{X}, \mathbf{Y}]] + [\mathbf{Y}, [\mathbf{Z}, \mathbf{X}]] = 0$

前文提到的李群 $SO(3)$ ，其对应的李代数 $so(3)$ 为定义在 $\mathbb{R}^3$ 上的向量，用 ϕ 表示，每个 ϕ 都对应一个反对称矩阵：

$$\Phi = \phi^\wedge = \begin{bmatrix} 0 & -\phi_3 & \phi_2 \\ \phi_3 & 0 & -\phi_1 \\ -\phi_2 & \phi_1 & 0 \end{bmatrix} \in \mathbb{R}^{3 \times 3} \quad (2-15)$$

式中： $\phi^\wedge$ 是由三维向量 $\phi = [\phi_1, \phi_2, \phi_3]^T$ 变换成的反对称矩阵。定义 ϕ_1 与 ϕ_2 的李括号为：

$$[\phi_1, \phi_2] = (\Phi_1 \Phi_2 - \Phi_2 \Phi_1)^\vee \quad (2-16)$$

由公式(2-15)知， $so(3)$ 中的元素都为三维向量，或者说是三维反对称矩阵，则 $so(3)$ 的集合可以表示为：

$$so(3) = \{\phi \in \mathbb{R}^3, \Phi = \phi^\wedge \in \mathbb{R}^{3 \times 3}\} \quad (2-17)$$

对于李群 $SE(3)$ 的李代数 $se(3)$ ，其表现形式与 $so(3)$ 类似， $se(3)$ 位于 $\mathbb{R}^6$ 空间中，本文在这不过多赘述，直接给出 $se(3)$ 的集合：

$$se(3) = \left\{ \xi = \begin{bmatrix} \rho \\ \phi \end{bmatrix} \in \mathbb{R}^6, \rho \in \mathbb{R}^3, \phi \in so(3), \xi^\wedge = \begin{bmatrix} \phi^\wedge & \rho \\ \mathbf{0}^T & 0 \end{bmatrix} \in \mathbb{R}^{4 \times 4} \right\} \quad (2-18)$$

式中： ξ 为六维向量，由 $se(3)$ 元素组成， ρ 为前三维平移向量， ϕ 为后三维旋转向量。

2.3 相机数学理论

在视觉 SLAM 系统中，机器人是依赖于相机拍摄的每帧图片，以实现对自身位姿进行准确估计，因此，相机对于视觉 SLAM 系统至关重要。相机成像遵循小孔成像原理，但相机的镜头由透镜制成，在成像时会产生畸变，从而影响成像效果，所以需要通过技术手段对相机进行畸变矫正以获得正常的图像。本节将对相机成像原理以及纠正图像畸变过程作详细分析。

2.3.1 小孔成像原理

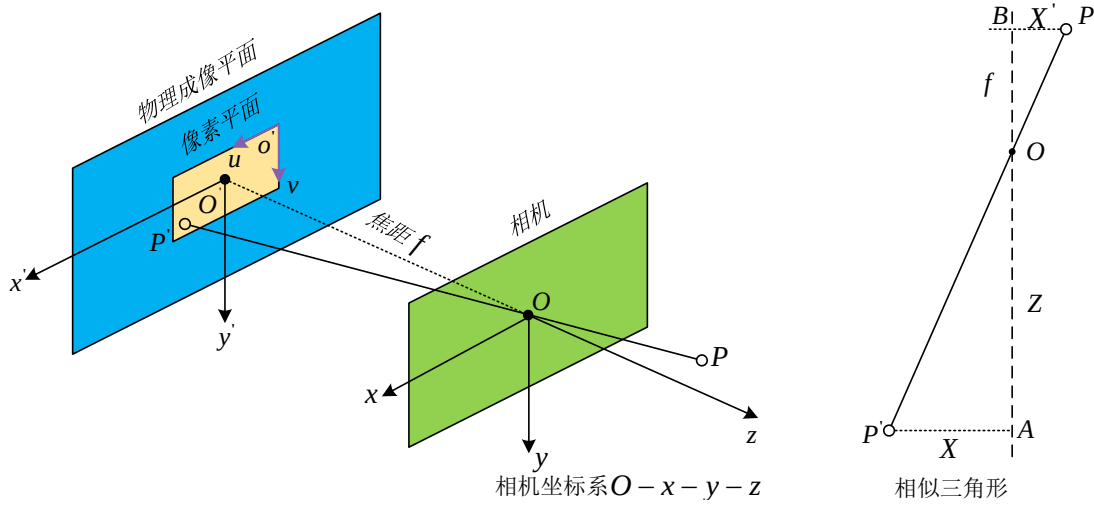


图 2-2 小孔成像原理图

图 2-2 为小孔成像原理图，在图中，有三个平面，分别相机平面、物理成像平面、像素平面（图中阴影部分），在对应的三个平面中分别建立相机坐标系 $O-x-y-z$ 、图像坐标系 $O'-x'-y'$ 、像素坐标系 $o-u-v$ 。相机坐标系的原点为相机的光心 O ， x 轴方向水平向左， y 轴方向竖直向下， z 轴为相机的主光轴；图像坐标系是以投影在物理成像平面的相中点 O' 为坐标原点， x' 轴、 y' 轴与 x 轴、 y 轴同方向；而像素坐标系的原点 o 定义为像素平面的右上角，其 u 轴、 v 轴与 x 轴、 y 轴同方向； O 与 O' 之间的距离定义为焦距 f 。

假设一点 P 在三维场景中的坐标为 $[X, Y, Z]^T$ ，其通过相机光心 O 投射到物理成像平面的点为 P' ，坐标为 $[X', Y', Z']^T$ ，根据三角形相似，可以得到关于点 P 与点 P' 的关系式：

$$\frac{Z}{f} = -\frac{X}{X'} = -\frac{Y}{Y'} \quad (2-19)$$

式中：负号表示像是倒立的，而实际上相机拍摄的图像并非倒立，是通过软件算法翻转所成正立的像，为了使模型更贴切实际，把物理成像平面对称地置于相机前方，与空间点一起置于相机坐标系同侧，如图 2-3，则公式(2-19)可简化为：

$$\frac{Z}{f} = \frac{X}{X'} = \frac{Y}{Y'} \quad (2-20)$$

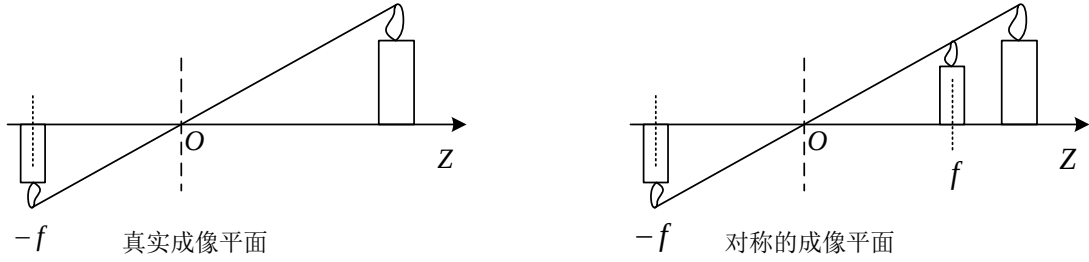


图 2-3 真实、对称的成像图示

对公式(2-20)进行整理，得：

$$\begin{cases} X' = f \frac{X}{Z} \\ Y' = f \frac{Y}{Z} \end{cases} \quad (2-21)$$

于是就得到了三维场景中点 P 与物理成像平面中点 P' 之间的坐标变换关系。但是相机最终获取的每一帧图像其实是一个个像素点，是由数字组成的矩阵，所以需要在相机成像平面对所成的像进行采样和量化，这就需要借助图 2-2 中得像素坐标系来处理。在像素坐标系中，点 P' 的坐标可以表示为 $[u, v]^T$ 。通过对成像平面进行缩放和原点的平移，可以将其转换为像素坐标系。设像素坐标在 u 轴与 v 轴上分别缩放了 α 倍和 β 倍，且原点平移了 $[u_x, v_y]^T$ ，则 P' 在成像平面的坐标与在像素平面的坐标的转换关系就可以表示为：

$$\begin{cases} u = \alpha X' + u_x \\ v = \beta Y' + v_y \end{cases} \quad (2-22)$$

联立公式(2-21)与公式(2-22)，并将 αf 和 βf 记为 f_x 和 f_y ，得：

$$\begin{cases} u = f_x \frac{X}{Z} + u_x \\ v = f_y \frac{Y}{Z} + v_y \end{cases} \quad (2-23)$$

将公式(2-23)表示成矩阵的形式：

$$Z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_x \\ 0 & f_y & v_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \mathbf{K} \mathbf{P} \quad (2-24)$$

上式中间的 3×3 矩阵一般被称为相机内参数(Camera Intrinsics)矩阵 $\mathbf{K}$ 。相机的内参由其硬件结构决定，不会发生改变。通常在出厂时会提供这些参数，如果厂商未提供，也可以通过标定来确定内参。因此，在使用过程中，这些内

参可以视为固定参数。

在公式(2-24)中，点 P 的坐标是在相机坐标系下获得的，由于相机的运动，所以点 P 的相机坐标是它的世界坐标 P_w 根据相机的当前位姿变换到相机坐标系的。用旋转矩阵 R 和平移向量 t 来描述相机的位姿，则像素坐标与世界坐标的关系就可以表示为：

$$ZP_{uv} = Z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K(RP_w + t) = KTP_w \quad (2-25)$$

式中： R ， t 称为相机的外参数，两者会随着相机的运动而发生改变，代表着相机的真实运动轨迹。准确计算相机的外参数，尽可能还原相机的真实运动轨迹，是 SLAM 的主要任务之一。

公式(2-25)描述的就是三维场景中的一点从世界坐标系转换到相机坐标系，再进行归一化的过程，所谓的归一化，就是将点的三维向量最后一位数值去除的处理方式，其具体公式如下：

$$(RP_w + t) = \underbrace{[X, Y, Z]^T}_{\text{相机坐标}} \rightarrow \underbrace{[X/Z, Y/Z, 1]^T}_{\text{归一化坐标}} \quad (2-26)$$

2.3.2 图像畸变矫正

在实际应用中，相机制造厂商为了使用户获取更高质量图片，通常在相机镜头前方加上透镜，但是透镜会使光线的传播产生影响，使实际获得的图像产生畸变。此类畸变根据产生的原因可分为径向畸变和切向畸变。径向畸变是由于透镜自身的形状引起的，当光线通过透镜后，形状就会发生变化。此类畸变又分为桶形畸变和枕形畸变两类。桶形畸变是由于图像放大率随着光轴之间的距离增加而减小，枕形畸变则相反，如图 2-4 所示。而切向畸变的产生是成像平面不平行与透镜所导致的，如图 2-5 所示。

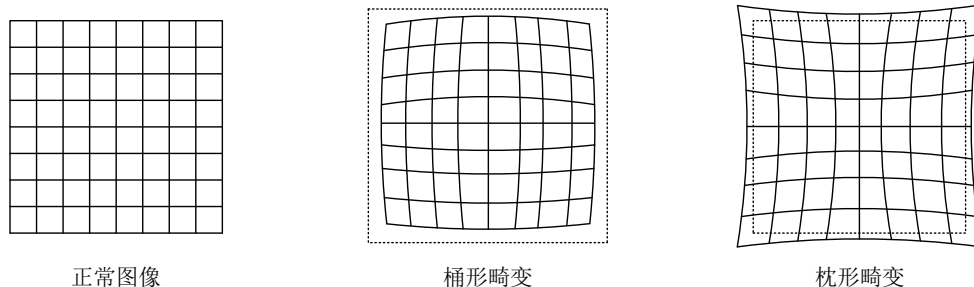


图 2-4 径向畸变两种类型

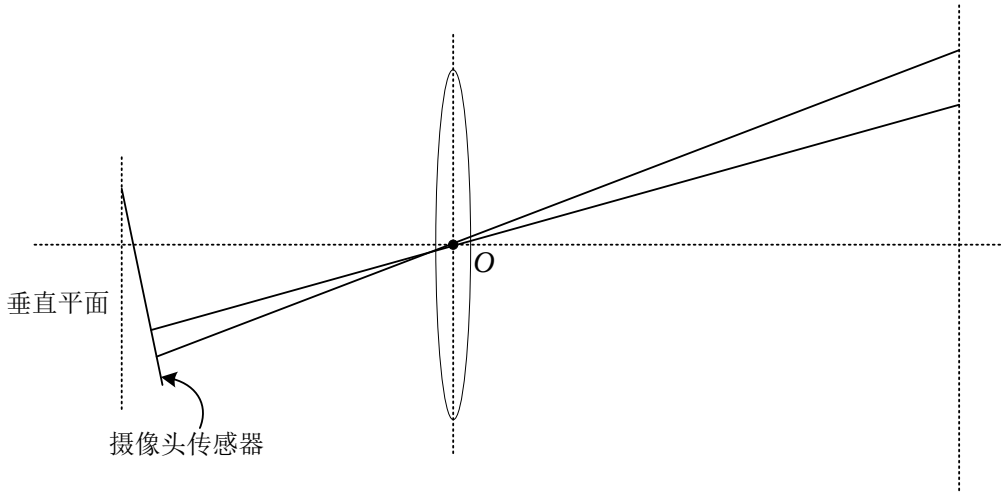


图 2-5 切向畸变来源示意图

解决畸变的问题，需要为畸变问题构建数学模型。径向畸变可以看作是坐标点距离原点的长度发生了变化，如图 2-4 所示，而切向畸变可以看作是水平夹角产生了变化，如图 2-5 所示。对于径向畸变，可以用多项式表示为：

$$\begin{cases} x_{distorted} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \\ y_{distorted} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \end{cases} \quad (2-27)$$

式中： $[x, y]^T$ 与 $[x_{distorted}, y_{distorted}]^T$ 分别表示畸变前和畸变后的成像点在归一化平面坐标。 r 为成像点与坐标系原点之间的距离。对于切向畸变，可以引入参数 p_1 与 p_2 ，用另外的多项式来表示：

$$\begin{cases} x_{distorted} = x + 2p_1 xy + p_2(r^2 + 2x^2) \\ y_{distorted} = y + p_1(r^2 + 2y^2) + 2p_2 xy \end{cases} \quad (2-28)$$

联立公式(2-27)与(2-28)，对两种畸变归一化处理，得：

$$\begin{cases} x_{distorted} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_1 xy + p_2(r^2 + 2x^2) \\ y_{distorted} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + p_1(r^2 + 2y^2) + 2p_2 xy \end{cases} \quad (2-29)$$

将畸变后的点用相机的内参矩阵投射到像素平面上，就可以求解出相机坐标系中的一点 P 在像素坐标系中的正确位置：

$$\begin{cases} u = f_x x_{distorted} + c_x \\ v = f_y y_{distorted} + c_y \end{cases} \quad (2-30)$$

2.4 IMU 数学模型

IMU 是视觉惯性 SLAM 系统中必要的组成部件，包含加速度计和陀螺仪，用于测量机器人的加速度（包含重力加速度）与角速度。加速度能通过积分与

二重积分的运算得到速度、位移，即加速度计可以间接测量机器人的速度与位置；而陀螺仪测量的角速度反映的是机器的运动方向。

然而，IMU 也存在不可忽视的误差，在实际应用时，会产生高斯白噪声和零偏，因此在建模时需要考虑这些误差因素，对 IMU 建立模型如下：

$$\begin{cases} \hat{\omega}_w^b(t) = \omega_w^b(t) + b_g(t) + \eta_g(t) \\ \hat{a}_b(t) = R_w^b(a_w(t) - g_w) + b_a(t) + \eta_a(t) \end{cases} \quad (2-31)$$

式中： $\hat{\omega}_w^b(t)$ ， $\hat{a}_b(t)$ 为陀螺仪与加速度计的测量值， $\omega_w^b(t)$ ， $a_b(t)$ 为陀螺仪与加速度计的真实值， g_w 为重力加速度， R_w^b 是世界坐标系与 IMU 坐标系之间的旋转系数， $b_g(t)$ ， $b_a(t)$ 为陀螺仪与加速度计的零偏值， $\eta_g(t)$ ， $\eta_a(t)$ 为器件的高斯白噪声。

由 IMU 的加速度与角速度，可以求解机器人的运动状态，那么在某时刻 t ，机器人的运动状态方程可以写为：

$$x_t = [p_{b_t}^w, v_{b_t}^w, q_{b_t}^w, b_{a_t}, b_{g_t}] \quad (2-32)$$

式中： $p_{b_t}^w$ ， $v_{b_t}^w$ ， $q_{b_t}^w$ 表示在 t 时刻时，机器人的位置、速度和旋转，对公式(2-31)进一步关于对 t 求导数，得：

$$\begin{cases} \dot{p}_{b_t}^w = v_{b_t}^w \\ \dot{v}_{b_t}^w = a_{b_t}^w \\ \dot{q}_{b_t}^w = q_{b_t}^w \otimes [0 \quad 0.5\omega_t]^T \end{cases} \quad (2-33)$$

