

分类号: TP242.6

密 级: 公开

学校代码: 10363

学 号: 2200220163



安徽工程大学

Anhui Polytechnic University

硕士学位论文

题目 基于 BWO 闭环检测的改进
视觉 SLAM 算法研究

论 文 作 者 姚满宇

校 内 导 师 陈孟元 (教授)

校 外 导 师 袁学超 (高级工程师)

专业学位类别 电子信息

研究方向 (领域) 人工智能与系统集成

论文提交日期: 2024 年 5 月 31 日

分类号：TP242.6

密 级：公开

单位代码：10363

学 号：2210330151

题 目 基于 BWO 闭环检测的改进视觉 SLAM
算法研究

题 目 Research on Improved Visual SLAM Algorithm
Based on BWO Closed-Loop Detection

学生姓名：姚满宇

校内导师：陈孟元（教授）

校外导师：袁学超（高级工程师）

专业学位类别：电子信息

研究方向（领域）：人工智能与系统集成

论文答辩日期：2024 年 5 月 16 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结果由本人承担。

学位论文作者签名：姚佩宇

日期：2024年5月30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：如满宇

指导教师签名：陈吉云

日期：2024年5月30日

日期：2024年5月30日

基于 BWO 闭环检测的改进视觉 SLAM 算法研究

摘 要

为了实现自主导航和探索未知环境，智能机器人需要实时感知周围环境并进行精确的自身状态估计。同步定位和建图（Simultaneous Localization and Mapping, SLAM）作为机器人感知的基础技术，促进了这些能力的实现。视觉 SLAM 系统使用摄像头作为感知设备不仅可以降低成本，还可以提供丰富的环境信息。然而，在面对复杂环境时，基于特征点的视觉 SLAM 算法存在特征跟踪较为耗时、大视角运动下易出现跟踪丢失以及难以精确识别闭环场景等问题，造成位姿估计误差较大最终影响系统定位精度。本文针对基于 BWO 闭环检测的改进视觉 SLAM 算法进行深入的理论与探索，并重点解决视觉 SLAM 算法在面临复杂场景导致跟踪丢失、轨迹漂移以及闭环检测效率较低等问题。

针对基于特征点法的视觉 SLAM 特征匹配时效过低、传统光流与特征点融合算法跟踪精度较差以及机器人面对大视角运动时易导致关键帧选取不准确造成系统定位精度较低等问题，提出一种改进视觉里程计与关键帧选取的视觉 SLAM 算法。该算法将光流法与特征点法进行融合，通过在视觉里程计中采用双阈值约束特征匹配策略，减少计算冗余并进行准确的位姿估计，提高了系统实时性与跟踪精度。与此同时，在关键帧选取环节提出一种基于活跃特征点的关键帧选取策略，通过计算当前帧中活跃特征点数量判断当前帧是

否属于高质量图像帧进而决定是否将当前帧选取为新的关键帧，最终提高算法的定位精度。

针对传统闭环检测方法在面对复杂场景时效率较低且易产生闭环误检测等问题，引入白鲸优化算法（Beluga Whale Optimization, BWO），通过模拟白鲸行为活动将闭环检测中寻找与当前帧图像相似度最大的历史帧图像的问题转换为最大值优化问题，采用种群适应度筛选个体最优位置进行闭环候选帧选取，得到候选帧集，有效提高了算法的速度。与此同时，提出一种候选帧检测方法进行闭环判断，通过判断阈值得到精确闭环帧，进而提升闭环检测的精度。

为验证本文算法有效性，在公开的 TUM、KITTI 数据集分别对改进视觉里程计与关键帧选取的视觉 SLAM 算法、基于 BWO 闭环检测的视觉 SLAM 算法进行验证，并与 ORB-SLAM2、ORB-LK-SLAM、OV2-SLAM 等先进算法对比，结果表明本文算法在复杂场景及大视角变化下的定位精度与实时性更具优势。通过在自主搭建的视觉移动机器人硬件平台上进行真实场景测试，验证了本文算法的有效性，为视觉 SLAM 技术的应用研究提供了参考依据。

关键词：视觉 SLAM，特征匹配，光流法，关键帧选取，闭环检测

RESEARCH ON IMPROVED VISUAL SLAM ALGORITHM BASED ON BWO CLOSED-LOOP DETECTION

ABSTRACT

In order to achieve autonomous navigation and explore unknown environments, intelligent robots need to sense the surrounding environment in real time and perform accurate estimates of their own state. Simultaneous Localization and Mapping (SLAM), as the basic technology of robot perception, promotes the realization of these capabilities. The use of cameras as perceptual devices in visual SLAM systems can not only reduce costs, but also provide rich environmental information. However, in the face of complex environments, the feature-point based visual SLAM algorithm has some problems, such as time-consuming feature tracking, easy tracking loss under large-angle motion, and difficult to accurately identify closed-loop scenes, resulting in a large pose estimation error and ultimately affecting the positioning accuracy of the system. This paper conducts in-depth theoretical research and exploration on the improved visual SLAM algorithm based on BWO closed-loop detection, and focuses on solving the problems of visual SLAM algorithm in the face of complex scenes leading to track loss, trajectory drift and low closed-loop detection efficiency.

Aiming at the low feature matching time of visual SLAM based on

feature point method, the poor tracking accuracy of traditional optical flow and feature point fusion algorithm, and the low positioning accuracy of the system caused by inaccurate key frame selection when the robot is moving in a large Angle of view, a visual SLAM algorithm with improved visual odometer and key frame selection is proposed. The algorithm combines the optical flow method with the feature point method, and adopts the feature matching strategy with double threshold constraints in the visual odometer to reduce the computational redundancy and perform accurate pose estimation, which improves the real-time performance and tracking accuracy of the system. At the same time, a key frame selection strategy based on active feature points is proposed in the key frame selection process. The number of active feature points in the current frame is calculated to determine whether the current frame is a high-quality image frame, and then whether to select the current frame as a new key frame, and finally improve the positioning accuracy of the algorithm.

Aiming at the problems such as low efficiency of traditional closed-loop detection methods and easy to produce closed-loop false detection when facing complex scenes, Beluga Whale Optimization (BWO) was introduced. By simulating the behavior of beluga whale, the problem of finding the historical frame with the greatest similarity to the current frame in the closed-loop detection is transformed into the maximum optimization problem, and the optimal position of individuals is screened by population

fitness for closed-loop candidate frame selection, and the candidate frame set is obtained, which effectively improves the speed of the algorithm. At the same time, a candidate frame detection method is proposed for closed-loop judgment. By judging the threshold value of the closed-loop frame, the precision of closed-loop detection is improved.

In order to verify the effectiveness of the proposed algorithm, the visual SLAM algorithm based on improved visual odometer and keyframe selection and the visual SLAM algorithm based on BWO closed-loop detection were respectively verified in the published TUM and KITTI data sets, and compared with the advanced algorithms such as ORB-SLAM2, ORB-LK-SLAM and OV2-SLAM. The results show that the proposed algorithm has more advantages in location accuracy and real-time under complex scenes and large Angle changes. Through the real scene test on the hardware platform of the self-built vision mobile robot, the validity of the proposed algorithm is verified, which provides a reference for the application research of visual SLAM technology.

Yao Manyu (Electronic Information)

Supervised by Professor Chen Mengyuan

KEY WORDS: Visual SLAM, Feature Matching, Optical Flow Method, Key Frame Selection, Closed-loop detection

目 录

第 1 章 绪论	1
1.1 研究背景及意义	1
1.2 国内外研究现状	3
1.2.1 视觉 SLAM 技术研究现状	3
1.2.2 闭环检测研究现状	6
1.3 本文研究内容与章节安排	7
第 2 章 视觉 SLAM 基本原理	9
2.1 视觉里程计	9
2.1.1 特征点法	9
2.1.2 光流法	10
2.2 关键帧筛选策略	13
2.3 闭环检测	13
2.3.1 闭环检测概述	13
2.3.2 闭环检测过程	14
2.4 本章小结	15
第 3 章 改进视觉里程计与关键帧选取的视觉 SLAM 算法	16
3.1 融合光流与特征点的双阈值视觉里程计	16
3.1.1 基于光流跟踪难度阈值的特征匹配	17
3.1.2 基于复合旋转量阈值的特征匹配	18
3.2 基于活跃特征点的关键帧选取策略	20
3.2.1 关键帧补偿	20
3.2.2 冗余关键帧剔除	22
3.3 实验结果与分析	23
3.3.1 融合光流法与特征点法的有效性实验分析	23
3.3.2 关键帧选取实验	25
3.3.3 系统定位精度与轨迹对比实验	26
3.4 本章小结	28

第 4 章 基于 BWO 闭环检测的视觉 SLAM 算法	30
4.1 BWO 算法概述	30
4.2 基于 BWO 算法的闭环检测	32
4.2.1 优化模型与适应度函数.....	32
4.2.2 基于 BWO 算法的候选帧选取	34
4.2.3 候选帧检测.....	35
4.3 实验设计与结果分析.....	36
4.3.1 基于 BWO 算法的闭环检测效率实验	36
4.3.2 闭环检测性能对比.....	38
4.3.3 KITTI 和 TUM 数据集轨迹对比实验.....	39
4.4 移动机器人真实场景实验.....	43
4.5 本章小结.....	45
第 5 章 总结与展望.....	46
5.1 全文总结.....	46
5.2 研究展望.....	47
参考文献	48
攻读学位期间发表的学术论文目录	53
攻读学位期间取得的荣誉奖励目录	53
致 谢	54

第 1 章 绪论

1.1 研究背景及意义

机器人技术与理论不断发展，使得机器人在众多行业中崭露头角，并展现出卓越性能^[1]。机器人技术的重要性体现在为人类带来了更加高效的生产与生活方式^[2]。如图 1-1 所示，在教育方面，机器人作为教学辅助工具能够促进学生更加高效地学习。在生活领域，机器人提供多样化的生活体验服务于人们。在工业领域，机器人主要是负责接替人工执行一些重复性、危险以及劳动强度高的任务，这不仅提升了生产的效率，也提高了产品的质量。在医疗行业，机器人应用于精密手术中，以增强治疗效果。在航天领域，机器人可代替人类执行高难度或高危险性任务。SLAM 作为移动机器人实现环境感知与导航功能的重要技术，已成为当前人工智能和机器人领域研究的热点^[3]。



图 1-1 智能移动机器人

移动机器人在执行 SLAM 任务时，需要准确获取环境信息，同时使用自身搭载的传感器感知并理解环境。根据所需传感器功能构建算法，并高效利用机器人传感器收集的环境数据，从而在计算和优化后获取地图，为移动机器人实

现的功能奠定基础。1986 年, Smith 等人^[4]首次将移动机器人在位置不确定的条件下, 根据环境信息估计自身位置和构建环境地图的过程, 称为“同步定位与建图”。视觉 SLAM 通过相机捕捉环境图像序列, 并应用图像处理技术识别关键特征点, 从而完成机器人的实时定位与地图构建^[5]。移动机器人通过视觉传感器得到图像序列, 根据图像帧之间的数据建立约束, 经过计算和优化准确获取场景特征信息, 将这些特征信息联系起来, 高效构建出三维地图。三维地图的建立, 对路径规划和避障导航有重要意义^[6]。另外, 视觉 SLAM 技术也能融合计算机生成的虚拟环境与世界, 从而满足大部分需求。随着智能设备的普及, 和科技服务于人类的迫切需求, 移动机器人视觉 SLAM 技术在不断更新, 优秀开源算法如层出不穷^[7]。

早期的视觉 SLAM 技术利用滤波方法建立约束模型, 通过对约束模型求解来获取移动机器人的位置。常见的滤波方法包括卡尔曼滤波和粒子滤波。此外, 根据需求对卡尔曼滤波方法进行改进, 从而得到扩展卡尔曼滤波方法, 丰富了基于滤波器的方法^{[8][9]}。基于滤波器的方法是建立在概率模型之上, 遵循马尔可夫性原则, 即认为当前的状态仅与上一状态有关, 与更早之前的状态则没有联系。因而, 这些滤波算法仅专注于连续两帧之间的图像关联, 难以与历史图像帧建立联系, 导致无法实现闭环检测。此外, 当轨迹路径呈现弯曲或许多特征点消失时, 经常出现轨迹漂移问题, 导致预测轨迹的误差逐渐积累。同时, 执行滤波算法相对复杂, 并且这种算法对处理能力和内存空间的要求较高, 导致这些算法的无法应用在高实时性的场景中。因此, 基于滤波的 SLAM 技术普遍被认为不适合应用于复杂环境的地图构建^{[10][11][12]}。为此, 有效提取及应用图像数据以提升 SLAM 算法的实时性和减少路径漂移误差逐步成为研究的重点^[13]。基于特征点法的视觉 SLAM 在前端视觉里程计方面有相对完整的框架, 但这种方法同样存在一定的局限性, 例如特征点法需要在图像中检测和描述关键特征点, 涉及大量的计算操作, 包括特征点提取、描述符计算以及特征匹配过程, 因此相对而言计算量较大^[14]。基于光流法的视觉 SLAM 适用于相机位移较小的情况下进行局部运动估计, 对于大场景和大位移的处理能力相对较弱^[15]。基于视觉 SLAM 的移动机器人在面对复杂环境或产生大视角变化时, 特征点无法进行稳定跟踪, 且面对复杂场景时闭环检测准确率与效率较低, 造成了系统定位精度的下降, 无法高效构建出系统运行轨迹^[16]。因此如何减少计算耗时以及面

对大视角变化运动中如何有效进行特征点稳定跟踪，提高闭环检测的准确率与系统的定位精度和实时性，同时高效地恢复相机的运动结构是视觉 SLAM 领域亟需解决的问题。基于此，本文提出一种融合特征点法与光流法的双阈值视觉里程计，并改进关键帧选取策略，提高跟踪线程的计算速度与系统定位精度。最后，引入BWO算法和候选帧检测方法提高闭环检测的效率与准确率，进而完成复杂场景下的任务。

1.2 国内外研究现状

1.2.1 视觉 SLAM 技术研究现状

视觉 SLAM 根据机器人搭载的传感器收集外界环境信息，进而估算出机器人的具体位姿，同时通过构建环境地图，机器人得以实现独立的自主导航功能。传统 SLAM 方法通常借助激光雷达来探测环境中的障碍物，但是激光雷达的高成本在一定程度上限制了 SLAM 技术的广泛应用。随着传感器技术的不断进步，新一代的 RGB-D 相机和惯性测量单元（IMU）等传感器广泛应用于机器人视觉 SLAM 系统中，提供了更丰富的环境信息，有助于提高定位和建图的精度。为机器人在复杂环境中的定位、导航和交互提供了强大的支持。视觉 SLAM 基本框架如图 1-2 所示。

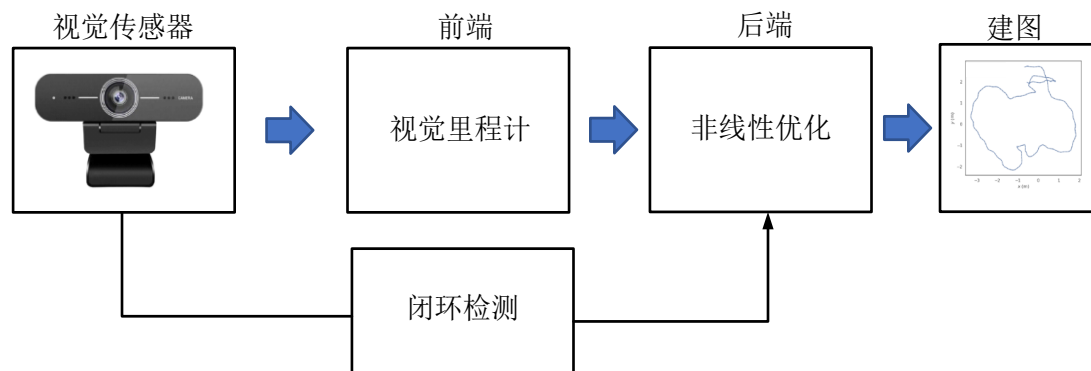


图 1-2 视觉 SLAM 基本框架

为了更高效地获取环境信息，视觉 SLAM 依据环境要求不断更新算法。对于视觉 SLAM 系统，相机是获取外界环境信息的重要依据，从而实现更高层次任务。目前视觉 SLAM 技术主要支持单目、双目以及 RGB-D 三种模式。根据 SLAM 构建的地图又可划分为稀疏、半稠密及稠密三种类型。单目相机通过三

角化估计深度，双目相机需要依据左右图像视差计算出深度值，RGB-D 相机能直接生成与 RGB 图像对应的深度图像。

在单目 SLAM 系统中，MonoSLAM^[17]采用特征点法获取图像特征信息，并利用卡尔曼滤波技术追踪稀疏特征点并估计运动状态，然而，由于系统中所有模块在单一线程中运行，随着稀疏特征点数量增加，负荷量将会显著上升。在应对这一问题时，Klein 及其团队^[18]提出了 PTAM (Parallel Tracking And Mapping) 的图像优化方法。为了增强系统的实时响应能力，该算法采用了一种简洁高效的关键帧筛选机制，并通过将跟踪与地图构建任务分配到两个不同的线程中去执行，实现了并行处理。但随着运行时间的增加，误差也逐渐累积。Endres 等人^[19]提出了 RGB-DSLAM 来解决视觉 SLAM 中累积误差的问题，该算法依赖深度相机捕捉图像深度信息，在优化系统误差的同时完成三维稠密重建。Angeli 等人^[20]提出的算法中，将视觉词袋模型引入到系统以实时检测是否回到相同地方，提高了回环精度。Galvez 等人^[21]提出的方法中，首先获取角点与描述符记录图像特征，然后依据索引策略获取图像之间特征信息，最后依据视觉字典判断是否回环，显著增强了闭环检测过程中特征匹配的效率与实时性。2015 年，Mur-Artal 等人^[22]提出了划时代的 ORB-SLAM 系统，这是首个集成了闭环检测功能的系统，有效降低了 SLAM 操作中误差的累积。此系统采用了 ORB 特征进行算法实现，在提取速度与匹配效率方面实现了较为良好的平衡。基于此，2017 年，该研究团队提出了 ORB-SLAM^[23]系统的改进版本，在先前模型的基础上扩展了对包括单目、双目以及 RGB-D 在内的多种摄像机类型的支持。在 2020 年，ORB-SLAM3^[24]系统依据环境的复杂性，设计了多传感器数据融合技术，从而增强了 SLAM 系统在复杂环境下的处理和适应性。ORB-SLAM3 在大规模环境和动态场景下的性能和稳定性得到了显著提升，能够实现高精度的定位和地图构建，并且在复杂环境中保持良好的鲁棒性。

直接法跟踪方式与光流法类似，都是基于光度不变假设进行。2011 年，Newcombe 等人^[25]提出了 DTAM，该方法使用单目相机采集图像，同时根据直接法构建出位姿约束获取系统当前图像帧所处位置，对每帧图像执行稠密地图构建。但是，此方法需要大量计算，导致系统的实时性较低。为了增强稠密地图构建的效率，Engel 等人^[26]在 2014 年提出了一种实时的半稠密 LSD-SLAM 算法。与 DTAM 不同的是，LSD-SLAM 主要通过选取含有丰富梯度信息的像素点