式中： $a_{b_t}^w$ 表示 t 时刻 IMU 坐标系到世界坐标系得加速度， ω_t 为 t 时刻机器人的角速度， $\otimes$ 为四元数运算符号。

2.5 本章小结

本章分析了视觉惯性 SLAM 算法相关的数学理论知识。在第一节中分析了如何采用旋转与平移、四元素表示机器人的刚体运动，并且对机器人运动过程进行了数学推导。第二节对 SLAM 中常用的李群和李代数进行了说明。第三节详细解释了相机的成像原理，并对相机产生的径向畸变与切向畸变的矫正模型作了分析。第四节给出了 IMU 的数学模型并分析了其原理。本章为后文的研究提供了理论基础，对本文算法的结构设计具有非常重要的影响。

第3章 基于自适应点线特征提取的视觉惯性 SLAM 算法

在视觉惯性 SLAM 系统中，视觉前端的特征信息对机器人的位姿估计有至关重要的作用，从而系统的定位精度会直接受到特征信息质量的影响，传统的线特征的提取算法集成在计算机视觉库 OpenCV 中，在暗纹理场景时提取质量差，因此本章给出基于动态点线耦合的单目视觉惯性 SLAM 算法，并验证其有效性。

3.1 LSD 线特征提取

3.1.1 线特征提取原理

与传统算法相比，LSD 算法计算速度快，能在较短线性时间内获得高精度的直线检测结果，其核心思路是对同一梯度相近的像素点进行合并，从而得到线段支持域 R ，每个线段支持域可能对应一条线段，需要构建最小外接矩形来确定线段支持域是否为线段。LSD 线段支持域示意图如图 3-1 所示，图中每个红色区域是梯度相近的像素点，构成了一个线段支持域。

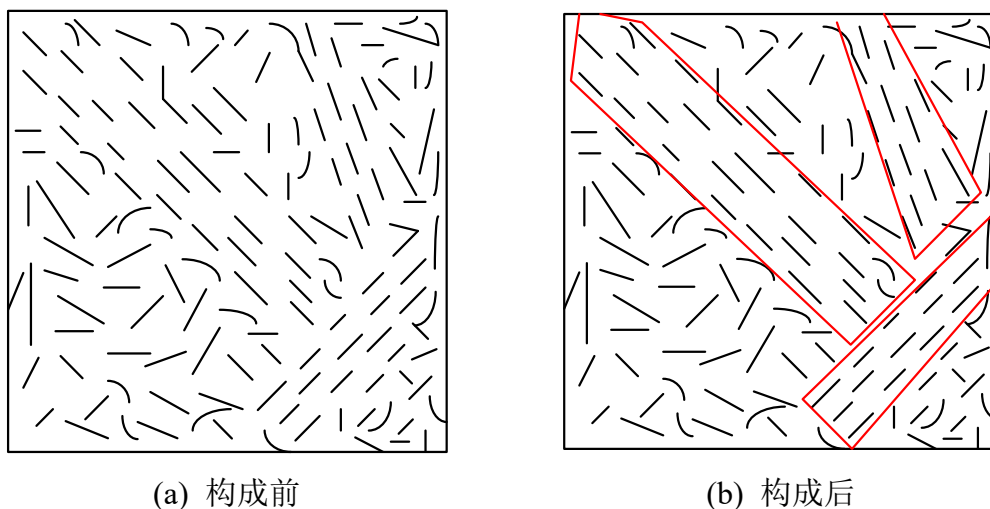


图 3-1 LSD 线段支持域示意图

LSD 线特征提取算法流程可分为以下六个步骤，其步骤如下文所述。

(1) 图像缩放：对原图按一定比例进行缩放，排除因原图像边缘出现锯齿状而产生断裂直线段现象。

(2) 梯度计算：采用图像中各像素点的右下四个像素计算梯度，以尽可能

小的计算模板减小其他像素点对梯度计算的影响，其计算公式如下：

$$\begin{cases} g_x(x, y) = \frac{i(x+1, y) + i(x+1, y+1) - i(x, y) - i(x, y+1)}{2} \\ g_y(x, y) = \frac{i(x, y+1) + i(x+1, y+1) - i(x, y) - i(x+1, y)}{2} \\ G(x, y) = \sqrt{g_x^2(x, y) + g_y^2(x, y)} \\ \theta(x, y) = \arctan\left(-\frac{g_x(x, y)}{g_y(x, y)}\right) \end{cases} \quad (3-1)$$

式中： i_0 为像素点灰度值， $G(x, y)$ 为像素点梯度值， $\theta(x, y)$ 为像素点梯度方向。

(3) 梯度排序：取梯度值最大的像素点作为线段检测的种子点。

(4) 梯度阈值设置：设定梯度阈值 u ，将梯度值小于 u 的标记为 USED，大于 u 的标记为 UNUSED，若像素点的梯度值小于 u ，则构建线段支持域时不考虑该像素点。

(5) 线段支持域更新：从排序列表中选取未使用的像素值点(UNUSED)作为种子点，计算其梯度方向与邻域生成的线段支持域的梯度方向容忍值是否小于 τ ，若满足，将被加入到线段支持域内。 τ 的值如公式(3-2)所示。

$$\tau = \frac{\sum_{j \in R} \sin \theta(x, y)}{\sum_{j \in R} \cos \theta(x, y)} \quad (3-2)$$

(6) 矩形估计：将更新后的支持域进行矩形近似计算，以得到一个规整的直线区域，每条直线对应一个矩形区域，矩形区域的中心坐标可表示为式(3-3)。

$$\begin{cases} C_x = \frac{\sum_{j \in R} G(j) \cdot x(j)}{\sum_{j \in R} G(j)} \\ C_y = \frac{\sum_{j \in R} G(j) \cdot y(j)}{\sum_{j \in R} G(j)} \end{cases} \quad (3-3)$$

式中： $G(j)$ 为像素 j 的梯度值， j 代表区域内所有的像素。

矩形的主轴方向被设置为与矩阵 $\mathbf{P}$ 的最小特征值相关联的特征向量的角度。

$$\mathbf{P} = \begin{bmatrix} p^{xx} & p^{xy} \\ p^{xy} & p^{yy} \end{bmatrix} \quad (3-4)$$

式中： p^{xx} 、 p^{xy} 、 p^{yy} 的具体表达形式如式(3-5)至(3-7)所示

$$p^{xx} = \frac{\sum_{j \in R} G(j) \cdot (x(j) - c_x)^2}{\sum_{j \in R} G(j)} \quad (3-5)$$

$$p^{xy} = \frac{\sum_{j \in R} G(j) \cdot (x(j) - c_x) \cdot (y(j) - c_y)}{\sum_{j \in R} G(j)} \quad (3-6)$$

$$p^{yy} = \frac{\sum_{j \in R} G(j) \cdot (y(j) - c_y)^2}{\sum_{j \in R} G(j)} \quad (3-7)$$

图 3-2 为 LSD 算法流程图，展示了 LSD 算法的执行过程。

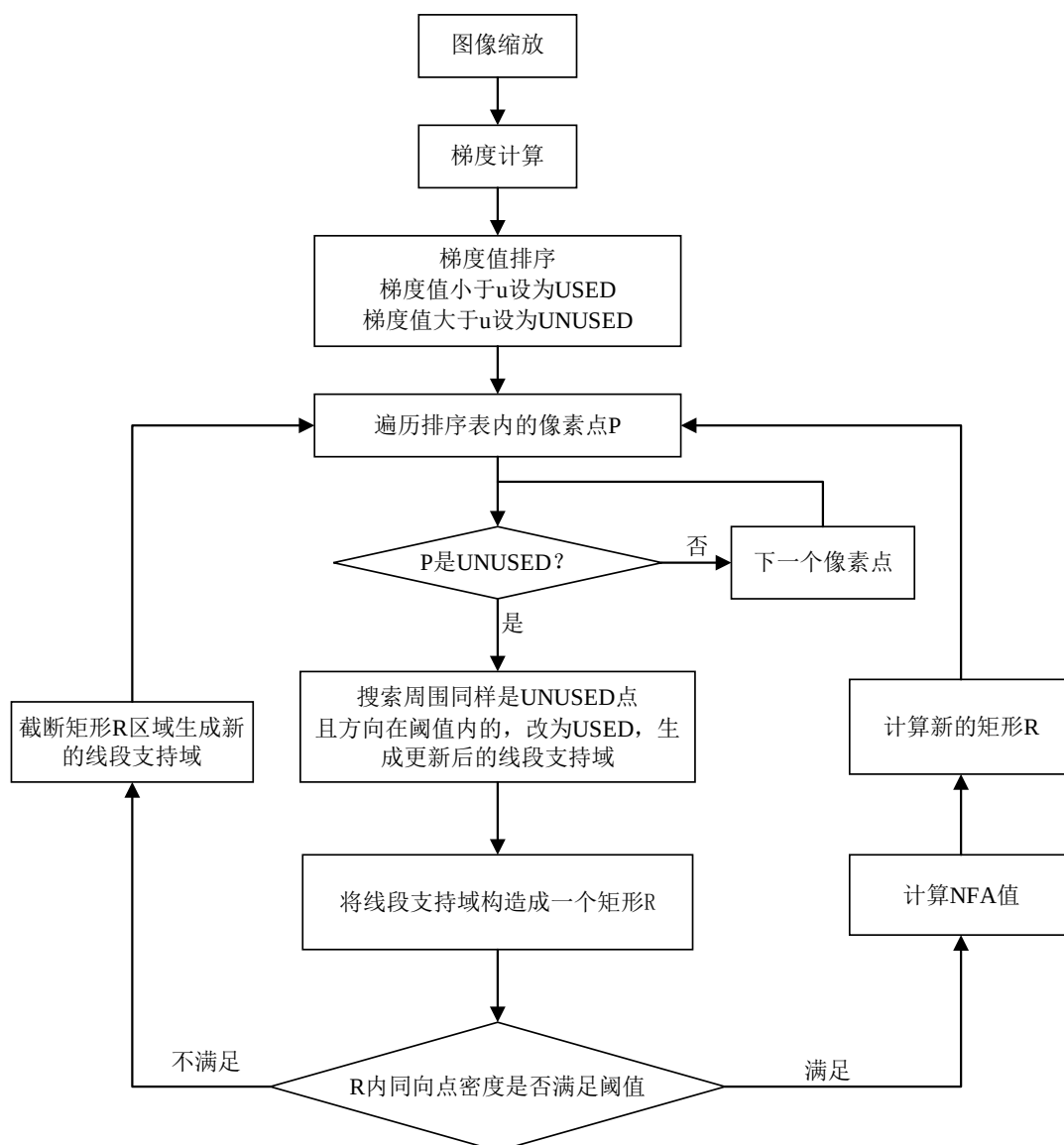


图 3-2 LSD 算法流程图

3.1.2 设定线特征检测的自适应阈值

传统的线段提取算法存在多短和重叠的线段不能被有效剔除的问题，这些

问题会导致视觉传感器对相机位姿的估计不够准确，且难以精确进行全局地图构建。对于点线融合的 SLAM 系统，线特征是对点特征的补充，只需要保留较长的线特征而剔除较多的短线就可以对相机位姿进行精确估计。本文构造了一种改进的线段提取算法，在线特征提取阶段，设定一个提取线段的自适应阈值，这样做的目的是对视觉传感器检测到实际场景的线特征进行保留和剔除，进一步提高相机位姿估计的准确性，从而提升了全局建图的精确度。

定义线段提取的动态阈值为 $L_{\min}$ ，则 $L_{\min}$ 满足如下公式：

$$L_{\min} = \frac{1}{2n + \max(W_m, H_m)} \cdot \min(W_m, H_m) \quad (3-8)$$

式中： n 表示在第 m 帧中线特征的提取数量， W 与 H 分别表示当前帧的宽度与高度。

以上公式是根据每帧图像的宽度、高度以及提取到的线特征数量设计而成，是其优点所在之处。通过修复传统线特征提取算法设置固定阈值的缺陷，可以有效避免提取过少有效线特征或提取多余短线段的问题，从而对线特征合理且高效的利用。

再将提取到的所有线特征定义为一个集合 $\{l_1, l_2, \dots, l_n\}$ ，则线段长度筛选机制遵循以下公式：

$$L_i \geq L_{\min}, i \in \{1, 2, 3, \dots, n\} \quad (3-9)$$

式中： L_i 表示提取到线段的长度，当 L_i 大于或等于阈值 $L_{\min}$ 时，线段被保留下来，反之线段会被剔除。

3.1.3 短线拼接

由于线段越长，其线特征越稳定，更容易在多帧图像中被重复检测到，且数量越多，对定位精度的贡献越大。因此，在线段经过一次筛选后，可对线段进一步处理，将方向相同的短线拼接成长线段，其示意图如图 3-3 所示。具体过程分为以下几个步骤。

(1) 将经过一次筛选后保留下来的线段按其长度进行降序排列，得到线段的集合 $\{L_1, L_2, \dots, L_n\}$ ，其中 L_n 表示第 n 条线段。因为线段 L_1 最长，根据较长线特征通常来源于具有连续强梯度的图像区域，故从线段 L_1 开始进行线段分组。

由于最长线段为 L_1 ，则剩下的线段分组的集合为 $L = \{L_2, L_3, \dots, L_n\}$ 。

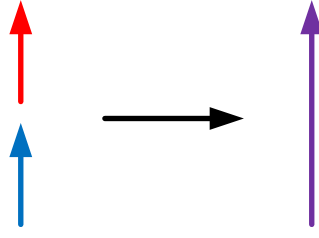


图 3-3 短线拼接示意图

(2) 按角度对线段集合 L 进行筛选，得到新的线段集合 L_r ：

$$\begin{cases} L_r = \{|\psi_n - \psi_1| < \delta\psi_{\min}, L_n \in L\} \\ \delta = \frac{1}{1 + e^{4\lambda_n - 5}}, \lambda_n = \frac{l_n}{l_1} \end{cases} \quad (3-10)$$

式中： L_n 为 L 中的任意一条线段，其长度为 l_n ， ψ_n 为 L_n 与水平方向的夹角，判断线段位置相似的角度筛选阈值为 $\psi_{\min}$ ，通过反复实验，当 $\psi_{\min}$ 的值设为 2° 时的效果较好， δ 为自适应比例系数，与 L_n 的长度成反比，即 δ 的大小随 l_n 的减小而增大， l_n 越小，线段越容易拼接。

(3) 对 L_r 进行水平、垂直筛选。由于线段存在起点和终点，所以对短线拼接时会存在位置的不同，分为首尾两种情形。如图 3-4 所示。

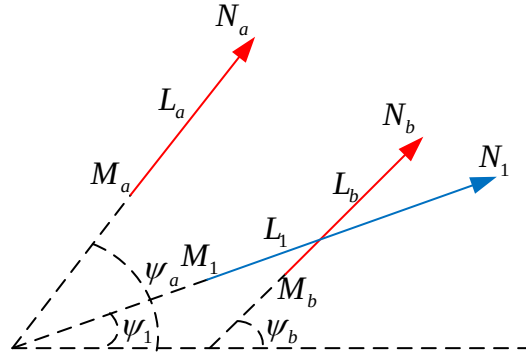


图 3-4 短线拼接算法图

如图 3-4，设 L_a 与 L_b 为两条待拼接的短线，端点坐标分别表示为 $M_a(x_a^s, y_a^s)$ 、 $N_a(x_a^e, y_a^e)$ 、 $M_b(x_b^s, y_b^s)$ 、 $N_b(x_b^e, y_b^e)$ ，则水平与垂直筛选后的线段集合分别为 L_s 、 L_t ：

水平筛选：

$$L_s = \left\{ \left(|x_a^e - x_1^s| < s \right) \vee \left(|x_b^s - x_1^e| < s \right), L_a, L_b \in L_r \right\} \quad (3-11)$$

垂直筛选:

$$L_t = \left\{ \left(\left| y_a^e - y_1^s \right| < s \right) \vee \left(\left| y_b^s - y_1^e \right| < s \right), L_a, L_b \in L_s \right\} \quad (3-12)$$

式中: (x_1^s, y_1^s) 和 (x_1^e, y_1^e) 为线段 L_1 的起点与终点, s 为判断线段水平和垂直距离相似的筛选阈值, 通过实验论证其取值为 3.2。

(4) 通过上述步骤逐步筛选, 最后得到线段集合 L_t , 由于其角度和空间与 L_1 相似, 可将 L_1 加入到集合 L_t , 组成一个新的集合 $\{L_1, L_t\}$ 。从新集合中选取偏离较远的首尾端点构成新线段 L_e 并计算其与水平方向角度 ψ_e , 如果 $\psi_e < \psi_{\min}$, 则满足拼接条件, 用 L_e 替代 $\{L_1, L_t\}$, 反之, $\psi_e > \psi_{\min}$ 时, 偏离角度太大, 不能进行短线拼接。

(5) 重复前面的步骤, 直到短线无法再拼接为止, 最终提取高质量的线段。

3.2 线特征描述匹配

在处理图像特征信息过程中, 线特征的匹配难度要大于点特征。最初的线匹配方法主要利用线的空间几何信息进行连续帧跟踪。Schmid^[55]利用灰度信息和图像之间的多视图几何关系来匹配线段。这些方法的主要缺点是需要获得极线几何信息或准确的初始位姿, 需要高精度的位姿约束和先验信息。为了保证被提取到的线特征能在毗邻帧间顺利匹配以便复原相机运动, 需要构建线特征描述子对线段进行描述, 本文采用了 LBD 描述子描述线特征的方法, 具有不依赖先验位姿信息的优点, 通过引入局部高斯函数 f_l 和全局高斯函数 f_g , 并计算对应的权重系数, 具有更好的匹配效果。

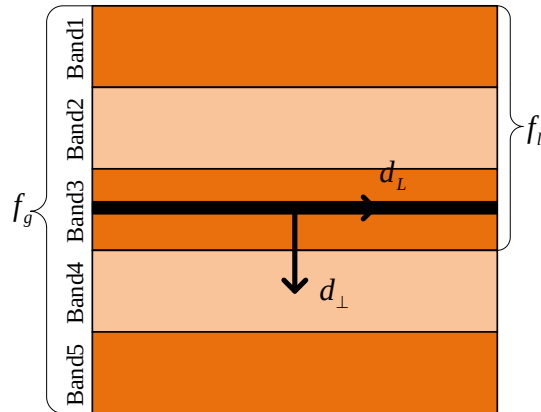


图 3-5 LBD 描述线段支持域

对于线特征描述子的构建, 需要围绕线特征建立线段支持域(Line Support

Region, LSR), 如图 3-5 所示, LSR 被分为了 n 个条带 $B_i (1 \leq i \leq n)$, 各条带像素宽度为 w , 黑色粗实线表线段, 沿线段的方向为 d_L , 与线段正交方向为 $d_\perp$ 。在该 LSR 中, 以线段的中点为坐标原点, d_L 与 $d_\perp$ 分别为横、纵坐标的正方向建立坐标系, 可求出 LSR 中各像素点梯度 g 在所建坐标系中的投影梯度 g_{pro} , 如式(3-13)所示。

$$g_{pro} = (g^T d_L, g^T d_\perp)^T = (g_{pro}^{d_L}, g_{pro}^{d_\perp})^T \quad (3-13)$$

计算 LSR 中各条带 B_i 对应的特征向量 BD_i , 所有特征向量的组成的整体就是 LBD 描述子:

$$LBD = [BD_1^T, BD_2^T, \dots, BD_n^T]^T \quad (3-14)$$

对于 LSR 中各条带 B_i , 可以通过计算其相邻条带的梯度对它进行描述, 那么对于第 k 行, 有:

$$\begin{cases} v1_i^k = \varepsilon \sum_{g_{pro}^{d_\perp} > 0} g_{pro}^{d_\perp} \\ v2_i^k = \varepsilon \sum_{g_{pro}^{d_\perp} < 0} -g_{pro}^{d_\perp} \\ v3_i^k = \varepsilon \sum_{g_{pro}^{d_L} > 0} g_{pro}^{d_L} \\ v4_i^k = \varepsilon \sum_{g_{pro}^{d_L} < 0} -g_{pro}^{d_L} \end{cases} \quad (3-15)$$

式中: ε 为高斯系数, 将所有行合并在一起, 组成了特征向量 BD_i 对应的条带描述矩阵 BDM_i :

$$BDM_i = \begin{bmatrix} v1_i^1 & v1_i^2 & \dots & v1_i^n \\ v2_i^1 & v2_i^2 & \dots & v2_i^n \\ v3_i^1 & v3_i^2 & \dots & v3_i^n \\ v4_i^1 & v4_i^2 & \dots & v4_i^n \end{bmatrix} \in \mathbb{R}^{4 \times n} \quad (3-16)$$

由上式的描述矩阵 BDM_i 可计算出其均值方差 D_i^T 和均值向量 S_i^T , 得到特征向量 $BD_i = [D_i^T, S_i^T]^T$, 则 LBD 描述子的特征表示为:

$$LBD = [D_1^T, S_1^T, D_2^T, S_2^T, \dots, D_n^T, S_n^T]^T \in \mathbb{R}^{8 \times n} \quad (3-17)$$

由于最终得到的 LBD 描述子为 $8 \times n$ 维的浮点型向量, 这会增加距离的计算成本, 为了更快速地计算, 需要将 $8n$ 维的浮点型向量转换为仅含 0 和 1 的 256

维二进制型向量，如图 3-6 所示。

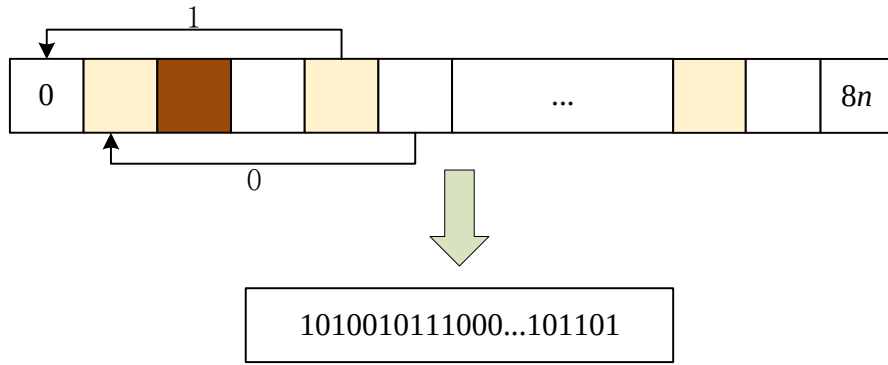


图 3-6 二进制描述子

3.3 线特征数学表述

在 SLAM 系统中，线特征主流的表示方法是用具有 6 自由度的 Plücker 坐标^[56]来表示，又根据数学知识可知，空间线段最简单的表示方法是由两个端点来确定，数学方程为 $Ax + By + Cz + D = 0$ ，具有四个变化量 (A, B, C, D) ，即具有 4 自由度。若直接用 6 自由度的 Plücker 坐标去表示 4 自由度的空间线，这会造成计算量增大，因此要引入正交表示法将 Plücker 坐标用四参数表出。本节内容分别对 Plücker 坐标与正交表示法^[57]进行阐述。

3.3.1 Plücker 坐标

假设在世界坐标系下存在一条空间直线 L_w ，其两端点其次坐标分别为 $P_1 = (x_1, y_1, z_1, \lambda_1)$ 、 $P_2 = (x_2, y_2, z_2, \lambda_2)$ ，则该空间线的 Plücker 矩阵和坐标定义为：

$$T = P_2 P_1^T - P_1 P_2^T = \begin{bmatrix} 0 & y_1 x_2 - y_2 x_1 & z_1 x_2 - z_2 x_1 & \lambda_1 x_2 - \lambda_2 x_1 \\ x_1 y_2 - x_2 y_1 & 0 & z_1 y_2 - z_2 y_1 & \lambda_1 y_2 - \lambda_2 y_1 \\ x_1 z_2 - x_2 z_1 & y_1 z_2 - y_2 z_1 & 0 & \lambda_1 z_2 - \lambda_2 z_1 \\ x_1 \lambda_2 - x_2 \lambda_1 & y_1 \lambda_2 - y_2 \lambda_1 & z_1 \lambda_2 - z_2 \lambda_1 & 0 \end{bmatrix} \in \mathbb{R}^{4 \times 4} \quad (3-18)$$

$$L_w = \begin{bmatrix} \tilde{P}_1 \times \tilde{P}_2 \\ \lambda_1 \tilde{P}_1 - \lambda_2 \tilde{P}_2 \end{bmatrix} = \begin{bmatrix} n_w \\ s_w \end{bmatrix} \in \mathbb{R}^6 \quad (3-19)$$

式中： $\tilde{P}_1$ 和 $\tilde{P}_2$ 为线段端点的非齐次坐标， s_w 与 n_w 分别表示空间线 L_w 的方向向量和法向量， λ_1 与 λ_2 为线段的齐次因子。Plücker 坐标示意图如图 3-7 所示。

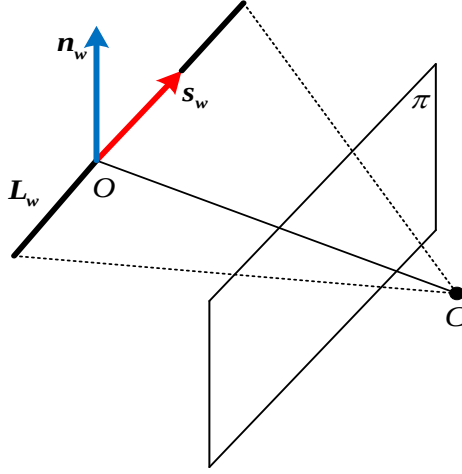


图 3-7 Plücker 坐标示意图

空间直线 L_w 的 Plücker 矩阵和坐标存在转换关系，可以表示为式(3-20)：

$$T = \begin{bmatrix} \hat{n} & s \\ -s^T & 0 \end{bmatrix} \quad (3-20)$$

世界坐标系下的一条空间线 L_w 可通过转换矩阵 T_{cw} 转换为相机坐标系下的直线 L_c ，其公式如下：

$$L_c = T_{cw} \cdot L_w = \begin{bmatrix} R_{cw} & t_{cw}^\wedge \\ 0 & R_{cw} \end{bmatrix} \begin{bmatrix} n_w \\ s_w \end{bmatrix} = \begin{bmatrix} n_c \\ s_c \end{bmatrix} \quad (3-21)$$

式中： R_{cw} 与 t_{cw} 分别表示空间线 L_w 从世界坐标系转换到相机坐标系的旋转矩阵和平移向量， $t_{cw}^\wedge$ 为 t_{cw} 的反对称矩阵。

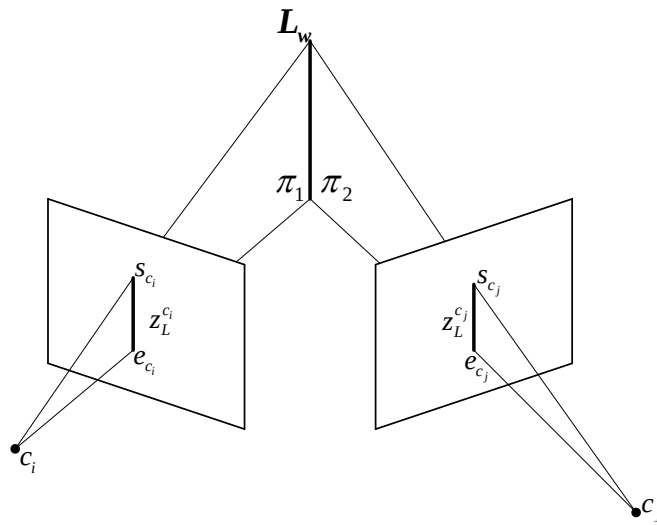


图 3-8 线残差模型

如图 3-8 所示，一条空间直线 L_w 在第 c_i 、 c_j 图像帧中被观测，该直线在两

帧中的投影线为 $z_L^{c_i}$ 、 $z_L^{c_j}$ ，分别用他们的两端点 s_{c_x} 与 e_{c_x} 来表示，设坐标系的原点 $C_o = [a, b, c]^T$ ，由于三个点可以确定一个平面，则可以在空间中得到一个平面 $\pi = [\pi_x, \pi_y, \pi_z, \pi_w]^T$ ，根据空间解析几何，有以下关系：

$$\begin{cases} \pi_x(x-a) + \pi_y(y-b) + \pi_z(z-c) = 0 \\ [\pi_x, \pi_y, \pi_z]^T = s_{c_x} e_{c_x} \\ \pi_w = a\pi_x + b\pi_y + c\pi_z \end{cases} \quad (3-22)$$

再由平面 π_1 和 π_2 可求出直线的在世界坐标系下的坐标 L^* ：

$$L^* = \begin{bmatrix} s^\wedge & n \\ -n^T & 0 \end{bmatrix} = \pi_1 \pi_2^T - \pi_2 \pi_1^T \in \mathbb{R}^{4 \times 4} \quad (3-23)$$

3.3.2 正交表示法

正交表示法是一种更为紧凑的表示方法，能对 6 自由度的 Plücker 坐标进行四参数正交化，精简参数表达式。对公式(3-19)的 Plücker 坐标 QR 分解，得到正交表示 $(U, W) \in SO(3) \times SO(2)$ ：

$$U = R(\phi) = \begin{bmatrix} \frac{n_w}{\|n_w\|} & \frac{s_w}{\|s_w\|} & \frac{n_w \times s_w}{\|n_w \times s_w\|} \end{bmatrix} \in \mathbb{R}^{3 \times 3} \quad (3-24)$$

$$W = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} = \frac{1}{\sqrt{\|n_w\|^2 + \|s_w\|^2}} \begin{bmatrix} \|n_w\| & -\|s_w\| \\ \|s_w\| & \|n_w\| \end{bmatrix} \in \mathbb{R}^{2 \times 2} \quad (3-25)$$

式中： U 和 W 分别表示三维、二维旋转矩阵， $\phi = [\phi_x, \phi_y, \phi_z]^T$ 为相机坐标系到世界坐标系的旋转向量， α 表示旋转角。

同样，正交表示 (U, W) 也可以转化为原来的 Plücker 坐标：

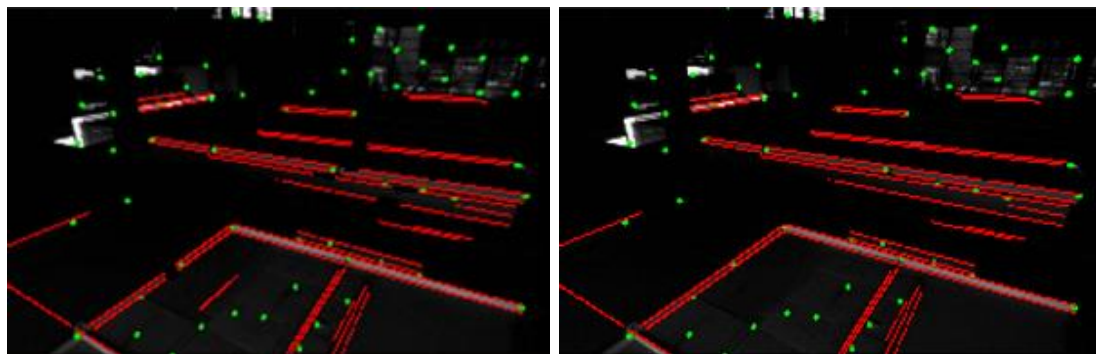
$$L_w = \begin{bmatrix} \cos \alpha u_1 \\ \sin \alpha u_2 \end{bmatrix} \quad (3-26)$$

式中： u_1 、 u_2 分别为矩阵 U 的第一列与第二列元素。

3.4 点线提取算法实验与分析

为了验证改进的线特征提取算法，将改进前后的线特征提取算法在 EuRoC 数据集的 MH_04_difficult 序列进行了仿真实验，并截取了在暗纹理环境中的效

果图,如图 3-9(a)、(b)所示,其中图 3-9(a)为改进前的效果,从中可以看出存在部分短线和断线,这会对线特征的提取质量产生影响,而图 3-9(b)是改进后的效果,从图上可以发现部分短线被剔除,短线被拼接成为了长线,这是由于引入了短线剔除和断线拼接的方法,使短而无用的线段被剔除,断线被重新拼接为长线段,提升了线特征的提取质量。