来估算深度，避免了对全部像素的依赖，增加了系统处理速度。尽管如此，该算法对图像光度变化极为敏感，当相机快速移动或者场景中缺乏足够的纹理时，经常会出现跟踪丢失的情况。针对这个问题，在 2016 年，Engel 等人^[27]提出了 DSO 视觉里程计方法。该方法通过策略进行双重优化，增强了相机在复杂运动条件下跟踪稳定性，同时能够适应纹理稀疏场景。在一定条件下，该方法在不利用 GPU 加速的情况下也能够实时构建半稠密地图。但是该方法无法创建适合机器人导航使用的八叉树地图，并且在处理效率上不及稀疏点云地图。因此，针对实际应用需求，研发一个更有效率、更轻量级的 SLAM 算法尤为重要。为此，Forster 团队提出了 SVO 算法^[28]，该算法考虑到特征点法在计算描述子时相对耗时，因此将特征点直接进行像素灰度值匹配以减少计算描述子花费的时间，同时依据直接法构建出亮度误差函数的方法来确定相机的位置。这一算法的优点在于其较快的处理速度，能达到每秒处理最多 100 帧的性能，极大地满足了系统对实时性的需求，即使是在硬件配置较低的设备上也可高效运行。

目前视觉 SLAM 技术依据人工设计算法获取图像特征点，该方法的稳定性较高。在获取特征点后，系统获取连续两帧图片之间的联系，依据特征点的联系完成匹配与跟踪，同时加入优化策略获取相机位姿和所处环境的 3D 结构图。此外，采用闭环检测机制对全局路径进行校正成为一种高效的方法^[29]。张峻琦等人^[30]则设计了一种结合随机抽样一致性光流跟踪法和改进的特征点匹配法的视觉里程计。该方法通过 RANSAC 光流跟踪法对关键帧之间的小规模运动进行估计，而关键帧之间的运动估计则利用改进的特征点匹配法。然而，该方法忽略了帧之间的联系，在弯道运动时，其算法鲁棒性较差。许广富等人^[31]提出的算法中结合了 ORB 特征和多尺度 LK 光流，有效提高了系统的稳定性，但其采用了固定帧执行特征匹配方法，可能导致计算冗余甚至跟踪丢失，从而降低了系统的精度。齐乃新等人^[32]通过将特征点法与光流法融合，提高了系统的实时性，但在大视角运动下易发生跟踪丢失，从而导致算法跟踪精度较差。张洪等人^[33]针对移动机器人相机与机器人相对运动引发的误拍摄问题，设计了劣质关键帧剔除算法，有效消除了相机与机器人之间相对运动产生的影响，从而降低了机器人在大视角下的定位误差。然而，该算法易造成关键帧冗余，从而降低了系统的实时性。Alonso 及团队^[34]引入了一套关键帧选取机制，该机制依据图像质量及语义内容进行判断，利用建立在 CNN 架构上的 Mininet 网络执行图像

的语义筛选，但其对硬件有一定要求。Lin 等人^[35]设计了一种基于传感器配准问题的关键帧选择机制，有效地降低了系统的跟踪误差。

1.2.2 闭环检测研究现状

在视觉 SLAM 领域，机器人通过前端视觉里程计来估计自己的位置和姿态。该估计过程采用递进式方法，即利用前一帧的数据来计算当前帧的位姿。若机器人在某个点的位置估计发生偏差，该误差会随着时间的推移而累积，进而导致总体误差的增加，这种累积误差极大地影响了机器人的地图构建和定位的精确度，而采用闭环检测被认为是纠正累积误差最为有效的策略。闭环检测主要作用是识别机器人当前所处场景之前是否已经历过^[36]，并以此为基础调节其地图构建和位置定位策略。当机器人确认发生闭环时，系统会比对新获取的图像与历史中的图像，以便准确重定位并纠正地图误差，有效提高定位准确度。因此，实现高效闭环检测对于提升视觉 SLAM 系统的性能至关重要。

闭环检测的方法众多，例如，通过设置路标信息来识别闭环场景。这种方法在机器人经过的区域内预设路标，当机器人再次抵达某个已知路标时，便能识别出闭环的存在。这种方法虽然简单，但需要手动放置路标，并且路标的设置会限制机器人的活动区域。另一方法是装备机器人用 GPS 导航，利用位置信息来检测闭环的发生。此方法虽然解决了依赖路标的局限性，但同时也会使机器人增加额外重量，且 GPS 的定位精度受限。

当机器人再次路过相同的位置，通过相机获取到的图像往往会显示出高度的相似性，因此研究重心转向了如何判定两张图像的相似度^[37]。相似度较高意味着形成闭环的可能性也较大。通常情况下，比较两张图像的最基本方法是进行像素的对比，即通过分析像素的灰度值来评估图像之间的相似度。尽管如此，这种方式涉及到庞大的数据量，因此在计算上非常耗时，在实际的视觉 SLAM 应用中并不满足实时性的要求。

闭环检测中图像相似度评估的方法主要是基于传统的词袋模型算法进行闭环识别。2003 年，Sivic 等人^[38]引入了词袋模型，通过该模型利用经过降噪处理的图像特征来构建一个特征字典，并依据字典内的特征单词来描述环境，进而判断闭环的存在。2005 年，在 Newman 及其团队^[39]提出的算法中，通过构建一

种存储库，及时存储相机获取图像帧的特征信息，随后对目前图像帧获取的特征信息与存储库中的特征信息构建约束并进行匹配，依据策略判定是否发生闭环。2006 年，Niser 等人^[40]考虑到图像特征区域，对局部区域描述符进行量化处理和分层管理，为提升效率，构建出容量更大的词汇树，同时考虑到系统在进行描述符转化时的效率问题。2008 年，Angeli 等学者^[41]在 BoW (Bag of Word) 模型的基础上，增加了图像中形状与颜色信息两种表达方式，同时依据该两种表达信息建立出一种增量式字典，增强了特征信息的表达。在回环判定阶段，在贝叶斯概率模型的基础上，构建出图像分类模型，同时联合该两种模型计算是否发生回环，该方法在一定程度上提高了准确性。2012 年，Yang 及其团队^[42]提出了一种使用低维图像描述符来表示单幅图像的方法，该方法通过 PCA 算法将高维数据转化为低维形式，不仅减少了计算机存储资源的需求，还提升了处理速度。2013 年，Labbe^[43]利用图像的颜色和形状特征进行增量式的字典本构建，满足了机器人需要在线计算的要求，但是增量式的字典本构建过程需要强大的内存管理，不能满足闭环检测的实时性要求。2014 年，Murartal 等人^[44]根据场景需求，选择了更具鲁棒性的 ORB 特征点表达图像特征，使得系统能够适应多种视角下的闭环检测，提高了闭环检测的准确性。2016 年，Lilly 等人^[45]提出的算法中，首先构建出图像帧二进制的表达形式，依据其表达形式进一步进行细化，提高了闭环检测准确率。2018 年，林俊钦等人^[46]通过利用空间信息筛选出候选图像，减少了搜索范围，接着对比当前帧与候选图像的相似性来确认闭环的存在，有效提升了检测的速度。2020 年，夏林^[47]通过在二进制描述符中加入区域编码信息，为描述符赋予了空间定位能力，显著提升了闭环检测的精度及其鲁棒性。

1.3 本文研究内容与章节安排

本论文为实现机器人在复杂环境场景下的高精度定位与建图，并提高系统的实时性，对现有视觉 SLAM 算法进行改进。首先将特征点法与光流法融合，提高跟踪线程的计算速度，然后提出一种基于活跃特征点的关键帧选取策略，提升系统的定位精度与鲁棒性，最后，引入基于 BWO 算法的闭环检测方法，并提出一种候选帧检测方法，提高闭环检测效率与准确率，论文总体技术路线如

图 1-3 所示。

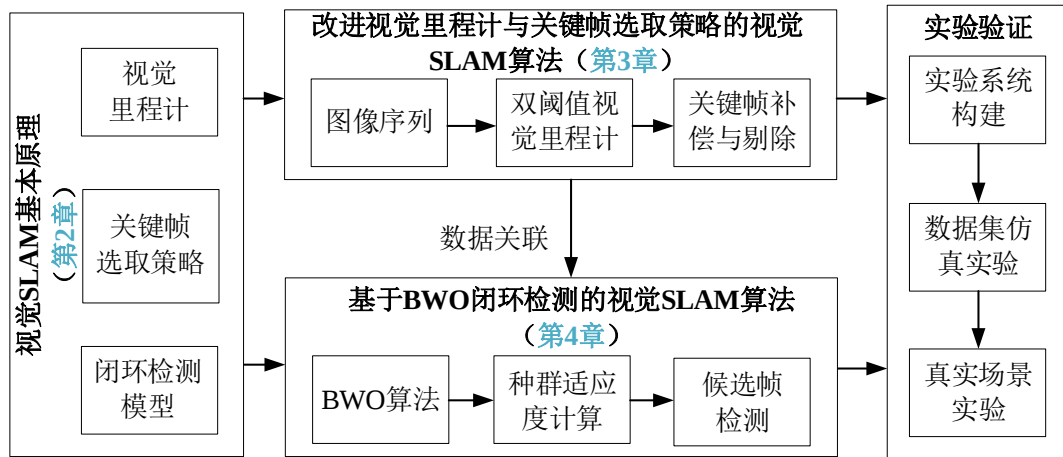


图 1-3 基于 BWO 闭环检测的改进视觉 SLAM 算法总体技术路线图

各章节主要内容如下：

第 1 章：绪论。主要介绍了本文的研究背景及意义，分别概述了特征点法下和直接法下的视觉 SLAM 算法及闭环检测的国内外研究现状，最后对本文主要研究内容进行简要概述。

第 2 章：主要讲述了视觉 SLAM 的相关基础知识。首先对视觉里程计的主要方法做出了总结和分析，然后对关键帧的筛选策略进行简要描述，最后详细介绍了闭环检测的相关知识。

第 3 章：改进视觉里程计与关键帧选取的视觉 SLAM 算法。提出了融合光流与特征点的双阈值视觉里程计，有效解决了特征点法 SLAM 特征匹配时效过低以及传统光流与特征点融合算法跟踪精度较差的问题。提出一种基于活跃特征点的关键帧选取策略，避免算法在大视角运动下易发生关键帧选取不准确导致系统定位精度较低等问题，在公开数据集中进行验证。

第 4 章：基于 BWO 闭环检测的视觉 SLAM 算法。本章提出了一种基于 BWO 闭环检测的视觉 SLAM 算法，首先，通过引入白鲸优化算法（BWO）将闭环检测相似度计算问题转换为最大值优化问题，提升了闭环检测的效率，其次，提出一种候选帧检测方法，提高了闭环检测的性能。最后，在 KITTI、TUM 数据集和真实场景验证了本章所提算法的有效性。

第 5 章：总结与展望。本章对本文主要研究内容和工作进行总结，并讨论了尚待解决的问题，并对今后的研究方向进行展望。

第 2 章 视觉 SLAM 基本原理

本章对视觉 SLAM 中视觉里程计、关键帧的筛选策略和闭环检测三个模块进行介绍。首先介绍了特征点法与光流法的原理，依据特征点法与光流法的优点建立模型，提高 SLAM 系统稳定性。然后介绍了关键帧筛选策略，根据设定阈值，及时获取关键帧。最后介绍了闭环检测基本原理，根据图像相似度，及时获取闭环位置，为本文算法的研究奠定理论基础。

2.1 视觉里程计

视觉里程计（Visual Odometry, VO）是一种利用相机的视觉信息来估计相机在运动过程中的轨迹和姿态的技术，在视觉 SLAM 中起着至关重要的作用。首先从相邻帧中提取特征，然后使用特征描述子对这些特征进行描述，并在相邻帧之间进行匹配，最后利用匹配成功的特征，计算相机之间的相对运动。目前主要采取特征点法和光流法求解视觉里程计问题，因此本节对特征点法和光流法进行详细阐述。

2.1.1 特征点法

特征点是记录图像特征的重要方式，可以准确表示图像特征。特征点法主要通过检测图像特征细节明暗程度获取关键点。特征点检测过程中需要预先设定的阈值，依据阈值响应获取特征点，视觉 SLAM 通常采用 ORB 算法进行特征提取^[48]。系统在获取当前图像的特征点集合后，与上一帧图像的特征点集合进行特征匹配，在特征匹配过程中，将特征点间的描述子进行比较，找出最相似的两个点，然后根据两点间的关联获取位姿矩阵，完成 SLAM 系统定位功能。对于图像帧 I_1 和帧 I_2 ，假设 I_1 中的特征 p_1 点与 I_2 中的特征点 p_2 相匹配，它们分点 P 为特征点 p_1 投影到世界坐标系产生的三维坐标。 I_1 和 I_2 通过特征点的匹配建立的几何约束^[49]关系如图 2-1 所示。

假设图像帧 I_1 和图像帧 I_2 的相机中心分别为 O_1 和 O_2 ，为构建出几何约束获取帧与帧之间的位姿矩阵，将 O_1 、 O_2 和 P 三点连线区域构建成一个平面。得到

点 p_1 和 p_2 的对极几何约束关系式，如下式所示：

$$p_2^T K^{-T} t^\wedge R K^{-1} p_1 = 0 \quad (2-1)$$

其中， K 表示相机的内参数矩阵， $t^\wedge$ 中的向量 t 对应的反对称矩阵。

考虑到表达式 $K^{-T} t^\wedge R K^{-1}$ ，将其转换为：

$$F = K^{-T} t^\wedge R K^{-1} \quad (2-2)$$

为了获取位姿矩阵，需要计算出旋转矩阵 R 以及平移向量 t ，因此，构造出以下约束：

$$\begin{cases} E = t^\wedge R \\ E = K^T F K \end{cases} \quad (2-3)$$

根据矩阵 E 构建算法，获取旋转矩阵 R 以及平移向量 t ，该方法与 2D-2D 法相比，只需要至少 3 对特征点求解相机位姿，从而完成系统定位。

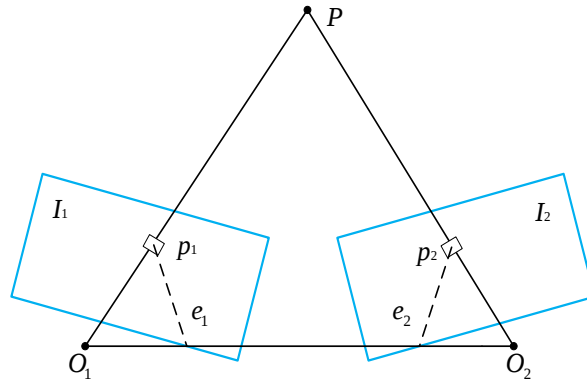


图 2-1 对极几何约束图

2.1.2 光流法

光流是指在图像序列中，描述像素在相机坐标系下的运动速度的概念^[50]。光流法通过分析像素强度在时间上的变化来推断像素的运动情况，从而估计物体的运动。在光流法中，通常包含稀疏光流和稠密光流两种方式。稀疏光流是选择一部分图像中特征较明显的像素进行分析，只跟踪这些像素的运动情况。这样做的好处是可以降低计算量，因为只需要处理图像中的一小部分像素。常见的稀疏光流算法包括 Lucas-Kanade 算法等。稠密光流考虑整幅图像中的所有像素点，将图像分成不同的区域，并计算每个像素点的运动量，从而实现像素级别的图像配准。稠密光流能够提供更为细致的运动信息，但计算量较大，特

别是对于高分辨率的图像来说。常见的稠密光流算法包括基于基本约束的 Horn-Schunck 算法和基于梯度的 Farneback 算法等。

(1) 光流法的基本假设条件

首先需假设亮度恒定不变。在图像序列中，同一空间点在不同帧之间的灰度值保持不变。这意味着目标物体表面的亮度不会随时间发生变化，或者说在相邻帧中，同一点的灰度值在运动过程中保持不变。

其次是空间平滑性假设。该假设认为图像中相邻像素之间的光流在空间上是平滑的。换言之，相邻像素之间的运动速度变化应该是连续的，这是因为在真实世界中，物体通常是连续运动的。

最后应假设目标点为“小运动”。假定在相邻帧之间，物体的运动是小的，即目标物体在短时间内只发生微小的位移。这个假设使得光流法能够使用局部近似来估计像素的运动，从而简化了问题的复杂性。

(2) 基本约束方程

对于图像的当前位置 (x, y) ，分别沿 x 轴和 y 轴添加位移变化量，分别为 dx 和 dy ，同时对当前时间 t 添加时间变化量 dt 。假设当前图像帧中每个位置的像素值在下一帧对应位置保持不变，构建光度变化。如下式所示：

$$FR(x, y, t) = FR(x + dx, y + dy, t + dt) \quad (2-4)$$

其中， $FR(\bullet)$ 表示在 t 时刻，获取图像帧对应位置像素值函数。

依据泰勒级数对公式 (2-4) 展开可得：

$$FR(x, y, t) = FR(x, y, t) + \frac{\partial FR}{\partial x} dx + \frac{\partial FR}{\partial y} dy + \frac{\partial FR}{\partial t} dt + \varepsilon \quad (2-5)$$

式中， ε 代表无穷小项。将公式 (2-5) 两边同除以 dt ，可得：

$$\frac{\partial FR}{\partial x} \frac{dx}{dt} + \frac{\partial FR}{\partial y} \frac{dy}{dt} + \frac{\partial FR}{\partial t} \frac{dt}{dt} = 0 \quad (2-6)$$

依据公式 (2-6)，分别获取像素沿 x 轴和沿 y 轴速度，如下式所示：

$$\begin{cases} v_x = \frac{dx}{dt} \\ v_y = \frac{dy}{dt} \end{cases} \quad (2-7)$$

为方便公式表示，令 $I_x = \frac{\partial FR}{\partial x}$ ， $I_y = \frac{\partial FR}{\partial y}$ ， $I_t = \frac{\partial FR}{\partial t}$ ，得到的约束方程为：

$$I_x v_x + I_y v_y + I_t t = 0 \quad (2-8)$$

要求解像素运动速度 v_x 和 v_y ，需要进一步设计算法。在LK光流算法中，它会在图像中选取局部窗口，在这个窗口内假设灰度变化是线性的。然后，利用局部邻域内的像素来构建一个光流模型，并利用最小二乘法来拟合模型，从而得到该局部窗口内的光流速度 v_x 和 v_y ，重复这个过程，可以在整幅图像上估计每个像素的运动。用红色和蓝色长方形表示已经检测到的特征点，特征点在三维空间的对应的地图点用圆形表示，在前一帧与当前帧中，特征点在两帧之间对应的位置用带箭头的实线表示。

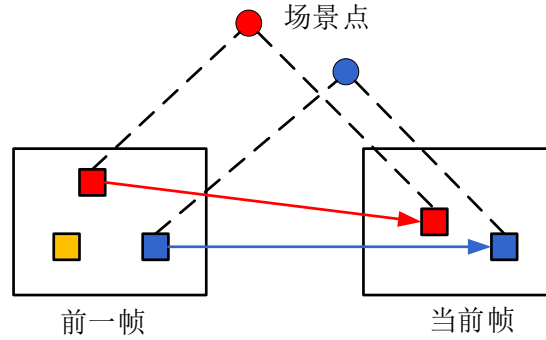


图 2-2 LK 光流跟踪图

根据空间一致性假设原理，设面积为 $m*m$ 的图像包含像素点的数量 m^2 ，则可列出方程：

$$\begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -I_{tk}, \quad k = 1, \dots, m^2 \quad (2-9)$$