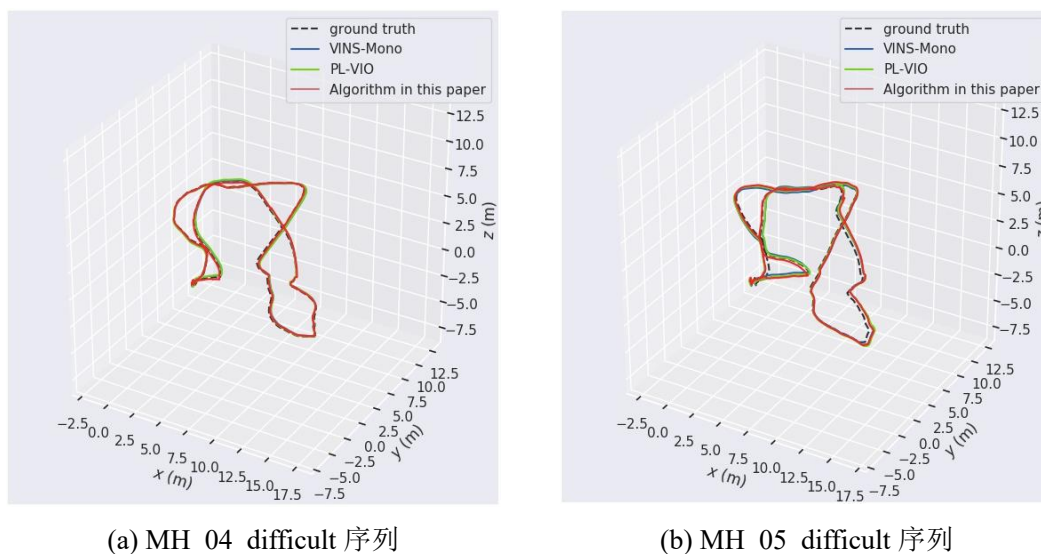


(a) 改进前

(b) 改进后

图 3-9 改进前后线特征的提取效果

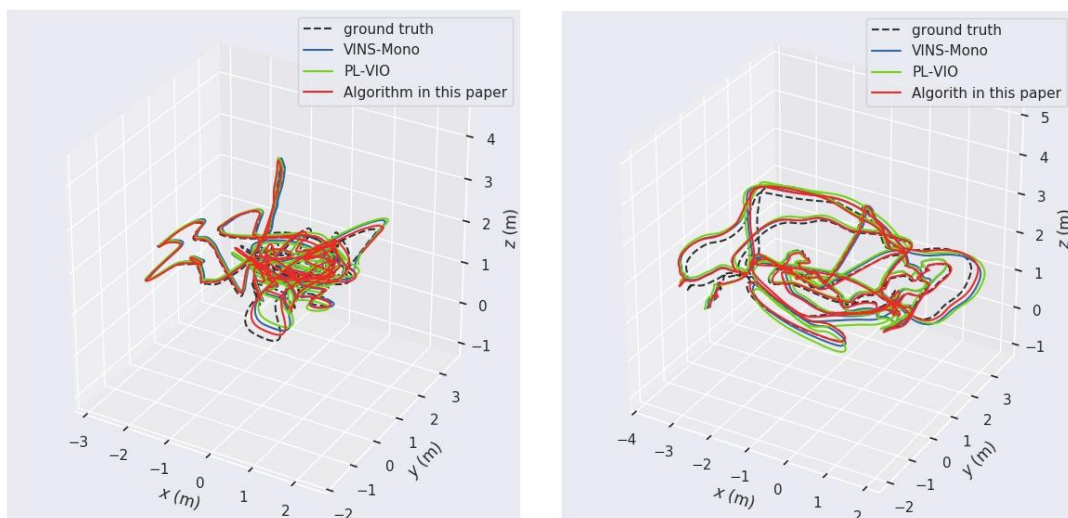
进一步,验证改进的线特征提取算法对机器人位姿估计的影响,将本文算法在大型工厂环境 MH_04_difficult、MH_05_difficult 序列以及小型室内环境 V1_03_difficult、V2_03_difficult 序列中进行了实验验证,并与之前的 VINS-Mono 算法、PL-VIO 算法进行对比,通过 EVO 评估工具绘制出各自的轨迹,如图 3-10 至图 3-11 所示。图中黑虚线表示机器人位姿真实轨迹,蓝、绿、红三种实线分别表示 VINS-Mono 算法、PL-VIO 算法以及本文算法中机器人位姿估计轨迹。



(a) MH_04_difficult 序列

(b) MH_05_difficult 序列

图 3-10 大型工厂环境中三种算法位姿轨迹



(a) V1_03_difficult 序列

(b) V2_03_difficult 序列

图 3-11 小型室内环境中三种算法位姿轨迹

从图 3-10 与图 3-11 可看出，无论是在大型工厂环境下，还是在小型室内环境中，本文算法中的机器人位姿轨迹与另外两种算法相比，其位姿估计轨迹更接近于真实的轨迹，表明本文算法提升了机器人位姿估计精度。

3.5 本章小结

本章对视觉惯性 SLAM 系统前端算法进行了改进，在特征信息提取阶段引入线特征。首先对 LSD 线特征提取算法及线特征匹配所需的 LBD 描述子的原理进行了探究与分析，然后给出了一种自适应线特征提取方法，通过设定线特征提取自适应阈值，剔除多而过短的线段，避免因线段冗余而影响线特征的匹配，再通过短线合并的方法将剩余的短线合并成长线，提高线段的利用率，最后用 Plücker 坐标和正交表示法对线特征进行了数学描述。经局部实验表明，本章给出的自适应线特征提取方法能有效提取线特征，其效果优于传统方法。

第 4 章 基于动态点线耦合的单目视觉惯性 SLAM 算法

针对传统视觉 SLAM 算法在暗纹理环境下鲁棒性差、定位精度低的问题，给出了一种基于动态点线耦合的单目视觉惯性 SLAM 算法。在第 3 章工作研究的基础上，利用点与线特征位置的相似性，提取新的点线耦合特征并构造点线耦合残差模型，用点线耦合残差来搜索点与线特征之间的关系。然后，将点线耦合残差集成到后端滑动窗口优化中，用以构造最小化成本函数，进一步提升了系统的定位精度。

4.1 算法框架

本论文算法的基本结构是基于 VINS-Mono 算法框架改进的，是 IMU 与相机构成的视觉惯性系统，系统主要包括三个模块，分别是前端数据预处理、后端全局滑动窗口优化和闭环检测，其框架结构如图 4-1 所示。

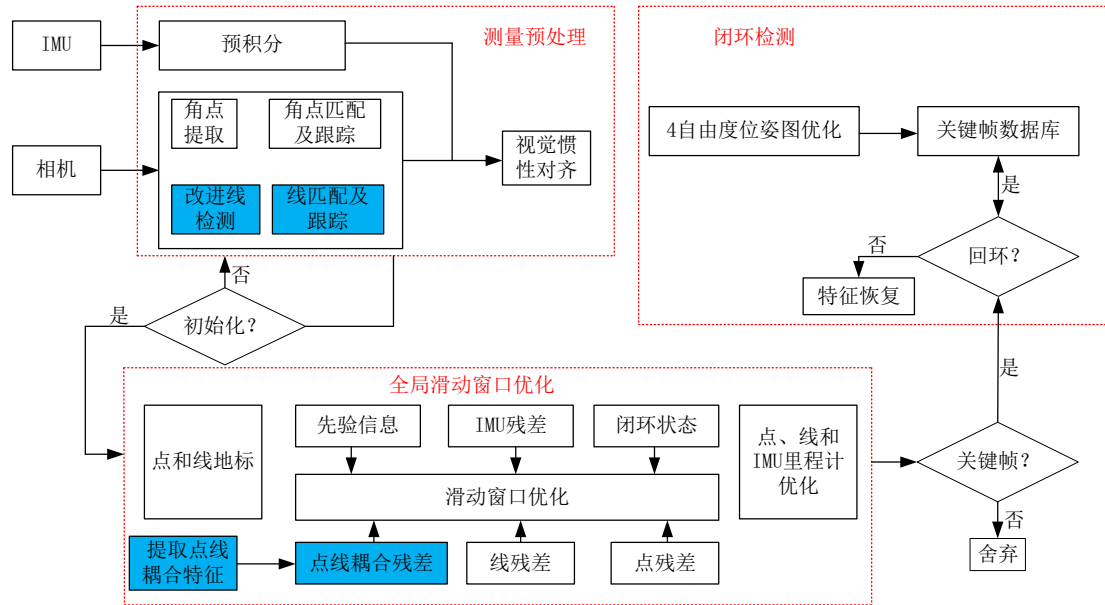


图 4-1 视觉惯性 SLAM 系统框架结构

系统各模块的具体功能如下：

前端数据测量预处理：在该模块中，对 IMU 和视觉传感器（相机）采集到的数据及图像进行处理。IMU 测量机器人的加速度和速度，通过对数据预积分，推算出机器人的位置和姿态。相机负责处理图像信息，对图像中的特征信息进行提取及匹配，本文先使用 Shi-Tomas 算法检测角点特征，再用 KLT 算法^[58]执

行点的跟踪。对于线特征，在原来的 LSD 算法上进行改进，根据特征点与特征线具有相似性的原理，提取新的点线耦合特征并构造点线耦合残差模型，然后采用 LBD 描述符来描述已检测到的线特征，由于 IMU 的采样频率要高于相机，所以最后须将提取到的点线特征与 IMU 预积分视觉惯性对齐。

后端全局滑动窗口优化：随着系统的持续运行，视觉传感器获取到的特征信息不断增多，计算量也就随之增大，从而影响机器人的位姿估计精度，这时就需要引入滑动窗口，对位姿进行非线性优化。首先，采用三角测量方法对相邻图像帧之间的点线特征进行三角化，构建点、线残差模型。然后，滑动窗口的非线性状态估计器可以根据先验信息，IMU 残差，闭环约束，点、线残差构造联合优化函数，实时计算滑动窗口中的帧的位置、速度、旋转与偏差^[59]。

闭环检测：尽管滑动窗口可优化系统的计算量减小误差，但还是存在信息损失而产生累积误差的现象。当累积误差超过某一限度时，会导致系统得到的估计位姿与真实位姿存在严重偏差，甚至是出现程序跑飞，无法得到估计位姿的情况。针对这一问题，需要在系统引入闭环检测来解决。本文采用的是点线特征 DBoW3 词袋模型^[60]这一主流方法来实现系统闭环检测。通过计算两帧图像的相似度来确定当前图像帧是否为关键帧，如果确定当前帧为关键帧，则启动闭环检测过程，此时根据 DBoW3 词袋模型搜索并判断是否产生了闭环。一旦确定闭环产生，就会进行全局位姿图优化，矫正系统位姿，消除累积误差，从而得到一致的位姿信息。

4.2 点线耦合

4.2.1 提取点线耦合特征

图像中的角点特征和线特征具有几何相似性。角点特征通常是从对象或结构的边缘尖端提取。Shi-Tomasi 角点提取算法设计是在边缘方向快速变化的位置处提取，一般在两条或多条线的交点处检测。同样地，在对象或结构的边缘或边界处检测线特征。LSD 算法是检测线特征的常用方法之一，由于该算法计算灰度图像中每个像素的水平线角度并分割具有相似水平线角度的连接区域，则可以在线特征的端点处检测角点特征。因此在线段的末端检测到的角点特征与线特征存在位置关系，这种位置关系被定义为点线耦合特征。

在特征处理过程中，将特征点与距它最近的特征线进行耦合，然后提取融

合后的特征，具体的做法如下表述。这里使用了 K-D 树(K-Dimensional Tree)快速搜索算法，在以特征点为圆心，半径为 r 的圆中搜索线特征的端点，将距离圆心最近的线特征端点所在的特征线视为距离特征点最近的特征线，并将其与特征点融合并提取。示意图如图 4-2 所示。

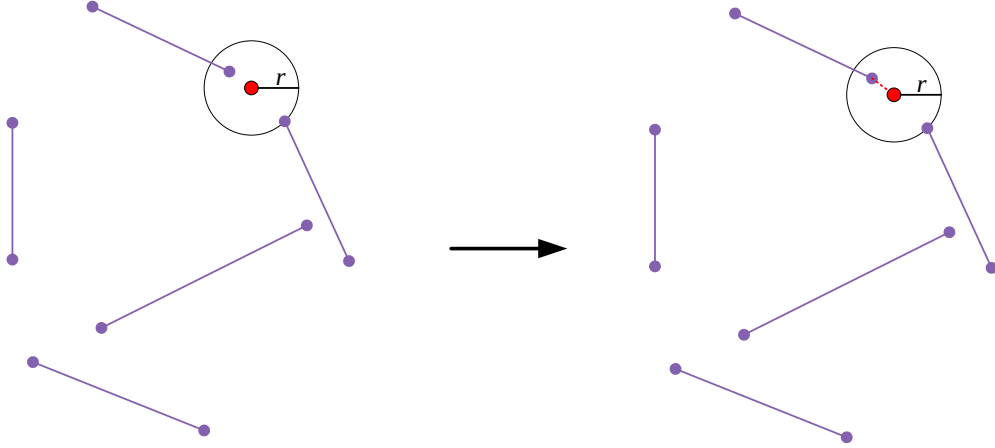


图 4-2 点特征搜索线特征示意图

如果点线特征成功耦合后，可以得到一个关于点特征、线特征和图像帧三者之间的关系式，如公式(4-1)所示。

$$H = \{h | h = \{p, l, c_i, c_j\}\} \quad (4-1)$$

式中： p 表示点特征， l 表示线特征， c_i 与 c_j 表不同的两图像帧， h 为点线耦合特征的描述符， H 表示为 h 的集合。

4.2.2 点线耦合残差模型

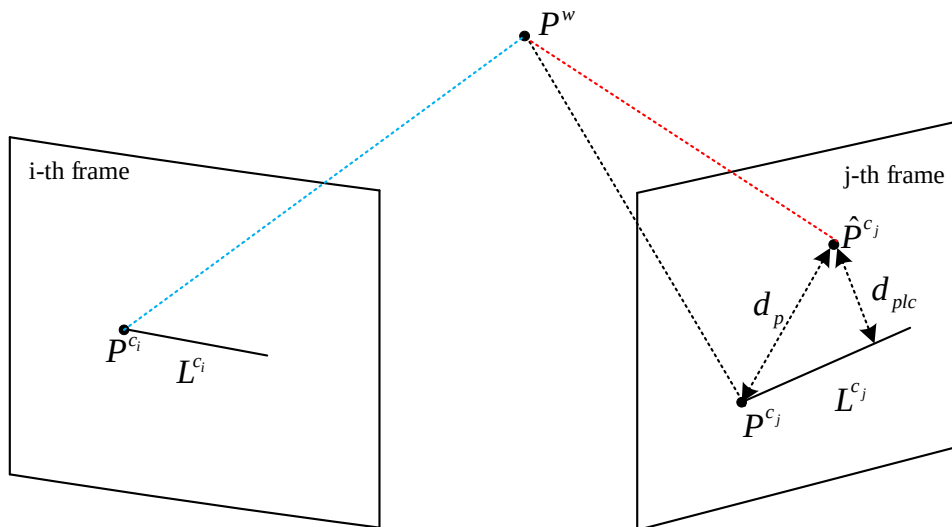


图 4-3 点线耦合残差模型

在全局优化中，采用了 BA 算法估计点特征的深度，点特征的深度估计的

准确性和稳定性受跟踪帧数的影响。跟踪帧数越多，深度估计就越准确越稳定，相反，若跟踪帧数少，就可能会导致不稳定。在系统的后端滑动窗口内设置一个点跟踪帧数的阈值 μ ，若点跟踪的帧数小于 μ ，就在滑动窗口优化中引入所构造的点线耦合残差，利用所提出的点线耦合残差进行鲁棒定位。点线耦合残差模型如图 4-3 所示。

在图 4-3 中， P^i 为第 i 个观察帧中的点特征， $P^{c_j} = (x^{c_j}, y^{c_j})$ 表示在第 j 帧中观察到的点特征，为第 i 帧中点特征 P^i 的跟踪点特征， $\hat{P}^{c_j} = (x^{c_j}, y^{c_j}, 1)$ 为 P^i 的重投影点。 L^i 与 L^j 分别表示为在第 i 帧和第 j 帧中观察到的线特征。直线可以由一个直线方程表示，如公式(4-2)所示。

$$ax + by + c = 0 \quad (4-2)$$

若已知直线的起点 $L_s^{c_j} = (x_s^{c_j}, y_s^{c_j})$ 和终点 $L_e^{c_j} = (x_e^{c_j}, y_e^{c_j})$ ，则可以计算出式(4-2)所表述直线的常系数，如公式(4-3)所示。

$$\begin{cases} a = -\left(\frac{y_e^{c_j} - y_s^{c_j}}{x_e^{c_j} - x_s^{c_j}}\right)b \\ c = -a \cdot x_e^{c_j} - b \cdot y_e^{c_j} \end{cases} \quad (4-3)$$

将直线的常系数集合定义为 $M(L^{c_j})$ ，称之为直线的性质，表示形式如公式(4-4)所示。

$$M(L^{c_j}) = [a, b, c] \quad (4-4)$$

根据在第 j 帧中的重投影点和观测线，就可以得到点线耦合残差，如图 4-3 所示，将属于集合 H 的重投影点($\hat{P}^{c_j}$)和观测线(L^{c_j})之间的距离 d_{plc} 定义为点线耦合残差，如公式(4-5)所示。

$$r_F(z_p^{c_j}, z_l^{c_j}, H, \chi) = d_{plc} \cong \frac{\left(M(L^{c_j}) \cdot [\hat{P}^{c_j}]^T\right)^2}{\sqrt{a^2 + b^2}} \quad (4-5)$$