记为：

$$A = \begin{bmatrix} \begin{bmatrix} I_x & I_y \end{bmatrix}_1 \\ \vdots \\ \begin{bmatrix} I_x & I_y \end{bmatrix}_k \end{bmatrix}, b = \begin{bmatrix} I_{t1} \\ \vdots \\ I_{tk} \end{bmatrix} \quad (2-10)$$

进一步整理可得：

$$\mathbf{A} = \begin{bmatrix} u \\ v \end{bmatrix} = -\mathbf{b} \quad (2-11)$$

采用最小二乘法求解可得：

$$\begin{bmatrix} u \\ v \end{bmatrix}^* = -(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} \quad (2-12)$$

在视觉里程计中，LK 光流通常被应用于估计相机的位姿。通过时间的离散化，以求解目标像素在一系列图像帧中的位置。若一次迭代无法得出解答，则通常需要进行多次迭代处理。

2.2 关键帧筛选策略

在视觉 SLAM 中，关键帧是指在视觉传感器采集的图像序列中被选择的特定帧，并且这些帧在图像序列中具有显著视角变化和丰富特征点的帧，用于构建地图和进行定位。关键帧的目的在于减少信息的冗余，降低电脑资源的损失，确保系统的流畅运行。通常采用 ORB-SLAM2 中的关键帧方法作为关键帧选取策略，策略如下：

关键帧的筛选涉及两种判断方式：不插入关键帧和插入关键帧。首先需要通过以下三个不插入关键帧的条件进行判断：（1）当前处于定位模式；（2）局部地图处于全局闭环情况下；（3）距离上一次重定位较近。

若不满足不插入关键帧条件，则进行插入关键帧判断，插入关键帧需要满足一个必要条件和一个三选一条件，内容如下：

以内点数必须超过设定的最小阈值为必要条件，在重叠度不能过大的前提下，满足以下任一条件：（1）假设 MAX 是 ORBSLAM2 预设的最大值，而离上次插入关键帧已经过去 MAX 帧；（2）假设 MIN 是 ORB-SLAM2 预设的最小值，距离上一次插入关键帧至少经过了 MIN 帧，且局部建图线程处于空闲状态；（3）关键帧队列在局部建图线程中的数量不超过 3 个。

2.3 闭环检测

2.3.1 闭环检测概述

闭环检测是指在 SLAM 系统中的一个重要步骤，用于识别和纠正相机轨迹

中可能出现的闭环或循环。当相机返回到之前经过的地点时，闭环检测会发现并利用地图中已有的信息，通过匹配当前观测到的特征与先前观察到的特征，来识别这一循环，并对地图和轨迹进行校正，以减少误差累积。为获取帧与帧之间的位姿矩阵，完成系统定位功能，视觉 SLAM 系统中的视觉里程计需要建立相邻图像帧之间的约束。但随着系统运行时间的增加，系统在位姿估计过程产生的误差会不断累积。虽然非线性优化能够依据更多图像帧之间的联系对位姿进行优化，但由于消耗计算资源过大，在进行优化的过程依然存在较大误差，从而降低位姿精度。闭环检测模块在获取系统确定为回环的判断后，及时将系统计算的位姿与历史图像帧库建立数据联系，产生更多约束，从而提高位姿精度，同时构建出与移动机器人所处环境一致的地图。

然而，传统闭环检测方法主要将当前图像帧与图像帧库中的每帧进行特征匹配，通过设定阈值反馈出可能存在的闭环。假设图像帧库中有 N 张图像，由于是帧与帧之间的特征匹配计算，因此在特征匹配计算过程中，需要进行 C_N^2 次数，复杂度为 $O(N^2)$ 。当图像数量增加到一定程度时，其计算量将增大，将占用大部分计算机资源。另一种方法直接对图像帧之间的特征外观进行比较。该方法在一定程度上降低了时间复杂度，随着时间的推移，系统从外界获取的图像数量增多，当图像帧数量增多时，闭环判断准确率会降低。目前闭环检测方法主要依据图像外观获取相似度函数，利用词袋模型构建图像帧间的相似度判定被广泛用于场景识别，从而根据相似度判定是否发生闭环。该方法鲁棒性较强，因此被用于视觉 SLAM 系统中。

2.3.2 闭环检测过程

基于图像外观的闭环检测方法通常包括三个步骤。首先，对当前帧图像进行特征提取，常用的方法包括 SIFT、SURF、ORB 等，再利用这些特征计算图像的描述子，用于表示图像的特征。其次，将当前帧图像的描述子与历史帧图像的描述子进行比较，以寻找与当前帧相似的历史帧图像。这一步通常会利用基于词袋模型的图像检索算法。对于与当前帧相似的历史帧图像，通过计算它们的描述子之间的相似度来评估相似程度。常用的相似度计算方法包括余弦相似度、汉明距离等。最后，根据图像相似度的计算结果，判断是否存在闭环。

通常，如果找到的历史帧图像与当前帧图像相似度很高，并且它们之间的时间间隔足够大，则可以认为闭环检测成功。闭环检测的流程如图 2-3 所示。

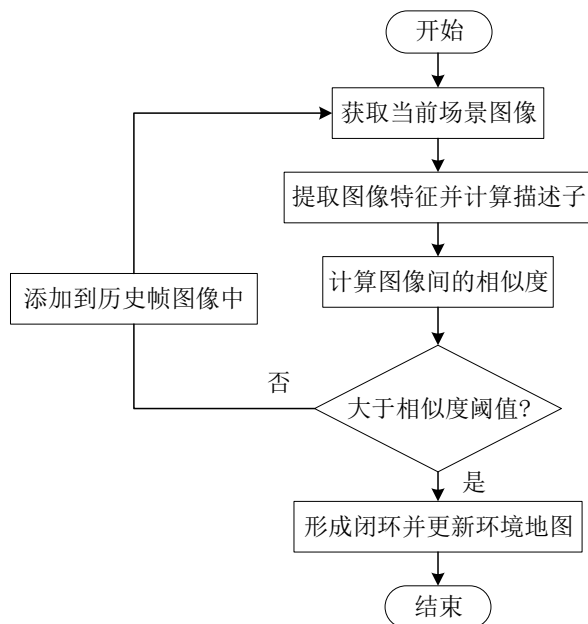


图 2-3 闭环检测流程图

图像相似度计算过程中如何选取关键帧是闭环检测的关键问题。本文采用了改进的关键帧选取策略。在提取图像特征后，对图像进行二进制描述子的计算。然后，根据图像的特征描述子来判断当前场景与历史帧图像所在场景是否相似。为判定是否发生闭环，将图像相似度与设定阈值进行比较，并将闭环之后的信息进行优化，同时更新到环境地图。

精确的闭环检测可提高地图一致性和准确性，改进轨迹估计，增强定位鲁棒性，减少闭环纠正错误，并改善系统长期稳定性。这种改善使系统更能适应复杂环境，提高定位准确性和系统整体性能，在系统长时间运行时尤为重要。

2.4 本章小结

本章介绍了视觉 SLAM 中视觉里程计、关键帧的筛选策略和闭环检测三个模块。首先，探讨了特征点法和光流法的基本原理，并结合这两种方法的优势来构建模型，以增强 SLAM 系统的稳定性。然后，本文阐述了如何根据阈值条件来选取关键帧，以便及时获取图像的关键信息。最后，重点介绍了闭环检测模块，通过图像的相似度分析，从而有效地确定闭环的位置。对视觉 SLAM 系统进行分析，为本文算法的研究奠定了基础。

第 3 章 改进视觉里程计与关键帧选取的视觉 SLAM 算法

本章提出了一种改进视觉 SLAM 算法，着重解决了特征点 SLAM 算法中特征匹配效率较低，以及传统光流与特征点融合跟踪算法精度不足，并且在大视角运动时关键帧选取不准确导致定位精度下降的问题。本章算法在 ORB-SLAM2 的基础上，提出一种双阈值约束特征匹配策略。通过将光流法与特征点法结合，以改进视觉里程计，提升系统实时性和并减少了图像跟踪丢失问题的发生。与此同时，在关键帧选取环节提出一种基于活跃特征点的关键帧选取策略，根据图像的活跃特征点数量判断出高质量图像并作为关键帧，提高系统整体定位精度。系统整体框架如图 3-1 所示。

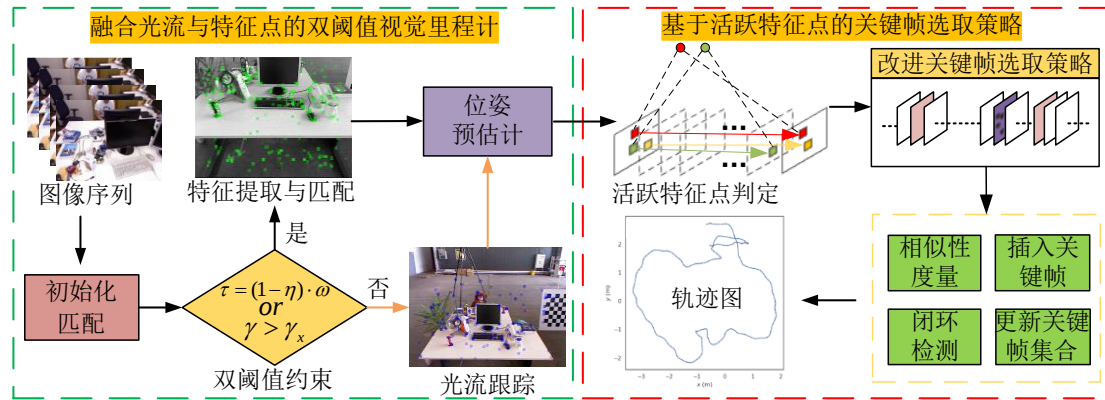


图 3-1 系统整体框架

本章在融合光流与特征点的双阈值视觉里程计模块中，采用双阈值约束特征匹配策略，从当前帧的跟踪难度阈值和复合旋转量决定是否插入图像帧执行特征匹配，进而提高位姿估计的精度，同时在光流帧上通过最小化光度误差执行光流跟踪，提高了算法的实时性与跟踪能力。在基于活跃特征点的关键帧选取策略模块中，依据特征点的活跃值进行活跃特征点的选取，并根据当前帧中的活跃特征点数量选取高质量图像为新关键帧，提升了系统的定位精度。

3.1 融合光流与特征点的双阈值视觉里程计

ORB-SLAM2 在进行特征匹配和跟踪时耗费大量资源，使得系统的实时性会受到影响，且传统光流特征融合的 ORB-LK-SLAM 算法仅关注系统实时性，在面对复杂环境如大视角运动时，以每 20 帧进行特征匹配的策略易产生跟踪丢

失和累积误差较大的问题，从而导致算法的跟踪精度下降。因此，本章在前端视觉里程计采用双阈值约束特征匹配策略如图3-2所示。首先输入图像序列，并进行图像初始化，然后对新生成的图像帧进行自适应跟踪次数 τ 判断，若满足阈值判断条件，则插入匹配帧，执行特征提取与匹配，若不满足阈值 τ 判断条件，再进行复合旋转量阈值 γ 判断，满足判断条件插入匹配帧，执行特征提取与匹配；否则插入光流帧，执行光流跟踪。

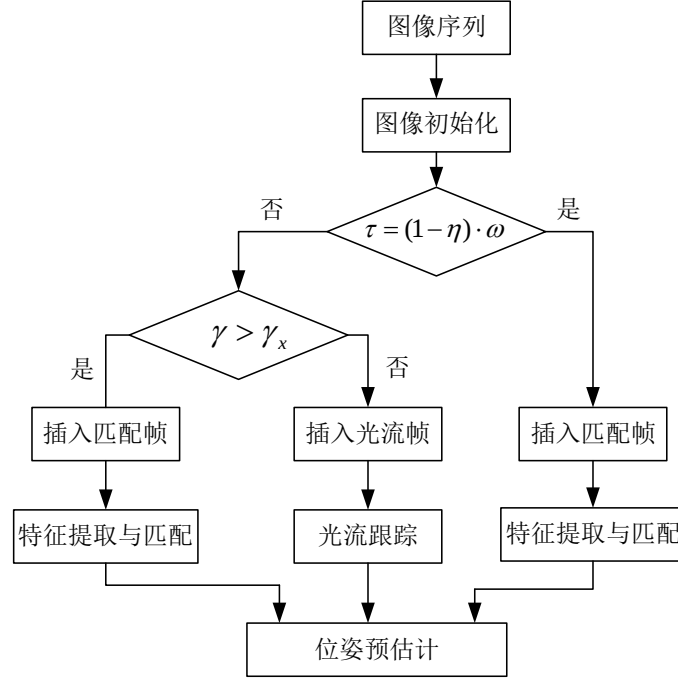


图 3-2 双阈值约束特征匹配策略

3.1.1 基于光流跟踪难度阈值的特征匹配

本章将执行特征匹配的图像作为匹配帧，执行光流跟踪的图像作为光流帧。随着 SLAM 系统的运行，光流跟踪到的特征点数量逐渐减少甚至产生跟踪丢失，从而导致系统无法稳定运行。为解决这一问题，提出一种双阈值约束特征匹配策略，其中第一类阈值是基于光流跟踪难度决定是否插入匹配帧。为了量化光流跟踪的难度，使用跟踪难度系数 η 定量描述上述跟踪过程的困难度。由式(3-1)可知难度系数 η 与跟踪过程中图像帧特征点的数量 δ_{z_i} 成反比：

$$\eta = \frac{\delta_{z_1} - \delta_{z_Q}}{\delta_{z_1}} \quad (3-1)$$

本节用 z 表示帧，假设存在持续 Q 帧的光流跟踪过程，其中 $i=1, \dots, Q$ ，并且

$\{Z_1, \dots, Z_Q\}$ 用于表示该过程。策略步骤如下：将初始帧 Z_1 设置为匹配帧并执行特征匹配；将下一个 H 帧（即 $Z_2, \dots, Z_{H+1}$ ）设置为光流帧并执行光流跟踪，其中 H 是用于启动匹配帧插入方案的预设值，将 Z_{H+2} 设置为匹配帧并执行特征匹配（本文将两个相邻匹配帧的集合和它们之间的所有光流帧称为光流跟踪过程），计算图像帧中的跟踪难度 η ，计算第一个光流跟踪过程（ $Z_2, \dots, Z_{H+1}$ ），将接下来的 τ 个帧设置为光流帧并执行特征跟踪。其计算方法如下：

$$\tau = (1 - \eta) \cdot \omega \quad (3-2)$$

式中， τ 表示自适应跟踪次数， ω 表示不添加新特征点的平均连续跟踪次数。基于光流跟踪难度阈值的特征匹配示意图如图 3-3 所示。

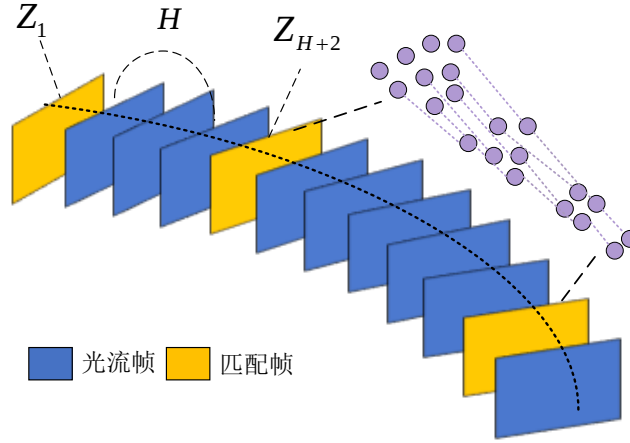


图 3-3 基于光流跟踪难度阈值的特征匹配示意图

3.1.2 基于复合旋转量阈值的特征匹配

光流跟踪在面向大视角运动时跟踪效果较差以及可能陷入局部极值，从而降低系统的定位精度，导致系统的轨迹误差过大。因此，在第一类阈值的基础上增加第二类阈值进行判断，通过当前帧与上一匹配帧之间的复合旋转量来描述旋转过程，解决大视角运动中的匹配帧插入困难的问题。复合旋转量为两帧之间位姿变化的范数之和，反映机器人的运动状态。任意两帧的复合旋转量为：

$$\gamma = \sum_{k=i}^{j-1} \|\phi_{k+1} - \phi_k\|, (i < j), \phi = \begin{bmatrix} p \\ d \end{bmatrix} \in \mathbb{R}^6 \quad (3-3)$$

式中， ϕ 是两个坐标系位姿的李代数形式，它是一个 6×1 矩阵，前三个维度 p 与平移有关，后三个维度 d 与旋转有关， p 的计算公式如式（3-4）， d 的计算

公式如式（3-6）所示。

$$p = tJ^{-1} \quad (3-4)$$

$$J = \frac{\sin \theta}{\theta} I + \left(1 - \frac{\sin \theta}{\theta}\right) ii^T + \frac{1 - \cos \theta}{\theta} i^\wedge, \quad (3-5)$$

$$d = \theta i = \ln R^\vee \quad (3-6)$$

式中， R 是旋转矩阵， t 是平移向量， I 是单位矩阵， i 是单位向量， J 是雅各比矩阵。

第二类阈值是作为相机在遇到旋转等困难场景时获得数值反射的辅助策略。该阈值计算结果为标量，并忽略旋转的方向性，基于复合旋转量阈值的特征匹配示意图如图 3-4 所示。

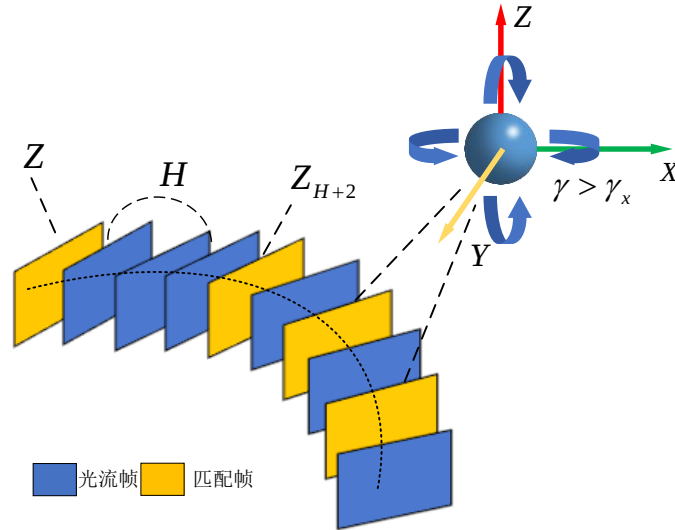


图 3-4 基于复合旋转量阈值的特征匹配示意图

每当机器人发生旋转时，都会增加复合旋转量的值。本章设置一种旋转阈值 γ_x ，当机器人在运行过程中，当前帧相较于上一匹配帧有 $\gamma > \gamma_x$ ，则对当前帧重新提取特征点，并设置为匹配帧。其次，在每次插入匹配帧后，将当前的复合旋转量设置为 0，当再次插入匹配帧之前必须再次满足阈值条件 $\gamma > \gamma_x$ ，进而减少系统的操作数量。该复合旋转量是两个匹配帧之间所有图像帧的复合旋转量之和，这反映了相机位姿在相邻两个匹配帧中发生的变化。相机位姿和场景中的特征点是相互关联的，通过相机位姿和特征点的位置，可以将相机观测到的特征点映射到三维空间中的坐标系中，并根据这些特征点构建场景地图。同时，在之后的运行过程中，特征点的位置信息也可以被用来辅助相机位姿的估