式中： z 为整体状态向量测量值， χ 为滑动窗口中待优化的状态向量真实值。

4.3 联合优化求解

4.3.1 IMU 残差模型

考虑到滑动窗口中两个连续帧 b_k 和 b_{k+1} 内的 IMU 测量，预积分 IMU 测量的

残差的公式可以定义为：

$$r_I(\hat{z}_{b_{k+1}}^b, \chi) = \begin{bmatrix} \delta\alpha_{b_{k+1}}^{b_k} \\ \delta\beta_{b_{k+1}}^{b_k} \\ \delta\theta_{b_{k+1}}^{b_k} \\ \delta b_a \\ \delta b_g \end{bmatrix} = \begin{bmatrix} R_w^{b_k} (p_{b_{k+1}}^w - p_{b_k}^w + \frac{1}{2} g^w \Delta t_k^2 - v_{b_k}^w \Delta t_k) - \hat{\alpha}_{b_{k+1}}^{b_k} \\ R_w^{b_k} (v_{b_{k+1}}^w + g^w \Delta t_k - v_{b_k}^w) - \hat{\beta}_{b_{k+1}}^{b_k} \\ 2[q_{b_k}^{w^{-1}} \otimes q_{b_{k+1}}^w \otimes (\hat{\gamma}_{b_{k+1}}^{b_k})^{-1}]_{xyz} \\ b_{a,b_{k+1}} - b_{a,b_k} \\ b_{w,b_{k+1}} - b_{w,b_k} \end{bmatrix} \quad (4-6)$$

式中： $\delta\alpha_{b_{k+1}}^{b_k}$ 、 $\delta\beta_{b_{k+1}}^{b_k}$ 、 $\delta\theta_{b_{k+1}}^{b_k}$ 、 δb_a 、 δb_g 分别表示为 IMU 在 b_k 与 b_{k+1} 两个连续图像帧间的位置、速度、旋转角度、加速度、陀螺仪偏置的误差状态， $[\hat{\alpha}_{b_{k+1}}^{b_k}, \hat{\beta}_{b_{k+1}}^{b_k}, \hat{\gamma}_{b_{k+1}}^{b_k}]$ 是两连续图像帧之间预先积分的 IMU 测量项， $[\cdot]_{xyz}$ 为四元数 q 的矢量部分误差状态表示。

4.3.2 点残差模型

如图 4-4 所示，在相机坐标系中，立体空间中有一点 P 同时被相机的不同两帧所观测，该点在第 i 帧和第 j 帧中的投影点分别为 P_i 、 P_j ，投影点 P_i 通过相机位姿估计在第 j 帧中的投影点为 $\hat{P}_j$ 。

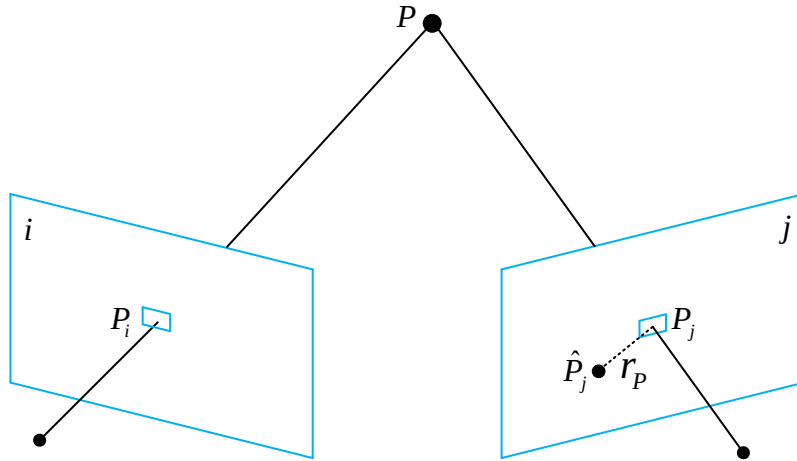


图 4-4 点残差模型

图 4-4 中，点 P 在第 j 帧中的投影点 P_j 与投影点 P_i 经过相机位姿估计在第 j 帧中的投影点 $\hat{P}_j$ 间存在一个误差 r_p ，即把 r_p 定义为点残差。令点 P 的空间坐标为 $(x, y, z)^T$ ，在相机坐标系下的归一化坐标是 $(u, v, 1)^T$ ，点 P 在第 i 帧被观测到的归一化坐标为 $(u_i, v_i, 1)^T$ ，然后进行初始化可获取点路标的逆深度 λ ，那么在

第 j 帧中点 P 再次被观测到并通过位姿估计第 j 帧中的空间坐标如公式(4-7)。

$$P_l^{c_j} = \begin{bmatrix} x_{c_j} \\ y_{c_j} \\ z_{c_j} \\ 1 \end{bmatrix} = T_c^{b-1} T_{b_j}^{w-1} T_{b_i}^w T_c^b \begin{bmatrix} \frac{1}{\lambda} u_{c_i} \\ \frac{1}{\lambda} v_{c_i} \\ \frac{1}{\lambda} \\ 1 \end{bmatrix} \quad (4-7)$$

式中： T_c^b 为 IMU 坐标到相机坐标系的转换矩阵， $T_{c_x}^w$ 表示世界坐标系到第 x 帧 IMU 坐标系转换矩阵。

根据公式(4-7)，可对投影点 P_i 通过相机位姿估计在第 j 帧中的投影点 $\hat{P}_j$ 的坐标做归一化处理：

$$\begin{bmatrix} \frac{x_{c_j}}{z_{c_j}} & \frac{y_{c_j}}{z_{c_j}} & 1 \end{bmatrix}^T \quad (4-8)$$

同样对点 P 在第 j 帧中的投影点 P_j 的坐标归一化： $\begin{bmatrix} u_{c_j} & v_{c_j} & 1 \end{bmatrix}^T$ ，则可以求得点残差为：

$$r_p = \begin{bmatrix} \frac{x_{c_j}}{z_{c_j}} - u_{c_j} \\ \frac{y_{c_j}}{z_{c_j}} - v_{c_j} \end{bmatrix} \quad (4-9)$$

4.3.3 线残差模型

由两个相机坐标系创建的平面与空间线相交。线测量的重投影误差是从端点到投影线的距离，误差通过在普吕克坐标中找到最优成本函数来表示。对于针孔相机模型，3D 空间线到相机平面的 2D 投影表示为：

$$L_p = K n_c = \begin{bmatrix} f_y & 0 & 0 \\ 0 & f_x & 0 \\ -f_y c_x & -f_x c_y & f_x f_y \end{bmatrix} n_c \quad (4-10)$$

式中： K 为投影矩阵， n_c 可由公式(3-21)求得。

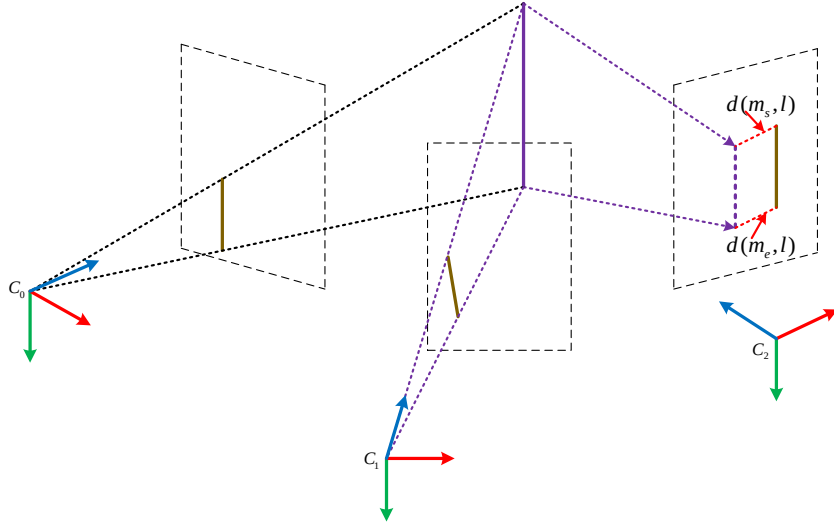


图 4-5 线残差测量模型

如图 4-5 为线残差测量模型，投影的空间线特征没有端点。误差函数^[61]是通过检测二维线的两个端点之间的距离来构造的。常见的处理方法是在世界坐标系中设置空间线 L_w 的正交表示。通过式(3-21)将世界坐标系转换为相机坐标系 c_j 。将帧 c_j 投影到像平面上得到的投影线 l^{c_j} 。相机帧中线残差的定义为：

$$r_L(z_{l_k}^{c_j}, \chi) = \begin{bmatrix} d(m_s^{c_j}, l^{c_j}) \\ d(m_e^{c_j}, l^{c_j}) \end{bmatrix} = \begin{bmatrix} \frac{(m_s^{c_j})^T l^{c_j}}{\sqrt{l_1^2 + l_2^2}} \\ \frac{(m_e^{c_j})^T l^{c_j}}{\sqrt{l_1^2 + l_2^2}} \end{bmatrix} \quad (4-11)$$

式中： z_l^c 是检测到的线 l 相对于相机坐标 c 的观测值， $d(m, l)$ 是 l 到端点之间的距离。

4.4 后端滑动窗口优化

随着运行时间的变长，系统提取到的特征信息变得越来越多，从而增大了系统的计算量，致使系统的实时性与位姿优化的准确性受到了严重的挑战。因此，为了保持系统状态的长期稳定，需要进行滑动窗口优化，其示意图如图 4-6 所示：

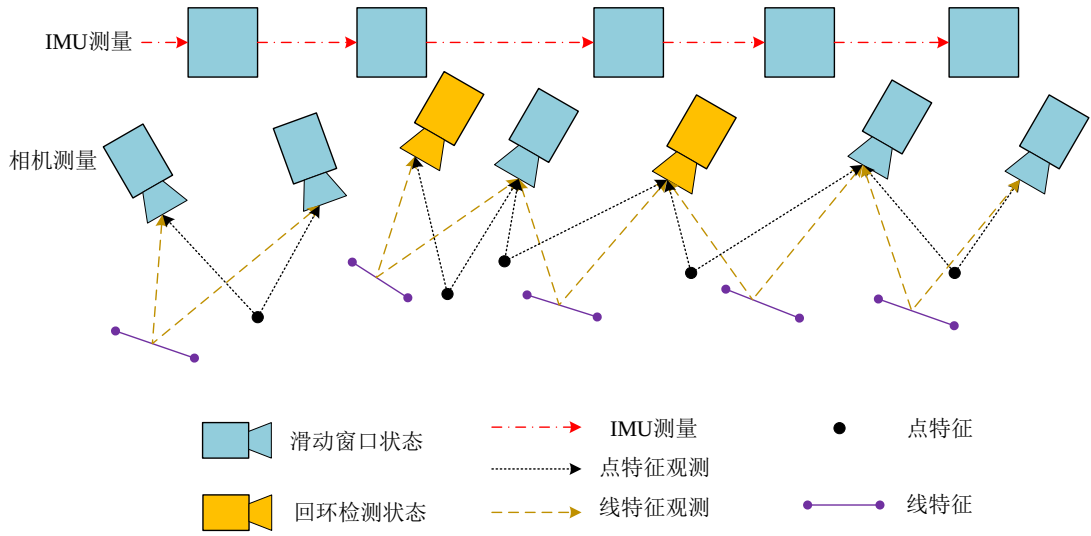


图 4-6 滑动窗口优化示意图

首先，需要定义滑动窗口中的各状态变量，由 VINS-Mono 可知，可将完整的状态变量定义为：

$$\begin{cases} \lambda = [x_0, x_1, \dots, x_m, \lambda_0, \lambda_1, \dots, \lambda_n, O_0, O_1, \dots, O_l] \\ x_k = [p_{b_k}^w, b_{b_k}^w, b_a, b_g], k \in [0, m] \end{cases} \quad (4-12)$$

式中： m 、 n 、 l 分别表示滑动窗口中关键帧，特征点，特征线的数量， x_k 表示第 k 个滑动窗口的 IMU 状态，分别描述了在世界系中 IMU 的位置、姿态方向、速度、加速度计偏差和陀螺仪偏差。 λ_m 表示空间点的逆深度， O_l 表示空间线的正交。

然后最小化边缘化先验残差、IMU 测量残差、闭环信息残差、点残差、线残差和点线耦合残差这六个残差项的和，构造后端最小化成本函数 C ，达到优化滑动窗口中各状态变量的目的。

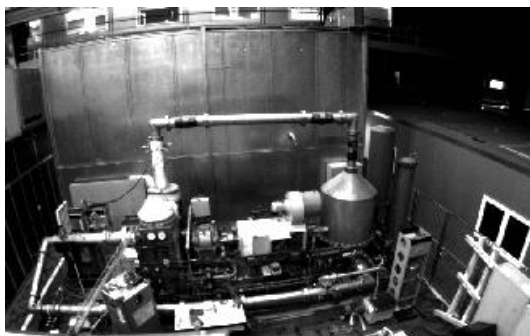
$$\begin{aligned} C = \min_{\chi} \left\{ \sum_{(p,l,j) \in F} \rho \left(\left\| r_F(z_p^{c_j}, z_l^{c_j}, H, \chi) \right\|^2 \right) + \rho \left(\left\| r_p - J_p \chi \right\|_{\Sigma_p}^2 \right) \right. \\ \left. + \sum_{i \in I} \rho \left(\left\| r_i(z_{b_{i+1}}^{b_i}, \chi) \right\|_{\Sigma_{b_{i+1}}}^2 \right) + \sum_{(j,k) \in S} \rho \left(\left\| r_s(z_{s_k}^{c_j}, \chi) \right\|_{\Sigma_s}^2 \right) \right. \\ \left. + \sum_{(j,k) \in P} \rho \left(\left\| r_p(z_{p_k}^{c_j}, \chi) \right\|_{\Sigma_{p_k}^{c_j}}^2 \right) + \sum_{(j,k) \in L} \rho \left(\left\| r_L(z_{l_k}^{c_j}, \chi) \right\|_{\Sigma_{l_k}^{c_j}}^2 \right) \right\} \end{aligned} \quad (4-13)$$

式中： Σ 表示协方差因子， $r_F(z_p^{c_j}, z_l^{c_j}, H, \chi)$ 为本文所构造的点线耦合残差， $\{r_p, J_p\}$ 表示边缘化一帧后的先验信息， J_p 为 Jacobi 矩阵， $r_i(z_{b_{i+1}}^{b_i}, \chi)$ 、

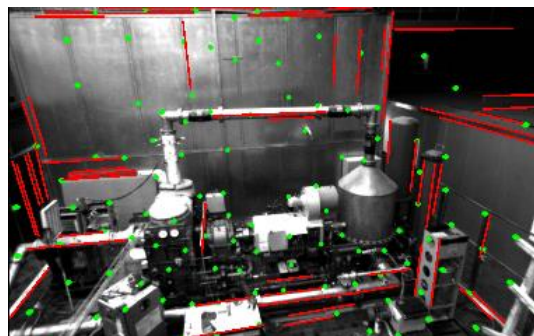
$r_s(z_{s_k}^{c_j}, \chi)$ 、 $r_p(z_{p_k}^{c_j}, \chi)$ 和 $r_L(z_{l_k}^{c_j}, \chi)$ 分别表示 IMU、闭环信息、点和线残差， $\rho(\cdot)$ 表示抑制异常值的柯西鲁棒函数。

4.5 点线耦合特征提取实验与分析

对于点线耦合特征提取算法，分别在有代表性的三种困难场景进行了实验，实验结果如图 4-7 至 4-9 所示。



(a) MH_05 场景

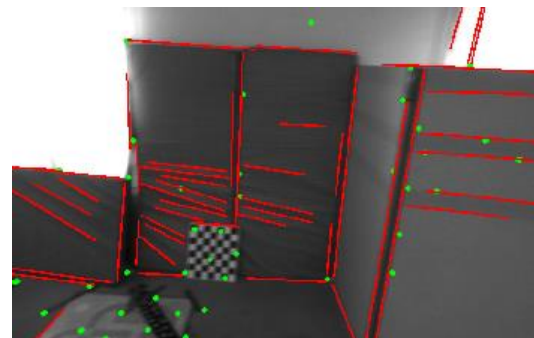


(b) MH_05 点线耦合特征提取

图 4-7 特征提取算法在 MH_05 验证



(a) V1_03 场景

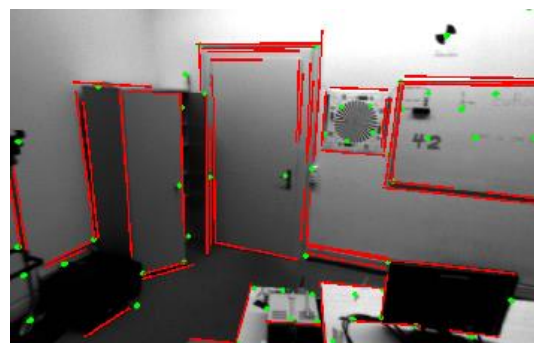


(b) V1_03 点线耦合特征提取

图 4-8 特征提取算法在 V1_03 验证



(a) V2_03 场景



(b) V2_03 点线耦合特征提取

图 4-9 特征提取算法在 V2_03 验证

如图 4-7, 为算法在 MH_05_difficult 序列的验证结果。该序列的特点是运动速快, 且光线昏暗, 而图 4-8 与图 4-9 分别为算法在 V1_03_difficult、V2_03_difficult 序列的验证结果, 二者序列特点为运动速度快, 运动模糊, 且伴有震荡。根据在三个序列的实验结果可以看出, 无论是在光线昏暗, 还是在运动速度快, 纹理模糊的环境, 使用本文的线特征提取算法都能够提取到丰富的点线特征信息, 使系统能保持在稳定状态。