计。因此当第二类阈值用于大视角运动精准插入匹配帧时，会减少光流法产生的位姿估计误差较大的问题，并提高系统的跟踪精度。

本章节中融合光流与特征点的双阈值视觉里程计的步骤如算法 1 所示。

算法 1 融合光流与特征点的双阈值视觉里程计

输入:连续帧图像 Z_i

输出:当前机器人的位姿估计

参数:跟踪难度系数 η ，光流跟踪自适应次数 τ ， ω 为不添加新特征点的平均连续跟踪次数，复合旋转量 γ 给定旋转阈值 γ_x 。

FOR new Z_i available THEN

ORB 算法提取图像特征点，初始化特征匹配

执行光流跟踪并计算当前帧的光流跟踪难度系数 η 与复合旋转量 γ

$$\eta = \frac{\delta_{Z_1} - \delta_{Z_Q}}{\delta_{Z_1}} \text{ and } \gamma = \sum_{k=i}^{j-1} \|\phi_{k+1} - \phi_k\|, (i < j), \phi = \begin{bmatrix} p \\ d \end{bmatrix} \in \mathbb{R}^6$$

IF $\tau = (1 - \eta) \cdot \omega$ or $\gamma > \gamma_x$

当前帧重新提取特征点，计算描述符，执行特征匹配，估计当前机器人位姿

ELSE

继续执行光流跟踪，估计当前机器人位姿

END IF

END FOR

3.2 基于活跃特征点的关键帧选取策略

3.2.1 关键帧补偿

针对传统视觉 SLAM 算法进行大视角运动时选取的关键帧质量较差，关键帧上的特征点无法稳定跟踪，导致位姿估计累积误差增大以及系统定位精度下降的问题，本章提出一种关键帧补偿决策。该决策通过图像中包含的活跃特征点数量决定是否将该图像帧选取为新的关键帧，首先将特征点在跟踪线程中可以连续跟踪的帧数作为活跃值，然后根据每个特征点的活跃值过滤出活跃特征点，最后通过当前帧中活跃特征点的数量选取新关键帧。与基于图像模糊和像素亮度方法不同，本章将图像中活跃特征点数量作为判断该图像帧质量优劣的条件，活跃特征点数量越多，图像质量越高。通过这种方法选取的关键帧包含了更多的运动状态信息，进而提升算法轨迹精度。

为提高特征点跟踪的准确性与算法的定位精度，本文选择图像中更稳定和

持续时间更长的特征点为活跃特征点，该类特征点具有稳定值和活跃属性，能够简化跟踪匹配过程。此外，图像中有部分特征点会在系统操作期间被优化，使它们无法映射到地图上，从而在建图中不提供任何效用。对于这类特征点，即使达到规定的活跃值，也不作为活跃特征点。

当发生特征点持续出现在连续帧中并且相机一直可以观察到该特征点时，开始将活跃值添加到特征点属性中。将首次出现的特征点的活跃值设为 0，当此特征点出现在下一帧中时，此特征点的活跃值增加 1。相机内参矩阵为 C ，当前帧为 F_c ， F_n 为前 n 帧。两个相邻的帧有 N 个特征点，并且彼此特征对应。当前帧 F_c ，前一帧 F_1 与活跃特征点 F_{act}^* 表示为：

$$F_c = \{F_c^1, F_c^2, \dots, F_c^N\} \quad (3-7)$$

$$F_1 = \{F_1^1, F_1^2, \dots, F_1^N\} \quad (3-8)$$

$$F_{act}^* = F_{act}^k, k \in [1, N], act(F_{act}^k) \geq l \quad (3-9)$$

式中， $act(F_{act}^i)$ 表示特征点 F_{act}^i 的活跃值， l 表示成为活跃特征点的规定值，特征点 F_{act}^* 可以在帧集合 $\{F_1, \dots, F_l\}$ 中进行跟踪。本章将一帧中活跃特征点的数量指定为满足规定活跃值的所有特征点的总和如下式所示：

$$M = \sum_{i \geq l} \sum_{j \in (0, N)} f(F_i^j, l) \quad (3-10)$$

$$f(F_i^j, l) = \begin{cases} 1, act(F_i^j) \geq l \\ 0, act(F_i^j) < l \end{cases} \quad (3-11)$$

式中， i 表示可以跟踪的帧数， M 表示活跃特征点数量， $f(F_i^j, l)$ 表示特征点 $act(F_i^j) \geq l$ ， F_i^j 表示活跃特征点，该特征点可以连续跟踪 l 帧。每一帧内出现的活跃特征点的数量可以通过 $act(F_i^j) \geq l$ 计算出来。在筛选出活跃特征点后，计算当前帧的活跃特征点数量，达到阈值 S ，则满足新关键帧选取条件，此时插入活跃关键帧 $act(F_i)$ 。

活跃特征点判定策略示意图如图 3-5 所示。 F_1^1 和 F_c^1 相互特征对应，并在地图中能找到对应的地图点 p_1 ，满足相隔 l 帧后特征对应， F_c^1 为活跃特征点。而 F_1^2 和 F_{c-1}^2 相距帧少于 l 帧，不满足活跃特征点的筛选条件，所以 F_{c-1}^2 不作为活跃特征点。对于 F_1^3 和 F_c^3 ，特征点连续存在 l 帧，但在地图中并没有找到对应的

地图点，表明特征点 F_1^3 在系统运行时被优化。此时，特征点 F_1^3 被排除，不作为有效的活跃特征点。

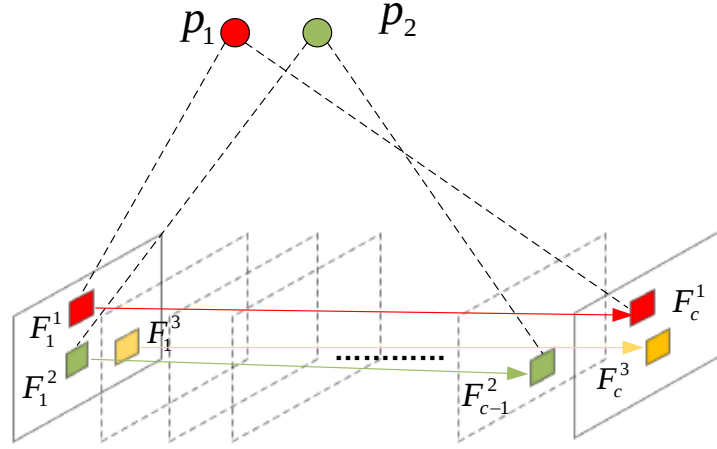


图 3-5 活跃特征点判定策略示意图

通常活跃特征点的数量小于传统特征点提取的数量，使用筛选后的活跃特征点作为选取新关键帧的条件能更精准的进行位姿估计，同时减少在遮挡处检测到的不稳定特征点所产生的系统误差，关键帧选取策略示意图如图 3-6 所示。

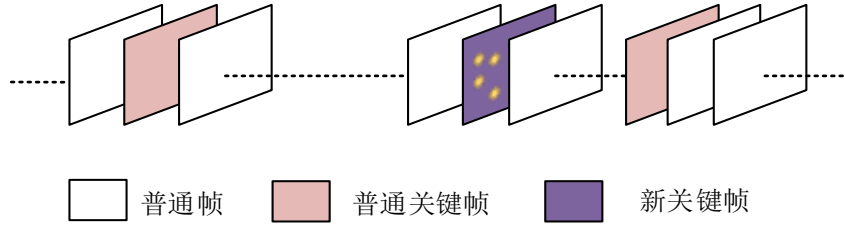


图 3-6 关键帧选取策略示意图

3.2.2 冗余关键帧剔除

为避免新选取的关键帧和普通关键帧重叠度过高，造成关键帧冗余问题，本节对关键帧的重叠程度进行评估。将当前关键帧与上一关键帧进行比较，计算两帧之间重叠的特征点比例，计算公式如下：

$$n_{ratio} = \frac{n_t}{n_{cur}} \quad (3-12)$$

式中， n_t 为当前关键帧和上一关键帧共有的地图点数， n_{cur} 为在当前关键帧中观察的地图点数。当比率接近于 1 轨迹上的映射点就越多，相反，越接近 0 轨迹上的映射点就越少。如果 $n_{ratio} > \lambda$ ， λ 为重叠度阈值，则允许该帧进入关键帧候选序列。基于活跃特征点的关键帧选取策略的步骤如算法 2 所示。

算法 2 基于活跃特征点的关键帧选取策略

输入:连续帧图像 F_i

输出:关键帧 $act(F_i)$

参数: 连续跟踪阈值 l , 新关键帧阈值 S , 关键帧重叠度 n_{ratio}

FOR new F_i available THEN

 定义图像帧 F_i^j 与当前帧 F_c^k , 计算当前帧特征点活跃值 $act(F_c^k)$, 选取活跃特征点 F_{act}^*

 IF $act(F_c^k) \geq l$

 则有 $F_{act}^* = F_c^k, k \in [1, N]$

 ELSE

 当前帧 F_c^k 中第 k 点不作为活跃特征点

 return null

 计算图像帧 F_i^j 中活跃特征点的数量 M

$$M = \sum_{i \geq l} \sum_{j \in (0, N]} f(F_i^j, l), f(F_i^j, l) = \begin{cases} 1, act(F_i^j) \geq l \\ 0, act(F_i^j) < l \end{cases}$$

 IF $M > S$, F_i^j 为新关键帧 THEN

 比较当前关键帧与上一关键帧的重叠度, $n_{ratio} = \frac{n_t}{n_{cur}}$

 IF $n_{ratio} > \lambda$

 则插入新关键帧, 进入关键帧候选序列

 ELSE

 return null

 END IF

END IF

END FOR

3.3 实验结果与分析

本章实验电脑配置为: CPU 为 InterCorei5-9400F, 6 核处理器, 主频 2.9 GHz, 运行内存 8GB, 系统为 Ubuntu18.04 为验证本章算法的有效性, 本章使用 TUM 公开数据集对所提策略进行验证, 该数据集包含的大视角场景较多, 且光照变化不明显, 满足本章算法的实验环境。

3.3.1 融合光流法与特征点法的有效性实验分析

为验证本章算法中融合光流与特征点的双阈值视觉里程计的有效性, 将 ORB-SLAM2, ORB-LK-SLAM 和本章算法实际运行中跟踪线程的直观图展示出来。选取 TUM 数据集 fr2/desk 中的 4 帧图像对本章算法进行验证, 如图 3-7 所

示，绿色图像为执行特征提取与匹配的匹配帧，蓝色图像为使用光流跟踪的光流帧。如图 3-7 中(a)-(d)所示为 ORB-SLAM2 算法运行效果图，从实验结果中能看出，该 4 帧进行了特征提取，执行了完整的特征匹配。图 3-7(e)-(h)为 ORB-LK-SLAM 运行效果图，将光流法与特征点相结合，在固定帧如图 3-6(g)中进行特征提取，表明了该算法的可行性。图 3-7(i)-(l)为本章算法运行效果图，由于执行了双阈值约束特征匹配策略，相比 ORB-LK-SLAM 只在第 3 帧执行了特征匹配，本章算法能够根据当前图像帧的光流跟踪难度与复合旋转量自适应执行特征匹配，如该算法在第 2 帧与第 3 帧均进行了特征提取，并与上一匹配帧进行特征匹配，有效提高了系统的跟踪精度。

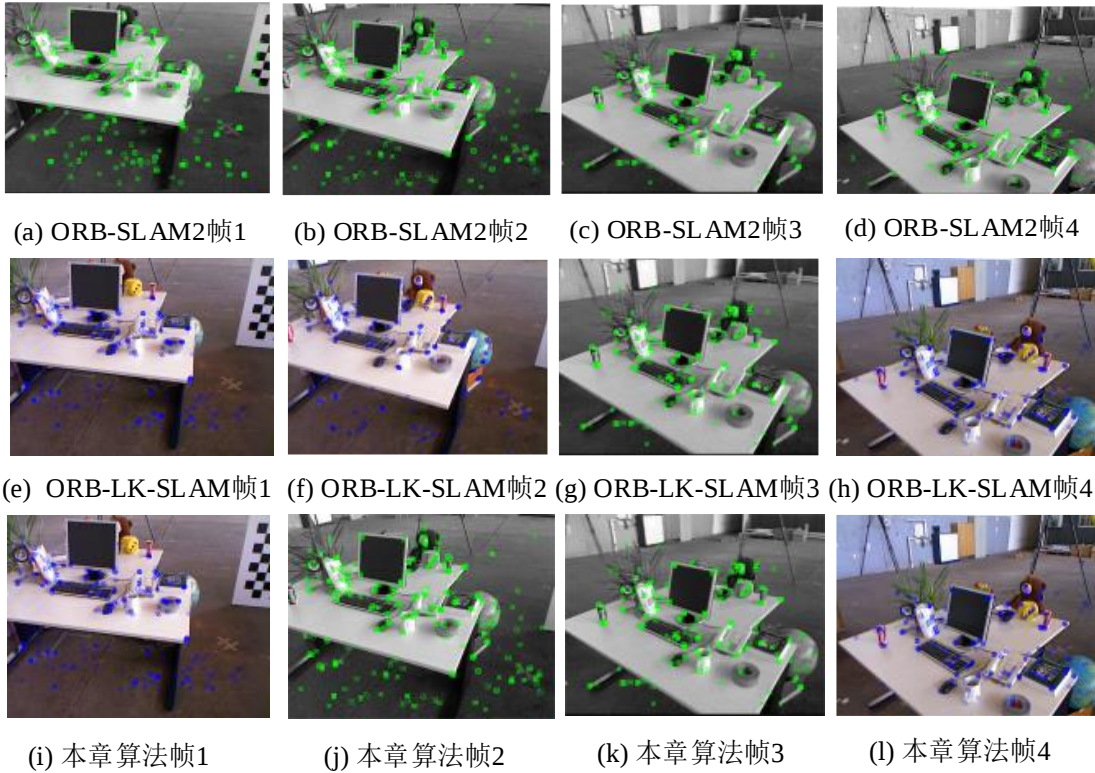


图 3-7 3 种算法跟踪线程运行直观图

为了更进一步验证本章算法的实时性，3 种算法跟踪线程特征匹配帧数 f_n 与平均每帧处理时间 t_s 选取对比结果如表 3-1 所示。其中，ORB-SLAM2 算法在 SLAM 系统运行过程中，由于每帧均执行特征匹配，导致平均每帧计算时间最长为 0.0262s，ORB-LK-SLAM 采用了光流与特征点法融合策略，在进行图像跟踪时每 20 帧执行特征匹配，该策略可以提高系统的实时性，但在发生大视角变化时算法会产生跟踪丢失或者特征匹配冗余等问题。而本章算法在跟踪线程中执行双阈值约束自适应特征匹配策略后，能在特征点跟踪丢失时自适应插入匹

配帧，并且平均每帧处理时间 t_s 与 ORB-SLAM2 相比降低了 47.7%,与 ORB-LK-SLAM 相比不超过 4%，算法实时性较好，因此本文算法在大视角运动场景长时跟踪优势明显。

表 3-1 3 种算法跟踪线程的特征匹配帧数与平均每帧处理时间选取对比结果

数据集	图像数量	ORB-SLAM2		ORB-LK-SLAM		本章算法	
		f_n /特征匹配	t_s /s	f_n /特征匹配	t_s /s	f_n /特征匹配	t_s /s
fr1/desk1	613	613	0.0294	31	0.0145	59	0.0159
fr1/desk2	640	640	0.0263	32	0.0152	67	0.0165
fr2/desk	2965	2965	0.0282	148	0.0138	95	0.0126
fr2/xyz	3669	3669	0.0209	183	0.0094	235	0.0102
fr3/l/c	1011	1011	0.0215	50	0.0134	44	0.0122
fr3/l/o/h	2586	2586	0.0311	129	0.0128	179	0.0145
平均值	1914	1914	0.0262	96	0.0132	113	0.0137

3.3.2 关键帧选取实验

为验证基于活跃特征点的关键帧选取策略的有效性，本章节选取 TUM 数据集下三个大视角场景较多的序列进行实验。3 种算法运行时关键帧选取效果对比如图 3-8 所示，蓝色方框为选取的关键帧，绿色方框为相机运行轨迹，其中 ORB-LK-SLAM 算法在曲线拐点处关键帧较为稀疏，未能形成对真实轨迹的细致还原，从而导致地图构建的精度较差。ORB-SLAM2 算法依赖特征匹配生成相机位姿，然而运行在 fr1/desk、fr1/desk2 序列中时，选取的关键帧数量变少，造成局部出现了轨迹误差，使得系统产生了较大漂移误差。ORB-SLAM2 算法在 fr2/desk 序列中剧烈弯道运动时出现特征跟踪丢失且未完成轨迹的重新寻回，缺失了部分相机的运行轨迹，从而无法高效定位。本章算法利用关键帧补偿决策，使选取的关键帧在面对任何场景时都能被均匀选取，尤其在大视角运动甚至剧烈转弯时选取了高质量的图像作为关键帧，生成了较为精确的轨迹地图，提高了 SLAM 系统运行的鲁棒性及准确性。

3 种算法关键帧选取对比结果如表 3-2 所示，本章算法由于采取关键帧补偿决策提高了在大视角运动下（如图 3-8 红色虚线所示运动过程）中有效关键帧的质量及数量，与 ORB-SLAM2 和 ORB-LK-SLAM 关键帧选取策略相比有效关键

帧数量得到提升，且本章所提方法在此阶段绝对轨迹均方根误差 RMSE 的平均值与 ORB-SLAM2 和 ORB-LK-SLAM 相比分别降低了 35.4%和 47.8%，减小了局部轨迹误差，提升了系统定位精度，表现了该决策在大视角运动过程中关键帧选取的优越性。