4.6 本章小结

本章详细论述了基于动态点线耦合的单目视觉惯性 SLAM 算法。基于第 3 章的研究, 根据点与线特征位置相似的性质, 定义了一种额外的新点线耦合特征, 并对该特征信息进行提取和点线耦合残差模型的构建, 并搭建了本文 SLAM 系统整体的算法结构框架, 且对系统框架的各个模块功能进行了详细地阐述。给出了 IMU 残差模型、点线残差模型, 把新构建的点线耦合残差模型、IMU 残差模型、点线残差模型、闭环信息以及前端的预处理信息集成到后端的滑动窗口中, 构造了最小化成本函数。

第 5 章 基于 ROS 的视觉惯性 SLAM 算法仿真与平台验证

本章利用搭载 ROS(Robot Operating System)的平台对本论文构建的视觉惯性 SLAM 算法进行仿真实验验证与实物平台验证，并将其实验结果与经典的 SLAM 算法作了对比。

5.1 仿真实验验证

本论文的算法仿真实验在一台处理器为 Intel® Core™i3-8100，运行内存是 8 GB，操作系统为 Ubuntu 18.04 LTS 64 位（安装 ROS melodic 版本）的计算机上进行，实验时所要用的库有 Eigen 3.3.4、Ceres 1.14.0、OpenCV 3.2.3，并使用 EuRoC 公开数据集对该算法进行评估。该数据集是采用 Asctec Firefly 六角旋翼直升机进行录制，使用到的传感器参数如表 5-1 所示。

表 5-1 数据集传感器参数

传感器	型号	频率	特点
双目相机	MT9V034	40Hz	全局快门，单色
IMU	ADIS16448	200Hz	内部已标定
位置获取	LEICA MS50	20Hz	精度=1 毫米
姿态获取	VICON	100Hz	精度=1 毫米

EuRoC 数据集既有大型工厂环境，又有小型房间环境，总共包含简单、中等和困难三个等级的 11 个数据序列，综合考虑了光照、运动速度等飞行条件，可以全面评价算法的性能。其不同场景图如图 5-1 所示。

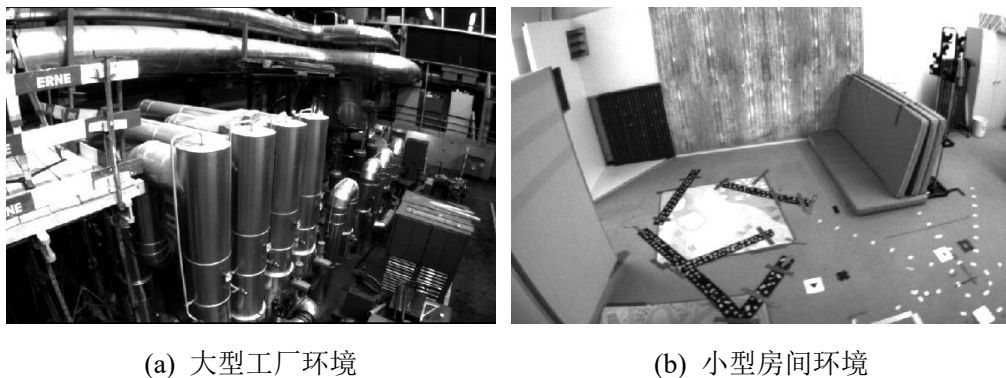


图 5-1 数据集不同场景图

对于本文算法的定位精度的判断，采用了 EVO 工具对算法进行评价。该评估工具可以对算法在数据集各序列的绝对轨迹误差^[62](ATE)——相机的真实位姿与估计位姿的均方根误差(RMSE)进行直观的数值化反映，并且能自动绘制出误差的曲线图、箱线图、误差密度分布图等。下面对绝对轨迹误差做一个说明。

假设机器人在第*i*个关键帧中的真实位姿与估计位姿分别是 P_i , P'_i ，那么其绝对轨迹误差可表述为：

$$ATE_i = P_i(P'_i)^{-1} \quad (5-1)$$

由在实验的时间为一段时间，存在 n 个关键帧的 ATE，则这些 ATE 的均方根误差可以表示为：

$$RMSE(ATE) = \sqrt{\frac{1}{n} \sum_{k=1}^n \|trans(ATE_i)\|^2} \quad (5-2)$$

式中： $trans(ATE_i)$ 表示在每帧中的 ATE 与一段时间内全部关键帧的 ATE 平均数的差值。

5.1.1 SLAM 算法整体运行结果

为验证所研究 SLAM 算法的整体性能，对算法进行了整体仿真运行。图 5-2 至图 5-4 对应算法在不同的三个数据集序列整体运行轨迹，图中红色曲线表示算法运行时的估计轨迹，绿色曲线表示真实轨迹，可以看出算法在各序列中的运行轨迹平滑、完整，且与原本真实轨迹保持一致。

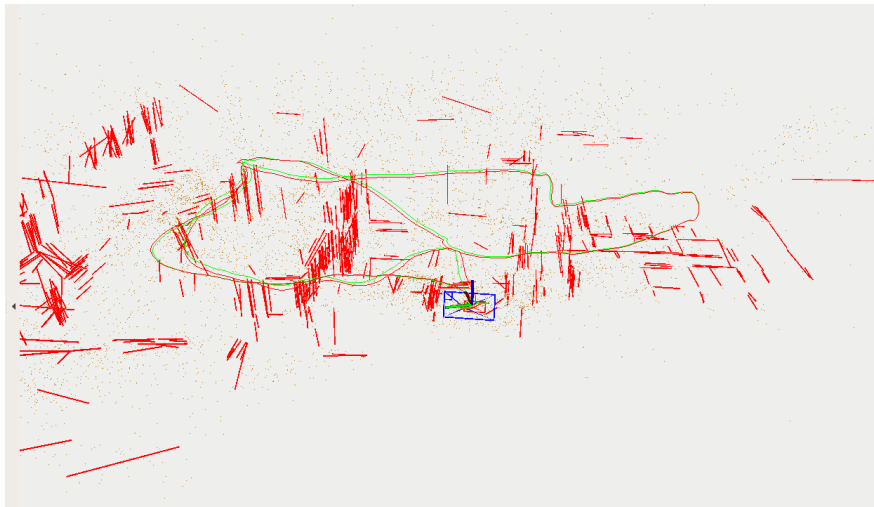


图 5-2 本文算法在 MH_05_difficult 运行轨迹

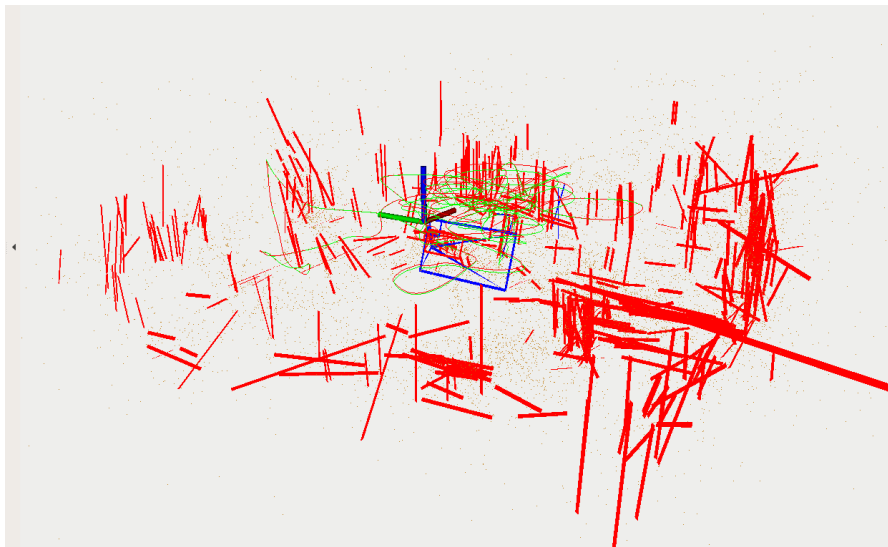


图 5-3 本文算法在 V1_03_difficult 运行轨迹

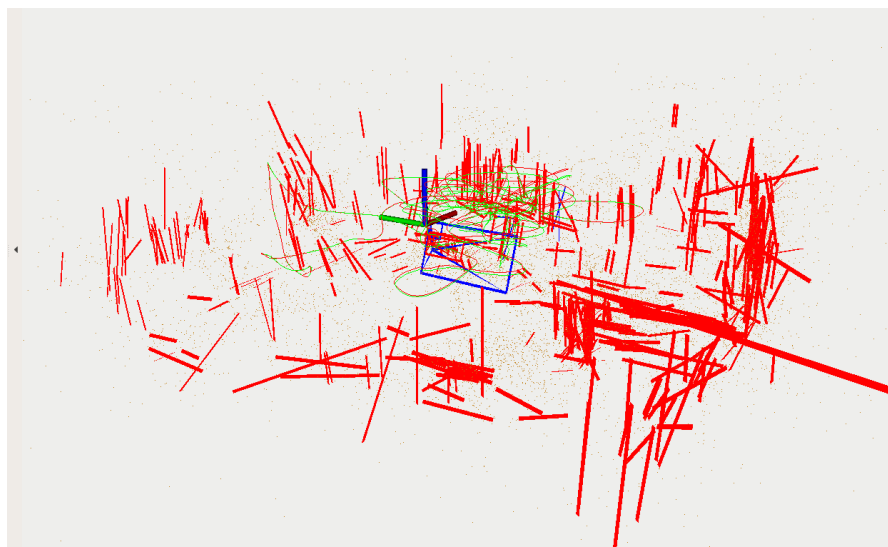


图 5-4 本文算法在 V2_03_difficult 运行轨迹

5.1.2 仿真实验评估分析

在 MH_04_difficult、V1_03_difficult 和 V2_03_difficult 三个难度大的序列中，将本文算法与 VINS-Mono 算法、PL-VIO 算法作对比，得到各自的三维绝对轨迹误差，实验结果如图 5-5 至图 5-7 所示。图中的黑色虚线与彩色实线分别表示真实轨迹和估计轨迹，竖直彩色条表示误差等级，颜色越深误差越小，颜色越深误差越大。从图中可以看出，本文算法的估计轨迹蓝色居多，表明本文算法相对于 VINS-Mono 算法与 PL-VIO 算法的定位误差更小，从而具有更高的定位精度。

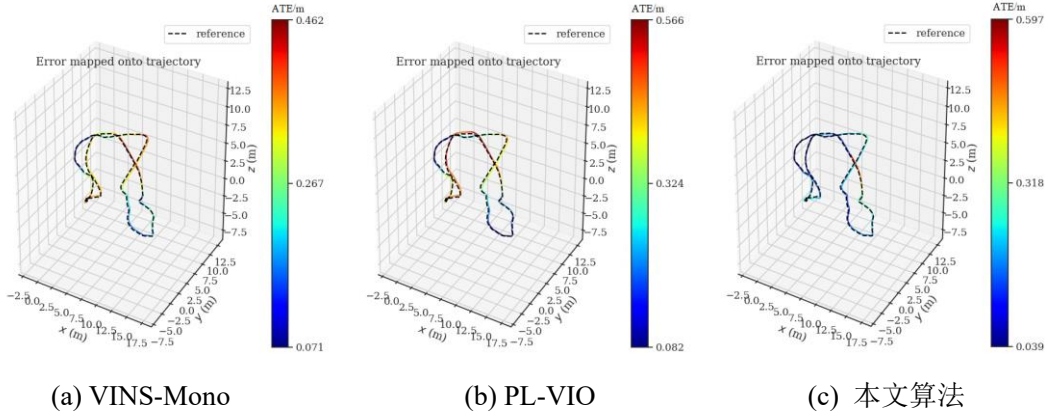


图 5-5 三种算法在 MH_04 序列的三维绝对轨迹误差图

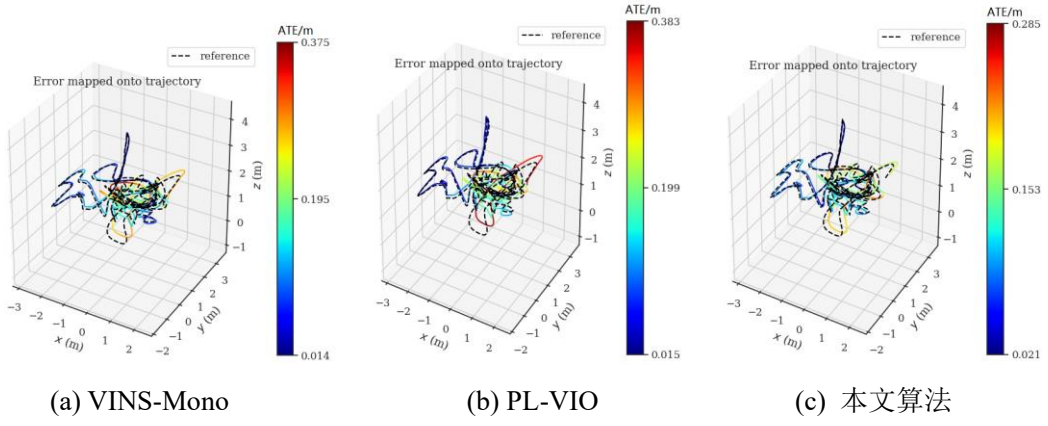


图 5-6 三种算法在 V1_03 序列的三维绝对轨迹误差图

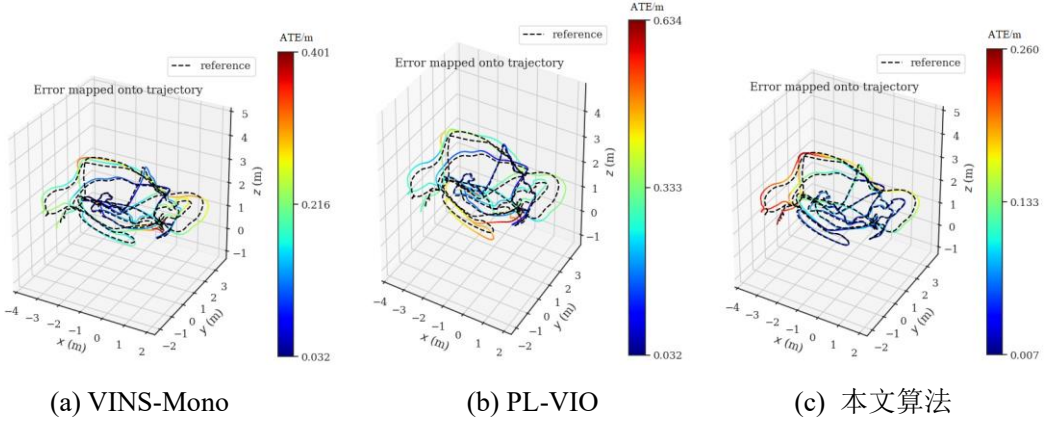


图 5-7 三种算法在 V2_03 序列的三维绝对轨迹误差图

进一步，选取了数据集两个难度大的序列，对本文算法与另外两种经典算法进行了对比试验，并用 EVO 绘制了三者的曲线。如图 5-8(a)、(b)是 VINS-Mono 算法、PL-VIO 算法和本文算法在两个序列中的实时绝对位姿误差曲线图，VINS-Mono 算法、PL-VIO 算法和本文算法的实时绝对位姿误差曲线分别用蓝线、绿线、红线表示。

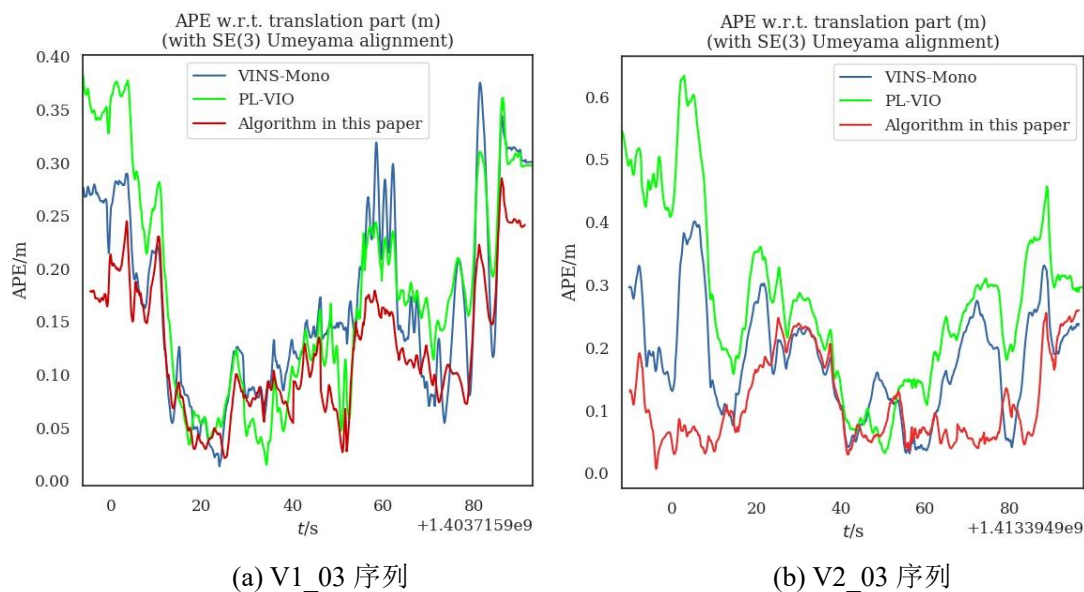


图 5-8 三种算法在两序列的实时绝对位姿误差曲线图

从图 5-8 可以直观的看出本文算法的轨迹误差曲线整体上是低于另外两种算法，且绝对位姿误差(APE)的峰值比另外两种算法小，表明本文算法与 VINS-Mono 算法、PL-VIO 算法相比，取得了更小的定位误差，再次验证了本文算法要优于另两种算法。

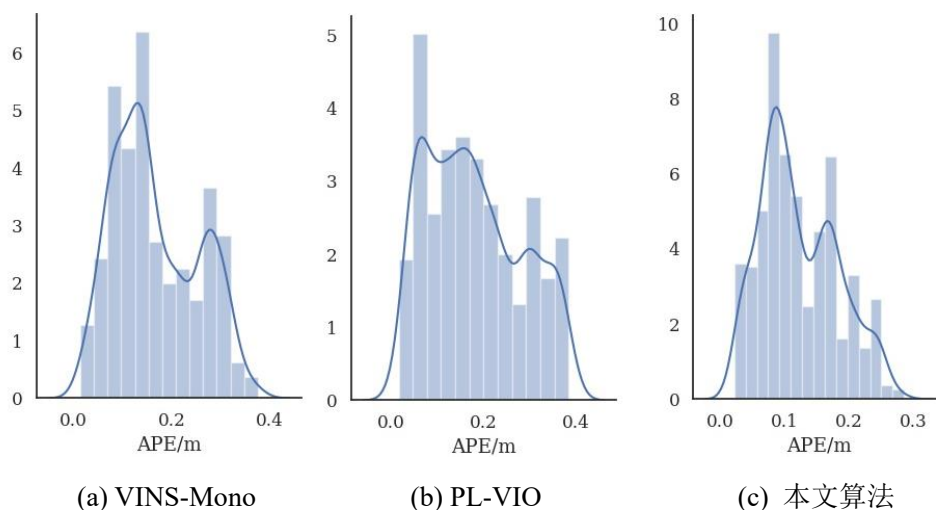


图 5-9 三种算法在 V1_03 序列绝对位姿误差分布

图 5-9 与图 5-10 是 VINS-Mono 算法、PL-VIO 算法和本文算法在 V1_03 和 V2_03 序列中绝对位姿误差的分布情况。图 5-9 可以看出，VINS-Mono 算法、PL-VIO 算法和本文算法在 V1_03 序列的绝对位姿误差分布分别为 0.05m~0.4m、0.05m~0.4m、0.05m~0.3m，从图 5-10 可知，三种算法在 V2_03 序列的对位姿误差分布依次为 0.05m~0.4m、0.05m~0.4m、0.05m~0.3m。从而可知本文算法的绝对位姿误差的分布区间均比 VINS-Mono 算法和 PL-VIO 算法小，进而可得本文

算法的绝对位姿误差分布范围更集中，说明采用本文算法的系统鲁棒性更强，从而使得系统的稳定性更好。

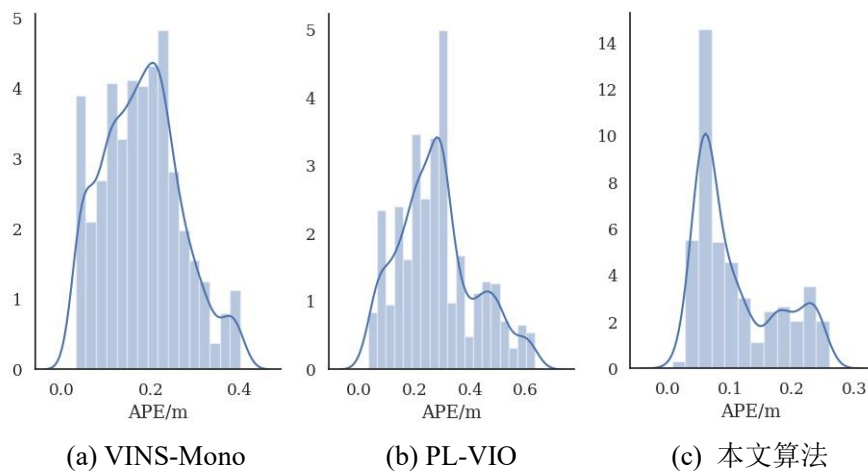


图 5-10 三种算法在 V2_03 序列绝对位姿误差分布

为了在数值上更好说明本文算法的优点，将本文算法在 11 个 EuRoC 公共数据集上进行多次仿真实验后取平均值，并与之前的 VINS-Mono 算法、PL-VIO 算法的结果作对比。在 EuRoC 数据集下的不同算法的绝对轨迹的均方根误差 (RMSE) 和标准差 (STD) 如表 5-2 所示。

表 5-2 不同算法的位姿估计误差

数据集	VINS-Mono		PL-VIO		本文算法	
	RMSE	STD	RMSE	STD	RMSE	STD
MH_01_easy	0.155	0.056	0.132	0.038	0.148	0.077
MH_02_easy	0.178	0.118	0.144	0.082	0.088	0.117
MH_03_medium	0.195	0.095	0.263	0.120	0.100	0.052
MH_04_difficult	0.313	0.091	0.367	0.130	0.221	0.095
MH_05_difficult	0.316	0.053	0.278	0.114	0.242	0.100
V1_01_easy	0.089	0.041	0.075	0.038	0.055	0.026
V1_02_medium	0.112	0.047	—	—	0.087	0.043
V1_03_difficult	0.187	0.086	0.206	0.102	0.111	0.059
V2_01_easy	0.083	0.044	0.089	0.053	0.072	0.045
V2_02_medium	0.158	0.091	0.157	0.081	0.089	0.042
V2_03_difficult	0.201	0.088	0.311	0.138	0.128	0.066

注：“—” 表示特征丢失

从表 5-2 数据可以直观地看出，本文算法相较于 VINS-Mono 和 PL-VIO 算法，绝对轨迹误差在大多数序列中有明显降低，这表明算法的定位准确度有了明显提升。本文算法在 MH_03_medium 序列中的轨迹精度比 VINS-Mono 算法提升了 48.7%，在 V2_03_difficult 序列中的轨迹精度比 PL-VIO 提升了 58.8%。

5.2 实物平台实验验证

前面一节只是介绍了在公开数据集上仿真的结果，是一种较为理想的状态，为了使本文算法更有说服力，在实物平台上进行了实验。如图 5-11 所示，为基于视觉开源作业的移动智能机器人实物平台，该平台预装了机器人操作系统，并融合了相机和激光雷达等多种传感器，可辅助机器人进行路径规划与自主导航。在本次实验验证实验中，主要使用平台中含 IMU 的 INDEMINFD 相机，依靠该相机实时采集实物场景信息。

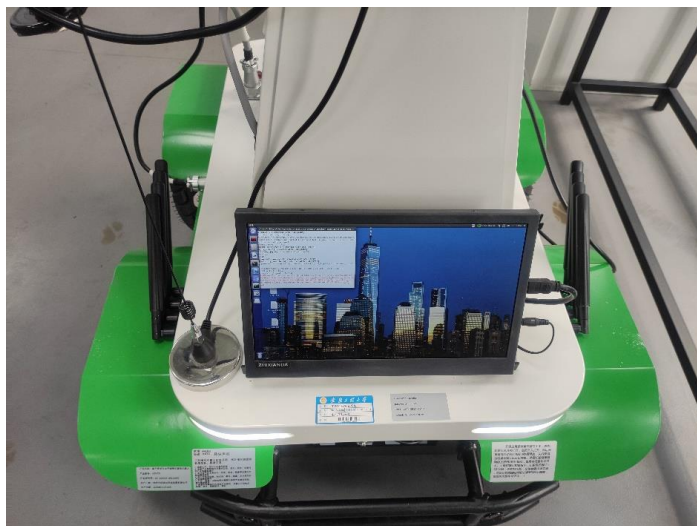


图 5-11 实物平台

本文研究的单目 SLAM，而平台所搭载的相机为双目相机，所以只需用到其中一目，本文在实验时用的是左目。由于相机的内参矩阵在出厂时厂商并未给出，所以在实验时需要自行标定，所采用的标定方法为常用的 Kalibr 标定法，通过标定，得到相机的左目内参矩阵如下：

$$K = \begin{bmatrix} 252.834 & 0 & 291.235 \\ 0 & 252.454 & 192.533 \\ 0 & 0 & 1 \end{bmatrix}$$

相机与 IMU 之间的相对变换矩阵为：

$$T_b^c = \begin{bmatrix} -0.999871 & -0.012026 & -0.010724 & -0.000958 \\ -0.001972 & -0.999926 & -0.001972 & -0.005812 \\ -0.010746 & -0.001843 & -0.999941 & -0.026513 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

参数标定完成后，就可以把标定结果写入算法，以便进行下一步实验。

把设计的算法导入到实物平台，分别选取室内房间、走廊通道和室外场景进行实验验证，得到三种实际场景的实验结果图如图 5-12 所示。

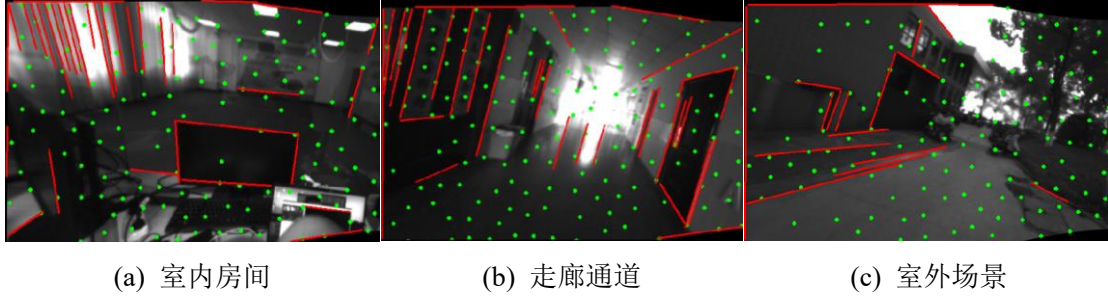


图 5-12 基于实物平台的实验

从以上三幅图可以看出，图 5-12(a)、(b)的场景都有光线弱，纹理不清晰的特点，若仅依靠点特征，无法对系统进行准确的位姿估计，而设计的算法除了提取了点特征外，还提取到了丰富的线特征信息，使系统能稳定运行。而图 5-12(c)为室外场景，光线充足，纹理清晰，就不需要过多的线特征就可以使系统稳定运行。通过在三种环境中的实验，证明了算法的可行性。

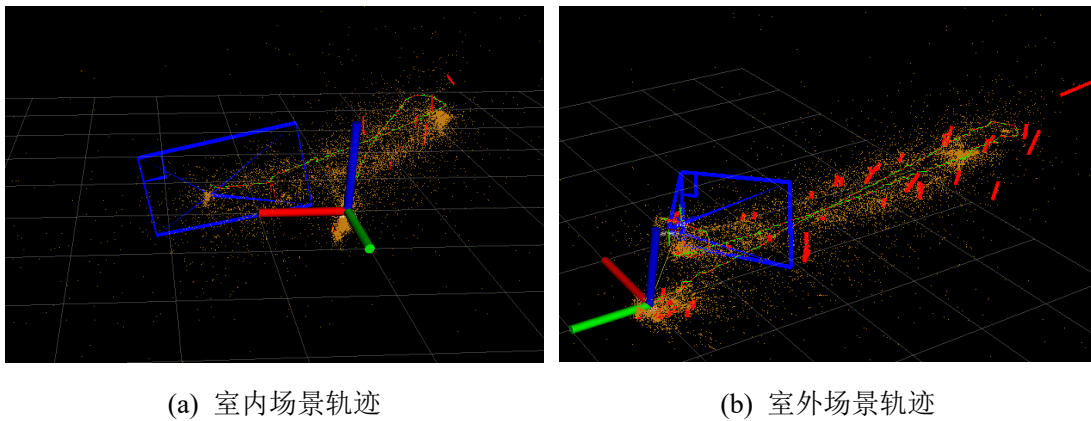


图 5-13 基于实物平台的算法运行轨迹

图 5-13 是本文算法基于实物平台在室内与室外不同场景的运行轨迹图，图中的红色曲线与绿色曲线分别表示算法运行的估计轨迹与真实轨迹。从图中的结果看，无论是在室内还是室外场景，算法的估计轨迹与真实轨迹基本吻合，且轨迹表现平滑，没有出现大幅度漂移现象，代表系统的绝对轨迹误差处于一

个很小的范围，说明了本文算法具有定位精度高、稳定性强的特性。

5.3 本章小结

本章对基于点线融合特征的视觉惯性 SLAM 算法进行了仿真实验与实物平台实验验证。首先在本章第一节，对本文算法进行仿真实验时所要用到的 EuRoC 数据集以及本文算法的评估指标(绝对轨迹误差)进行了讨论与分析，随后对本文中的点线特征提取算法在不同的场景中进行了实验展示；再基于数据集，整体运行了本文算法，得到不同数据集的运行轨迹；再采用了 EVO 工具对其轨迹进行了评估，并从不同的方面与两种经典的算法进行完整比较和详细分析。在第二节中，先给出了 IMU 相机的标定结果，然后展示的是本文算法基于实物实验平台的算法验证，得到本文算法在三种实际环境下点线特征信息的提取效果及算法整体的运行效果，实验结果充分证明了本文算法的有效性。

第 6 章 总结与展望

6.1 总结

本文对点线融合特征的视觉惯性 SLAM 算法开展研究，以视觉惯性 SLAM 技术在纹理复杂场景中的应用作为其研究背景。首先对视觉惯性 SLAM 相关的数学理论作了详细阐述，再对 SLAM 算法进行了局部改进，然后基于公开数据集进行仿真实验，最后将本文算法导入实物平台进行实验验证，证明了本文算法的可行性。本文主要工作总结如下：

(1) 前期对国内外关于视觉 SLAM 的研究进行了大量调研工作，比较几种视觉 SLAM 算法的优缺点，最后选择了视觉惯性 SLAM 算法作为研究内容。在本文的前两个章节，首先分别对视觉 SLAM 算法、视觉惯性 SLAM 算法、基于点线视觉惯性 SLAM 算法的研究现状进行了总体概述，紧接着对本文研究算法所涉及到的机器人运动时的旋转和平移、相机模型、IMU 模型的数学理论作了详细阐述并完成了数学公式推导。

(2) 为解决移动机器人在暗纹理和运动快速环境下点线特征信息提取质量差导致鲁棒性差、定位精度低的问题，给出了一种基于自适应点线特征提取的视觉惯性 SLAM 算法。在基于点特征提取的基础上，加入线特征，通过设置自适应阈值动态调整线特征检测，并采用短线拼接的方法来提取场景中的线特征，提高线特征信息的利用率。将提取到的高质量线特征加入后端优化，确保机器人估计自身位姿状态的精确性。通过局部实验证明，该算法线段提取质量相较于之前的算法有明显提升，表明该算法的有效性。