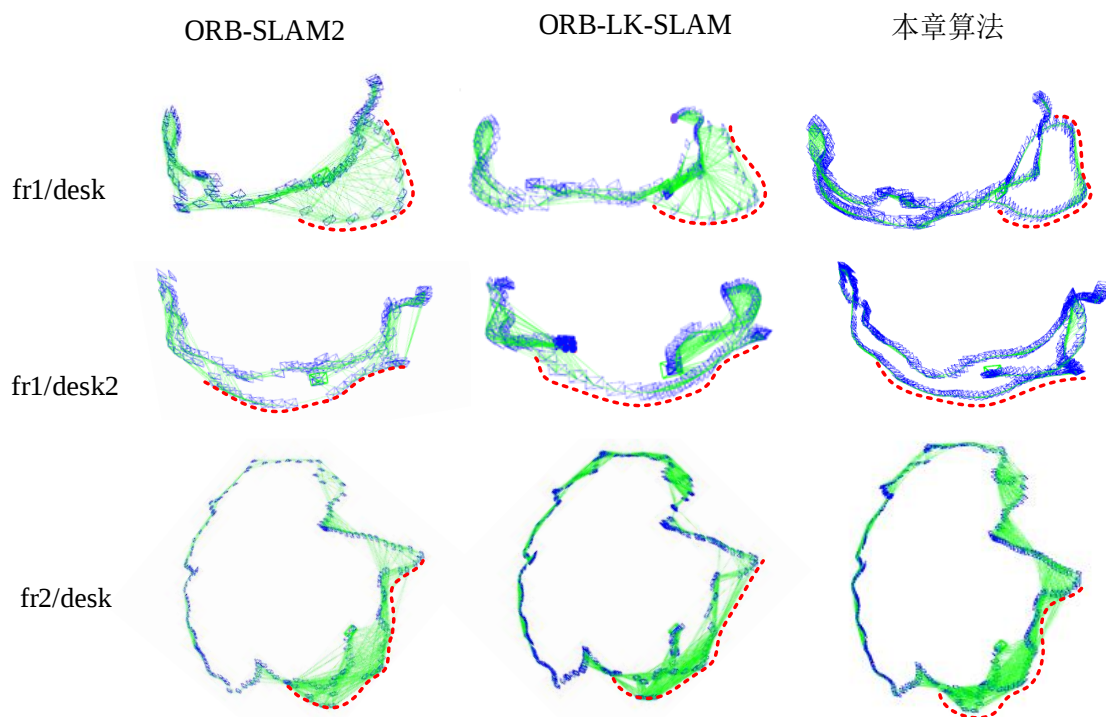


图 3-8 3 种算法运行时关键帧选取效果对比

表 3-2 3 种算法关键帧选取对比结果

数据集	图像数量	ORB-SLAM2		ORB-LK-SLAM		本章算法	
		关键帧 (个)	RMSE(m)	关键帧 (个)	RMSE(m)	关键帧 (个)	RMSE(m)
fr1/desk	613	14	0.009565	10	0.010591	33	0.004522
fr1/desk2	640	23	0.006832	20	0.009235	46	0.002459
fr2/desk	2965	39	0.004904	36	0.006908	52	0.004930
fr2/xyz	3669	46	0.002421	53	0.001262	63	0.000420
fr3/l/c	1011	22	0.009526	29	0.010288	44	0.008609
fr3/l/o/h	2586	36	0.007409	24	0.012042	57	0.005312
平均值	1914	30	0.006776	29	0.008388	49	0.004375

3.3.3 系统定位精度与轨迹对比实验

3 种算法在不同序列下的平移误差和旋转误差对比如表 3-3 所示，本章算法在视觉里程计上提出的双阈值约束特征匹配策略，提高了算法的泛用性以及复杂环境中的定位精度，并且提出的基于活跃特征点的关键帧选取策略不仅在大

视角运动中有良好的表现，在平缓场景中也同样适应，使得系统在应对不同场景时都能稳定精准的运行。从表中数据可知，本文算法的平移误差与相比 ORB-SLAM2 减少了 36.4%，与相比 ORB-LK-SLAM 减少了 67.4%。同时，本文算法的旋转误差与 ORB-SLAM2 相比减少 17.3%，与 ORB-LK-SLAM 相比减少 48.5%。

表 3-3 三种算法在 TUM 数据集下的平移误差和旋转误差对比

TUM sequence	ORB-SLAM2		ORB-LK-SLAM		本章算法	
	平移误差 (m)	旋转误差 (°)	平移误差 (m)	旋转误差 (°)	平移误差 (m)	旋转误差 (°)
fr2/desk	0.007539	1.016275	0.013856	0.975936	0.006845	0.935919
fr2/xyz	0.003297	1.438558	0.003094	1.521360	0.003015	1.239391
fr3/l/c	0.074512	1.846059	0.053857	2.299539	0.042618	1.395482
fr3/l/o/h	0.007609	0.575130	0.110379	3.033221	0.006596	0.459654
平均值	0.023239	1.219006	0.045297	1.957514	0.014769	1.007612

为进一步验证本章算法在大视角运动下的性能，在相同实验环境下与 ORB-SLAM2、ORB-LK-SLAM 的运行轨迹做出对比。3 种算法在 TUM 数据集下的绝对轨迹误差的对比如表 3-4，由于本章算法改进了视觉里程计和关键帧选取策略，使得算法的绝对轨迹误差相比 ORB-SLAM2 减少了 31.8%，相比 ORB-LK-SLAM 减少了 60.1%，有效提升了系统的定位精度。

表 3-4 三种算法在 TUM 数据集下的绝对轨迹误差的对比(单位: m)

TUM sequence	ORB-SLAM2	ORB-LK-SLAM	本章算法
fr2/desk	0.009310	0.015473	0.006945
fr2/xyz	0.003666	0.003783	0.003307
fr3/l/c	0.090140	0.060393	0.059426
fr3/l/o/h	0.008152	0.110728	0.006219
平均值	0.027817	0.047594	0.018974

本章节实验选取的序列 fr2/desk 和 fr2/xyz 具有较多的旋转场景，TUM 数据集序列轨迹对比图如图 3-9 所示，图 3-9 (a) 中 1 和图 3-9 (b) 中的红色区域存在旋转运动，ORB-LK-SLAM 在此场景下的定位精度较差，产生了轨迹漂移，而 ORB-SLAM2 在图 3-9 (a) 的 1 和 2 也表现出了较大的轨迹误差。本章算法在达到复合旋转量阈值时执行特征匹配并结合关键帧补偿决策，使得算法获取的轨迹图更加接近真实轨迹。图 3-9 (c) 为三种算法在平缓序列 fr3/l/c 下的轨迹对比图，本章算法与 ORB-SLAM2 的轨迹接近真实轨迹，但在图 3-9 (c) 的 1 和 2 中 ORB-SLAM2 算法出现了轨迹震荡，表明算法的鲁棒性相较于 ORB-SLAM2 更高。从图 3-9 (d) 中 fr3/l/o/h 序列的红色区域可以看出 ORB-SLAM2

在大视角运动时轨迹漂移的问题，本章算法在实行基于活跃特征点的关键帧选取策略后，选取新关键帧进行轨迹重建，使得本章算法获取的轨迹更加精准。

从图 3-9 中四个序列轨迹对比可以直观看出，ORB-SLAM2 与 ORB-LK-SLAM 算法存在较大的轨迹误差，而本章算法与真实的运动轨迹有较高的吻合度，且在大视角运动等场景下优势更为明显，说明所估计的轨迹误差相对更小，证明了本章算法相较于传统的视觉 SLAM 算法有着更好的鲁棒性与定位精度。

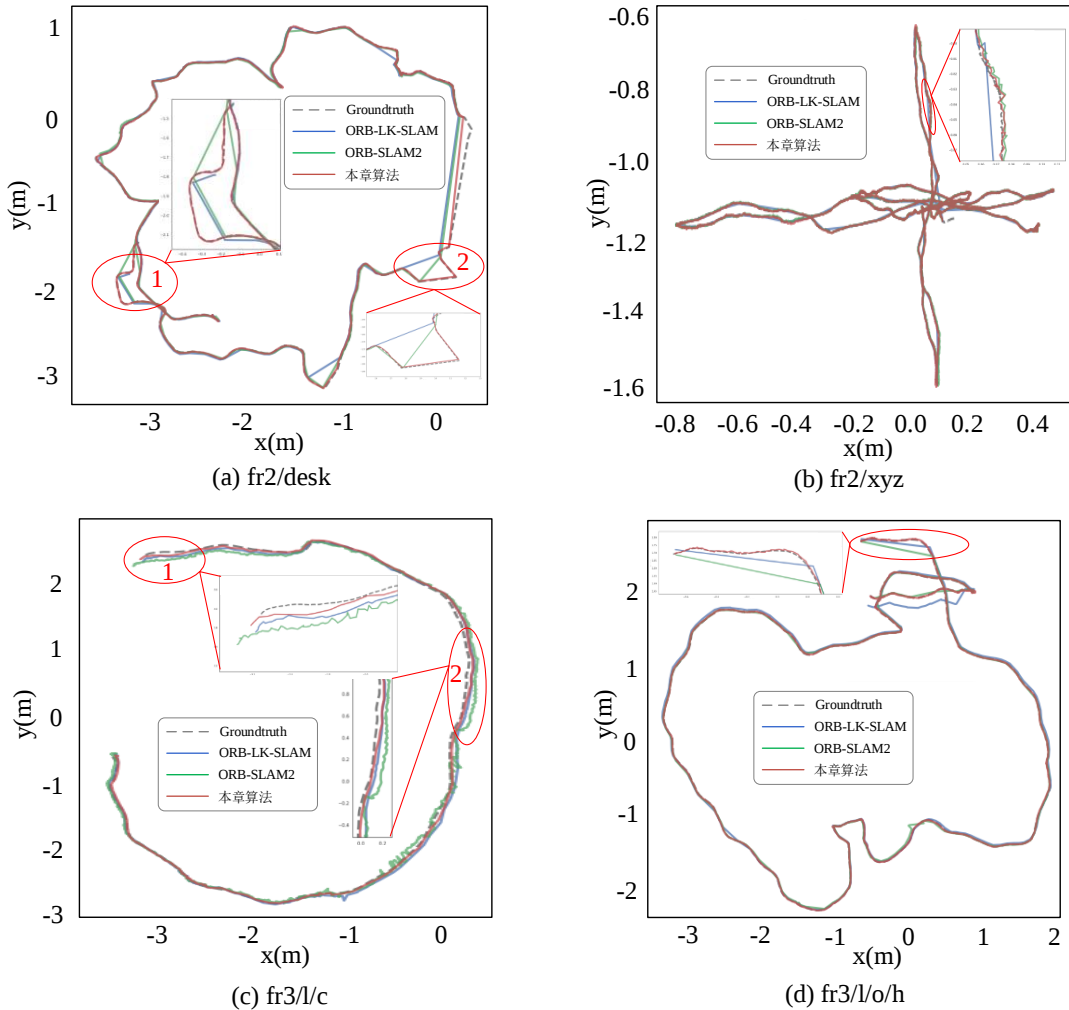


图 3-9 3 种算法在 TUM 数据集下的序列轨迹对比图

3.4 本章小结

本章提出了一种改进视觉里程计与关键帧选取的视觉 SLAM 算法，该算法具有以下优点。首先，提出了一种融合光流与特征点的双阈值视觉里程计，通过在跟踪线程实行双阈值约束特征匹配策略，解决了视觉 SLAM 算法特征匹配的时效过低和传统光流与特征点融合算法跟踪精度较低等问题。其次，提出了

基于活跃特征点的关键帧选取策略，通过帧中活跃特征点的数量来选取高质量图像作为新关键帧，解决了视觉 SLAM 算法在面对大视角运动时位姿估计误差较大导致无法精确定位以及轨迹漂移的问题，提高了系统的定位精度和鲁棒性。最后，在 TUM 公开数据集下进行实验，结果表明本章算法与 ORB-SLAM2、ORB-LK-SLAM 算法相比，系统的定位精度和实时性有较大提升，展现出良好的构图能力。

第 4 章 基于 BWO 闭环检测的视觉 SLAM 算法

白鲸优化算法（BWO）是一种群智能优化算法，具有较好的全局搜索能力和收敛性能，通过模拟白鲸的觅食行为和社会行为，能够在搜索空间中进行全局搜索，并逐步收敛到最优解附近。本章在第 3 章改进视觉里程计和关键帧选取策略的基础上对闭环检测进行研究。针对传统闭环检测方法在面对复杂场景时检测效率较低且易产生闭环误检测等问题，本章提出基于 BWO 闭环检测的视觉 SLAM 算法，将闭环检测问题转化为最大值优化问题。首先设计出合适的适应度函数，然后再设计出该优化问题的 BWO 算法求解最大值问题，最后提出一种候选帧检测方法，进而提高算法在闭环检测时的运行效率和精度。算法整体框架如图 4-1 所示，其中本章算法包括 BWO 候选帧选取和候选帧检测两部分。

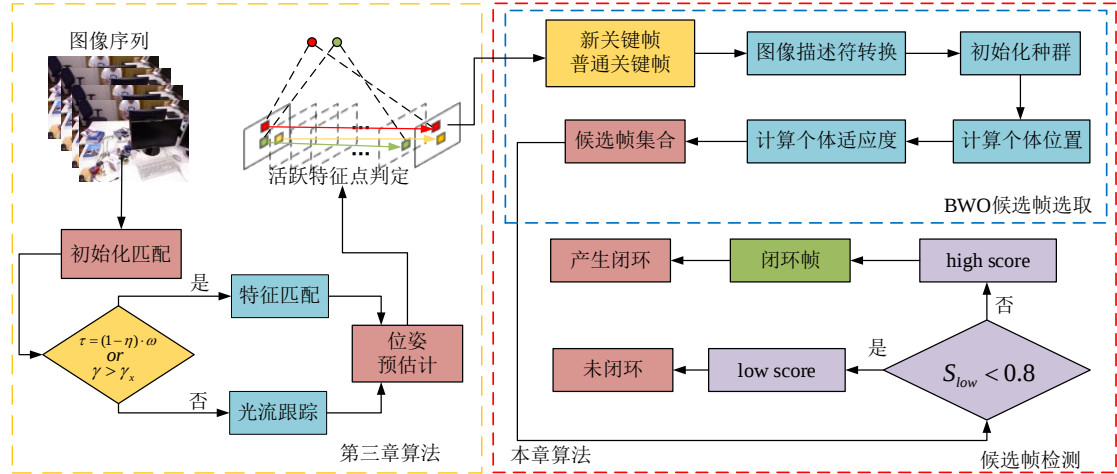


图 4-1 本章算法整体框架

4.1 BWO 算法概述

BWO 算法包含了探索、开发和鲸鱼坠落的三个阶段，并且分别对应着白鲸的游泳，捕食和坠落等自然行为。该算法包含两个自适应变量和 W_f ，其中 B_f 为平衡因子，决定白鲸进入探索阶段还是开发阶段； W_f 为鲸落概率，决定白鲸是否进入鲸落阶段。 B_f 和 W_f 的表达式如下：

$$B_f = B_0 \left(1 - \frac{t}{2T}\right) \quad (4-1)$$

$$W_f = 0.1 - 0.05t / T \quad (4-2)$$

其中， t 为当前迭代次数， T 为最大迭代数， B_0 在每次迭代时，其数值将

在(0,1)的范围内随机改变。当平衡因子 $B_f > 0.5$ 白鲸处于探索阶段，而 $B_f \leq 0.5$ 时转向开发阶段。随着迭代 t 的增加，进入开发阶段的几率也逐渐增大。如果 $B_f < W_f$ ，则进入鲸落阶段。

(1) 探索阶段

探索阶段基于白鲸游泳行为建立。在进行成对游泳时，两只白鲸能够以同步或者镜像的形式紧密地协同行动。

通过该行为，建立探索阶段下白鲸位置更新的数学模型：

$$\begin{cases} X_{i,j}^{t+1} = X_{i,p_j}^T + (X_{r,p_1}^T - X_{i,p_j}^T)(1+r_1)\sin(2\pi r_2), & j = \text{even} \\ X_{i,j}^{t+1} = X_{i,p_j}^T + (X_{r,p_1}^T - X_{i,p_j}^T)(1+r_1)\cos(2\pi r_2), & j = \text{odd} \end{cases} \quad (4-3)$$

其中， t 为当前迭代次数， $X_{i,j}^{t+1}$ 是第 i 条白鲸在第 j 维上的新位置， $P_j (j=12\dots,d)$ 是从 d 维中随机选择的， $X_{i,j}^t$ 是第 i 条白鲸在第 j 维度上的位置， X_{i,p_j}^t 和 X_{r,p_1}^t 是第 i 和第 r 条白鲸(r 代表随机选择的白鲸)的当前位置， r_1 和 r_2 为(0,1)的随机数，是用于增强探索阶段的随机算子。 $\sin(2\pi r_2)$ 和 $\cos(2\pi r_2)$ 为平均鱼鳍之间的随机数。通过奇偶数选定维度，更新位置模拟出白鲸在进行游泳或潜水活动时展现的同步或镜像行为。

(2) 开发阶段

开发阶段基于白鲸捕食行为建立。白鲸通过分享彼此的位置信息来合作觅食，此阶段引入了 Levy 飞行策略捕捉猎物，提高算法收敛性。其数学模型表示为：

$$X_i^{t+1} = r_3 X_{best}^t - r_4 X_i^t + C_1 \cdot L_F \cdot (X_i^t - X_i^t) \quad (4-4)$$

式中， T 为当前迭代次数， X_i^t 和 X_r^t 是第 i 条白鲸和一条随机白鲸的当前位置， X_i^{t+1} 是第 i 条白鲸的新位置， X_{best}^t 是鲸鱼中最优位置， r_3 和 r_4 是范围在(0,1)的随机数， $C_1 = 2r_4(1-t/T_{\max})$ 用于测量 Levy 飞行强度的随机跳跃强度。 L_F 为 Levy 飞行函数，其计算方法如下：

$$L_F = 0.05 \times \frac{u \times \sigma}{|v|^{1/\beta}} \quad (4-5)$$

$$\sigma = \left(\frac{\Gamma(1+\beta) \times \sin(\pi\beta/2)}{\Gamma((1+\beta)/2) \times \beta \times 2^{(\beta-1)/2}} \right)^{1/\beta} \quad (4-6)$$

其中， u 和 v 是正态分布的随机数， β 是默认的常数，等于 1.5。

(3) 鲸鱼坠落

鲸落阶段基于白鲸坠落行为建立。当白鲸受到外界威胁时，可以通过彼此分享信息来逃避威胁。然而，仍然有少数白鲸无法幸存，这被称为“鲸鱼坠落”。根据白鲸所处位置以及鲸鱼坠落过程中的步长，构建一个数学模型用于描述白鲸在其坠落阶段位置变化的过程：

$$X_i^{t+1} = r_5 X_i^t - r_6 X_r^t + r_7 X_{step} \quad (4-7)$$

其中， r_5 、 r_6 和 r_7 是(0,1)之间的随机数， X_{step} 是鲸鱼坠落的步长，确定为：

$$X_{step} = (u_b - l_b) \exp(-C_2 \frac{t}{T}) \quad (4-8)$$

其中， C_2 为与鲸落概率和种群大小相关的步长因子 ($C_2 = 2W_f \times n$)， u_b 和 l_b 分别为变量的上界和下界。这个模型中，鲸鱼坠落的概率 W_f 为一个线性函数：

$$W_f = 0.1 - 0.05t / T \quad (4-9)$$

当鲸落概率由初始的 0.1 减少至 0.05 时，表明白鲸在寻优过程中越靠近食物源，其面临的风险越少。根据 BWO 算法的终止条件，通过不断地更新白鲸个体位置，计算相应的适应度值，并以此判断是否达到适应度阈值。若达到则算法结束，否则继续执行。如果迭代次数达到上限，算法便停止，并得到最优解。

4.2 基于 BWO 算法的闭环检测

针对传统闭环检测方法在面对复杂场景时效率较低且易发生闭环误检测等问题，本章提出一种基于 BWO 算法的闭环检测方法。首先设计优化模型与适应度函数，建立 BWO 算法和闭环检测相对应的数学模型，从而将闭环检测问题转化成最大值优化问题，然后给出 BWO 算法闭环检测实现方法的步骤，通过种群适应度函数计算出局部最优个体位置，得到相似度最高的候选图像，最后提出一种候选帧检测方法，降低非相似图像对系统的干扰并简化相似度比较过程，减少了系统计算量和闭环误检测问题的发生。

4.2.1 优化模型与适应度函数

本章对提取的图像特征进行描述子计算，再将其压缩成鲁棒描述子，以增强特征稳定性，并用鲁棒描述子的相似性表示图像相似度。

在本章节中，采用 BRIEF 算法，此过程主要包括以下三个步骤：

(1) 在图像帧中选取特征点 $p=(u,v)$ ，以该点为圆心画圆。同时以该圆的半径为标准，获取以点 p 为中心的正方形区域 S 。为方便后续处理，将区域 S 大小设置为 31×31 ，并对区域 S 进行高斯模糊；

(2) 在经过处理的区域 S 中，依据高斯分布原理选取点对，点对数量设置为 256。为获取点对之间的联系，对于其中的点对 (x,y) ，获取 $PV(x)$ 和 $PV(y)$ ，若 $PV(x) > PV(y)$ ，将该点对之间的联系记为 1，反之，记为 0。其中， $PV(\cdot)$ 表示获取像素值函数；

(3) 对 (2) 中所有点对进行比较，并将比较结果按顺序排序得到字符向量，从而得到以特征点 p 为先验条件的描述子；

为获取更为鲁棒的描述子，本章利用 PCA 算法对经过上述三个步骤得到的描述子进行降维，同时进行压缩。

对于当前帧 A 和历史帧库 $\{B_1, B_2, \dots, B_N\}$ ，构建出当前帧 A 与历史帧库的关系模型。该模型通过计算图像相似度，并获取与当前帧相似度最高的历史帧，具体计算方式如公式所示(4-10)所示：

$$\max \text{SIMY}(\text{BRF}(A), \text{BRF}(B_j)), j = 1, \dots, N \quad (4-10)$$

式中， $\text{SIMY}(\cdot)$ 为计算两幅图像相似度的函数， $\text{BRF}(\cdot)$ 获取图像帧的鲁棒描述子函数， N 为历史帧图像的数目。

本章将 BWO 算法种群中的每个对象构造图像帧，并依据描述子的表达形式，将 BWO 算法中的每个个体 X 表示为：

$$X = \{i_{\text{index}}, D(i_{\text{index}})\} \quad (4-11)$$

式中， i_{index} 表示图像索引序号， $D(\cdot)$ 表示获取序号图像帧的鲁棒描述子向量。

为提升回环检测效率，本章引入了适应度函数。适应度函数主要构建图像间鲁棒描述子的欧氏距离模型，以此衡量图像间的相似度，适应度函数的构建过程如下：

(1) 对于图像 A_1 和图像 A_2 ，对应的鲁棒描述子分别为 $\text{BRF}(A_1)$ 和 $\text{BRF}(A_2)$ ，则图像 A_1 和图像 A_2 可构建出以下约束，如公式所示：

$$\text{dis}(\text{BRF}(A_1), \text{BRF}(A_2)) = \sqrt{\sum_{i'=1}^n (d_{1,k} - d_{2,k})^2} \quad (4-12)$$

其中， $\text{BRF}(A_1) = \{d_{1,1}, d_{1,2}, \dots, d_{1,n}\}$ ， $\text{BRF}(A_2) = \{d_{2,1}, d_{2,2}, \dots, d_{2,n}\}$ ， $d_{i,j}$ 为描述子向

量里面的元素，通常为 0 或 1。

(2) 构建出图像 A_1 和图像 A_2 间的相似度，如式(4-13)所示：

$$s_1(A_1, A_2) = \frac{1}{dis(BRF(A_1), BRF(A_2))} \quad (4-13)$$

对公式(4-12)，进一步优化公式，如公式(4-13)所示：

$$s_2(A_1, A_2) = \frac{1}{1 + dis(BRF(A_1), BRF(A_2))} \quad (4-14)$$

(3) 将图像的相似度函数作为适应度函数，如式(4-15)所示：

$$F = s_2(A_1, A_2) \quad (4-15)$$

由式（4-15）可知式（4-12）值越小，适应度值越大，图像相似度越大。

4.2.2 基于 BWO 算法的候选帧选取

为了提高闭环检测的效率并获得准确的闭环候选图像，本章采用 BWO 算法将闭环检测问题转化为最大值优化问题，用种群适应度函数计算代替传统基于词袋模型的相似度计算。首先，将图像的描述子转化成图像鲁棒描述子；其次对白鲸个体进行图像编码，随机生成种群个体位置，初始化种群并通过白鲸的平衡因子 $B_f(i)$ 得出白鲸当前所处的阶段，得到白鲸在各个阶段的个体位置；最后计算白鲸个体的适应度 F_{fit} ，得出最优个体位置，并将最优个体位置作为相应的相似度最高的图像。基于 BWO 算法的候选帧选取的步骤如算法 3 所示。

算法 3 基于 BWO 算法的候选帧选取

输入：当前帧图像、历史帧图像和序列号

输出：相似图像编号 i_{best_index}

参数：设定初始种群的个体数量 Q ，最大迭代次数 D_{max} 和全局最优适应度阈值 T

提取当前帧图像和历史帧图像的特征并将其压缩为鲁棒性描述符。

随机产生 Q 个个体位置（历史帧图像序列号），并初始化种群。

While $t < D_{max}$ do

 由式（4-1）得到平衡因子 B_f ，由式（4-9）得到鲸鱼坠落概率 W_f

 For $i=1$ to Q do

 根据公式（4-15），计算 F_{fit} ，将适应度值最大的个体位置作为最优个体位置。

 End

 If ($F_{fit} > T$)

 输出最优个体位置，即 i_{best_index} ；

 Break

 End

For $i=1$ to Q do

续算法 3 基于 BWO 算法的候选帧选取

```

If  $B_f(i) > 0.5$  then
    使用公式 (4-3) 更新第  $i$  个白鲸的新位置;
Else if  $B_f(i) \leq 0.5$  then
    更新随机跳跃强度  $C_1$ , 计算 Levy 飞行函数;
    使用公式 (4-4) 更新第  $i$  个白鲸的新位置;
End if
    检查新位置的边界, 并评估适应度值;
End For
For  $i = 1$  to  $Q$  do
    If  $B_f(i) \leq W_f$  then
        使用公式 (4-8) 更新第  $i$  个白鲸的新位置;
        检查新位置的边界, 并评估适应度值;
    End if
End For
求最优个体位置  $i_{best\_index}$ 
 $t = t + 1$ 
End While
输出与当前帧图像相似的所有候选帧图像编号集合

```

4.2.3 候选帧检测

依据 BWO 检测结果预测当前图像是否构成闭环, 若预测显示当前图像不涉及闭环, 则无需将其与之前出现的图像逐一对照比较, 若当前图像被预测为闭环, 则对当前图像的闭环候选集进一步的预测分析。在此基础上提出候选帧检测方法, 该方法将闭环图像的搜寻限定在一系列连续且可能性更高的图像集之中, 降低了逐帧对比的数量, 并减少了闭环误检测的发生, 从而有效地提升了闭环检测的效率与准确率。

候选帧检测得分判断图如图 4-2 所示, 若当前图像与之前所有图像的适应度得分均低于特定阈值 S_{low} 时, 该图像被视为 low score 图像。反之, 高于阈值时 S_{low} , 该图像被视为 high score 图像, 则可判定为闭环帧。经多次实验可知阈值 S_{low} 取 0.8 时闭环候选帧相似度最高。

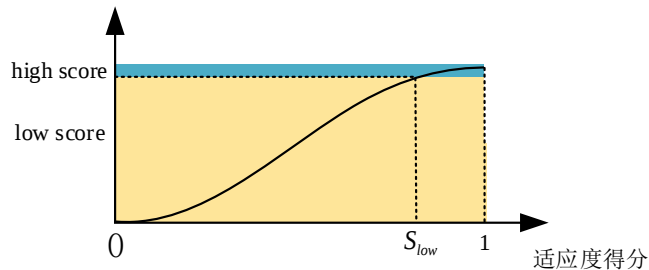


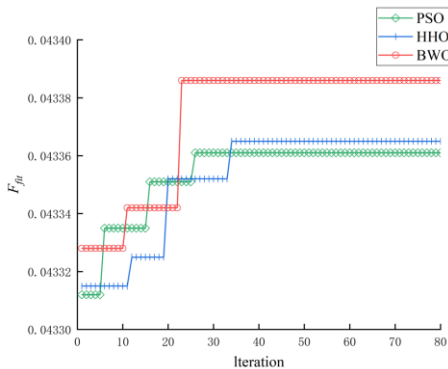
图 4-2 候选帧检测得分判断图

4.3 实验设计与结果分析

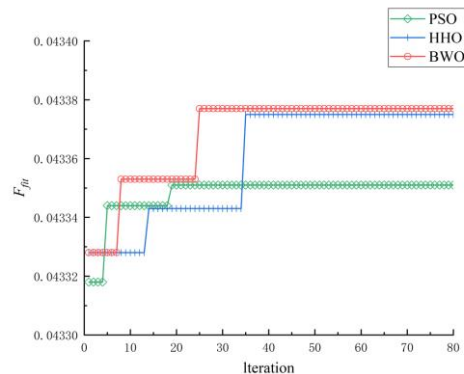
本章实验所用的电脑软硬件配置：CPU 为 InterCorei5-9400F，6 核处理器，主频 2.9GHz，运行内存 8GB，系统为 Ubuntu18.04。本章使用 KITTI 和 TUM 公开数据集进行验证。

4.3.1 基于 BWO 算法的闭环检测效率实验

本章节选取室内数据集 TUM 的 fr1/desk1 序列与室外数据集 KITTI 的 02 序列进行实验，两种序列包含较多的闭环场景，且运动轨迹较为复杂，符合本章算法实验条件。为验证 BWO 算法的有效性，本章节将 BWO 算法与粒子群优化算法（Particle Swarm Optimization, PSO），哈里斯鹰算法（Harris Hawk Optimization, HHO）以及 BOW 词袋模型方法进行闭环检测的相关实验验证。本章节 BWO 算法中参数设置：使用 fr1/desk1 序列时种群规模 Q 设为 80，使用 KITTI/02 序列时种群规模 Q 设为 100；最大迭代次数 $D_{\max}$ 设为 80；适应度阈值 T 设为 0.1。在 fr1/desk1 序列下，本章所提方法与基于 PSO 算法和基于 HHO 算法实施的闭环检测的适应度收敛曲线如图 4-3 所示。在 KITTI/02 序列下，本章所提方法与基于 PSO 算法和基于 HHO 算法实施的闭环检测的适应度收敛曲线如图 4-4 所示。从图 4-3（a）和图 4-4（a）的三种算法比较中可以看出，本章所提的方法不仅适应度更高，而且在收敛速度上也更快。从图 4-3（b）和图 4-4（b）可以看出，虽然三种算法在适应度上相近，但是本文所提方法在收敛速度上明显快于 PSO 算法和 HHO 算法。实验结果充分证明了本章所提方法在效率和精度上的优势。

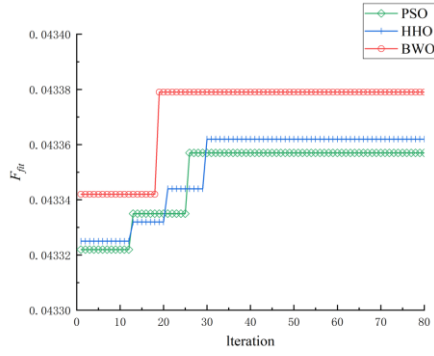


（a）图像序号为 497 适应度收敛曲线

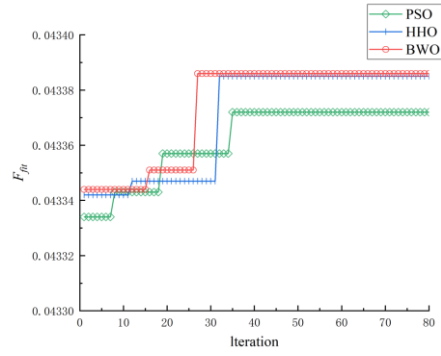


（b）图像序号为 1295 的适应度收敛曲线

图 4-3 fr1/desk1 下的适应度收敛曲线



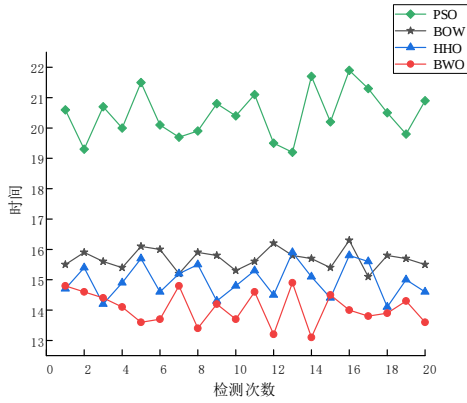
(a) 图像序号为 207 的适应度收敛曲线



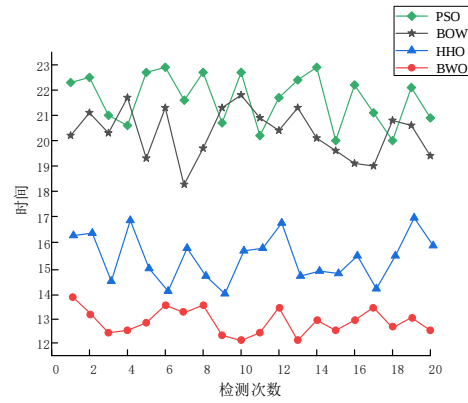
(b) 图像序号为 319 的适应度收敛曲线

图 4-4 KITTI/02 序列下的适应度收敛曲线

从图 4-5 和图 4-6 的数据对比中可以看出本章提出的方法耗时最少，并且，由图 4-5 (a)、图 4-5 (b) 和图 4-6 (b) 可见，本章所提闭环检测方法的时间波动程度较低，说明其具有较好的稳定性。从实验结果来看，不论是在上述哪个数据集中，本章所提方法都显示了较短的检测时间，这说明了本章所提基于 BWO 算法的闭环检测方法在效率上具的优越性。

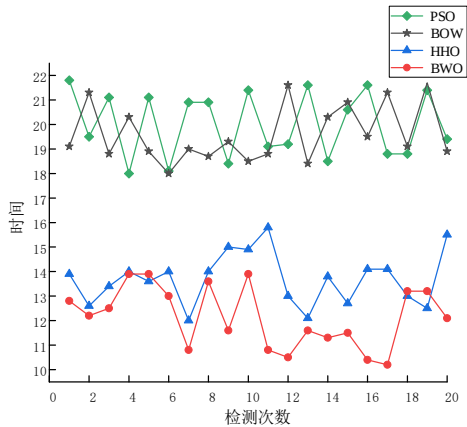


(a) 图像序号为 497 的检测时间

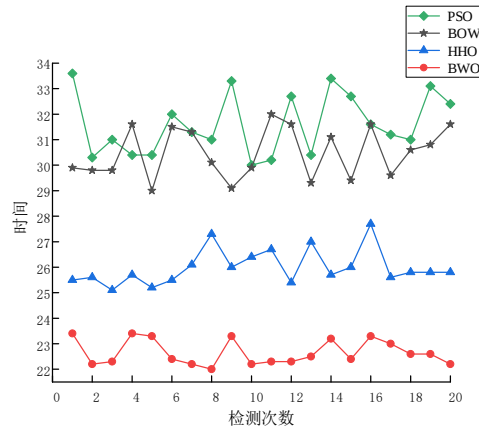


(b) 图像序号为 1295 的检测时间

图 4-5 三种闭环检测方法在 fr1/desk1 下的时间对比图



(a) 图像序号为 207 的检测时间



(b) 图像 319 序号为的检测时间

图 4-6 三种闭环检测方法在 KITTI/02 序列下的时间对比图

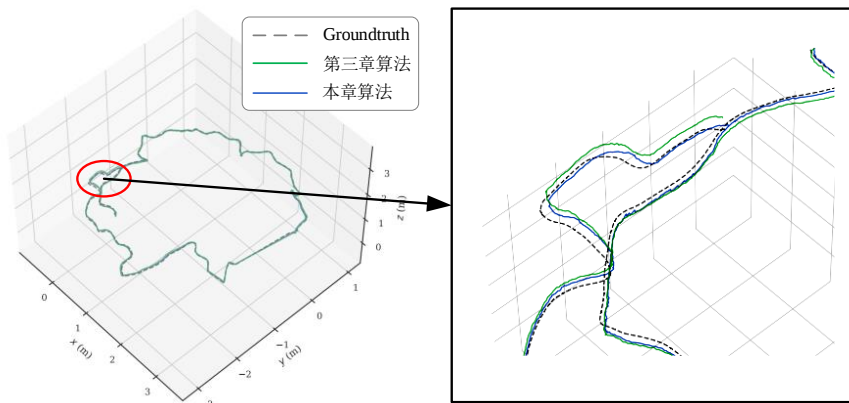
4.3.2 闭环检测性能对比

为了验证候选帧检测的有效性，实验使用室外 KITTI 数据集中包含闭环场景的 02、05、06 序列以及室内 TUM 数据集 fr1/desk1、fr2/desk、fr3/l/o/h 序列进行试验，对 DBOW 和本章算法进行闭环检测精度与实时性对比。闭环精度实验选取每个序列的 20 幅图像进行闭环检测，从这些图像中选 3 幅图像作为闭环图像，并且这些图像的前 5 幅图像顺序间隔相同，后 15 幅图像包含与闭环图像相似的特征。系统实时性实验表示算法通过运行整个序列，得出最终耗时。如表 4-1 所示，本文算法由于能精准选取闭环帧，提升了正确检测闭环图像的准确率，且本章算法减少了系统的计算量，在实时性方面表现一定优势。

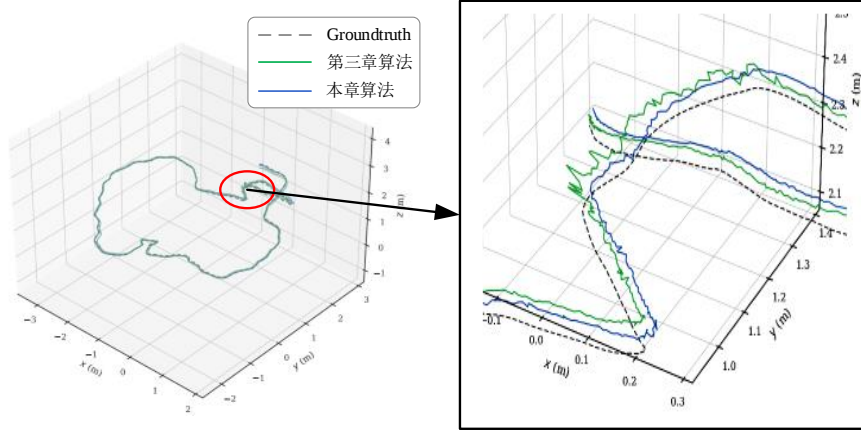
表 4-1 闭环检测正确率与实时性对比

序列	正确检测图像/幅		耗时/s	
	DBOW	本章算法	DBOW	本章算法
KITTI/02	12	14	619.58	573.92
KITTI/05	7	9	190.71	179.61
KITTI/06	13	15	67.30	62.49
fr1/desk1	12	15	29.18	27.32
fr2/desk	15	12	95.41	88.74
fr3/l/o/h	12	13	129.27	114.84