(3) 在前面工作的基础上，给出了一种基于动态点线耦合的单目视觉惯性 SLAM 算法，解决了传统视觉 SLAM 算法在暗纹理环境下鲁棒性差、位姿估计误差大的问题。根据点特征与线特征相似的性质，引入并提取了一种新的点线耦合特征，且构造出点线耦合残差模型，用点线耦合残差来搜索点与线特征之间的关系。为了检验本文算法对整个 SLAM 系统的鲁棒性、定位精度的影响，设计了一个完整的基于点线融合特征的视觉惯性 SLAM 系统，把本文算法框架分为了前端测量与处理、后端全局滑动窗口优化、闭环检测三个线程。

(4) 基于点线融合特征的视觉惯性 SLAM 算法的实验验证。首先, 讨论分析了本次仿真实验需要用到的 EuRoC 公开数据集的情况以及评估本文算法效果的指标——绝对轨迹误差。其次, 基于不同序列的数据集, 进行了点线特征提取的局部实验, 并展示了本文算法在不同场景中特征信息的提取效果。再对本文算法进行全局运行, 得到了本文算法在不同序列的整体运行轨迹, 记录了各轨迹对应的数据, 采用 EVO 工具对数据进行处理评估, 得到了绝对轨迹误差并绘制了相关的曲线图以及误差密度分布图, 与传统的 VINS-Mono、PL-VIO 算法作了对比, 实验数据表明, 本文算法在定位精度和鲁棒性方面均有显著提升, 相较传统算法有明显优势。最后, 将本文算法导入基于视觉作业开源移动智能机器人实物平台, 开展实地场景测试, 从测试的结果可以看出本文算法在实地场景中可以平稳运行, 并取得了预期的效果, 表明本文算法的可行性。

6.2 展望

本文搭建的基于点线融合特征的 SLAM 系统通过仿真与实物验证实验, 相较于传统算法有显著提升。但由于现实场景的多样化及环境越趋复杂化, SLAM 系统在实际场景中的定位仍是一个值得深入研究的课题。为此, 基于本论文的研究基础, 后续可以展开以下几点研究:

(1) 本论文中仅用了 IMU 与相机结合方式解决机器人定位精度差的问题, 但是还有如 GPS、红外视觉等多种传感器, 后期的研究考虑在现有的 SLAM 系统中融合更多的传感器吗, 帮助机器人定位, 提升定位精度。

(2) 本论文中基本上是对视觉前端进行算法改进的, 缺乏对后端算法的改进工作, 后续会加强对后端算法的研究, 改进后端算法优化机制, 获取更高的定位精度。

(3) 当下深度学习, 人工神经网络以及智能算法研究火热, 可以结合深度学习等在 SLAM 中的研究现状, 将其加入到本文现有的 SLAM 算法中, 加强机器人在现实场景中的定位效果。

参考文献

- [1] Liu Z, Shi D X, Li R H, etc. PLC-VIO: Visual inertial odometry based on point line constraints[J]. IEEE Transactions on Automation Science and Engineering, 2022,19(3):1880-1897.
- [2] Wei S F, Shi X J, Liu Z B, etc. Point-and-line joint optimization visual inertial odometer[J]. Science of Surveying and Mapping,2021,46(4):20-27.
- [3] Esfahani M A, Wang H, Wu K Y, etc. AbolDeepIO: A novel deep inertial odometry network for autonomous vehicles[J]. IEEE Transactions on Intelligent Transportation Systems,2020,21(5):1941–1950.
- [4] Giubilato R, Chiodini S, Pertile M, etc. An evaluation of ROS-compatible stereo visual SLAM methods on a nVidia Jetson TX2[J]. Measurement,2019, 140(1):161–170.
- [5] 关翔中,蔡晨晓,翟文华,等. 基于神经网络补偿的室内无人机组导航系统[J]. 航空学报,2020,41(S1):206-213.
- [6] 任伟. 基于 SLAM 和虚拟现实的综采工作面巡检系统[J]. 工矿自动化,2023,49(5):59-65.
- [7] Smith R C, Cheeseman P. On the representation and estimation of spatial uncertainty[J]. The International Journal of Robotics Research, 1986, 5(4):56-68.
- [8] Davison A J, Reid I D, Molton N D, etc. MonoSLAM: Real-time single camera SLAM[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007, 29(6):1052-1067.
- [9] Shi J, Tomasi C. Good features to track[C]// 1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition,1994:593-600.
- [10] Ribeiro M I. Kalman and extended Kalman filters: Concept, derivation and properties[J]. Institute for Systems and Robotics, 2004, 43(46):3736-3741.
- [11] Klein G, Murray D. Parallel tracking and mapping for small AR workspaces[C]// 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, 2007:225-234.

- [12] Li Y Y, Fan S Y, Sun Y B, etc. Bundle adjustment method using sparse BFGS solution[J]. Remote Sensing Letters, 2018,9(8):789-798.
- [13] 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.
- [14] 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, 2017, 33(5):1255-1262.
- [15] Newcombe R A, Lovegrove S J, Davison A J. DTAM: Dense tracking and mapping in real-time[C]// 2011 International Conference on Computer Vision. IEEE, 2011:2320-2327.
- [16] Engel J, Schöps T, Cremers D. LSD-SLAM: Large-scale direct monocular SLAM[C]// Computer vision—ECCV 2014: 13th European conference, Zurich, switzerland, september 6-12, 2014, Proceedings, Part II 13. Springer International Publishing, 2014:834-849.
- [17] Forster C, Pizzoli M, Scaramuzza D. SVO: Fast semi-direct monocular visual odometry[C]// 2014 IEEE International Conference on Robotics and Automation (ICRA), 2014:15-22.
- [18] Engel J, Koltun V, Cremers D. Direct sparse odometry[J]. IEEE Transactions on Pattern analysis and Machine Intelligence, 2017, 40(3):611-625.
- [19] Yu C, Liu Z X, Liu X J, etc. DS-SLAM: A semantic visual SLAM towards dynamic environments[C]// 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),2018:1168-1174.
- [20] Badrinarayanan V, Kendall A, Cipolla R. SegNet: A deep convolutional encoder-decoder architecture for image segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(12):2481-2495.
- [21] Zhan H, Weerasekera C S, Bian J W, etc. DF-VO: What should be learnt for visual odometry?[J]. arXiv preprint arXiv:2103.00933, 2021.
- [22] Wang J, Rünz M, Agapito L. DSP-SLAM: Object oriented SLAM with deep shape priors[C]// 2021 International Conference on 3D Vision (3DV), 2021:1362-1371.

-
- [23] 张杰. 基于点线特征视觉惯性 SLAM 算法研究[D]. 西北师范大学,2022.
 - [24] Chen G, Zhang H, Fu H, etc. Monocular visual-inertial odometry based on point and line features[C]// 2020 IEEE 6th international conference on computer and communications (ICCC), Chengdu, China, 2020:1443-1449.
 - [25] Shen W, Jia Y, Zhu Y, etc. A fast monocular visual-inertial odometry using point-line features[C]// International Conference on Signal and Image processing (ICSIP), 2022:591-595.
 - [26] 袁国帅,齐咏生,刘利强,等. 一种基于因子图消元优化的激光雷达视觉惯性融合 SLAM 方法[J]. 电子学报,2023,51(11):3042-3052.
 - [27] Mourikis A I, Roumeliotis S I. A multi-state constraint Kalman filter for vision-aided inertial navigation[C]// Proceedings 2007 IEEE International Conference on Robotics and Automation, 2007:3565-3572.
 - [28] Li M, Mourikis A I. High-precision, consistent EKF-based visual-inertial odometry[J]. The International Journal of Robotics Research, 2013,32(6):690-711.
 - [29] Leutenegger S, Lynen S, Bosse M, etc. Keyframe-based visual-inertial odometry using nonlinear optimization[J]. The International Journal of Robotics Research, 2015, 34(3):314-334.
 - [30] Bloesch M, Omari S, Hutter M, etc. Robust visual inertial odometry using a direct EKF-based approach[C]// 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015:298-304.
 - [31] Mur-Artal R, Tardós J D. Visual-inertial monocular SLAM with map reuse[J]. IEEE Robotics and Automation Letters,2017, 2(2):796-803.
 - [32] Qin T, Li P L, Shen S J. VINS-Mono: A robust and versatile monocular visual-inertial state estimator[J]. IEEE Transactions on Robotics, 2018,34(4):1004-1020.
 - [33] Bloesch M, Burri M, Omari S, etc. Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback[J]. International Journal of Robotics Research,2017,36(10):1053-1072.
 - [34] Gao K, Wang R, Demmel N, etc. LDSO: Direct sparse odometry with loop closure[C]// 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018:2198-2204.

-
- [35] Sun K, Mohta K, Pfrommer B, etc. Robust stereo visual inertial odometry for fast autonomous flight[J]. IEEE Robotics and Automation Letters.2018,3(2):965-972.
 - [36] Qin T, Cao S Z, Pan J, etc. A general optimization-based framework for global pose estimation with multiple sensors[J]. 2019. doi:10.48550/arXiv.1901.03642.
 - [37] Kuutti S, Fallah S, Katsaros K, etc. A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications[J]. IEEE Internet of Things Journal, 2018,5(2):829-846.
 - [38] Motlagh H D K, Lotfi F, Taghirad H D, etc. Position estimation for drones based on visual SLAM and IMU in GPS-denied environment[C]// 2019 7th International Conference on Robotics and Mechatronics (ICRoM),2019:120-124.
 - [39] Campos C, Elvira R, Gómez Rodríguez J J, etc. ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multimap SLAM[J]. IEEE Transactions on Robotics,2021,37(6):1874-1890.
 - [40] Grompone von Gioi R, Jakubowicz J, -M. Morel J, etc. LSD: A fast line segment detector with a false detection control[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,2010,32(4):722-732.
 - [41] Zhang G, Lee J H, Lim J, etc. Building a 3-D line-based map using stereo SLAM[J]. IEEE Transactions on Robotics,2015,31(6):1364-1377.
 - [42] Pumarola A, Vakhitov A, Agudo A, etc. PL-SLAM: Real-time monocular visual SLAM with points and lines[C]// IEEE International Conference on Robotics and Automation (ICRA), Singapore, 2017:4503-4508.
 - [43] He Y J, Zhao J, Guo Y, etc. PL-VIO: Tightly-coupled monocular visual-inertial odometry using point and line features[J]. Sensors,2018,18(4):1-25.
 - [44] Zhao Y, Vela P A. Good line cutting: Towards accurate pose tracking of line-assisted VO/VSLAM[C]// European Conference on Computer Vision,2018:527-543.
 - [45] Gomez-Ojeda R, Moreno F A, Zuñiga-Noël D, etc. PL-SLAM: A stereo SLAM system through the combination of points and line segments[J]. IEEE Transactions on Robotics, 2019,35(3):734-746.
 - [46] Yang Y, Geneva P, Eckenhoff K, etc. Visual-inertial odometry with point and line

- features[C]// 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019,2447-2454.
- [47] Zou D, Wu Y, Pei L, etc. StructVIO: Visual-inertial odometry with structural regularity of man-made environments[J]. IEEE Trans. Robot., 2019,35(4):999-1013.
- [48] Wen H, Tian J, Li D. PLS-VIO: Stereo vision-inertial odometry based on point and line features[C]// 2020 International Conference on High Performance Big Data and Intelligent Systems(HPBD&IS),2020:1-7.
- [49] 方琪,王晓华,苏杰. 基于分组策略的点线特征融合同步定位与地图构建算法[J]. 激光与光电子学进展,2021,58(14):405-413.
- [50] 赵良玉,金瑞,朱叶青,等. 基于点线特征融合的双目惯性 SLAM 算法[J]. 航空学报,2022,43(3):363-377.
- [51] 刘建军,卢大威,胡雪花,等. 基于点线特征的快速单目惯性 SLAM 算法[J]. 国外电子测量技术,2022,41(3):14-19.
- [52] 付煜,郑爽,别桐,等. 融合点线特征的视觉惯性 SLAM 算法[J]. 计算机应用研究, 2022, 39 (2): 349-355.
- [53] Jia C, Zheng X, Wang X, etc. Visual inertial odometer based on GNSS assisted fusion of point line features in large scale environments[C]// 2023 IEEE International Conference on Mechatronics and Automation (ICMA), Harbin, Heilongjiang, China, 2023:2112-2116.
- [54] 高翔,张涛,刘毅,等. 视觉 SLAM 十四讲[M]. 电子工业出版社:201703.399.
- [55] Schmid C, Zisserman A. Automatic line matching across views[C]// Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition,1997:666-671.
- [56] Dong B, Zhang K. A tightly coupled visual-inertial GNSS state estimator based on point-line feature[J]. Sensors,2022,22(9):1-24.
- [57] Xu L, Yin H, Shi T, etc. EPLF-VINS: Real-time monocular visual-inertial SLAM with efficient point-line flow features[J]. IEEE Robotics and Automation Letters,2023,8(2):752-759.
- [58] Micheal A A,Vani K. Comparative analysis of SIFT and SURF on KLT tracker for

- UAV applications[C]// 2017 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 2017:1000-1003.
- [59] Wang Z H, Cheng X H. Adaptive optimization online IMU self-calibration method for visual-inertial navigation systems[J]. Measurement, 2021,180(1):1-16.
- [60] Galvez-Lpez D, Tardos J D. Bags of binary words for fast place recognition in image sequences[J]. IEEE Transactions on Robotics,2012,28(5):1188-1197.
- [61] Xu B, Wang P, He Y J, etc. Leveraging structural information to improve point line visual-inertial odometry[J]. IEEE Robotics and Automation Letters, 2022,7(2):3483-3490.
- [62] 才子建. 基于多源信息融合的智能车辆同步定位与建图技术研究[D]. 吉林大学,2023.

作者攻读学位期间发表的学位论文及参加的科研项目

一、读研期间发表的学术论文

[1] 吴桐,黄宜庆,张开平. 基于动态点线耦合的单目视觉惯性 SLAM 算法[J]. 长春师范大学学报,2024,43(02):59-67.

[2] Wu T, Zhang K P, Huang Y Q. Visual inertial SLAM algorithm based on adaptive point and line feature extraction [C]. CSSC2024.

[3] 张开平,黄宜庆,吴桐.基于目标检测与深度信息关联的 RGB-D SLAM 算法[J].哈尔滨商业大学学报(自然科学版),2023,39(06):685-693.

二、读研期间参与的科研项目

1. 参与安徽省自然科学基金面上项目“基于增益调度切换的线性变参数系统输入饱和控制”(编号: 2108085MF220)。

2. 安徽省高等学校自然科学研究项目“四旋翼无人机集群鲁棒仿生约束跟随控制与合作型博弈优化研究”(项目编号: 2022AH050962), 第三参与人。

3. 安徽高校协同创新项目“自主智能无人系统视觉感知与控制”(项目编号: GXXT-2020-069)。

致 谢

三年千日，聚散有时。这篇致谢之词的开始，意味着我的研究生生活即将结束。纵使有万般不舍，也要告别校园，告别老师和同学，告别我那呼朋引伴的读书岁月。立在山巅，回望来路，自己的思想和学识都有了很大的进步。在此，要多谢陪伴我一路走来的人们，有你们的陪伴，才使我的研究生生活更加丰富多彩。

一朝沐杏雨，一生念师恩。感恩我的导师黄宜庆教授，您细致严谨的治学态度深深感染了我，让我在三年的硕士学习中进步良多。黄老师对待学生细心又耐心，当我们在课题上感到迷茫的时候，帮助我们找到研究的方向。在科研过程中遇到问题时，黄老师都会认真作答，会从专业的角度提出建议并为我们加油鼓劲。在论文撰写中，传授给我们写论文的方法和技巧。在生活上，黄老师不但关心我们的身体健康，更关心我们的心理健康，希望我们在读研期间，认真科研，开心科研。我相信今后与黄老师不仅是短暂的师徒关系亦是一生的挚友。同时感谢研究生三年所有的授课老师，为我传授诸多知识，在学习中给了我极大的鼓励。在此表示真挚的感谢，祝愿老师们工作顺利，平安幸福。

慈母手中线，游子身上衣。我要感谢我的家人在生活上对我的照顾，让我能够安心学习。感谢你们一直以来的爱与支持，感谢你们在我每一个开心与不开心的日子里的陪伴。家，永远是最温暖的港湾，是我在遇到困难与压力的时候最最想要回到的地方。从蹒跚学步到远行深思，对父母的恩情自是没齿难忘，二十余年的养育之恩，不是一朝一夕，只希望父母长乐久安。

山河不足重，重在遇知己。感谢师兄师姐在学习中的建议，感谢我的同门和好友在生活上的照顾和学业上的支持。是你们在我这三年里留下了无数的温暖，是你们让我对这个校园又多了一份留念。初见乍惊欢，久处仍怦然，人海茫茫，得以相遇，何其有幸。祝你们未来平安喜乐，前程似锦。

一程山水一年华，独愿此去经年，一生磊落，一生纯善。