为验证本章所提闭环检测方法的有效性，将本文第三章未改进闭环检测算法与本章改进后的闭环检测算法在 TUM 公开数据集下进行实验验证。如图 4-7 所示，本章算法与第三章未改进的闭环检测算法进行了轨迹对比，从图 4-7 (a) 中红色区域可以看出，两种算法轨迹比较相似，但在弯道场景下本章算法接近真实轨迹。从图 4-7 (b) 的红色区域可以看出，第三章算法在面对复杂场景时易出现轨迹震荡的情况，无法精准重建轨迹。而本章算法在运行过程中能够精准选择闭环，降低轨迹图漂移误差，较为精准稳定的进行轨迹重建。



(a) fr2/desk



(b) fr3/l/o/h

图 4-7 轨迹对比实验

在 TUM 公开数据集下对第三章算法、和本章算法的闭环修正轨迹进行评估，用平移误差（m）与旋转误差（°）表示算法闭环性能，用耗时（s）表示算法的实时性，对比结果如表 4-2 所示。从表 4-2 可以看出，本章算法相比于第三章算法在 TUM 数据集四个序列的平移误差与旋转误差分别平均降低 9.97%、5.48%，定位精度具有较大优势，且耗时平均减少 20.20%，实时性得到较大提升。

表 4-2 TUM 数据集下的闭环检测性能对比

TUM 序列	第三章算法			本章算法			提升幅度/（%）		
	平移 误差（m）	旋转 误差（°）	耗时 （s）	平移 误差（°）	旋转 误差（m）	耗时 （s）	平移 误差	旋转 误差	耗时
fr2/desk	0.0068	0.9359	132.95	0.0061	0.8835	104.53	10.29	5.60	21.38
fr2/xyz	0.0030	1.2394	225.33	0.0026	1.1905	179.32	13.33	3.95	20.42
fr3/l/c	0.0426	1.3955	93.12	0.0402	1.2987	75.18	5.63	6.94	19.27
fr3/l/o/h	0.0066	0.4597	165.47	0.0059	0.4348	132.85	10.61	5.42	19.71
平均值	0.0148	1.0076	154.22	0.0137	0.9519	122.97	9.97	5.48	20.20

4.3.3 KITTI 和 TUM 数据集轨迹对比实验

本章节实验在第三章的基础上将改进的视觉里程计与关键帧选取策略结合本章基于 BWO 算法的闭环检测共同构成一个 SLAM 系统进行实验。首先选取 KITTI 数据集不同序列对本文改进视觉算法进行测试，并利用开源工具 EVO 对得到的实验效果进行可视化。为验证本章基于 BWO 闭环检测的改进视觉 SLAM 算法的定位精度，实验中将本章算法与 ORB-SLAM2 算法及 OV2-SLAM 算法在 KITTI 数据集进行实验比较，ORB-SLAM2 是一种基于 ORB 特征点法的视觉 SLAM 算法，在室内以及室外场景都有较好的建图效果，OV2-SLAM 是一种高

速视觉 SLAM，在视觉里程计中使用 FAST 提取与 LK 光流进行跟踪，且该算法的闭环检测使用的是 iBowLCD 在线构建字典，可以适应更多的场景。KITTI 数据集序列下的轨迹图如图 4-8 所示。图 4-8 中虚线为数据集内的真实轨迹，图中本章算法与 OV2-SLAM 算法均能完成不同精准度的轨迹跟踪，其中 09 序列场景内容较大并且轨迹内容较为复杂，相机长时跟踪易不稳定且面对闭环场景易产生误判等问题。因此，ORB-SLAM2 在运行时跟踪效果较差，OV2-SLAM 算法凭借在线构建词典，有效减少了部分轨迹漂移的问题，基本实现了轨迹复原。而本章算法引入了关键帧补偿决策与 BWO 闭环检测方法，克服了较差质量关键帧中特征点跟踪不稳定以及传统闭环检测方法在面对复杂场景时准确率较低的问题，对整条轨迹进行了精确的跟踪。在 02 序列中，受到场景复杂性增加和广泛的大视角场景影响，ORB-SLAM2 和 OV2-SLAM 均出现了轨迹漂移，而本章算法在长时跟踪下有明显优势，面对复杂场景时能够精准识别闭环，进而提高了算法的定位精度。

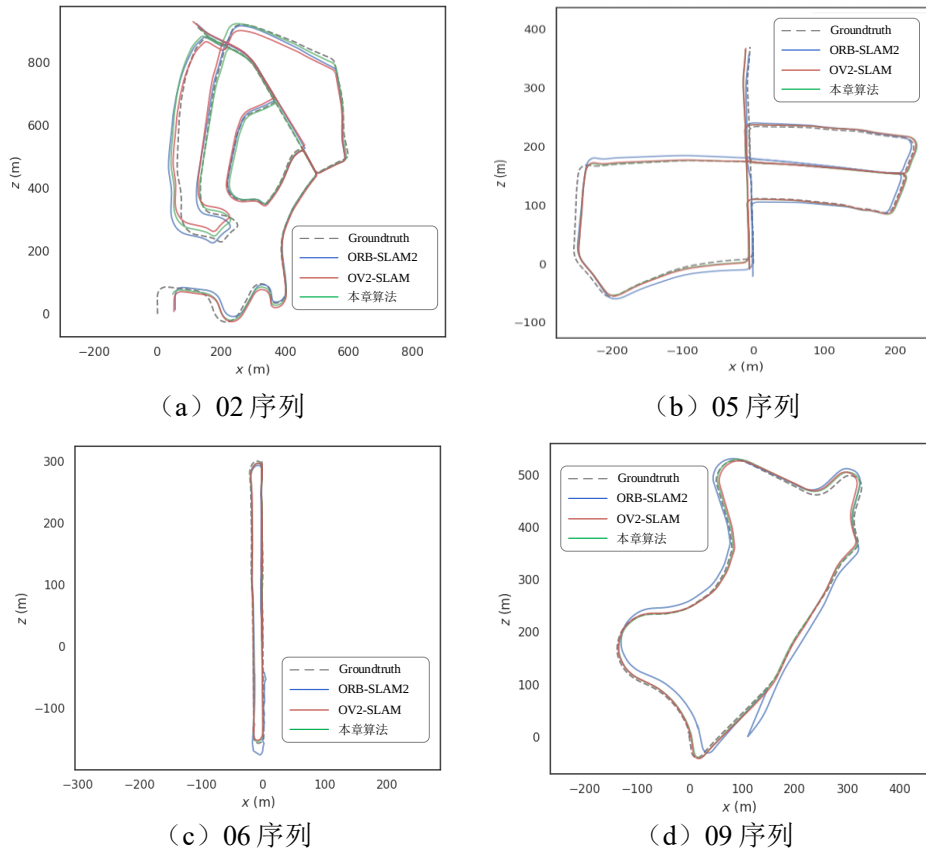


图 4-8 KITTI 数据集序列下的轨迹

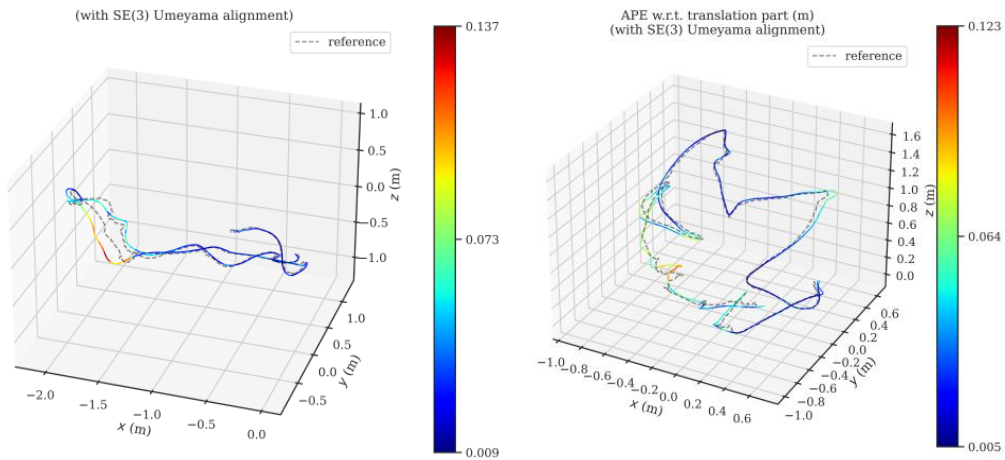
表 4-3 为 90%召回率下的 3 种算法轨迹均方根误差和准确率的对比结果，其中 ORB-SLAM2 是基于词袋模型进行闭环检测，OV2-SLAM 使用在线构建词典

的闭环检测方法，而本章算法是结合第三章改进视觉里程计和关键帧选取策略以及基于 BWO 闭环检测方法共同构成一个改进的视觉 SLAM 系统，从表中可以看出，除了 06 序列，本章算法在大部分情况下都能提升 10%-20%。06 序列相比于其他序列较为简单，三种算法闭环准确率相差不大。此外除了 02 与 07 序列，三种算法的闭环准确率都达到了 80%左右。在 02 序列下，相似场景的广泛存在（尤其是道路边相同的树木）导致了三种算法的准确率都相对较低，即便如此，本章算法相较于 ORB-SLAM2 算法还是取得了 17.39%的提升。在 07 序列下，本章算法相比其他两种算法也有明显提升。

表 4-3 90%召回率下的 3 种算法轨迹均方根误差和准确率对比结果

序列	准确率 (%)			轨迹均方根误差 (RMSE/m)		
	ORB-SLAM2	OV2-SLAM	本章算法	ORB-SLAM2	OV2-SLAM	本章算法
00	69.12	63.52	82.17	7.43	6.52	6.07
01	72.31	79.58	80.04	15.72	8.44	7.59
02	33.58	35.92	39.42	9.29	7.61	7.15
05	67.49	71.65	81.91	5.45	4.43	4.12
06	85.34	88.43	88.93	9.11	7.73	5.82
07	63.18	71.37	75.17	4.65	4.40	3.74
09	79.25	74.93	88.26	5.82	4.93	4.79

为进一步验证本文算法的性能，在 TUM 数据集下与 ORB-SLAM2、OV2-SLAM 的运行轨迹做出对比。图 4-9 表现了三种算法在 TUM 数据集下各序列的轨迹对比，从图中可以看出，本文算法在 TUM 序列中有更优表现，相比 ORB-SLAM2 而言，本文算法提取的关键帧质量更高，减轻了复杂运动时关键帧跟踪丢失造成的误差，并且减少了闭环检测的误判，使其更加接近真实轨迹。相对于 OV2-SLAM 算法，本文算法能更有效地利用场景信息，从而减小误差。



(a) ORB-SLAM2 fr1/desk1

(b) ORB-SLAM2 fr1/plant

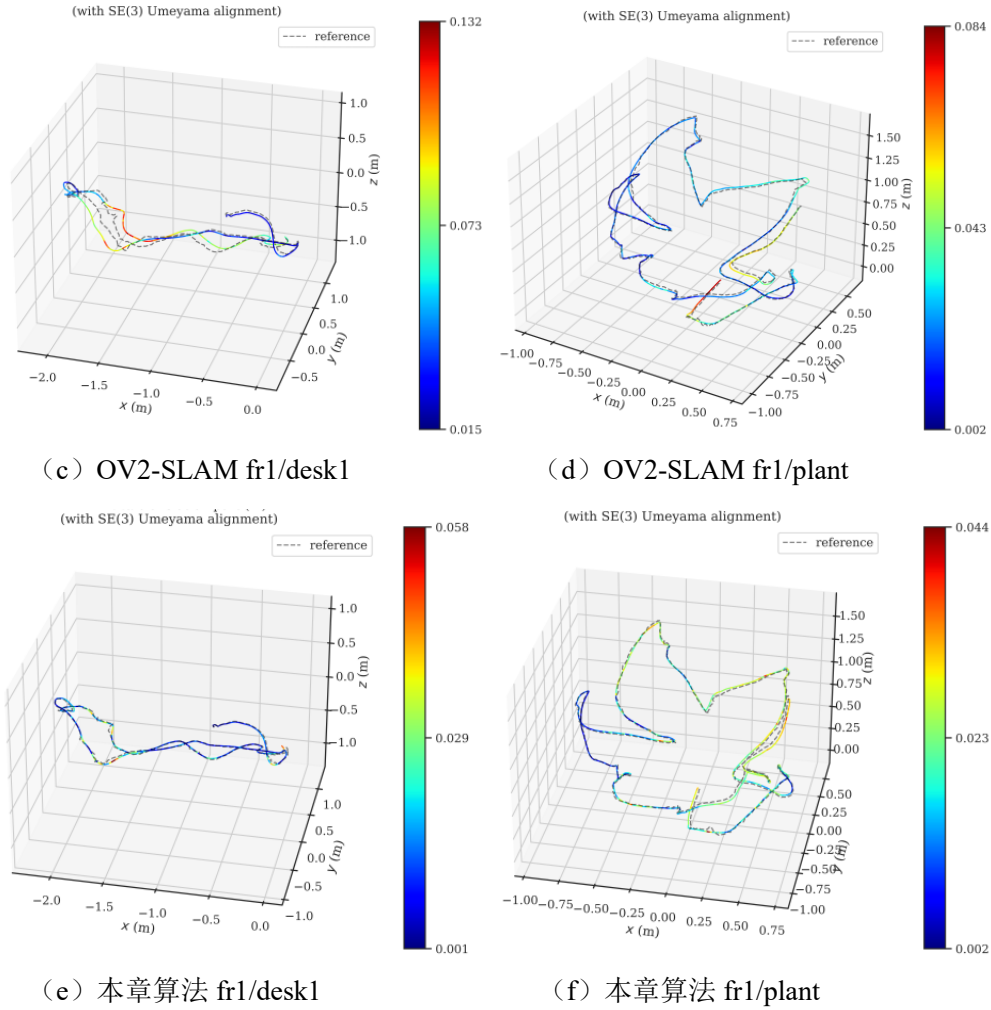


图 4-9 TUM 数据集下三种算法轨迹对比图

表 4-4 为 3 种算法在 TUM 数据集下的绝对轨迹误差的对比，从表中可以看出，本章节算法在 4 个序列中都获得了最佳结果。整体来看，本章算法的轨迹误差相比 ORB-SLAM 减少了 39.4%，相比 OV2-SLAM2 减少了 35.8%。从序列细分来看，3 种算法在 fr1/desk、fr1/plant 与 fr1/room 场景中表现效果较好，其中由于本章算法采用了关键帧补偿决策以及改进的闭环检测方法，在室内具有回环的场景下优势更为明显。而 fr3/cabinet 场景包含可用的图像信息较少，3 种算法的精度较为相近。

表 4-4 3 种算法在 TUM 数据集下的绝对轨迹误差的对比(单位: m)

TUM sequence	ORB-SLAM2	OV2-SLAM	本章算法
fr1/desk1	0.058	0.047	0.026
fr1/plant	0.045	0.031	0.021
fr1/room	0.115	0.130	0.072
fr3/cabinet	0.065	0.058	0.053
平均值	0.071	0.067	0.043

图 4-10 为 TUM 数据集中，本章算法与其他算法的闭环准确率与召回率对比。从图 4-10 中变化曲线可以看出，在 fr2/xyz 和 fr1/plant 两个序列上，在保证三种算法的准确率均为 100%的同时，本章算法的召回率的峰值分别达到了 84.1%和 87.7%，这与 ORB-SLAM2 算法相比，分别提高了大约 12.4%和 15.2%。

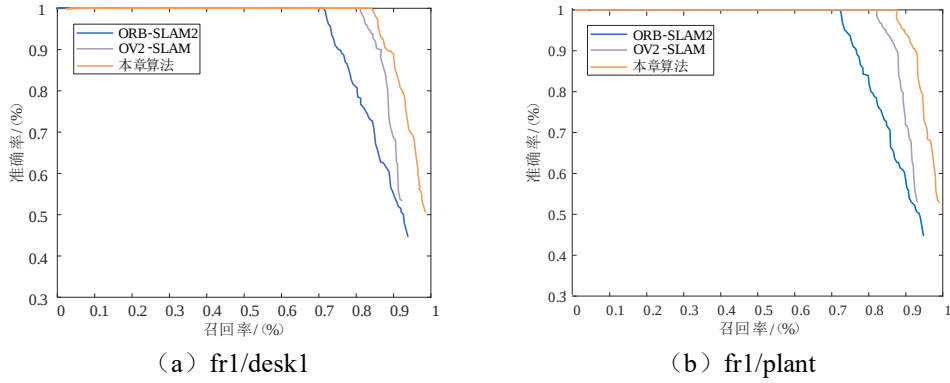


图 4-10 TUM 数据下的准确率-召回率对比图

4.4 移动机器人真实场景实验

为验证本文算法在实际应用场景中的效果，文中构建了基于 BWO 闭环检测的改进视觉 SLAM 系统的实验平台。该平台的软件部分负责实现机器人的自我定位和地图构建，为之后的数据分析工作提供基础，而硬件部分则确保机器人能够平稳地执行任务，在移动过程中收集必要的图像数据。文中设置真实场景进行移动机器人面对复杂场景及大视角运动 SLAM 算法实验，如图 4-11 所示，移动机器人平台以 Husky 机器人硬件为基础，配备了 Velodyne VLP-16 激光雷达、D435i 视觉传感器和 NVIDIA TX2 等设备。这台机器人主要由传感器模块、ROS 主机模块和底盘驱动模块组成。通过视觉传感器收集运动中的数据，并将这些数据实时发送给处理器进行处理，其中 Husky 机器人的扩展板起到底层运动控制的功能。



图 4-11 移动机器人平台

本章算法在办公室环境数据集上测试，该数据集包含了多种环境特征，如大视角场景和闭环场景。视觉 SLAM 系统在操作过程中，累积误差会逐渐增大，这将直接影响位姿估计的准确性和算法环境的适应性。累积误差的增加导致位姿估计精度下降，环境适应性减弱；而累积误差减少则意味着更高的估计精度和更强的环境适应性。图 4-12 为办公室真实场景测试，图 4-12 (a) 展示了办公室场景中机器人的移动轨迹，实验安排机器人从 A 点星形位置开始，按箭头指示的预设路径绕过会议桌、椅子和屏幕，最终在 B 点星形位置结束行程。实验采集了一组数据，包含了彩色图像和深度图像，大小为 480×640 。

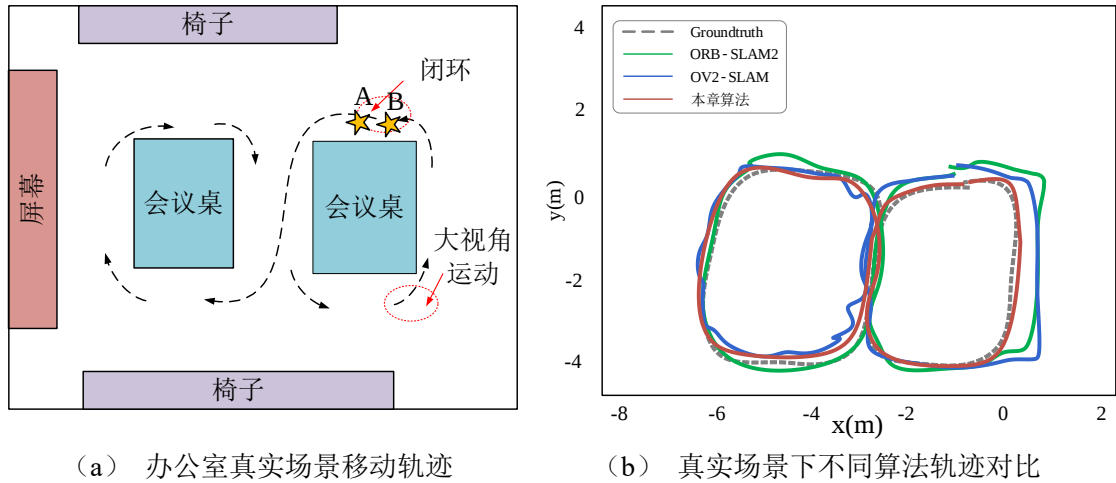


图 4-12 办公室真实场景测试

实验中分别采用 OV2-SLAM, ORB-SLAM2 与本文算法对机器人的运动轨迹进行解算，真实场景下移动机器人运行不同算法生成轨迹对比如图 4-12 (b) 所示。从实验结果可见在移动机器人行驶到拐点附近时，由于视角持续改变，ORB-SLAM2 和 OV2-SLAM 算法与之前帧图像之间的联系变得较弱，最终造成了闭环检测失败。而本章算法联合 BWO 算法和候选帧检测方法改善了闭环误检测的问题，进而提高算法在闭环时的运行效率和精度。OV2-SLAM 在大视角运动处存在跟踪丢失现象，导致复原轨迹缺少关键相机位姿，产生严重轨迹漂移问题。在大视角运动状态下，ORB-SLAM2 发生了尺度偏差与轨迹漂移，导致部分相机位姿丢失，对运行轨迹进行了错误的校正。然而本文算法在关键帧选取方面，提出了一种关键帧补偿决策，移动机器人在面临大视角变化时仍能够准确恢复其运动结构，达到高精度的定位和建图。根据表 4-5 中的真实场景测试结果，通过比较相对轨迹误差与绝对轨迹误差的数据，可以看出本文所提出的算法在办公室这种结构化环境中展现了优秀的闭环性能，同时具备了较高的轨

迹精度和实时性。

表 4-5 真实场景运行结果对比

算法	相对轨迹误差 (m)	绝对轨迹误差 (m)	总时间 (s)
ORB-SLAM2	0.53	0.42	29.84
OV2-SLAM	0.44	0.34	26.73
本章算法	0.37	0.28	24.18

4.5 本章小结

针对传统闭环检测方法在面对复杂场景时效率和闭环误检测的问题，提出一种基于 BWO 闭环检测的改进视觉 SLAM 算法。首先将图像特征描述子压缩成图像鲁棒描述子，再将闭环检测在历史帧图像中寻找与当前帧图像相似度最大的图像问题转化为最大值优化问题，然后给出基于 BWO 算法求解闭环检测问题的具体实现方法，最后提出一种候选帧检测方法，精准选择闭环帧，提高闭环性能。本章在 KITTI、TUM 等公开数据集上进行实验对比。结果表明，本章所提的基于 BWO 闭环检测的视觉 SLAM 算法相较于 ORB-SLAM2 和 OV2-SLAM 闭环检测方法具有更高的准确率、效率以及定位精度。最后算法在办公室真实场景中对本文算法进行实验验证，实验结果证明了本文算法的定位精度与实时性相较于 ORB-SLAM2 和 OV2-SLAM 要更好。

第 5 章 总结与展望

5.1 全文总结

第一章主要介绍了本文的研究背景和意义，分别概述了特征点法下和直接法下的视觉 SLAM 算法及闭环检测的国内外研究现状，最后对本文主要研究内容进行简要概述。第二章对视觉 SLAM 中视觉里程计、关键帧的筛选策略和闭环检测三个模块进行介绍。在第三章和第四章中，针对当前基于特征点法与光流法的视觉 SLAM 算法缺陷，运行过程中实时性与精度较低、大视角运动下选取的关键帧不准确造成的跟踪丢失以及在面对复杂场景时闭环检测效率与准确率较低等问题，提出一种基于 BWO 闭环检测的改进视觉 SLAM 算法，该算法在复杂环境中闭环准确率和定位精度较高，且满足实时性需求。

论文主要研究内容与创新点概括如下：

(1) 针对基于特征点法的 SLAM 算法特征匹配时效过低，传统光流与特征点融合算法跟踪精度较差，以及机器人在面对大视角运动时易出现关键帧上特征点跟踪不稳定导致位姿估计误差较大，定位精度降低等问题，提出一种改进视觉里程计和关键帧选取策略的视觉 SLAM 算法。首先，本文基于 ORB-SLAM2 算法框架提出一种改进融合光流与特征点的视觉里程计，通过双阈值约束特征匹配策略在算法运行过程中进行特征匹配或光流跟踪的切换，提高算法在跟踪线程的计算速度。其次，提出一种基于活跃特征点的关键帧选取策略，通过特征点的活跃值判断该点是否属于活跃特征点，再根据图像中活跃特征点的数量选取新关键帧，减少位姿估计误差，提高机器人在大视角运动下的定位精度和鲁棒性。实验结果表明本文算法能有效提升系统跟踪线程的计算速度并能构建准确的轨迹地图。

(2) 针对传统基于视觉词袋模型的闭环检测方法在面对复杂场景时效率不足、闭环误检测等问题，提出一种基于 BWO 闭环检测的视觉 SLAM 算法。首先，该算法通过引入 BWO 算法将闭环检测问题转化为最大值优化问题，然后，采用种群适应度函数代替传统的相似度计算方法，最后提出一种候选帧检测方法，通过判断阈值得到精确闭环帧，提升系统闭环检测精度并减少计算量。该

闭环检测方法结合第三章的改进视觉里程计与关键帧补偿决策共同构成一个改进的视觉 SLAM 系统，在面对复杂环境时有较高的效率与准确率。

(3) 通过在公开数据集及实际场景中验证本文所提算法基于 BWO 闭环检测的改进视觉 SLAM 算法研究的可行性，结果表明了本文算法有效地增强了系统的定位精度和鲁棒性，同时确保了系统的实时运行能力，提高机器人在实际场景中的闭环检测效率和精度。

5.2 研究展望

本文围绕“基于 BWO 闭环检测的改进视觉 SLAM 算法”进行研究，通过在公开数据集和自建的移动机器人硬件平台上进行实验，验证了本文所提算法的有效性。实验结果证明了该算法面对复杂环境时有良好的适应能力，然而，在实验的探索中还发现了一些亟待解决的问题，主要有以下几个方面：

(1) 构建语义 SLAM。本文基于 BWO 闭环检测的改进视觉 SLAM 算法在面临动态物时，实验环境中动态物体的特征点会影响视觉 SLAM 系统的定位精度。因此，未来研究的方向将集中在结合语义信息来识别并剔除场景中的动态物体方面。

(2) 构建基于深度学习的闭环检测。在视觉 SLAM 系统中，闭环检测的传统方法主要依赖于人工设计的图像特征来判断图像间的相似性，以此来确定闭环。然而，这些人工特征的鲁棒性通常较弱，且未能充分利用图像信息。面对大规模数据集，通过训练卷积神经网络提取图像的深层特征，能够有效利用图像内容，进而提升特征提取的效率和准确性。因此，本文下一步会将深度学习方法与闭环检测结合，提取图像的多尺度特征，从而提高闭环检测的准确率。

参考文献

- [1] 阴贺生,裴硕,徐磊等. 多机器人视觉同时定位与建图技术研究综述[J]. 机械工
程学报, 2022,58(11):11-36.
- [2] Chaka C. Fourth industrial revolution—a review of applications, prospects, and
challenges for artificial intelligence, robotics and blockchain in higher education[J].
Research and Practice in Technology Enhanced Learning, 2023, 18(2): 1-39.
- [3] 王耀南,江一鸣,姜娇等. 机器人感知与控制关键技术及其智能制造应用[J].自
动化学报,2023,49(03):494-513.
- [4] Smith, R C, Cheeseman P. On the representation and estimation of spatial
uncertainty [J]. International Journal of Robotics Research, 1986, 5(4): 56-68.
- [5] 张辉,王盼,肖军浩等. 一种基于三维建图和虚拟现实的人机交互系统[J].控制
与决策,2018,33(11):1975-1982.
- [6] 苏晓杰,刘星雨,李睿等. 半结构化场景下移动机器人视觉边线检测综述[J].控
制与决策,2023,38(06):1491-1509.
- [7] Macario Barros A, Michel M, Moline Y, et al. A comprehensive survey of visual
slam algorithms[J]. Robotics, 2022, 11(1): 24.
- [8] Gupta A, Fernando X. Simultaneous localization and mapping (slam) and data
fusion in unmanned aerial vehicles: Recent advances and challenges[J]. Drones,
2022, 6(4): 85.
- [9] Gil Aparicio A, Reinoso O, Paya L. Assessing the influence in the parameters of a
Rao-Blackwellised particle filter to solve the SLAM problem[J]. IEEE Latin
America Transactions, 2008, 6(1) :18-27.
- [10] Li Y, Brash N, Wang Y, et al. Structure-SLAM: Low-drift monocular SLAM in
indoor environments[J]. IEEE Robotics and Automation Letters, 2020, 5(4): 6583-
6590.
- [11] Yuan S, Guo P, Han X. DSMT-based ultrasonic detection model for estimating
indoor environment contour[J]. IEEE Transactions on Instrumentation and
Measurement, 2019, 69(7): 4002-4014.

- [12] 王铨彬,李星星,廖健驰等.基于图优化的紧耦合双目视觉/惯性/激光雷达 SLAM 方法[J].测绘学报,2022, 51(08): 1744-1756.
- [13] Wang Y, Fu Y, Zheng R, et al. New trend in front-end techniques of visual SLAM: from hand-engineered features to deep-learned features[C]. Artificial Intelligence in China: Proceedings of the 3rd International Conference on Artificial Intelligence in China. Singapore: Springer Singapore, 2022: 298-307.
- [14] Chen Q, Yao L, Xu L, Yang Y, Xu T, Yang Y, Liu Y. Horticultural Image Feature Matching Algorithm Based on Improved ORB and LK Optical Flow[J]. Remote Sensing. 2022, 14(18): 4465.
- [15] S Wen et al. Joint optimization based on direct sparse stereo visualinertial odometry[C]. Auto Robots.Netherlands, 2020:791-809.
- [16] 李康宇,王西峰,徐斌等.非结构化环境下基于外观的闭环检测研究综述[J].机器人,2023,45(2):238-256.
- [17] Davison A J, Reid I D, Molton N D, et al. MonoSLAM: Real time single camera SLAM[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007, 29(6): 1052-1067.
- [18] 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. IEEE, 2007: 225-234.
- [19] 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.
- [20] Angeli A, Filliat D, Doncieux S, et al. Fast and incremental method for loop-closure detection using bags of visual words[J]. IEEE transactions on robotics, 2008, 24(5): 1027-1037.
- [21] [Gálvez-López D, Tardos J D. Bags of binary words for fast place recognition in image sequences[J]. IEEE Transactions on Robotics, 2012, 28(5): 1188-1197.
- [22] Mur-Artal R, Montiel J M M, Tardos J D. ORB-SLAM: A versatile and accurate monocular SLAM system[J]. IEEE Transactions on Robotics, 2015, 31(5): 1147-1163.
- [23] 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.
- [24] Campos C, Elvira R, Rodríguez J J G, et al. ORB-SLAM3: An accurate open-source library for visual, visual-Inertial, and multimap SLAM[J]. IEEE Transactions on Robotics, 2021, 37(6): 1874-1890.
- [25] 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.
- [26] Engel J, Schöps T, Cremers D. LSD-SLAM: Large-scale direct monocular SLAM[C]. European conference on computer vision. Cham: Springer International Publishing, 2014: 834-849.
- [27] Engel J, Koltun V, Cremers D. Direct sparse odometry[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018, 40(3): 611-625.
- [28] Forster C, Pizzoli M, Scaramuzza D. SVO: Fast semi-direct monocular visual odometry[C]. 2014 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2014: 15-22.
- [29] 王通典, 刘洁瑜, 吴宗收等. 基于多尺度光流融合特征点视觉-惯性 SLAM 方法[J]. 系统工程与电子技术, 2022, 44(03): 977-985.
- [30] 张峻琦, 胡立坤. 融合 RANSAC 光流跟踪法和特征点匹配法的视觉里程计[J]. 应用光学, 2020, 41 (06): 1214-1221.
- [31] 许广富, 曾继超, 刘锡祥. 融合光流法和特征匹配的视觉里程计[J]. 激光与光电子学进展. 2020, 57(20): 270-278.
- [32] 齐乃新, 杨小冈, 李小峰等. 基于 ORB 特征和 LK 光流的视觉里程计算法[J]. 仪器仪表学报. 2018, 39(12): 216-227.
- [33] 张洪, 于源卓, 邱晓天. 基于改进关键帧选择的 ORB-SLAM2 算法[J]. 北京航空航天大学学报. 2023, 49 (01): 45-52.
- [34] I. Alonso, L. Riazuelo and A. C. Murillo. Enhancing V-SLAM Keyframe Selection with an Efficient ConvNet for Semantic Analysis[C]. International Conference on Robotics and Automation (ICRA 2019). Montreal, QC, Canada, 2019: 4717-4723.
- [35] X. Lin et al. A Robust Keyframe-Based Visual SLAM for RGB-D Cameras in

- Challenging Scenarios[J]. in IEEE Access, 2023,11: 97239-97249.
- [36] 佟晓宇. 基于 Kinect 机器人的视觉 SLAM 方法研究[D].天津: 天津工业大学, 2019.
- [37] 胡年宗, 伍世虔, 张亦明. 基于卷积神经网络的 SLAM 回环检测算法研究[J]. 计算机仿真, 2020, 37(5): 282-286.
- [38] Sivic J, Zisserman A. Video Google: A text retrieval approach to object matching in videos[C]. IEEE International Conference on Computer Society. IEEE, 2003: 1470-1477.
- [39] Newman P, Ho K. SLAM-loop closing with visually salient features[C]. Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, 2005: 635-642.
- [40] Nister D, Stewenius H. Scalable recognition with a vocabulary tree[C]. 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR2006), New York, NY, USA, 2006: 2161-2168.
- [41] Angeli A, Filliat D, Doncieux S, et al. Fast and incremental method for loop-closure detection using bags of visual words[J]. IEEE transactions on robotics, 2008, 24(5): 1027-1037.
- [42] Liu Y, Zhang H. Visual loop closure detection with a compact image descriptor[C]. IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway, USA, 2012:1051-1056.
- [43] Labbe M, Michaud F. Memory Management for Real-Time Appearance-Based Loop Closure Detection[C]. IEEE International Conference on Intelligent Robots and Systems, 2011,1271-1276.
- [44] R. Mur-Artal and J. D. Tardós, Fast relocalisation and loop closing in keyframe-based SLAM[C]. IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 2014: 846-853.
- [45] Zhang G , Lilly M J , Vela P A. Learning binary features online from motion dynamics for incremental loop-closure detection and place recognition[C]. 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 2016: 765-772.

- [46] 林俊钦,韩宝玲,罗庆生,等. 基于 NDT 匹配和改进回环检测的 SLAM 研究[J]. 光学技术,2018,44(2):152-157.
- [47] 夏林. 基于改进回环检测的视觉惯性 SLAM 技术研究[D]. 北京: 北京林业大学, 2020.
- [48] 苑全德. 基于视觉的多机器人协作 SLAM 研究[D]. 哈尔滨工业大学, 2016.
- [49] Zhang Z, Dong P, Wang J, et al. Improving S-MSCKF with variational Bayesian adaptive nonlinear filter[J]. IEEE Sensors Journal, 2020, 20(16): 9437-9448.
- [50] 刘钰嵩,何丽,袁亮,等. 动态场景下基于光流的语义 RGBD-SLAM 算法[J]. 仪器仪表学报, 2022, 43(12): 139-148.

攻读学位期间发表的学术论文目录

- [1] 姚满宇, 陈孟元*, 徐韬. 曲线运动下基于关键帧补偿决策的快速 SLAM 算法. 安徽工程大学学报 (录用)
- [2] 徐韬, 姚满宇, 袁学超, 陈孟元*, 刘晓晓, 张玉坤. Dense 3D Reconstruction SLAM Algorithm Based on Improved Keyframe Selection. YAC2022 中国自动化学会青年学术年会议 (EI 收录, 检索)

攻读学位期间取得的荣誉奖励目录

- [1] 2022 年第十六届“西门子杯”中国智能制造挑战赛全国初赛, 智能制造工程设计与应用类赛项, 华东赛区二等奖. (第一完成人)
- [2] 2023 年第十届安徽省“挑战杯·华安证券”大学生课外学术科技作品竞赛一等奖. (主要完成人)
- [3] 2022 年安徽工程大学第十一届专利发明与创新大赛三等奖. (主要完成人)
- [4] 2021-2022 年度安徽工程大学硕士研究生学业奖学金一等奖.

致 谢

时光荏苒，我的硕士生涯已接近尾声。这几年的时光既漫长又短暂，其中充满了酸甜苦辣，更有收获和成长。几年来，感谢陪我一起度过美好时光的每位尊敬的老师和亲爱的同学，正是你们的帮助，我才能克服困难，正是你们的指导，我才能解决疑惑，直到学业的顺利完成。

本人的学位论文是在我的恩师陈孟元教授的殷切关怀和耐心指导下进行并完成的，衷心感谢我的恩师对我的谆谆教诲和悉心关怀。从课题的选择、项目的实施，直至论文的最终完成，陈老师都始终给予我耐心的指导和支持，我取得的每一点成绩都凝聚着恩师的汗水和心血。恩师开阔的视野、严谨的治学态度、精益求精的工作作风，深深地感染和激励着我，在此谨向陈老师致以衷心的感谢和崇高的敬意。此外，还要感谢孙新柱老师提出的宝贵意见，对我论文的写作起到了很重要的作用。

在此我要对实验室所有师兄姐妹们在平时开展相关工作中的支持和帮助并表示感谢。感谢我的伙伴们在实验过程和论文写作过程中提供的热心帮助!无论在炎热的夏天，还是寒冷的冬季，他们不辞劳苦地为我提供无私的帮助，没有他们的帮助就没有这篇论文的顺利完成。

最后，我要向百忙之中参与审阅、评议本论文各位老师、向参与本人论文答辩的各位老师表示由衷的感谢。人生的每个阶段都值得好好珍惜，这段美好岁月，因为有你们的关心和帮助，我很幸福。我会更加勤奋学习、认真研究，努力做得更好，我想这也是我能给你们的最好的回报吧。把最美好的祝福献给你们，愿永远健康、快乐。

尚德敏学 唯实惟新

